



THE UNIVERSITY *of* EDINBURGH

This thesis has been submitted in fulfilment of the requirements for a postgraduate degree (e.g. PhD, MPhil, DClinPsychol) at the University of Edinburgh. Please note the following terms and conditions of use:

This work is protected by copyright and other intellectual property rights, which are retained by the thesis author, unless otherwise stated.

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge.

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author.

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.



Digisprudence:
The affordance of legitimacy in code-as-law

Laurence E. Diver

Submitted for the degree of Doctor of Philosophy
The University of Edinburgh
April 2019

For my late mother and father,

Beatrix and Brian.

I wish you were here for this

– but I suppose, in some sense, you are.

Contents

Please note that heading and footnote numbers restart in each new chapter.

Declaration	vii
Abstract	ix
Lay summary	xi
Acknowledgements	xiii
Chapter 1 (Introduction)	1
1. Introduction	1
1.1 Code.....	2
1.2 Highlighting a tension.....	8
1.3 Questions, aims, and methodology.....	12
1.4 Legitimacy vs. ‘compliance by design’	18
1.4.1 Normative relationships online.....	22
2. Code is law(-like).....	25
2.1 The design of behavioural constraint	29
2.2 Computational legalism.....	31
3. Aspiring to legitimacy in code	34
3.1 From <i>operation</i> to <i>production</i>	34
3.1.1 Fuller’s internal morality of law	35
3.1.2 Wintgens’ legisprudence	36
3.2 Towards a <i>digisprudence</i> : the affordance of legitimacy in code-as-law	37
4. Case studies.....	39
4.1 Digital rights management	39
4.2 Blockchain applications	43
4.2.1 Blockchain design	44
4.2.2 ‘Smart contracts’?	47
4.3 The Internet of Things	53
5. Conclusion and recap.....	56

Chapter 2 (A design perspective: code is <i>more</i> than law)	59
1. Introduction	59
2. Affordance	60
2.1 Real and perceived affordance.....	63
2.1.1 Signifiers.....	65
3. Infusing code with normativity	66
3.1 Disaffordance	67
3.2 Postphenomenology and code’s mediation of reality.....	71
3.3 Code mediating perception.....	72
3.4 Code mediating action.....	73
3.4.1 Affordance and technological mediation.....	79
3.5 A spectrum of technological normativity.....	81
3.5.1 Constitutive and regulative normativity.....	85
3.6 Technological constitutionalism.....	87
3.6.1 Integrated development environments.....	90
4. Conclusion	92
Chapter 3 (A legal theory perspective: code is <i>less</i> than law)	95
1. Introduction	95
2. What is legalism?	97
2.1 Solipsism and positivism.....	101
2.2 Legalism according to legisprudence	103
3. Computational legalism	109
3.1 Ruleishness.....	111
3.1.1 Three consequences.....	112
3.2 Representationalism.....	117
3.2.1 Constitutive and regulative rules	121
3.3 Immediacy.....	126
3.3.1 Default configurations.....	127
3.3.2 Pervasiveness	130

3.3.3	Immutability	131
3.4	Opacity.....	133
3.4.1	Code as legal positivism.....	135
3.5	The veiling of code's production	137
4.	Conclusion	140
 Chapter 4 (Normative criteria for producing {law, code}).....		143
1.	Introduction	143
2.	Normative criteria for law-making: the aspirations of legality	144
2.1	Input and output legitimacy in law.....	148
2.2	Fuller's internal morality of law	152
2.3	Wintgens' jurisprudence.....	156
2.3.1	What is jurisprudence?	157
2.3.2	The requirement of justification	158
2.3.3	The principle of coherence (PC).....	162
2.3.4	The principle of alternativity (PA).....	168
2.3.5	The principle of temporality (PT)	170
2.3.6	The principle of normative density (PN).....	171
2.4	Conclusion	172
3.	Normative criteria for code-making.....	174
3.1	Input and output legitimacy in code.....	176
3.2	Output legitimacy	180
3.2.1	Brownsword's 'technological management'	180
3.2.2	Leenes' 'techno-regulation'	183
3.2.3	Koops' 'criteria for normative technology'.....	184
3.3	Input legitimacy	188
3.3.1	Hildebrandt's 'Legal Protection by Design'	188
3.3.2	The Fullerian principles applied to code.....	191
3.4	Conclusion	202
4.	Conclusion	203

Chapter 5 (Affording computational legitimacy) 205

- 1. Introduction 205
 - 1.1 Testing *decisions* or testing *design*? 206
- 2. Mapping the criteria..... 208
 - 2.1 Contestability as an overarching affordance 211
 - 2.1.1 Affording scrutiny to legal institutions..... 212
- 3. From characteristics to affordances 213
 - 3.1 Ruleishness..... 214
 - 3.1.1 Affordance: choice..... 214
 - 3.1.2 Discussion: blockchain applications 228
 - 3.1.3 Discussion: IoT 231
 - 3.2 Opacity..... 231
 - 3.2.1 Affordance: transparency of provenance & purpose 233
 - 3.2.2 Affordance: transparency of operation 235
 - 3.2.3 Discussion: blockchain applications 239
 - 3.2.4 Discussion: IoT 241
 - 3.3 Immediacy..... 243
 - 3.3.1 Affordance: delay..... 244
 - 3.3.2 Discussion: blockchain applications 253
 - 3.3.3 Discussion: IoT 255
 - 3.4 Immutability 256
 - 3.4.1 Affordance: oversight 257
 - 3.4.2 Discussion: blockchain applications 261
 - 3.4.3 Discussion: IoT 265
 - 3.5 Pervasiveness 266
- 4. Conclusion 268

Chapter 6 (Operationalisation: towards agile *digisprudence*) 271

- 1. Introduction 271
- 2. The programmer of the programmer..... 272

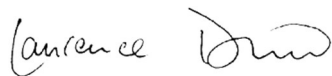
2.1	From primary and secondary rules to primary and secondary (dis)affordances and inscriptions.....	274
2.2	Agile development processes	277
2.3	Integrated development environments	279
2.3.1	Facilitating contestability in the IDE	282
3.	Conclusion	294
Chapter 7 (Conclusion).....		295
1.	Introduction	295
2.	The relevance of the thesis	296
3.	Research questions and contributions.....	297
3.1	Research question 1: How does code in fact regulate end-user behaviour? 298	
3.2	Research question 2: To what extent are the characteristics of legalistic illegitimacy reflected in the normativity of code?	299
3.3	Research question 3: Can mechanisms for designing legitimate <i>legal</i> normativity be adopted to ensure the design of legitimate <i>technological</i> normativity?.....	301
3.4	Research question 4: What might such mechanisms look like in the design context?	302
4.	Further research	303
4.1	Drawing law-making and code-making closer together – the future of compliance by design	303
4.2	Design and private law.....	306
4.3	‘Legitimacy Impact Assessment’	308
4.4	Application to data-driven applications	310
5.	Conclusion	312
Bibliography		313
1.	Primary sources	313
2.	Books.....	313
3.	Journal articles and book chapters.....	315
4.	Reports, technical publications, and whitepapers	327
5.	Grey literature (news articles, blogs, etcetera)	329

Declaration

I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where it states otherwise by reference or acknowledgement, the work presented is entirely my own.

Elements of this thesis have been published in the following articles:

- **Chapter 2:** L. Diver, 'Law as a User: Design, Affordance, and the Technological Mediation of Norms' (2018) 15(1) *SCRIPTed* 4
- **Chapter 6:** L. Diver and B. Schafer, 'Opening the Black Box: Petri Nets and Privacy by Design' (2017) 31(1) *International Review of Law, Computers & Technology* 68



Laurence E. Diver

Old College, University of Edinburgh

23rd April 2019

Abstract

This multidisciplinary thesis is located at the intersection of legal theory and design. It synthesises the practical question of how code regulates (using theories including James Gibson's/Donald Norman's affordance, Don Ihde's postphenomenology, and Madeleine Akrich's inscription) with a legal-theoretical view of what constitutes legitimate regulation (using theories including Lon Fuller's internal morality of law, Luc Wintgens' jurisprudence, and Mireille Hildebrandt's legal protection by design).

Proceeding from the notion that code is an a-legal normative order, I argue that even (and indeed especially) in the absence of suitable or sufficient legal regulation, the norms of that order ought to be legitimated. This is particularly true given the unique characteristics of code as a regulator, which include its ruleishness, opacity, immediacy, immutability, pervasiveness, and, perhaps most importantly, its production by private enterprise. Having set out how code regulates from the perspective of the design theories mentioned above, I explore these characteristics from a legal-theoretical perspective, developing the concept of *computational legalism*, a uniquely strong form of the undesirable ideological phenomenon of legalism. This is the first significant contribution of the thesis.

Having set up the parallel between legal and technological normativity, I explore the extent to which ex ante mechanisms for ameliorating legalism in the creation of *legal* norms can be translated into the 'legislature' of the design environment, to be applied in the creation of *code*-based norms. The motivating idea is that the standards that make legal norms legitimate ought *mutatis mutandis* to be applicable to other normative orders that enable and constrain individual behaviour. The literature has so far tended to focus on ex post assessments of code's operation, and to that extent it fails to account for computational legalism and the standards that must be met – by definition during the production process, ex ante – in order to mitigate or avoid it.

Taking all of this into account, the second significant contribution of the thesis is a ‘constitutional’ framework of *digisprudential affordances* that I argue ought to be present in all user-facing code, in order to ensure that certain foundational capabilities are provided by the design. The affordances I identify are: contestability; transparency of provenance, purpose, and operation; choice; oversight; and delay. They act as a formal mechanism for constraining what substantive code can possibly do, imposing ‘thin’ constitutional design standards that ought to be met regardless of the ‘thick’ purposes or functionality of the digital artefact. I discuss how these might be implemented in practice through an analysis of two contemporary technologies, the Internet of Things and blockchain applications. This practical element of the thesis connects with the last significant novel contribution, namely an exploration of Cornelia Vismann and Markus Krajewski’s concept of the *programmer of the programmer*, and how this ‘constitutional actor’ can be used to impose digisprudential limits – analogous to HLA Hart’s secondary rules – within the code development process.

Lay summary

This thesis considers the ways in which the architectures (user interfaces and general operating ‘rules’) created by computer code regulate behaviour, and how this form of regulation can and should be made legitimate. The specific notion of legitimacy that I adopt comes from the literature on legal theory, specifically in the area of legal rule creation (i.e. legislation).

I first describe, using design theory, the ways in which computer architectures enable and constrain user behaviour. I then look at these characteristics from a legal theoretical perspective, mapping them on to the philosophical concept of *legalism*. Legalism is concerned with the following of rules regardless of their quality or efficacy. It ‘veils’ the reasons behind a rule and the sources of power which have caused it to be created. In a similar way, I argue that code imposes the following of rules on its users and, because of its private production, the processes by which code is produced are also to a great extent ‘veiled’. This leads me to propose a theory of *computational legalism*, which is the first significant contribution of the thesis.

From that position I then explore how mechanisms that are designed to reduce legalism in the traditional legal context might be transferred into the code design context, given that in the metaphor I build up the design environment is essentially a type of legislature, or parliament, where rules are created. Like traditional legal rules, code rules also enable and constrain behaviour, and so they should be held to similar standards of legitimacy. I adopt two primary theories from legal philosophy, called the *internal morality of law* and *legisprudence*. Each of these is to an extent concerned with the *form* that legal rules ought to have in order to prevent their iniquity, and so I use them as a basis to develop similar formal standards for code rules.

This is the basis for the second significant contribution of the thesis. The existing literature has mostly tended to focus on the operation of code, whereas I argue that we should be much more closely concerned with its production. Here I come full-circle to make use of the design theories as tools to develop a framework of *digisprudential* design characteristics

(‘affordances’) that legitimate code ought to have. These characteristics are ‘constitutional’, or foundational, meaning they should be present in all user-facing code regardless of its commercial purposes. Broadly, the characteristics are: the ability to contest the code in court; transparency of the code’s provenance, purpose, and its operation; the ability of the user to exercise choice; appropriate delays in the code’s operation to allow the user to understand what is happening; and oversight of the code by its designer to fix problems or revoke it after launch. The framework is complemented by a discussion of ways in which some of its elements can be practically integrated into the code development process, to create architectural guides that prevent the creation of illegitimate code from the outset.

Acknowledgements

I'd like to thank some of the people who have supported me during the years of researching and writing this thesis, directly or indirectly. Friends in the Edinburgh Ph.D. community have provided many, many discussions, laughs, dinners, boulderings, room-escapes, and explorations of questions mundane and profound – thank you especially to Dani, Dawoon, Himani, Jiahong, Johan, Lucas, Matt, Pablo, Paul, and Xiaoou.

Towards the more senior end of the academic spectrum I'd like to thank my supervisors Burkhard Schafer and Judith Rauhofer. Burkhard in particular has been ever a great source of ideas, inspiration, and just the right amount of encouragement to keep me on track, especially in the final frantic months of completion when his willingness to parse a flurry of drafts made all the difference to my ability to submit on time. A big thank you to him also for his general support in career matters, from assisting with my application for the Principal's Career Development scholarship to tips on how best to round out my burgeoning academic CV. A thank-you is due also to the College of Arts, Humanities and Social Science at Edinburgh for the scholarship award, without which I wouldn't have been able to embark on this research. On a final academic note, I'd like to thank Lilian Edwards for the opportunity to work with her as a research assistant during her time at Strathclyde. It provided me with invaluable experience ranging from the exciting (discussing with an LL.M. class how to think critically about algorithms) to the somewhat more administrative (scanning airport sandwich receipts). But it has all been to the good: that broad range of experience has equipped me to embark on the next phase of my academic career, working in Brussels on a new project investigating the foundations of legal theory and practice in the new code- and data-driven paradigm. I couldn't have hoped for a more appropriate sequel to the work in this thesis.

Less academically, but no less importantly, I would like to thank the many friends I have made in The Edinburgh Samba School (TESS), before and during the Ph.D. You've supported me in so many ways, large and small, not least in facilitating the incredible experience (one I may have become somewhat accustomed to over the years, but which still has the power to create the most immense joy) of smashing out a cathartic *afôxê*, *maracatu*, or *Burn'em* in a bateria overflowing with infectious energy. There's been no better antidote to the

isolation of writing, and I've lost count of how many long weeks of research were made bearable by Friday night rehearsal or a weekend gig. My deepest thanks to the great orange ship of TESS for being a bright constant in my life over the last five years – one day I'll return to Edinburgh and we'll play *Algorhythm* together!

On a familial note, I'd like to thank my late parents, to whom this thesis is dedicated. Without the sense of possibility that my mother Beatrix instilled in me from a young age, I doubt very much that I'd be typing these words. Thank you.

Lastly, but by no means least, I'd like to thank HJD for your support, particularly over these past few months. It was your wit that got me first, but your kindness has proved to be something worth treasuring.

Chapter 1

Introduction

Given that code is not exactly like law, it is difficult in the realm of code to adopt a kind of rule of law (or ‘rule of code’) approach. Yet, we have also seen that when a particular code is ‘enacted’, it may be too late to remedy the violation of certain rights. This is why the accent should be put on the moment of production, rather than on the moment of distribution.¹

1. Introduction

This thesis is located at the intersection between law and design. It is concerned with how the rules, or normativity, that designers embed in the code of technological artefacts can be legitimately produced. The question of legitimacy is viewed from a legal-theoretical perspective, and the analysis asks whether the standards that make a *legal* rule legitimate might be imported into the realm of design to make a *computational* rule legitimate. Quintessentially multidisciplinary, the thesis contributes to knowledge both by deepening our understanding of the connection between law and design, and by developing a framework that can be used to critique and guide the production of digital artefacts which have a regulatory effect on behaviour. Additionally, it considers the constitutional role of those who create the technical environments that product designers use to produce code, making the novel argument that those environments are one suitable point at which effective constraints on design can be imposed.

Legal analyses of technology often focus on compliance with one or more areas of substantive doctrinal law, for example data protection or intellectual property. By contrast, I remain agnostic as to specific areas of doctrine, adopting instead a legal-

¹ M. Goldoni, ‘The Politics of Code as Law: Toward Input Reasons’ in J. Reichel and A.S. Lind (eds.), *Freedom of Expression, the Internet and Democracy* (Leiden: Brill, 2015) p. 128.

theoretical framing that considers how good law (in any field) is thought to be made. I use the existing literature on legalism and its alternatives to focus on constraining *form* as a mechanism to ensure legitimate *substance*, whatever the intended goal of the latter may be. This illuminates some parallels between the legal and computational spheres, parallels that I collectively term *computational legalism*. This sustained analysis of digital architecture through the prism of the literature on legalism is the first significant contribution of the thesis.

Following from that descriptive analysis, I consider how these challenges might be ameliorated by the adaptation to the computational realm of two primary theoretical frameworks that combat legalism in the realm of traditional rule-making, namely Fuller's *internal morality of law* and Wintgens' *legisprudence*. I review these theories alongside the existing literature on ex ante criteria for code production, creating the theoretical scaffolding for a synthesis that bridges the gap between legal analysis and design practice. That synthesis is the second significant contribution of the thesis. It is a novel and direct engagement with the design of digital systems whose normative architectures exhibit computational legalism, suggesting how a 'design turn' in legal thinking might be effected in practice. I discuss blockchain applications and the Internet of Things, anchoring the discussion in real-world application. I call the resulting theoretical and practical contribution *digisprudence*, based on the idea that where Wintgens' legisprudence is an evolution or extension of jurisprudence within the legislative realm, digisprudence is an evolution of legisprudence within the design realm.

1.1 Code

The starting point of the thesis is expressed in the quote at the beginning of this chapter. Code is like traditional law in its regulative capacity, but is also different from it in crucial ways. Code is like legislation, in that it is created to achieve some end; it

is 'enacted'. Code is capable of violating rights, whilst simultaneously resisting the aspirations and oversight of the rule of law. And finally, there are two pivotal moments at which assessments of code can most fruitfully be made: *ex ante* at the point of production, or *ex post* at the point of operation. Each of these elements plays an important role in the argument I am presenting, which runs as follows.

When commercial enterprises create digital artefacts, or code,² they create alternative normative orders that are apt to replace traditional institutional law as a primary source of behavioural regulation. Importantly, the private commercial contexts within which this code is created are not subject to the legitimising procedural standards of law-making found in constitutional democracies. This means that in the move from public to private rule-making, the resulting effects of that code on behaviour risk being illegitimate, whether or not this is intended.³ The question then arises of whether formal standards of law-making might be imported into the *sui generis* 'legislature' of the commercial design environment, in order to ensure that the code it produces is legitimate. Asscher summarises the spirit of the enquiry in these terms:

Code can present constraints on human behaviour that can be compared with constraints by traditional laws. We have argued that *even though code is not law, in some instances it can be useful to ask the same questions about code regulation as we do about traditional regulation*. Code as law must be assessed by looking at the results of regulation in terms of freedom and individual autonomy and compared to the balance struck by traditional law.⁴

² I further explain my use of the term 'code' below.

³ E. Bayamlioglu and R. Leenes, 'The "Rule of Law" Implications of Data-Driven Decision-Making: A Techno-Regulatory Perspective' [2018] *Law, Innovation and Technology* 1, p. 12.

⁴ L. Asscher, "'Code' as Law: Using Fuller to Assess Code Rules' in E. Dommering and L. Asscher (eds.), *Coding Regulation: Essays on the Normative Role of Information Technology* (The Hague: TMC Asser Press, 2006) p. 86 (my emphasis).

The thesis discusses various aspects of this insight, including the regulative capabilities of code, the ways in which code differs from law, and how autonomy might be maintained in the alternative normative order of code.

An analogy for the idea of a separate order that is subject to formal standards of legitimacy can be found in the millennia-old notion of the scientific method. According to its modern incarnation, a theory will only be accepted if it meets predefined formal standards of observation, hypothesis, prediction, experimentation, reproducibility, falsifiability, promulgation, and peer review.⁵ Even if the putative ‘fact’ happens to be true objectively, those formal requirements must be met in order for it to qualify as a fact from the scientific perspective. One can see therefore how legitimacy as a concept can be an internal formal requirement in various fields (although the precise steps required to achieve it will of course vary).⁶

My purpose here in framing code in terms of *legal* legitimacy derives from the point, made above, that it can so readily augment and even supplant law as a regulator. Those who create code, whose work is shielded by the private context of its production, ought therefore to wield that power legitimately.⁷ If sovereign legislatures are bound by constitutions so that they cannot arbitrarily impose regulations on citizens’ behaviour, then neither should this be possible for private enterprise, especially given the characteristics of code that render it qualitatively and quantitatively more problematic to comprehend and control than text-based legislation. We will see below and in Chapters 2 and 3 what those characteristics are and what effects they have.

⁵ ‘Scientific Method, n.’ <<http://www.oed.com/view/Entry/383323>>.

⁶ Wintgens makes reference to Karl Popper and the concept of ‘theory dependence’ in the natural sciences, importing it into jurisprudence as part of his principle of coherence. See L. Wintgens, *Legisprudence: Practical Reason in Legislation* (Surrey: Routledge, 2012) p. 253 *et seq.*, and also the discussion *infra* at Chapter 4, section 2.3.3 (The principle of coherence (PC)).

⁷ K. Yeung, ‘Can We Employ Design-Based Regulation While Avoiding Brave New World?’ (2011) 3 *Law, Innovation & Technology* 1.

The term ‘code’ draws on Lessig’s seminal work on the concept of ‘code as law’.⁸ His original argument was that individuals are regulated not just by law, but also by three other regulatory ‘modalities’, namely social norms, the market, and architecture. In ‘cyberspace’, architecture is constituted by software, or code, as opposed to the physical architecture of the ‘real’ world. The ‘architects’ of that code therefore have significant power in cyberspace, and because of the far greater potency of code than the other modalities to shape what is and is not possible in that space, the architects of code therefore have disproportionate power within the digital realm. Lessig summarises his thesis by stating that ‘[a]rchitecture is a kind of law: it determines what people can and cannot do. When commercial interests determine the architecture, they create a kind of privatized law’.⁹

The debate has moved on somewhat since Lessig and his predecessor Reidenberg – whose analysis of *lex informatica* influenced Lessig’s work¹⁰ – first introduced these concepts. The literature was initially concerned mainly with the regulation of the amorphous cyberspace as a location that is ‘out there’, i.e. the Internet as a ‘place’ and a platform. The discussion has since evolved to consider on the one hand the code of individual and/or networked applications, and on the other code that facilitates data-driven services based on machine learning. Both forms of code are ‘algorithmic’, but the distinction between the two is an important one.¹¹

Broadly speaking, modern data-driven applications are concerned with the use of machine learning algorithms and ‘big data’ to facilitate automated classification and decision-making. Such systems can be unpredictable because they are based on the

⁸ L. Lessig, *Code: Version 2.0* (New York: Basic Books, 2006) *passim*.

⁹ *Ibid.*, p. 77.

¹⁰ J.R. Reidenberg, ‘Lex Informatica: The Formulation of Information Policy Rules through Technology’ (1997) 76 *Texas Law Review* 553.

¹¹ M. Hildebrandt, ‘Algorithmic Regulation and the Rule of Law’ (2018) 376 *Philosophical Transactions of the Royal Society A* 20170355, pp. 2–4.

processing of vast, multi-dimensional, contingent datasets using computer-generated models that are not readily susceptible to human interpretation.¹² This includes a wide range of applications, from classifying images for searching to the prediction of recidivism.¹³ Code-driven applications, on the other hand, are tied to the predetermined ‘if-then’ logical building blocks that underpin all traditional computational systems. Data-driven applications necessarily operate on a level above code-driven applications, and indeed are one type of application that requires the foundation of the latter to operate. The general-purpose computing infrastructure that gathers, transmits, and stores the data that machine learning algorithms process is ultimately based entirely on that general-purpose computing infrastructure. The issues raised by data-driven applications are distinct from those of code-driven applications, and although there are some overlaps, my focus is on the implications of *human-designed* code, rather than *algorithmically-processed* data.¹⁴

As is common in the literature, and taking into account the distinction just described, I will hereafter use ‘code’ interchangeably with ‘software’ and ‘architecture’, to refer generally to digital systems that have a regulatory effect on behaviour. Throughout the thesis the term is intended to be contrasted with law as a competing regulator.

‘Regulation’ in the context of the thesis straddles two of the definitions identified by Black: (i) the promulgation of rules by government (i.e. traditional laws

¹² E. Bayamlioglu, ‘On the Possibility of Normative Contestation of Automated Data-Driven Decisions’ in I. Baraliuc et al. (eds.), *Being Profiled: Cogitas Ergo Sum - 10 Years of Profiling the European Citizen* (Amsterdam University Press, 2018); J. Burrell, ‘How the Machine “Thinks”: Understanding Opacity in Machine Learning Algorithms’ (2016) 3 *Big Data & Society*.

¹³ On the latter, see J. Angwin et al., ‘Machine Bias’ (*ProPublica*, 23 May 2016) <<https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>>.

¹⁴ The digisprudential framework of affordance could be applied in the design data-driven applications, perhaps with some adaptation to account for specific issues that arise in that context, for example around transparency.

and regulatory instruments), and (ii) all mechanisms of social control affecting behaviour, of whatever kind from whatever source, whether intentional or not.¹⁵ The phrase ‘social control’ in the second of these might be viewed as including ‘technical control’ or ‘commercial control’. This notion of control is connected with the definition of ‘normativity’ that I adopt, which is to say any mechanism, legal or otherwise, through which behaviour is enabled or constrained. As Goldoni puts it, ‘code as law is normative in the sense that it regulates and guides human behaviour.’¹⁶ This meaning is similar to the concept of ‘governance’ in the regulatory literature.¹⁷ I discuss technological normativity in greater detail in Chapter 2.

By ‘legitimacy’, I refer to the idea that rules that govern behaviour ought to be created according to pre-existing standards that embody values of accountability, transparency, and contestability.¹⁸ I refine and deepen this idea in section 3 below and in subsequent chapters. Despite a large literature on software as both a target and a conduit of regulation, the question of legitimacy is one that has received only minimal attention with regard to the normative standards to which the designers who create code might be held.¹⁹ Few scholars have considered the question directly, and the treatment so far has focused more on ex post assessments of code regulation (the effects of its operation in the world) rather than on ex ante normative standards (questioning

¹⁵ See J. Black, ‘Critical Reflections on Regulation’ (2002) 27 *Australian Journal of Legal Philosophy* 1, p. 11 (as an aside, Hildebrandt suggests that Black’s definition has become the ‘canonical reference’. See M. Hildebrandt, ‘Radbruch’s Rechtsstaat and Schmitt’s Legal Order: Legalism, Legality, and the Institution of Law’ (2015) 2 *Critical Analysis of Law* 42, p. 61).

¹⁶ Goldoni, *supra* n. 1, p. 119. See also M. Hildebrandt, ‘Legal and Technological Normativity: More (and Less) than Twin Sisters’ (2008) 12 *Techné: Research in Philosophy and Technology* 169, p. 173 *et seq.*

¹⁷ C. Reed and A. Murray, *Rethinking the Jurisprudence of Cyberspace* (Cheltenham, UK: Edward Elgar Publishing, 2018) p. 140.

¹⁸ See for example J. Waldron, ‘Can There Be a Democratic Jurisprudence?’ (2009) 58 *Emory Law Journal* 675.

¹⁹ Goldoni, *supra* n. 1, pp. 123–125.

how it was, and ought to have been, produced). Although the former are a necessary and important element of oversight, the characteristics of computational legalism make necessary an additional focus on the application of ex ante standards during the production process. The reasons for this are discussed below in section 2, and in detail in Chapter 3.

1.2 Highlighting a tension

This is of course a legal thesis. I am, however, arguing explicitly that there is a body of normativity that operates separately from and in parallel with institutional law.²⁰ In doing so my aim is not to follow some cyberlibertarian arguments that welcome, and seek to validate, the usurping of the state by private producers of code.²¹ Indeed, it is quite the opposite. We are at risk of finding ourselves mired in a ‘Collingridge dilemma’, where by the time consensus has been reached on the need for change (e.g. regulation of technology and the business models that push its development in certain directions) conditions have become such that implementing any change is expensive, difficult, and time-consuming.²²

My aim therefore is first to acknowledge the reality of this predicament and then, by adopting a precautionary approach, to suggest ways in which we might guide

²⁰ R. Brownsword, ‘In the Year 2061: From Law to Technological Management’ (2015) 7 *Law, Innovation and Technology* 1, pp. 10–14; R. Mohr and F. Contini, ‘Reassembling the Legal: “The Wonders of Modern Science” in Court-Related Proceedings’ (2011) 20 *Griffith Law Review* 994, p. 998.

²¹ The now-classic expression of this perspective is J.P. Barlow, ‘A Declaration of the Independence of Cyberspace’ (1996) <<https://www.eff.org/cyberspace-independence>>. For another classic argument in this vein see D.R. Johnson and D.G. Post, ‘Law and Borders - The Rise of Law in Cyberspace’ (1995) 48 *Stanford Law Review* 1367.

²² M.T. Young, ‘Artifacts as Rules: Wittgenstein and the Sociology of Technology’ (2018) 22 *Techné: Research in Philosophy and Technology* 377; A. Thierer, ‘The Pacing Problem, the Collingridge Dilemma & Technological Determinism’ *Technology Liberation Front* (16 August 2018) <<https://techliberation.com/2018/08/16/the-pacing-problem-the-collingridge-dilemma-technological-determinism/>>.

the practice of design towards outcomes that are more legitimate, as defined according to the legal-theoretical frame that I later adopt. If designers are going to create code that regulates human behaviour, that code ought to be legitimate: '[w]hen technologies are always influencing human actions, we had better try to give this influence a desirable form.'²³ While I therefore agree with some of the cyberlibertarians' *descriptions* of code, I expressly disagree with their *normative* positions on what should flow from those characteristics.

One of the traditional counter-arguments to the cyberlibertarian position is that code is readily susceptible to regulation by law. Arguments to that effect – about the regulability of code – are well-heeded as far as they go, but they neither adequately encompass the characteristics of recent technological developments (e.g. blockchain applications) nor address the question of how code is in fact produced. Those scholars who have argued against the idea that code is hegemonic have tended to focus on the infrastructure of the Internet and the large platforms that own and operate it,²⁴ rather than on the individual digital artefacts that constitute our daily lives online. This code is often produced by smaller enterprises²⁵ who are less easy targets for traditional regulation and who may view the benefits of traditional compliance as being

²³ P.-P. Verbeek, 'Materializing Morality: Design Ethics and Technological Mediation' (2006) 31 *Science, Technology, & Human Values* 361, p. 370.

²⁴ Goldsmith and Wu, for example, focus on the physical networks that underpin the Internet, noting that they are owned by 'some of the most regulated companies on earth'. See J.L. Goldsmith and T. Wu, *Who Controls the Internet?: Illusions of a Borderless World* (Oxford: Oxford University Press, 2006) p. 73. As discussed above, we also saw this focus in the 'code as law' literature.

²⁵ G. Papadopoulos et al., 'Statistics on Small and Medium-Sized Enterprises' (European Commission, 2018) <https://ec.europa.eu/eurostat/statistics-explained/index.php/Statistics_on_small_and_medium-sized_enterprises>.

outweighed by the cost,²⁶ particularly when they lack dedicated legal departments or expertise (indeed, a great deal of code is produced by individuals or microbusinesses²⁷).

Such scholarship has also tended not to engage in depth with the role and practices of the *designer* as the creator of the code-based norms that constrain and enable behaviour. Again, another counterargument to the thesis I am advancing might be that designers, just like any other legal person, should be the subjects of traditional regulative processes and therefore that any illegality in the code they produce should be dealt with using traditional ex post legal processes. In the computational context this is necessary but insufficient: as we shall see below and in subsequent chapters, the ex ante legitimation of code in addition to ex post legal remedial measures is crucial because of its *sui generis* nature as a regulator. Its ruleishness, opacity, immediacy, immutability, and pervasiveness, coupled with the overarching problems of privatised production, mean that the stakes are both qualitatively and quantitatively higher than with other instances of problematic behavioural regulation that can be ameliorated by traditional legal processes, for example unfair terms in a contract or conflicting legislative norms. Whereas a traditional law that is improperly enacted (i.e. does not achieve legality, in Fuller's terms) can be deemed invalid *ab initio* and therefore ignored and/or struck down, the characteristics of code as a regulator admit of no such possibility: once it is 'promulgated', code's (il)legality has no bearing on its ability to execute and impose its normativity. From the moment of 'shipping', the code will operate as though it was legitimately 'enacted', even where this is manifestly not the case. There is, therefore, a crucial difference between invalid laws and 'invalid' code –

²⁶ See for example 'Study on the Economic Benefits of Privacy-enhancing Technologies (PETs)' (London Economics, 2010).

²⁷ See 'Developer Survey 2019' (*Stack Overflow*) <<https://insights.stackoverflow.com/survey/2019#work--company-size>> (purportedly the largest survey of developers in the world, demonstrating that a quarter of developers work in companies with fewer than 20 employees).

with the former, the ‘hermeneutic gap’ that exists between text and action allows for a space in which validity can be considered, whereas with the latter there is no such opportunity, either to arrest execution or (in many cases) even to observe the invalidity. This in turn connects with the question of ex post contest. If the nature and extent of the code’s invalidity cannot be observed, traditional mechanisms of legal redress cannot meaningfully be invoked. Ultimately, then, we fall into a trap if we assume that institutional law is capable of operating with its usual force where code is the subject of regulation.

Power is shifting away from the publicly-accountable legislator onto private actors who have pervasive control over the digital products and infrastructures that permeate contemporary life.²⁸ The ability of code to supplant law as the dominant normative enterprise,²⁹ and the privatised nature of the environments within which code is produced, raise the question of how to ensure that that new normativity is legitimate. As Lessig puts it, ‘if code is a lawmaker, then it should embrace the values of a particular kind of lawmaking.’³⁰ Hildebrandt echoes this, suggesting that ‘if we agree on the need for democratic procedures to regulate the enactment of legal normativity, technological normativity requires similar democratic legitimacy.’³¹ Designers are in effect legislating *already*, and there is no sign that in a neoliberal economy focused on ‘innovation’ that they will cease unilaterally to do so³² (and,

²⁸ R. Brownsword, ‘Lost in Translation: Legality, Regulatory Margins, and Technological Management’ (2011) 26 *Berkeley Technology Law Journal* 1321, p. 1324; M.J. Radin, ‘Regulation by Contract, Regulation by Machine’ (2004) 160 *Journal of Institutional and Theoretical Economics (JITE)* 142.

²⁹ Bayamlioglu and Leenes talk about law being ‘challenged as a normative enterprise’ when mechanisms other than reason define norms. Although they are focused on data-driven systems, the same holds true for other types of code. See Bayamlioglu and Leenes, *supra* n. 3, p. 11 *et seq.*

³⁰ Lessig, *supra* n. 8, p. 328.

³¹ Hildebrandt, ‘Legal and Technological Normativity’, *supra* n. 16, p. 176.

³² R. von Schomberg, ‘A Vision of Responsible Research and Innovation’ in R. Owen, J. Bessant and M. Heintz (eds.), *Responsible Innovation* (Chichester, UK: John Wiley & Sons, Ltd, 2013) p. 58;

simultaneously, we of course should not wish societally-beneficial innovation to be prevented). As lawyers we must grapple with this difficult reality, and seek to guide the work of designers, which will require a shift in discourse ‘from distribution to production and [thus a] focus on how the digital environment is created.’³³ Thankfully, a recent turn in the literature – particularly in the sphere of privacy – sees the focus shifting toward the direct consideration of design and the production of code.³⁴ This thesis contributes to that emerging debate by considering, from a legal-theoretical perspective on what constitutes legitimacy,³⁵ the expectations we should have of designers and enterprise in their anticipation of the effects of the code they produce.

1.3 Questions, aims, and methodology

From the above discussion of the context of the research, we can discern the following questions:

M. Hildebrandt and B.-J. Koops, ‘The Challenges of Ambient Law and Legal Protection in the Profiling Era’ (2010) 73 *The Modern Law Review* 428, pp. 456–457. Daly argues that ‘authorities’ forbearance from introducing ex ante regulation in situations where there are extant or foreseeable problems for user autonomy can also be explained by the influence of neoliberal trends promoting minimalist “light touch” regulation of private economic power.’ See A. Daly, *Private Power, Online Information Flows, and EU Law: Mind the Gap* (Oxford, United Kingdom; Portland, Oregon: Hart Publishing, 2016) p. 137.

³³ Goldoni, *supra* n. 1, p. 129. Interestingly, Young suggests that the tendency in sociological studies of technology has been precisely the reverse, namely focusing on production rather than use. See Young, *supra* n. 22.

³⁴ See for example W. Hartzog, *Privacy’s Blueprint: The Battle to Control the Design of New Technologies* (Cambridge, Mass: Harvard University Press, 2018); S. Gürses and J. van Hoboken, ‘Privacy after the Agile Turn’ in E. Selinger, J. Polonetsky and O. Tene (eds.), *The Cambridge Handbook of Consumer Privacy* (1st edn., Cambridge University Press, 2018). See also P. Nemitz, ‘Constitutional Democracy and Technology in the Age of Artificial Intelligence’ (2018) 376 *Philosophical Transactions of the Royal Society A* 20180089, arguing for a design perspective on the effects that artificial intelligence is having on constitutional democracy.

³⁵ Cf. A. Murray, ‘Looking Back at the Law of the Horse: Why Cyberlaw and the Rule of Law Are Important’ (2013) 10 *SCRIPTed* 310. Murray retains the orthodox position of viewing code as the subject of law.

- (1) How does code in fact regulate end-user behaviour?
- (2) To what extent are the characteristics of good law reflected in code as a regulator?

Following Gürses and van Hoboken, I use the term ‘end-user’ to draw attention to the individual user’s position at the *end* of the product design process, emphasising her relative lack of agency in shaping its outputs.³⁶

Taken together, the answers to these first two questions constitute a ‘diagnosis’ of the problem, which I refer to as computational legalism. This is the contribution of Chapters 2 and 3. From that descriptive analysis arise the following normative questions:

- (3) Can mechanisms for designing legitimate *legal* normativity be adopted to ensure the design of legitimate *technological* normativity?
- (4) What might such mechanisms look like in the design context?

The aspiration then is the governance of the design process and its products by constitutive principles that are informed by a legal-theoretical perspective on what makes normativity legitimate.

Compliance with substantive doctrinal law (data protection, intellectual property, contract, etcetera) is of course important, but as a general aim it overlooks (i) the *sui generis* nature of code as a regulator of behaviour (i.e. it overlooks computational legalism), and (ii) how the translation from textual norms to code-based norms invariably involves some level of modification of the former.³⁷ The precise nature of the reality envisioned by legal text is not reflected in the reality constructed by code, partly because law itself is (and arguably should be) vague,³⁸ and partly because the two modes

³⁶ Gürses and van Hoboken, *supra* n. 34, p. 581.

³⁷ Goldoni, *supra* n. 1, p. 129; Hildebrandt and Koops, *supra* n. 32, p. 452 *et seq.*

³⁸ T. Endicott, ‘Law Is Necessarily Vague’ (2001) 7 *Legal Theory* 379; C. Reed, ‘How to Make Bad Law: Lessons from Cyberspace’ (2010) 73 *The Modern Law Review* 903, p. 904 *et seq.*

of representing meaning (text and software code) are by nature very different, both because language is vague where code is precise³⁹ and because words require translation into behaviour whereas code is simultaneously ‘words and actions’.⁴⁰

This lack of one-to-one mapping of meaning is true not only of attempts to interpret and instantiate substantive (textual) legal norms in code but is also demonstrated in the unintended constellations of legal and non-legal effect that are continually being reified by digital artefacts.⁴¹ This is what van den Berg and Leenes have referred to as ‘techno-effects’,⁴² or the aggregate normative impact of a technology considered regardless of the designer’s intent or any legal impetus behind its use or design. They suggest that while much of the literature focuses on ‘techno-regulation’, i.e. the use of technology as a tool to effect legal norms, there has been insufficient consideration of the wider spectrum of techno-effects, including a-legal regulation. This gap in the literature is an important one, particularly given that ‘[t]he “regulatory” potential of technologies – in the broadest sense – is tremendous, and daunting, indeed.’⁴³ Not only can it be difficult to discern the *intention* of the designer, but so too is the line between intentional and unintentional normativity difficult to detect – ‘[t]he

³⁹ Columbia contrasts human language and programming languages thus: ‘Programming languages, as Derrida knew, are codes: they have one and one only correct interpretation (or, at the absolute limit, a determinate number of discrete interpretations). Human language practice almost never has a single correct interpretation [... t]he use of the term [programming] *language* to describe them is a deliberate metaphor, one that is meant to help us interact with machines [...]’. See D. Columbia, *The Cultural Logic of Computation* (Cambridge, Mass: Harvard University Press, 2009) p. 19. This ‘representationalism’ is discussed further in Chapter 3.

⁴⁰ B. Latour, ‘Where Are the Missing Masses? The Sociology of a Few Mundane Artifacts’ in W.E. Bijker and J. Law (eds.), *Shaping Technology/Building Society: Studies in Sociotechnical Change* (MIT Press, 1992) n. 1. This is discussed *infra* in section 2.1 and in Chapter 3.

⁴¹ B. van den Berg and R.E. Leenes, ‘Abort, Retry, Fail: Scoping Techno-Regulation and Other Techno-Effects’ in M. Hildebrandt and J. Gaakeer (eds.), *Human Law and Computer Law: Comparative Perspectives* (Dordrecht: Springer Netherlands, 2013).

⁴² *Ibid.*, p. 81.

⁴³ *Ibid.*, p. 83.

affected individual cannot discern which part of the normativity (as could be inferred from the output) is intentional and which part is merely spin-off in the form [of] unforeseen or secondary effects'.⁴⁴ Whereas law benefits from delay and processes of interpretation that permit application across heterogeneous circumstances,⁴⁵ the code in a digital artefact tends by nature toward a fixed configuration of normativity, rather than an interpretable standard, which will be applied with unqualified force in every case where the necessary computational conditions exist, regardless of any other relevant consideration. The challenge therefore is to require designers to ensure *ab initio* that the fixity of their code is as legitimate as it can be (recall the discussion in section 1.2 above about the difference between invalid laws and 'invalid' code). This means that code-makers are quasi-legislators, creating the norms to which the end-users of their products are subject. As Goldoni puts it,

on the one side, code can be a norm-enforcing technology, as has been outlined several times in the debate; on the other side, code can also be a *norm-establishing technology* as well.⁴⁶

If both law and code create norms, and we as a society have expectations about the legitimacy of the former, then we ought to expect similar standards from the latter. Code, however, is not law, it is only law-like; but it is precisely because of the ways in which it is *not* law that this kind of analysis is necessary: code can control behaviour more directly than can 'true' law, but simultaneously it lacks the latter's mechanisms of ex ante legitimation and ex post remediation.

The need is therefore all the greater for it to be legitimated *ab initio*, within the design process, and not only in the aftermath of a high-profile data breach or other

⁴⁴ Bayamlioğlu and Leenes, *supra* n. 3, p. 12. They refer to this phenomenon as 'normative opaqueness'.

⁴⁵ Endicott, *supra* n. 38, pp. 382–383; R. Dworkin, 'Law as Interpretation' (1982) 9 *Critical Inquiry* 179.

⁴⁶ Goldoni, *supra* n. 1, p. 118 (my emphasis).

scandal. The opacity of code means that only the most egregious, conspicuous, and high-stakes illegitimacies are ever likely to be exposed; this demonstrates the problem of a retrospective focus centred on the operation of code rather than its production.

Consider, for example, the ongoing controversy surrounding Facebook and the sharing of its users' personal data with third-party application developers, who subsequently used it to micro-target election advertisements online. The case is a complex (and evolving) mix of business ethics, democratic politics, and doctrinal law, but at its heart lie decisions made by designers and concretised in code: a now-deprecated version of Facebook's application programming interface (API)⁴⁷ allowed developers to access the data of 'friends' of the primary end-user, which enabled the large-scale data harvesting that facilitated the voter profiling that lies at the centre of the controversy. Most end-users would be wholly unaware that the code enabled this kind of behaviour until it was uncovered and publicised as part of a much larger public-interest journalistic enquiry.⁴⁸ This example demonstrates the problem that code's opacity poses for ex post remedial measures: the fact that it took the UK Information Commissioner's Office – an expert regulator – several months to investigate the nature of Facebook's systems suggests it is highly unlikely individual end-users can be expected to understand what is *actually* going on. As Pasquale suggests, '[i]t could take weeks to fully map the flow of data from something as simple as commenting on

⁴⁷ APIs enable otherwise unconnected software systems to communicate with one another, allowing the combination of systems with different specialities. Examples include the embedding of a mapping and GPS navigation system into a bicycle rental application or using a payment platform's API to enable the purchase of virtual assets within a game.

⁴⁸ For a technical overview of how Facebook's system worked and enabled the harvesting of end-users' data, see 'Investigation into the Use of Data Analytics in Political Campaigns - Investigation Update' (Information Commissioner's Office, 2018) s. 4.3.1. For one of the press stories which broke the scandal, see C. Cadwalladr and E. Graham-Harrison, 'Revealed: 50 Million Facebook Profiles Harvested for Cambridge Analytica in Major Data Breach' *The Guardian* (17 March 2018) <<http://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election>>.

Facebook'.⁴⁹ Furthermore, Facebook is quite clearly a prominent target for regulators, and its alleged role in election tampering means the case is of the greatest possible public interest.

The question remains, however, of the extent to which less significant code infelicities might be operating all around us that are never detected or remedied, because the scrutiny and impetus to investigate ex post are relatively minimal or simply absent. As Gürses and van Hoboken note, 'the ideological markers, pools of desirable knowledge and practices of technology *production* that bring these sets of [ex post] conditions forth and not others tend to go unquestioned.'⁵⁰ The effects of computational legalism make code resistant to the modulating effects of interpretation and ex post remedial measures that are more readily effective in the realm of traditional text-based law. It is clear, therefore, that in addition to those traditional ex post methods of redress, we should aim for ex ante code legitimacy. In light of these observations and the recommendations of a handful of scholars – including Goldoni, who I quoted at the beginning of the chapter – I adopt this alternative ex ante focus on design and the production of code-based artefacts.

The legal-theoretical analysis I adapt is based in part on Wintgens' theory of *legisprudence*, an aspirational framework that considers how legislators can achieve formal legitimacy in the process of creating new legislative norms.⁵¹ Legal theory has in general been concerned more with ex post judicial reasoning and interpretation than with the process of legislative creation; as Wintgens notes, 'the way law is created

⁴⁹ See his *The Black Box Society: The Secret Algorithms That Control Money and Information* (Cambridge: Harvard University Press, 2015) p. 144. For a set of fascinating maps of Facebook's internal algorithms and data flows visualised by a team of media theorists see 'Immaterial Labour and Data Harvesting' <<https://labs.rs/en/facebook-algorithmic-factory-immaterial-labour-and-data-harvesting/>>.

⁵⁰ Gürses and van Hoboken, *supra* n. 34, p. 580 (emphasis supplied).

⁵¹ L. Wintgens, 'Legisprudence as a New Theory of Legislation' (2006) 19 *Ratio Juris* 1. For an explanation and history of the term 'legisprudence', see Wintgens, *supra* n. 6, pp. 231–235.

through the process of legislation does not appear on the screen of the legal theorist'.⁵² Legisprudence, on the other hand, is specifically aimed at the *ex ante* process of legislative law-making. Alongside other relevant theories that consider what makes a good rule, such as Fuller's internal morality of law, legisprudence is therefore an interesting platform for carrying out *ex ante* assessments of other forms of normative rule-making. It aims to engender sensitivity to numerous factors in the creation of legislation that are presently lacking in the privately-ordered sphere of 'code-as-law'. By adapting and importing the principles of legisprudence into the design context I will provide a mechanism through which the creation of a normative order by and through code can be made (more) legitimate from the outset, thus limiting the negative effects of computational legalism.

The remainder of this chapter sets out the overarching narrative of the thesis in more detail, as well as sketching the proposed synthesis and introducing case study technologies which exemplify some of the problems identified. Later in Chapter 5 these will be discussed in practical terms, viewed through the lens of the digisprudential framework.

1.4 Legitimacy vs. 'compliance by design'

The thesis may appear superficially to be concerned with 'compliance by design' (CbD). Although the term is open to interpretation, I understand it to refer to compliance with specific fields of substantive doctrinal law, for example data protection or intellectual property. This is in line with the terminological usage of initiatives like 'privacy by design', expressed doctrinally in Recital 78 and Article 25 of the EU General Data Protection Regulation under the rubric of 'data protection by design'.⁵³

⁵² Wintgens, *supra* n. 51, p. 1.

⁵³ Regulation on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 85/46/EC (General Data Protection Regulation) 2016.

This is a clear example of orthodox technology regulation, where the focus is the regulation *of* software *by* substantive doctrinal law. From the perspective of the current thesis, this is a limited and inherently legalistic view that looks upon the law as a set of rules that is ‘just there’, to be passively observed and obeyed by the designer of code. It also narrows our focus, directing it away from the broader range of techno-effects that play as important (and indeed larger) a part in regulating behaviour as does legally-sanctioned code.⁵⁴ A perspective of code based solely on this understanding of ‘compliance by design’ is unsatisfactory, or at least incomplete, because it elides the very active role that designers play, consciously or otherwise, in the creation of such normative ‘reality’ *in* and *through* the code that they produce.⁵⁵

Broader views of ‘by design’ that are closer to the focus of the thesis do exist, however. For example, Nemitz refers to ‘the principles of democracy, rule of law and human rights by design’.⁵⁶ Similarly, Hildebrandt defines her concept of ‘Legal Protection by Design’ as

a way to ensure that the technological normativity that regulates our lives: first, is compatible with enacted law, or even initiated by the democratic legislator; second, can be resisted; and third, may be contested in a court of law.⁵⁷

One can see how Nemitz’s and Hildebrandt’s concepts include more fundamental issues than simply compliance with a given area of substantive doctrine. Nevertheless, to avoid the possibility of causing confusion, I will avoid attaching yet another concept to the already over-burdened phrase ‘by design’. Hildebrandt’s and Koops’ previous formulation of ‘ambient law’ comes close to the idea at hand, where what matters is not (just) compliance with the substantive law, but the kinds of constitutional

⁵⁴ van den Berg and Leenes, *supra* n. 41.

⁵⁵ The creation of reality by artefacts is discussed in Chapter 2.

⁵⁶ Nemitz, *supra* n. 34 *passim*.

⁵⁷ M. Hildebrandt, *Smart Technologies and the End(s) of Law: Novel Entanglements of Law and Technology* (London: Edward Elgar Publishing, 2015) p. 218.

safeguards that law as an enterprise is expected to provide regardless of its substantive content. They describe the difference thus:

These requirements [democratic enactment and contestability] constitute the difference between our concept of Ambient Law on the one hand and the technological enforcement of legal rules on the other. Ambient Law represents the technological articulation of legal norms as a form of democratic legislation, requiring both democratic participation and built-in safeguards that guarantee the contestability of the decisions made within the legal-technical infrastructure.⁵⁸

While the authors are concerned with the reflection of state-sourced law in code, it can equally be said that code which embodies normativity that is not state-sourced ought also to embody ‘safeguards’ in order for it to be legitimate. In this context, where commercial enterprise is the source of the normativity, the requirement of democratic participation in the design of code is perhaps unlikely to be achievable by smaller enterprises with limited resources.⁵⁹ Initiatives connected with this goal include constructive technology assessment,⁶⁰ value sensitive design,⁶¹ and ideation.⁶² I explicitly do not consider such approaches in the thesis, because their focus tends to individualise the idea of ‘constitutional’ standards that I am aiming for, focusing on the practices of an individual designer and how they impact on a given design project,

⁵⁸ Hildebrandt and Koops, *supra* n. 32, p. 445. See also M. Hildebrandt, ‘A Vision of Ambient Law’ in R. Brownsword and K. Yeung (eds.), *Regulating Technologies: legal futures, regulatory frames and technological fixes* (Oxford; Portland, Or: Hart, 2008).

⁵⁹ Papadopoulos et al., *supra* n. 25.

⁶⁰ Verbeek, *supra* n. 23, p. 375 *et seq.*

⁶¹ B. Friedman, ‘Value-Sensitive Design’ (1996) 3 *interactions* 16.

⁶² E. Luger and M. Golembewski, ‘Towards Fostering Compliance by Design; Drawing Designers into the Regulatory Frame’ in M Taddeo and L Floridi (eds.), *The Responsibilities of Online Service Providers* (Springer International Publishing, 2017); B. Friedman and D. Hendry, ‘The Envisioning Cards: A Toolkit for Catalyzing Humanistic and Technical Imaginations’, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA: ACM, 2012) <<http://doi.acm.org/10.1145/2207676.2208562>>.

rather than at the commercial institutional level. Such initiatives seek to legitimise a design by the bare fact of having involved stakeholders in decisions as to its *substantive* characteristics. By contrast, the ‘constitutional’ perspective of digisprudence comes earlier than the question of whether stakeholders have had their views taken into account. Design *for* all need not require design *with* all;⁶³ the characteristics of legitimacy that I discuss are formal rather than substantive and ought to be present in all end-user-facing technologies regardless of their substantive purpose. In that vein, then, we can say that digisprudence is to participatory design approaches as legisprudence is to the democratic process; they are separate but complementary elements in the norm-creation process.

⁶³ A. Pols and A. Spahn, ‘Designing for the Values of Democracy and Justice’ in J. Van Den Hoven, P.E. Vermaas and I. van de Poel (eds.), *Handbook of Ethics, Values, and Technological Design: Sources, Theory, Values and Application Domains* (Dordrecht: Springer, 2015) p. 351.

1.4.1 Normative relationships online

To clarify the location of the enquiry further, we can visualise the normative relationships in the digital sphere using the vertical model below:⁶⁴

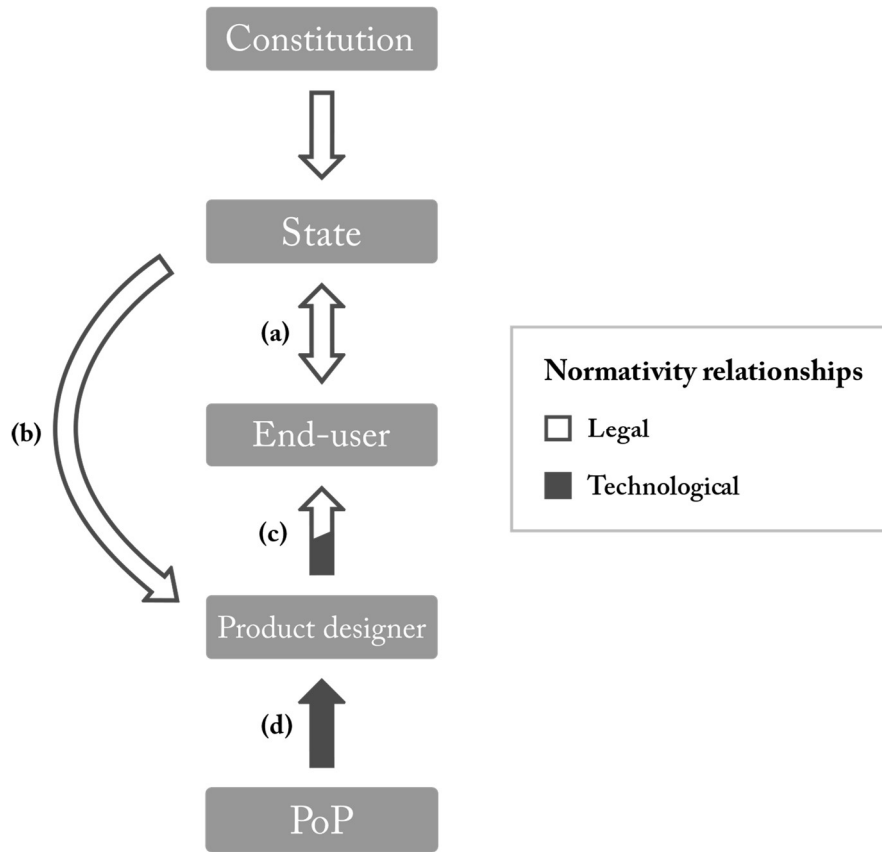


Figure 1: Legal and technological normativity relationships

Relationship (a) represents the classic compact between the citizen end-user and the state (the latter being bound by a constitution), where democratic participation results in the state's promulgation of legal norms through both that relationship and relationship (b). The latter represents the traditional understanding of 'compliance by design', where the state promulgates legal normativity in the form of substantive

⁶⁴ This model is adapted from the triangular diagram in M. Birnhack and N. Elkin-Koren, 'The Invisible Handshake: The Reemergence of the State in the Digital Environment' [2003] *Virginia Journal of Law & Technology* para. 119 *et seq*, adding the constitution, technological normativity, and the 'programmer-of-the-programmer' (discussed *infra* in section 1.4.1.2).

doctrinal laws that tell designers (and enterprise generally) of their duties in the creation of products and services. We have seen already how this relationship differs from the focus of the thesis.

1.4.1.1 The commixture of law and code

The thesis is focused on relationships (c) and (d). In relationship (c), the product designer imposes behavioural constraint through a mix of legal and architectural normativity. The legal normativity in this relationship can flow from public-order norms (legislation of various forms) on the one hand, or private-order contractual norms on the other. These will be operationalised by (i) the traditional force of law, operating outside the design of the artefact, (ii) their implementation in and through that design, or (iii) a mixture of the two. In scenarios (ii) and (iii), code complements law.⁶⁵ Examples of this include encryption used to implement data protection requirements (which flow from a public-order norm in relationship (b)), or a firewall containing rules that prevent an employee's computer from accessing social media websites (which flow from a private contractual norm).

Crucially, while code can complement legal norms, it can also supplant them altogether.⁶⁶ Separately from these architectural implementations of public or private legal norms, code can also implement normativity that is *purely* technological, which is to say rules that enable and constrain behaviour outside of any legal requirement, public or private, to do so. Here, norms are created, intentionally or otherwise, that have a regulative effect on end-users' behaviour.⁶⁷

⁶⁵ This is what Leenes calls 'state endorsed techno-regulation'. See R. Leenes, 'Framing Techno-Regulation: An Exploration of State and Non-State Regulation by Technology' (2011) 5 *Legisprudence* 143, p. 160.

⁶⁶ Leenes terms this 'non-state techno-regulation' (*ibid.*). See also Asscher, *supra* n. 4, pp. 67, 69 and Radin, *supra* n. 28.

⁶⁷ Goldoni, *supra* n. 1, p. 118.

Whether or not these assemblages of code rules aim explicitly to instrumentalise legal norms, they by definition exist separately from the law-system's⁶⁸ corpus of rules. Understood traditionally, the legal force of the data protection statute or the employment contract applies regardless of either instrument's implementation in or through code. But a corollary arises from this: it is precisely in the separateness of the two mechanisms of regulation that the *architectural* force of code, which implements some form of (potentially legally-significant) normativity, is able to 'supplant the legal infrastructure of the state'.⁶⁹ Whereas the data protection statute or employment contract 'hovers', waiting to be interpreted, complied with, and perhaps declared (in)applicable by a court, code simply goes ahead and enforces some alternative configuration of behavioural constraint which might not comport with either the substantive law (the specific statute, contract, or the whole corpus of legal rules) or, if its rules were not designed to implement a specific legal norm, the standards of legitimacy according to which I argue all substantive behaviour-constraining norms should be made.

1.4.1.2 The programmer of the programmer

Relationship (d) exists between what Vismann and Krajewski term the *programmer of the programmer*⁷⁰ ('PoP') and the product designer. The PoP designs the tools that the product designer in turn uses to create the products and services that are ultimately

⁶⁸ Both Schmidt and Reed discuss the concept of a 'law-system quality' in relation to public norms being better designed to fit technology (top-down regulation, or relationship (b) in Figure 1, *supra*). See A. Schmidt, 'Radbruch in Cyberspace: About Law-System Quality and ICT Innovation' (2009) 3 *Masaryk University Journal of Law and Technology* 195 and Reed, *supra* n. 38, respectively. By contrast, I am concerned with bottom-up instantiations of normativity (i.e. relationships (c) and (d) in Figure 1, *supra*).

⁶⁹ Radin, *supra* n. 28, p. 143.

⁷⁰ C. Vismann and M. Krajewski, 'Computer Juridisms' [2007] *Grey Room* 90. For an earlier discussion that alludes to a similar concept see J. Weizenbaum, *Computer Power and Human Reason: From Judgment to Calculation* (San Francisco: Freeman, 1976) p. 100 *et seq.*

destined for the end-user. Situated at a more ‘constitutional’ level of the product design process, the decisions made by the PoP fundamentally impact on what the product designer can and cannot do. In this way, the PoP has a crucial power to define the rules of the design game before it even begins. This idea of ‘technological constitutionalism’, which I link with Hart’s foundational concept of primary and secondary rules,⁷¹ suggests one locus for the operationalisation of formal principles that can constrain the substantive design of code to encourage legitimacy. This idea will be discussed again in Chapter 2, where it is described in greater detail,⁷² and Chapter 6, where it is suggested as a way of implementing the ‘constitutionalism’ of the digisprudential framework.⁷³ For now, I continue mapping out the overarching skeleton of the thesis, to be fleshed out more fully in later chapters.

2. Code is law(-like)

As mentioned above, a core element of Lessig’s thesis is that end-users’ behaviour is regulated by four modalities: law, social norms, the market, and architecture. His fear was that, given the power of code to define the rules of behaviour in cyberspace, and given the inherent flexibility of designers to choose those rules,⁷⁴ the risk arises that they might be captured by state interests mandating (through relationship (b)) backdoors and other anti-civil liberty measures.⁷⁵ His general prescription to avoid this risk is a culture of transparency, including actual transparency, in the form of open source code.⁷⁶ Mayer-Schönberger suggests that Lessig’s perspective is too dependent on free market orthodoxy, and that a fuller idea of how code regulates requires greater

⁷¹ H.L.A. Hart, *The Concept of Law* (2nd edn., Oxford: Clarendon Press, 1994) p. 91 *et seq.*

⁷² Chapter 2, section 3.6 (Technological constitutionalism).

⁷³ Chapter 6, section 2 (The programmer of the programmer).

⁷⁴ Lessig, *supra* n. 8, p. 32.

⁷⁵ *Ibid.*, pp. 79–80.

⁷⁶ *Ibid.*, ch. 5.

sensitivity to the social embeddedness of technological development.⁷⁷ He argues in favour of a less deterministic approach, and the integration of science and technology studies (STS) perspectives. This is echoed by Cohen, who argues that without the insights of STS ‘one cannot explain how code regulates.’⁷⁸ In any event, Lessig’s analysis does not include such perspectives, nor does it include the connected topic of design practice (all of which is the subject of Chapter 2). Here lies a point of departure: whereas Lessig is interested in the abuse of power by the state, my primary interest is in the abuse of the power of private enterprise in regulating end-user behaviour, and in how design practice can be shaped to avoid this.

But the question remains – is code *law*? Vismann and Krajewski refer to the ‘structural homologies’ between law and computers.⁷⁹ Code instantiates law in technological artefacts, and while those artefacts too are constituted to some extent by law (e.g. through contracts, intellectual property, etcetera), the relationship is lopsided. There exists an inherent ‘hermeneutic gap’ between the legal norm printed on the page and its instantiation in the physical world via interpretation and behavioural change. In the computational context, law is not nearly as powerful as we might suppose, because it is subject to the very medium it is attempting to regulate, and the immediacy and instrumental power of that medium and the ‘sovereignty’ of the designer in shaping its effects tip the balance against law as hegemonic regulator. The written law is rendered ‘a paper dragon in the age of the “digital tsunami”’,⁸⁰ with the social and rhetorical power of legal fictions making way for the representationalism of ‘digital virtuality’, whereby reality is constituted by and through the machine.⁸¹ Adjudication

⁷⁷ V. Mayer-Schönberger, ‘Demystifying Lessig’ [2008] *Wisconsin Law Review* 713.

⁷⁸ J.E. Cohen, *Configuring the Networked Self: Law, Code, and the Play of Everyday Practice* (Connecticut: Yale University Press, 2012) p. 27.

⁷⁹ Vismann and Krajewski, *supra* n. 70, p. 92.

⁸⁰ Hildebrandt and Koops, *supra* n. 32, p. 440.

⁸¹ Vismann and Krajewski, *supra* n. 70, p. 92.

is thus potentially collapsed into obedience,⁸² since the rule in the code also represents reality for the end-user. Representationalism is a key theme in legalism and is discussed in greater detail in Chapter 3.⁸³

Fuller defines law as ‘the enterprise of subjecting human conduct to the governance of rules’.⁸⁴ As Chapter 2 will demonstrate, code quite clearly subjects human conduct to the governance of rules; rules which increasingly constitute the ‘terms and conditions of existence and action’.⁸⁵ They may not be rules as we commonly understand them,⁸⁶ but they are designed by humans with a purpose in mind, and should therefore be subject to scrutiny as to their legitimacy. As Lessig puts it, ‘[a]rchitecture is a kind of law: it determines what people can and cannot do. When commercial interests determine the architecture, they create a kind of privatized law.’⁸⁷ The power that inheres in those who decide on that privatisation is significant:

*The quasi-sovereign power of the computer engineer’s code stems from the ease by which posing, implementing, and applying a norm are achieved in technology compared with the cumbersome procedures that legal code must pass through. The swift effectiveness of a technological code, which cannot, when seen through legal eyes, appear as anything other than uncanny, renders any possible competition between law and computer pointless.*⁸⁸

Architectural constitutions supplant legal constitutions; code is not just law-like, rather it is both *more* and *less* than law. As Chapter 2 will demonstrate, it is *more* than law

⁸² Z. Bańkowski and B. Schafer, ‘Double-Click Justice: Legalism in the Computer Age’ (2007) 1 *Legisprudence* 31, p. 48.

⁸³ Chapter 3, section 3.2 (Representationalism).

⁸⁴ L.L. Fuller, *The Morality of Law* (Yale University Press, 1977), *et passim*.

⁸⁵ G. Longford, ‘Pedagogies of Digital Citizenship and the Politics of Code’ (2005) 9 *Techné: Research in Philosophy and Technology* 68, p. 71.

⁸⁶ I discuss the question of code as rules in Chapter 4, section 3.3.2.3 (Does code contain rules *per se?*).

⁸⁷ Lessig, *supra* n. 8, p. 77.

⁸⁸ Vismann and Krajewski, *supra* n. 70, p. 93 (my emphasis).

because of the instrumental power of design to constitute and regulate end-user behaviour. But it is also simultaneously *less* than law, because as Chapter 3 explains it lacks the normative mechanisms designed to keep its textually-bound sister in check. This is what Hildebrandt points to when she says that ‘technologies that are constitutive for [sic] our interactions may enforce compliance beyond anything that a written law can achieve.’⁸⁹ It is precisely because architectural constitutions are not law *per se*, but nevertheless have law-like power to regulate, that it is necessary to instantiate the sorts of constitutional protections I am concerned with. While code constitutions are *not* law as traditionally defined, if we take a legal pluralist⁹⁰ perspective we can identify, through a comparison of the regulative aspects of traditional law and code, which of the checks and balances we expect to be present in the former are absent in the latter.

Whereas traditional regulative norms derive their legitimacy from the institutions and traditions of the rule of law within constitutional democracy, code-based norms do not necessarily have the same democratic connection. As Winner puts it, ‘technological innovations are similar to legislative acts [which require] the same careful attention one would give to the rules, roles, and relationships of politics.’⁹¹ As will become clear in Chapter 3, this kind of attention is precisely what traditional legalism prohibits. Whereas legal normativity *invites* the citizen to comply (she always has the notional option to interpret the norm, contest it, or to ignore it entirely), technological normativity can make compliance a necessity, either in the form of imposing a response to a circumstance or by constituting at the outset all the courses

⁸⁹ Hildebrandt, ‘Legal and Technological Normativity’, *supra* n. 16, p. 178.

⁹⁰ M.A.C. Dizon, ‘From Regulating Technologies to Governing Society: Towards a Plural, Social and Interactive Conception of Law’ in H.M. Morgan and R. Morris (eds.), *Moving Forward: Tradition and Transformation* (Cambridge Scholars Publishing, 2011). Dizon argues that ‘[w]hen Lessig uses his four modalities of control to describe the normative orders of cyberspace, he is in fact describing the condition of legal pluralism in the ICT field’.

⁹¹ L. Winner, ‘Do Artifacts Have Politics?’ [1980] *Daedalus* 121, p. 129.

of action that the end-user can possibly take. The fact that code is not ‘law’ *per se* is not relevant; as Fuller demonstrates in his discussion of a college dormitory’s parietal rules, law-systems exist in many contexts that have no explicit or implicit connection with the state;⁹² what matters is when they ‘subject human conduct to the governance of rules.’⁹³ As Le Sueur puts it in relation to automated public administration, ‘we should treat “the app” (the computer programs that will produce individual decisions) as “the law”... [i]t is this app, not the text of the legislation, that will regulate the legal relationship between citizen and state in automated decision-making.’⁹⁴ Precisely because of the supreme efficacy with which code achieves this regulation, what Lessig calls ‘perfect control’,⁹⁵ it is imperative that the creators of private code are, like public law-makers, constrained by *ex ante* standards that ensure both legitimacy during operation and the possibility of *ex post* remediation.

2.1 The design of behavioural constraint

How is behaviour in practice enabled and constrained by code? Chapter 2 considers three main theoretical perspectives on this question: affordance, inscription, and technological mediation. To summarise these, an artefact’s affordances are the ways in which it can be used by a particular end-user, determined according to the characteristics of both the artefact and the individual.⁹⁶ Affordances can (and, depending on the incentives involved, ought to) be signified to the end-user, so that she can understand what her behavioural possibilities are. By contrast with the

⁹² Fuller, *supra* n. 84, p. 125 *et seq.*

⁹³ *Ibid.*, p. 124 *et seq.*

⁹⁴ A. Le Sueur, ‘Robot Government: Automated Decision-Making and Its Implications for Parliament’ in A. Horne and A. Le Sueur (eds.), *Parliament - Legislation and Accountability* (Oxford; Portland, Oregon: Hart, 2016) p. 201.

⁹⁵ Lessig, *supra* n. 8, p. 4.

⁹⁶ D.A. Norman, *The Design of Everyday Things* (Cambridge, Mass; London, UK: MIT Press, 2013) p. 11.

enablement of affordance, the idea of disaffordance – which builds on Lessig’s concept of ‘architectures of control’⁹⁷ – is concerned with the conscious choice of a designer to embed mechanisms within the artefact which ‘enforce or restrict certain user behaviour’.⁹⁸ Inscription is the idea of embodying a particular ‘story’ in the design of the artefact, dictating the end-user’s behaviour by defining a narrative (of which affordances are the building blocks) of what she can and should do with it.⁹⁹ Taken in aggregate, (dis)affordances and inscriptions are the building blocks of technological mediation, the postphenomenological theory concerned with how the end-user’s perception and actions, and in turn her agency, are affected by the artefact.¹⁰⁰

The attributes of an artefact are designed with a particular class of end-user in mind, and so the code’s (dis)affordances, inscriptions, and mediations are all fundamentally affected by the choices made by the designers who produce them. Although some effects are emergent or open to (re)interpretation and/or resistance by end-users, it is nevertheless the case that design choices embed path-dependent ‘programs of action’¹⁰¹ in artefacts, thus highlighting the significant normative power that inheres in their creators. When a designer embeds (dis)affordances in the design of her artefact, she affects what it is possible to do with that artefact, either expanding or contracting those possibilities.

This design perspective is complemented in Chapter 3 by a legal-theoretical perspective that considers the characteristics of code from the perspective of legalism.

⁹⁷ Lessig, *supra* n. 8, ch. 4.

⁹⁸ D. Lockton, ‘Architectures of Control in Product Design’ [2006] *Engineering Designer: The Journal of the Institution of Engineering Designers* 28.

⁹⁹ M. Akrich, ‘The De-Description of Technical Objects’ in W.E. Bijker and J. Law (eds.), *Shaping Technology/Building Society: Studies in Sociotechnical Change* (MIT Press, 1992) p. 208; B. Latour, ‘The Berlin Key or How To Do Words with Things’ in P. Graves-Brown (ed.), *Matter, Materiality and Modern Culture* (London: Routledge, 2000).

¹⁰⁰ P.-P. Verbeek, *What Things Do: Philosophical Reflections on Technology, Agency, and Design* (Penn State Press, 2005) ch. 3.

¹⁰¹ Latour, *supra* n. 40.

2.2 Computational legalism

One of the central problematics of code from a legal-theoretical perspective is its ‘ruleishness’, meaning its application of defined rules in all instances where certain fixed conditions, specified in the code itself, obtain.¹⁰² In the technical context this is of course a major benefit: even the most complex body of rules can be expected to execute in pre-determined ways under what are ideally precisely-defined and controlled conditions, giving a notional predictability that facilitates rapid incremental innovation that has played a central role in the development of modern technological society.

In the legal context, however, the rote application of rules is undesirable, at least in a society built around the ideals of democracy and the concept of legality. Linked with the Kantian categorical imperative, legalism is the jural equivalent of software code’s ruleishness. Although it has more than one form in the literature, the conception that I adopt, closely connected with legal positivism,¹⁰³ is seen as an ideology under which rules and the strict adherence to them are the proper fundamentals of social ordering.¹⁰⁴ That the state defines what is legal is enough to legitimise the substance of the legal norms it chooses to declare; in constituting the field of play (the legal system), the state legitimises *de facto* that which it consequently promulgates as the rules of the game. Constitutive facts (natural laws, the social contract/constitution, or a mix of these) operate prospectively to legitimise any subsequent act of the sovereign.¹⁰⁵ The citizen is given the imperative to ‘not think about it’; the rule is ‘just

¹⁰² J. Grimmelmann, ‘Regulation by Software’ (2005) 114 *The Yale Law Journal* 1719.

¹⁰³ See Z. Bańkowski and N. MacCormick, ‘Legality without Legalism’ in W. Krawietz et al. (eds.), *The Reasonable as Rational? On Legal Argumentation and Justification; Festschrift for Aulis Aarnio* (Berlin: Duncker & Humblot, 2000) and J.N. Shklar, *Legalism* (Harvard University Press, 1964) p. 7.

¹⁰⁴ There is also a conceptual similarity to Dworkin’s ‘plain fact view’ of the law, where all that matters in adjudication are extant historical legal materials. See R. Dworkin, *Law’s Empire* (Cambridge, Mass: Belknap Press, 1986) p. 31, *et seq.*

¹⁰⁵ Wintgens, *supra* n. 6, chs. 5–6.

there' and she need only act in accordance with it as written,¹⁰⁶ since by virtue of those constitutive facts the pronouncement of the sovereign is 'imputed to [the people], as if they were its author.'¹⁰⁷ As an outlook, then, legalism tends towards a 'narrow governance of rules, unleavened by the principled approach to interpretation.'¹⁰⁸ This simplicity implies the notional possibility of abuse: the prioritisation of heteronomy militates against critical reflection and the application of other normative principles of legality that are aspirational characteristics in a democracy. The freedom of the citizen to interpret is seen as a crucial aspect of legality, without which rules become 'implements of tyranny' and legalism a 'vice of narrow governance'.¹⁰⁹

From this brief summary of legalism (the concept is expanded on in detail in Chapter 3¹¹⁰), one can begin to appreciate how code can exemplify these characteristics.¹¹¹ Even in the most tyrannical state there is space to interpret, and perhaps to disobey – the hermeneutic gap between the text of a norm on the page and its translation into behaviour in the world makes this at least a notional possibility. In the environments where code is designed, however, the elision of that gap is not only easy to do but is entirely standard, not necessarily through malice or intentional obfuscation (although they are certainly a problem¹¹²), but simply by the underlying characteristics of code, which by nature presents norms to the end-user that 'just are'. Even where the code does allow for choice via changeable settings, the default

¹⁰⁶ Z. Bańkowski, 'Don't Think About It: Legalism and Legality' in M.M. Karlsson, Ó. Páll Jónsson and E.M. Brynjarsdóttir (eds.), *Rechtstheorie: Zeitschrift für Logik, Methodenlehre, Kybernetik und Soziologie des Rechts* (Berlin: Duncker & Humblot, 1993).

¹⁰⁷ Wintgens, *supra* n. 6, p. 208.

¹⁰⁸ Bańkowski and MacCormick, *supra* n. 103, p. 194.

¹⁰⁹ *Ibid.*

¹¹⁰ Chapter 3, section 2 (What is legalism?).

¹¹¹ Bańkowski and Schafer, *supra* n. 82.

¹¹² See generally Pasquale, *supra* n. 49.

configuration of code tends to be seen by end-users as ‘a natural and immutable fact’.¹¹³ The hermeneutic gap is thus closed, or at least significantly narrowed, because the ‘text’ of the ‘rule’ (the source code) constitutes directly the geography of the artefact: they are not just isomorphic, they are one and the same. Unlike traditional law, whose ‘carrier’ has hitherto been the inherently passive medium of text, software code allows us to, in Latour’s words, ‘conceive of a text (a programming language) that is at once words and actions’.¹¹⁴ This potentially represents the apex of legalism: the normative collapses into the descriptive (what was once *requested* becomes simply what *is*), and there is no choice but to obey the rule as it is expressed by the designer, much less to view and contest it, since it by definition constitutes empirical as well as legal and technological reality.¹¹⁵ The characteristics of computational legalism – ruleishness, opacity, immediacy, immutability, and pervasiveness, each compounded by privatised production – mean that in many cases code is simultaneously more powerful and less adaptable than a law-system that is built around the characteristics of delay, flexible interpretation, and ex post remediation. Code is thus simultaneously *more*, and *less*, than law.

¹¹³ Goldoni, *supra* n. 1, p. 128. Boyle also hinted early on at this ‘legalistic’ nature of code, noting that ‘[t]he technology appears to be “just the way things are”; its origins are concealed, whether those origins lie in state-sponsored scheme or market-structured order, and its effects are obscured because it is hard to imagine the alternative.’ See J. Boyle, ‘Foucault in Cyberspace: Surveillance, Sovereignty, and Hardwired Censors’ (1997) 66 *University of Cincinnati Law Review* 177, p. 205.

¹¹⁴ Latour, *supra* n. 40, n. 1.

¹¹⁵ Bańkowski, *supra* n. 106; Bańkowski and Schafer, *supra* n. 82. Representationalism is a key element of the legalistic outlook – see Wintgens, *supra* n. 51, p. 5. Hildebrandt characterises this qualitative difference in her discussion contrasting architectural ‘constitutive rules’ (which define the scope of possible behaviour ex ante) and ‘regulative rules’ (which are akin to tradition laws that require to be interpreted to have any effect). See Hildebrandt, ‘Legal and Technological Normativity’, *supra* n. 16, p. 172 *et seq.*

3. Aspiring to legitimacy in code

If the rote rule-bound heteronomy of legalism is sited at one end of a spectrum, at the other end can be found the aspirational concept of *legality*, which aims to maintain a connection between the normative construct of law as a system of behavioural constraint and the legitimising principles that underlie the exercise of sovereign power in constitutional democracies. While legality is not a wholly settled concept, it has received some treatment from technology law scholars, and has a theoretical pedigree that includes influential analyses that fit well with the aspirational normative approach I adopt. Legality is considered to be of fundamental importance in constitutional democracies; Bańkowski goes so far as to say it is ‘something worth living for; something worth dying for.’¹¹⁶ Hildebrandt defines legality by what for her it is not: legal certainty, ‘justice’, and expediency on their own are insufficient; the characteristic of legality also encompasses the rule of law and the binding of the sovereign to constitutional rules in its wielding of legislative power.¹¹⁷ For Brownsword, legality is about human dignity and the creation and maintenance of conditions that ‘make moral community possible’. Legality, then, is not just about the substance of regulation, but also its form.¹¹⁸ This idea of purpose binding speaks to the ex ante, ‘constitutional’ nature of the analysis I am advancing. Through the guidance of designers’ *production* of technological normativity, we can help ensure that the negative outcomes toward which computational legalism tends are minimised as far as possible.

3.1 From *operation* to *production*

I have already mentioned briefly the importance of widening our focus to include the production of code, in addition to the orthodox ex post assessment of its operation.

¹¹⁶ Bańkowski, *supra* n. 106, p. 45.

¹¹⁷ Hildebrandt, *Smart Technologies*, *supra* n. 57, pp. 157–158.

¹¹⁸ Brownsword, ‘Lost in Translation’, *supra* n. 28.

This relates to the definitions of legality just set out – one can appreciate the relevance of the ‘design’ of a rule to the question of whether it meets those standards. Whereas legalism looks only to sources to discern validity, legality is something altogether more reflexive and rational,¹¹⁹ seeking evidence of certain requirements in the rule-making process. There is a clear alignment here between this view of legality and the shift in the literature towards design thinking I mentioned above.

3.1.1 Fuller’s internal morality of law

Exploring the legal theory further, this idea of rule production connects with Fuller’s seminal theory of the *internal morality of law*, which he argues is maintained by upholding his eight principles of legality.¹²⁰ These principles provide an underlying substrate which is necessary for the creation of good substantive laws, regardless of how reasonable it may be to dispute their substantive content – their ‘external morality’. I consider the principles and their connection to code in greater detail in Chapter 4.¹²¹ One question that several of the principles point towards is how best to *design* a legal norm, regardless of what its external morality is or ought to be. Indeed, Fuller uses the language of design on various occasions, referring to law-making as a ‘craft’¹²² and to the eight principles as ‘those laws respected by a carpenter who wants the house he builds to remain standing and serve the purpose of those living in it.’¹²³ His distinction between the internal and external moralities of law is connected with what Goldoni refers to as the difference between ‘input’ and ‘output’ reasons for decision-making,¹²⁴ and one can also see a connection to Hart’s theory of primary and secondary rules,

¹¹⁹ Bańkowski and Schafer, *supra* n. 82, pp. 31–32.

¹²⁰ Fuller, *supra* n. 84, ch. 2.

¹²¹ Chapter 4, section 2.2 (Fuller’s internal morality of law).

¹²² Fuller, *supra* n. 84, pp. 43, 156.

¹²³ *Ibid.*, p. 96.

¹²⁴ See Goldoni, *supra* n. 1, p. 127, citing J. Waldron, ‘The Core of the Case against Judicial Review’ (2006) 115 *Yale Law Journal* 1346.

which separate substantive ordinances from the rules which set out how they should be created, modified, and extinguished.¹²⁵ I discuss these concepts in greater detail at various points later in the thesis. For now, the second significant theory that I draw upon is Wintgens' *legisprudence*.

3.1.2 Wintgens' legisprudence

Wintgens places an even stronger focus than Fuller on the characteristics that proposed norms ought to have, and how guidance of their formal characteristics can have a bearing on their legitimacy. The theory is built around the idea that upholding individuals' subjective notions of freedom ought to be a guiding principle of both politics and law, and that any limitation on that freedom by rules (legislation) is only legitimate if it is justified (a standard met according to the requirements of the legisprudential framework of principles).¹²⁶ While fidelity to rules remains a necessary part of the legal order, this fidelity is via a 'weak' legalism which, unlike the stronger form described above (and in Chapter 3¹²⁷), requires those rules to be formulated according to principled *ex ante* standards and not simply through the exercise of the sovereign's raw power. Following rules thus becomes acceptable because those rules, if they have been legitimated by the application of the principles of legisprudence, cannot be arbitrary, and are otherwise legitimated by the framework. The legisprudential threshold of justification is met when the cumulative requirements of its four principles are taken into account in the design of a new legislative norm. In summary, the principles are concerned with whether a binary rule is desirable, whether the proposed norm is proportionate to the issue the legislator seeks to address, whether its design

¹²⁵ See generally Hart, *supra* n. 71, ch. V.

¹²⁶ Wintgens, *supra* n. 6, p. 220.

¹²⁷ Chapter 3, section 2.2 (Legalism according to legisprudence).

enables an on-going assessment of its efficacy, and finally whether it is coherent at the semantic, temporal, intra-systemic, and extra-systemic levels.¹²⁸

As suggested above, computational legalism can represent the apex of strong legalism. The impetus to legitimate the exercise of power by designers whose code vies with law to regulate behaviour is therefore all the greater. Designers limit individual and collective freedom in ways that have not been ratified by the democratic polity, via mechanisms that are technically and socially opaque, and which are not straightforwardly susceptible to public contest, redress, and (judicial) review. They are therefore potentially illegitimate exercises of power, and their negative effects are difficult to arrest or ameliorate when diffused across potentially millions of devices, often with little or no technical means of applying retrospective fixes.

3.2 Towards a *digisprudence*: the affordance of legitimacy in code-as-law

Koops suggests that ‘a good place to start looking for criteria for acceptability of normative technology is to study criteria for law.’¹²⁹ My search has taken me to the Fullerian and (especially) the legisprudential principles, concerned as they are with providing criteria for good law-making. Undoubtedly the two theories do not map directly onto the digital context, and so in Chapter 5 I translate them into the language of affordance, in order ultimately to set out a framework for ensuring legitimate rule-making in the commercial design environment.

Fuller’s principles have to an extent been applied to code before, with mixed results. Asscher was the first consider to them in the context of code, where he adapts

¹²⁸ Chapter 4 sets out the principles in greater detail in section 2.3 (Wintgens’ legisprudence).

¹²⁹ B.-J. Koops, ‘Criteria for Normative Technology: The Acceptability of “Code as Law” in Light of Democratic and Constitutional Values’ in R. Brownsword and K. Yeung (eds.), *Regulating Technologies: Legal Futures, Regulatory Frames and Technological Fixes* (Oxford; Portland, Or: Hart, 2008) p. 162.

them for the digital environment.¹³⁰ Although his subsequent analysis is far from exhaustive,¹³¹ it does provide a proof-of-concept for the kind of enquiry I am carrying out. Brownsword too considers the principles from the perspective not of design *per se*, but of the policies that are intended to be instrumentalised by it.¹³² In Chapter 4 I discuss in greater detail the literature in this area, including these two analyses,¹³³ but for now it can be said that neither application of Fuller to code engages with design theory or practice. In terms of jurisprudence, to my knowledge there has to date been no application of Wintgens' theory as a principled framework for guiding the design of code-based normativity.

As Chapter 5 sets out, the proposed framework consists of a set of digisprudential affordances that translate the principled goals that I distil from the literature into concrete suggestions for the design of code. In brief, these cover contestability, transparency of provenance and purpose, transparency of operation, choice, delay, and oversight.¹³⁴ The affordances are simultaneously general and concrete: they provide a design goal that should be reflected in all legitimate end-user-facing code, regardless of the form of technology, its substantive functionality, or the underlying business model. A corollary to be borne in mind is that certain functionalities or business models will therefore be illegitimate *a priori*.

There may be edge cases where the affordances are less easy to envisage or implement. However, like the legal-theoretical foundations upon which it builds,

¹³⁰ Asscher, *supra* n. 4, p. 85.

¹³¹ Asscher explicitly refers to Fuller's principles as a 'checklist' of requirements (see *ibid.*, p. 86). Wintgens suggests such a box-ticking approach is in general reductive of the aspiration that legality represents (see Wintgens, *supra* n. 6, p. 280). This accords with other literature on the overlap between design, engineering, and complex values. See for example S. Gürses, C. Troncoso and C. Diaz, 'Engineering Privacy by Design' (2011) 14 *Computers, Privacy & Data Protection*.

¹³² R. Brownsword, 'Technological Management and the Rule of Law' (2016) 8 *Law, Innovation and Technology* 100.

¹³³ Chapter 4, section 3 (Normative criteria for code-making).

¹³⁴ Chapter 5, section 3 (From characteristics to affordances).

digisprudence is aspirational: both legisprudence and the Fullerian principles are intended to encourage better (if not perfect) rule-making, and so similarly it is not expected that the digisprudential framework will cover every conceivable scenario where normative code is being produced. As Fuller suggests, perfect legality is ‘utopian’;¹³⁵ Wintgens notes in a similar vein that respect for the legisprudential principles is about ‘the aspiration to do the job as well as possible’.¹³⁶ The same can be said of its technological counterpart that I propose in this thesis.

4. Case studies

To give the thesis a grounding in contemporary technologies, in Chapter 5 I consider the digisprudence framework from the perspective of blockchain applications and the Internet of Things (IoT). Before I outline those case studies and their contemporary and future relevance, it is perhaps worth giving some historical context by mentioning an example that demonstrates many of the issues I am concerned with, namely digital rights management.

4.1 Digital rights management

As I discussed above in section 1.4, it is important to distinguish between compliance with substantive law (copyright, in the case of digital rights management), and broader and more fundamental questions of legitimacy. Digital rights management (DRM) systems have of course most commonly been concerned with preventing copyright infringement, but the term covers a broad range of approaches to code-based regulation.¹³⁷ This is demonstrated by the Sony BMG scandal in the mid-2000s.¹³⁸ To

¹³⁵ Fuller, *supra* n. 84, p. 41. See also p. 43.

¹³⁶ Wintgens, *supra* n. 6, p. 280.

¹³⁷ Radin, *supra* n. 28, p. 152. See also Lockton, *supra* n. 98.

¹³⁸ See for example ‘Sony Slated over Anti-Piracy CD’ *BBC News* (3 November 2005) <<http://news.bbc.co.uk/1/hi/technology/4400148.stm>>.

summarise, the record company Sony BMG included DRM software on several of its CD releases that was designed to limit the scope of playback, as well as of end-users' ability to copy the music either to digital files or to a blank CD. The code forced these operations to be performed via software on the CD, which installed itself surreptitiously on end-users' Windows PCs without notifying them. Upon insertion of the CD, if the code detected CD copying software already installed on the computer, it would cease playback and eject the CD.¹³⁹ Any copies made using the system were themselves protected by the same restrictions.¹⁴⁰

These measures might appear superficially reasonable vis-à-vis the prevention of copyright infringement. All was not as it appeared, however. Microsoft researcher Mark Russinovich discovered that the DRM system was also contacting Sony BMG's servers, ostensibly to update album artwork and lyrics, and in the process it was undermining end-user privacy by sending Sony BMG both the ID of the CD being played and the end-user's IP address.¹⁴¹ The end-user was not informed of this functionality or given the opportunity to disable it.¹⁴²

We can briefly consider the Sony BMG code through the lens of computational legalism. It bears repeating that those characteristics must all be viewed in light of the overarching issue of the code's private production. Not only was Sony BMG itself a commercial enterprise, but it outsourced production of the DRM systems to two third-party software firms, suggesting that Sony BMG may not have

¹³⁹ J.A. Halderman and E.W. Felten, 'Lessons from the Sony CD DRM Episode', *15th USENIX Security Symposium* (USENIX Association, 2006) p. 80.

¹⁴⁰ *Ibid.*, n. 8.

¹⁴¹ Mark Russinovich, 'More on Sony: Dangerous Decloaking Patch, EULAs and Phoning Home' <<http://blogs.technet.com/b/markrussinovich/archive/2005/11/04/more-on-sony-dangerous-decloaking-patch-eulas-and-phoning-home.aspx>>.

¹⁴² *Ibid.*

known the full extent of the software's (mis)behaviour.¹⁴³ Regarding the other characteristics, the system's code self-evidently imposed rules upon the end-user: her use of the music CD on her PC was constrained by policies set by the designer: music could be played, but only using the included software; the CD could be copied, but only if no other copying software was installed on the PC and again only using the included software. The system was opaque in its operation: those rules were not transparent, the system installed itself surreptitiously before the end-user granted permission (and even if she withheld consent),¹⁴⁴ and the full extent of its operation was not made clear even in the license agreement, which in any case was inaccurate.¹⁴⁵ While the end-user might reasonably have expected the system's operation to relate to the playing of the music, it is 'almost certain' that she would not anticipate that inserting the CD would 'open security backdoors into [her] computers and allow remote monitoring of the activities and knowledge of [her] machine configuration'.¹⁴⁶ Similarly, the system's immediacy was evidenced by the nature of its installation – end-users without deeper technical knowledge had no opportunity to refuse its installation on their machines, despite there being a license agreement requiring notional acceptance. Furthermore, the cumulative normativity of the system was felt most by those least likely to attempt to circumvent it: infringers were more likely to be technically adept and therefore capable of side-stepping the DRM, while lawful end-users with less technical literacy had their rights and convenience circumscribed despite not wishing to engage in unlawful copying.

¹⁴³ D.K. Mulligan and A. Perzanowski, 'The Magnificence of the Disaster: Reconstructing the Sony BMG Rootkit Incident' (2007) 22 *Berkeley Technology Law Journal* 1157, p. 1178 *et seq.* Halderman and Felten note though that the DRM vendors themselves almost certainly knew of the problematic behaviours of their software. See Halderman and Felten, *supra* n. 139, p. 90.

¹⁴⁴ Halderman and Felten, *supra* n. 139, p. 81.

¹⁴⁵ Mulligan and Perzanowski, *supra* n. 143, p. 1162.

¹⁴⁶ *Ibid.*, p. 1211.

The system's configuration of functionality, including the 'phoning home' behaviour, was imposed upon the end-user without consent or choice. The system was included on CDs which are impervious to change, and it had no pre-designed means of updating the software ex post to provide mechanisms to re-establish protection of the end-user's rights and/or enable her to express her preferences (this is evidenced by the release by Sony BMG, after the scandal started to gain prominence, of patches purporting to uninstall the software. These patches in fact caused further serious security problems¹⁴⁷). As Halderman and Felten note, '[i]f a particular version of DRM software is shipped on a new CD, that software version may well try to install and run decades after it was developed.'¹⁴⁸ Lastly, the system achieved significant distribution, if not pervasiveness: up to two million users were affected,¹⁴⁹ and in the fallout of the crisis Sony BMG recalled around 7.3 million CDs.¹⁵⁰

As I have already emphasised, these characteristics of the code's normativity can be critiqued separately from its implementation of the substantive norms of copyright law.¹⁵¹ As Halderman and Felten note,

The design of DRM systems is only weakly connected to the contours of copyright law. The systems make no pretense of enforcing copyright law as written, but *instead seek to enforce rules dictated by the label's and vendor's business models*. These rules, and the technologies that try to enforce them, implicate other public policy concerns, such as privacy and security.¹⁵²

Precisely because of the formal illegitimacies identified above, the ability of the end-user to be aware of and contest the mis-implementation of substantive copyright law

¹⁴⁷ See Mark Russinovich, *supra* n. 141 and Halderman and Felten, *supra* n. 139, p. 88 *et seq.*

¹⁴⁸ Halderman and Felten, *supra* n. 139, p. 89.

¹⁴⁹ Mulligan and Perzanowski, *supra* n. 143, p. 1158.

¹⁵⁰ *Ibid.*, p. 1169.

¹⁵¹ I discuss this in more detail in L. Diver, 'Law as a User: Design, Affordance, and the Technological Mediation of Norms' (2018) 15 *SCRIPTed* 4.

¹⁵² Halderman and Felten, *supra* n. 139, p. 91 (my emphasis).

was severely limited. To Halderman and Felten's suggested public policy concerns I would therefore add that of legitimacy, as embodied in the digisprudential affordances. By that standard the characteristics of the Sony BMG system were illegitimate regardless of substantive legal doctrine, and should not have been designed as they were, particularly given that if Russinovich had not uncovered the issues with the code's design – its embodiment of computational legalism – those issues may well have continued to operate for some considerable period without being detected and remedied.

As mentioned, DRM is not a case-study *per se*, but it does give a flavour of the issues I am concerned with. The next two sections introduce the 'forward-looking' case studies that will be returned to throughout Chapter 5 as a means of discussing the digisprudential framework.

4.2 Blockchain applications

The first case study focuses on so-called 'smart contracts' built upon the foundation of blockchain technology (later I shift from the term 'smart contract' to 'blockchain application', for reasons I will explain below). Like DRM, smart contracts represent another very explicit example of the embodiment of rules that have normative significance within the fabric of a digital artefact. Blockchain technology is still maturing¹⁵³ – the Bitcoin paper that proposed its initial design was published in 2008¹⁵⁴ and the first blockchain went live in January the next year¹⁵⁵ – but its implications and the massive publicity surrounding it¹⁵⁶ are the subject of increasing scrutiny from the

¹⁵³ P. De Filippi and A. Wright, *Blockchain and the Law: The Rule of Code* (Cambridge, Massachusetts: Harvard University Press, 2018) p. 209.

¹⁵⁴ S. Nakamoto, 'Bitcoin: A Peer-to-Peer Electronic Cash System' (2008).

¹⁵⁵ De Filippi and Wright, *supra* n. 153, p. 205.

¹⁵⁶ By 2016, blockchain had almost reached the 'peak of inflated expectations' in Gartner's Hype Cycle for New Technologies. See 'Gartner's 2016 Hype Cycle for Emerging Technologies Identifies Three

legal academy. While the publicity given to a particular technology should not in itself be enough for it to merit academic attention, the peculiarly ‘legal’ aspects of smart contracts raise questions that are of explicit interest to legal scholars.¹⁵⁷ Although increasing scepticism is being expressed about the practical value of blockchains,¹⁵⁸ these legal issues are worthy of scrutiny if only to confirm various aspects of the sceptics’ arguments.

4.2.1 *Blockchain design*

What follows is a summary of the main characteristics of blockchains,¹⁵⁹ which it is necessary to understand in order to appreciate the relevance of smart contracts, which are built upon blockchains, to the thesis. Blockchains are public¹⁶⁰ databases (or ‘ledgers’ – hence the term ‘distributed ledger technology’, or ‘DLT’, a term occasionally used instead of ‘blockchain’) which are stored on a number of computers (‘miners’) which together constitute a peer-to-peer network. To add to the chain requires consensus among the network’s nodes, and so a new ‘block’ of data will only be added if a majority of the miners agree that its addition is in accordance with the set of rules that govern how that particular blockchain should operate.¹⁶¹ These rules are known as

Key Trends That Organizations Must Track to Gain Competitive Advantage’ (*Gartner*, 16 August 2016) <<https://www.gartner.com/newsroom/id/3412017>>.

¹⁵⁷ R.H. Weber, “‘Rose Is a Rose Is a Rose Is a Rose’ – What about Code and Law?” (2018) 34 *Computer Law & Security Review* 701, p. 705.

¹⁵⁸ I. Kaminska, ‘Growing Scepticism Challenges the Blockchain Hype’ (*Financial Times*, 20 June 2017) <<https://www.ft.com/content/b5b1a5f2-5030-11e7-bfb8-997009366969>>.

¹⁵⁹ For a more in-depth primer on blockchains see M. Pilkington, ‘Blockchain Technology: Principles and Applications’ in F.X. Olleros and M. Zhegu (eds.), *Research Handbook on Digital Transformations* (Cheltenham, UK; Northampton, MA: Edward Elgar Publishing, 2016).

¹⁶⁰ Private (‘permissioned’) blockchains also exist, but because these are generally used internally within an organisation they mostly lack the focus on end-user behavioural regulation represented in relationship (c) in Figure 1 *supra*, and so I do not include them in this analysis. For more on private blockchains see V. Buterin, ‘On Public and Private Blockchains’ (*Ethereum Blog*, 8 June 2015) <<https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/>>.

¹⁶¹ De Filippi and Wright, *supra* n. 153, p. 2.

the blockchain's 'protocol', and they define how the blockchain operates and what the incentives and costs are for participants. These include the miners who provide the network's infrastructure, and the end-users who transact through it. Two prominent examples of different blockchain protocols are Bitcoin,¹⁶² a cryptocurrency and the original application of a blockchain design, and Ethereum,¹⁶³ the first blockchain to support sophisticated automation through the provision of a decentralised computing platform (the Ethereum Virtual Machine) which supports various general-purpose programming languages.

The protocol includes a mechanism for the miners to reach consensus on what should be included in the blockchain, including both metadata about transactions that have taken place and new smart contracts that will be executed by it. The question of how to reach consensus among anonymous computers is connected with what is known as the 'Byzantine fault problem', where the networked nodes each have a different understanding of the precise state of the chain (given that there are multiple simultaneous copies), but consensus must be reached for the system to be workable. Blockchain protocols overcome this using a combination of public key cryptography and 'hashing'.¹⁶⁴ The former is a mechanism for uniquely and conclusively identifying each node within the network by a public signature (key), while the latter is a method for generating a unique signature (hash) from any given volume of data (in this case, the existing prior state of the blockchain). Each block is given a unique hash, which is generated from a combination of that block's data and the hashes of all the blocks that are already on the chain. This means that, provided the last block in two copies of the chain have the same hash, one can be completely confident that the copies of the chain

¹⁶² Nakamoto, *supra* n. 154.

¹⁶³ 'Ethereum White Paper' (*The Ethereum Wiki*, 22 August 2018) <<https://github.com/ethereum/wiki/wiki/White-Paper>>.

¹⁶⁴ Pilkington, *supra* n. 159, p. 228.

are identical all the way back to the first block, and that therefore neither copy has been tampered with.

When a miner solves the mathematical challenge specified in the chain's protocol (this is how new blocks are added, for which the miner receives a reward), the proposed solution is broadcast to the network for the other miners to verify. They independently generate a new hash from the existing state of the chain and the proposed solution broadcasted by the 'winning' miner, and if the solution meets the requirements of the protocol's rules, each miner adds the block to their local copy of the chain. In this way the copies of the chain are kept identical and up-to-date across the many miners that store them. An important corollary of this proposal mechanism, particularly its use of hashes that represent the historical state of the chain, is that once a block has been added its contents are both immutable¹⁶⁵ and verifiable by observers.¹⁶⁶

Copies of the blockchain, including both its protocol and the data that it stores (e.g. transaction metadata, account balances, smart contract code) are replicated across the network, providing resilience through decentralisation.¹⁶⁷ Disabling (even physically) one of the network's computers will not delete the blockchain or prevent the code it stores from executing. In this way the design is similar to that of ARPANET, the precursor to the modern Internet, a communications network designed to be resistant to the underlying infrastructural network becoming damaged.¹⁶⁸ This lack of centralised authority controlling what gets added to the chain is part of the ideology behind the technology: provided participants follow the rules contained in the protocol, they get the benefits of a tamper-resistant, 'trustless' database with no centralised controlling authority.

¹⁶⁵ *Ibid.*, p. 233f.

¹⁶⁶ *Ibid.*, p. 227.

¹⁶⁷ De Filippi and Wright, *supra* n. 153, p. 2.

¹⁶⁸ B.M. Leiner et al., 'Brief History of the Internet' (Internet Society, 1997) p. 3 <<https://www.internetsociety.org/internet/history-internet/brief-history-internet/>>.

4.2.2 'Smart contracts'?

At present blockchains are probably best-known as the foundation of cryptocurrencies, but another related application that is potentially more disruptive from a legal perspective are so-called 'smart contracts' (SCs). SC platforms provide varying levels of sophistication. The Bitcoin protocol provides some very basic programming capabilities which can allow very limited SCs to be written. Some other platforms, known as 'sidechains', provide more sophisticated computation that runs separately from the primary Bitcoin blockchain but which rely on its relative stability as the ultimate store (ledger) of transaction activity.¹⁶⁹ Yet others are completely separate from the Bitcoin blockchain, providing both an independent transaction ledger and a protocol that is specifically designed to provide a more sophisticated programming foundation for SCs. Of these 'smarter' platforms, Ethereum is the most prominent¹⁷⁰ and therefore I focus on it in the following discussion.

According to the project's whitepaper, Ethereum is a 'next-generation smart contract and decentralized application platform.'¹⁷¹ What Ethereum seeks to do is compliment the powerful architectural characteristics of blockchains with a fully-fledged programming execution environment. Programming languages have been specially created¹⁷² that can compile code to be executed by the Ethereum platform, raising the possibility of supporting computationally-rich functionality with the immutability, robust decentralisation, and 'trustless trust' of blockchains. The notional

¹⁶⁹ An example of the former is Counterparty (see <<http://counterparty.io>>). For a discussion of sidechains, see T.I. Kiviat, 'Beyond Bitcoin: Issues in Regulating Blockchain Transactions' (2015) 65 *Duke Law Journal* 569, p. 604 *et seq.*

¹⁷⁰ For an empirical overview of the current major SC platforms, including Bitcoin and Ethereum, see M. Bartoletti and L. Pompianu, 'An Empirical Analysis of Smart Contracts: Platforms, Applications, and Design Patterns' [2017] *arXiv preprint arXiv:1703.06322* <<https://arxiv.org/abs/1703.06322>>.

¹⁷¹ 'Ethereum White Paper' *supra* n. 163.

¹⁷² Several such languages exist, including Solidity, Serpent, and LLL.

result is that the innovative possibilities of software's plasticity can be undergirded with the stability inherent in the 'anti-plasticity' of blockchains.

'Smart contracts' combine 'Turing-completeness, value-awareness, blockchain-awareness and state',¹⁷³ meaning they can define complex conditions, execute arbitrary behaviours when certain conditions are met, maintain and monitor states over time, and record the outcomes in the immutable blockchain. All of this takes place automatically; once conditions are defined in the 'contract' it remains 'live', awaiting the appropriate change(s) in conditions to trigger the rules it contains. In this sense smart contracts are not passive instructions on what the contracting parties *should* do, in the way that traditional legal contracts are, rather they are 'more like "autonomous agents" that live inside of the Ethereum execution environment, always executing a specific piece of code when "poked" by a message or transaction'.¹⁷⁴

Multiple SCs can be bundled together by a central business logic (itself written in code and stored on the blockchain) to create a 'distributed organisation' ('DO')¹⁷⁵ and even a 'distributed *autonomous* organisation' ('DAO'), which can operate without any human input.¹⁷⁶ These artefacts' logic enables, disables, and manages individual SCs, using them as tools to effect external changes according to the rules predefined in the code. A DO could, for example, require a majority vote from its (human) members as a condition of a given smart contract being triggered. Again, the decentralised and 'trustless' nature of blockchain design obviates the need for a trusted centralised authority (a traditional board or committee), and so notional governance of the organisation can be achieved even where the membership is geographically

¹⁷³ 'Ethereum White Paper' *supra* n. 163.

¹⁷⁴ *Ibid.*

¹⁷⁵ V. Buterin, 'DAOs, DACs, DAs and More: An Incomplete Terminology Guide' <<https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide/>>.

¹⁷⁶ *Ibid.*

dispersed, or even unknown.¹⁷⁷ A DAO, on the other hand, could consult external sources of data, known in this context as ‘oracles’,¹⁷⁸ to check for particular conditions in the real world, executing predetermined code-based logics when certain thresholds are met. This DAO code might then create real-world effects by interacting with the APIs of other services. When coupled with a cryptocurrency, it becomes possible to effect automated commercial transactions, even using APIs to automatically involve human actors, such as so-called ‘gig economy’ personnel,¹⁷⁹ or physical devices such as drones, for example to deliver goods.¹⁸⁰

The power of such code is intuitively appreciable. When specific conditions that are computationally representable are met, the code self-executes according to its internal logic, and the outcomes are enforced regardless of any (relevant) external circumstances or considerations. With the outcomes of the code’s execution being stored in the underlying blockchain alongside the code itself, what this means is both its logic and its results are immutable once they are ‘enacted’, executed, and stored. Thus code, in a very real and legally-significant sense, becomes ‘law’, through the ‘collapsing [of] contract formation and enforcement into a single instrument’.¹⁸¹ This coincidence of form and substance mean that when a smart contract is executed, the

¹⁷⁷ A. Wright and P. De Filippi, ‘Decentralized Blockchain Technology and the Rise of Lex Cryptographia’ (Rochester, NY: Social Science Research Network, 2015) SSRN Scholarly Paper ID 2580664 pp. 15–16 <<https://papers.ssrn.com/abstract=2580664>>.

¹⁷⁸ Cardozo Blockchain Project, ‘“Smart Contracts” & Legal Enforceability’ (New York: Benjamin N Cardozo School of Law, 2018) p. 6.

¹⁷⁹ Buterin describes a DAO as ‘an entity that lives on the internet and exists autonomously, but also heavily relies on hiring individuals to perform certain tasks that the automaton itself cannot do’. See Buterin, ‘DAOs, DACs, DAs and More’, *supra* n. 175.

¹⁸⁰ De Filippi and Wright, *supra* n. 153, p. 156. For a recent real-world example of the latter, see J. Perez, ‘XYO Game-Changer: We’ve Executed a Smart Contract With a Drone!’ *Medium* (21 November 2018) <<https://medium.com/xyonetwork/xyo-game-changer-weve-executed-a-smart-contract-with-a-drone-4deb414af67b>>. See also the discussion of the Internet of Things, *infra*.

¹⁸¹ K.E.C. Levy, ‘Book-Smart, Not Street-Smart: Blockchain-Based Smart Contracts and The Social Workings of Law’ (2017) 3 *Engaging Science, Technology, and Society* 1, p. 3.

material effects of that execution are governed by the dictates of pure code, regardless of any ambiguity or subjective understanding that might exist in the minds of the humans involved. One can appreciate the parallel with the discussion of computational legalism in section 2.2 above, noting that code is at once rule and reality; the normative collapsed into the merely descriptive.

4.2.2.1 ‘Contract’ and the problem of terminology

We saw above in section 1.2 the tension between the goal of regulating code using law (i.e. achieving compliance by design), and the idea that it can also be a separate normative order that while regulable by law also requires, because of computational legalism, additional ex ante formal standards that can guide its development. The use of the term ‘contract’ somewhat muddies the waters by implying that SCs are traditional legal contracts that have simply been rendered ‘smart’ by the addition of some technology, akin to a ‘smart’ electricity meter or a ‘smart’ doorbell.

However, that an SC is also a legal contract is by no means a given, and the use of the term ‘contract’ has served to confuse more than to enlighten. The freedom that SC designers have to create normative architecture is not limited by the formal rules required for contract formation within the institutional law (e.g. Hart’s secondary rules¹⁸² or MacCormick’s institutive rules¹⁸³). While it is possible that the institutional law will develop to declare that SCs will be recognised as traditional legal-institutional contracts, this is currently by no means a settled question.¹⁸⁴ It is potentially possible

¹⁸² Hart, *supra* n. 71, p. 94.

¹⁸³ N. MacCormick, *Law as Institutional Fact* (Edinburgh: University of Edinburgh, 1973) pp. 5–6.

¹⁸⁴ To underline just how uncertain the legal status of smart contracts currently is, the England and Wales Law Commission has recently initiated a law reform project to investigate the issue. See ‘Smart Contracts’ (*Law Commission*) <<https://www.lawcom.gov.uk/project/smart-contracts/>>. See also S. Murphy and C. Cooper, ‘Can Smart Contracts Be Legally Binding Contracts? White Paper Key Findings’ (R3 and Norton Rose Fulbright, 2016) <<http://www.nortonrosefulbright.com/files/norton-rose-fulbright--r3-smart-contracts-white-paper-key-findings-nov-2016-144554.pdf>> and Cardozo Blockchain Project, *supra* n. 178, p. 9

to automate parts of a traditional legal contract, but the point at which an SC shifts from a tool for enforcing a traditional contract to being a legal contract itself lies on a spectrum that is currently the subject of discussion across various jurisdictions.¹⁸⁵ On that spectrum, the extent of automation can be anything from the (minimal) wrapping of traditional legal contract text within code in order to facilitate provenance checking and to prevent tampering with the text,¹⁸⁶ to the (intermediate) inclusion of code parameters in a traditional legal contract that facilitate some computational interaction with the contract,¹⁸⁷ to the (maximal) embodiment of the entire contractual logic within code. The latter is of course what SCs aim to do (although work to document SC code for natural language intelligibility is ongoing – this is discussed in Chapter 6). Although some US jurisdictions have attempted to define the various terms for legal purposes, this has not been unproblematic.¹⁸⁸

At any rate, the technological normativity embodied in an SC in no way hinges on it being recognised as ‘legal’ *per se*. This is, of course, to restate the central argument of the thesis: the code in the SC can impose normativity and instrumental effects directly, without any consideration of the law’s requirements for validity. This confusion of legal/non-legal effect is arguably deepened by the fact that SCs are indeed *intentionally designed* to effect operations that *are* quintessentially legal, for example the transfer of funds or assets between parties. For this reason, Buterin recently lamented

(discussing the U.S. position). There appears to be no analysis of the question from the perspective of Scots contract law.

¹⁸⁵ Cardozo Blockchain Project, *supra* n. 178, p. 4.

¹⁸⁶ This is the goal behind the concept of the Ricardian Contract. See I. Grigg, ‘The Ricardian Contract’, *Proceedings of the First IEEE International Workshop on Electronic Contracting* (IEEE, 2004). I discuss this further in Chapter 5, section 3.1.2 (Discussion: blockchain applications).

¹⁸⁷ C.D. Clack, V.A. Bakshi and L. Braine, ‘Smart Contract Templates: Foundations, Design Landscape and Research Directions’ [2017] *arXiv:1608.00771 [cs]*.

¹⁸⁸ Cardozo Blockchain Project, *supra* n. 178, p. 22 *et seq.*

adopting the term ‘smart contract’ (originally coined by Szabo in the 1990s¹⁸⁹), suggesting that a better name would have been something ‘more boring and technical’, such as ‘persistent scripts’.¹⁹⁰ Similarly, Felten suggests that it is best to think of SCs not as contracts but as ‘mechanisms’.¹⁹¹ Noting the lack of consensus on terminology, Clack et al. make a useful distinction between on the one hand ‘smart legal contracts’, which are traditional legal contracts that are merely expressed in code and are intended to operate like their text-based counterparts (whose legal status is, as noted above, not yet settled), and on the other ‘smart contract code’, which simply intends to enforce the content of the code, regardless of any purported legal status.¹⁹² Although these differing goals might overlap in future, for now it is useful to view the former as a technology for legal practice (i.e. ‘legal tech’), and the latter as the form of ‘a-legal’ normative code with which I am concerned.

One can perhaps appreciate from the preceding discussion the subtleties of implementation and terminology that are involved, and the complications that arise from using legal language in an ‘a-legal’ context.¹⁹³ In any event, the moniker ‘contract’ (and the outcome of investigations into their legal status) notwithstanding, we can view SCs as simply as another species of privately-produced normative code that embodies the design characteristics of (dis)affordance and inscription discussed in the next chapter. Indeed, Buterin’s alternative name (‘persistent script’) is interesting in

¹⁸⁹ See N. Szabo, ‘Smart Contracts’ (1994) <<https://web.archive.org/web/19971008001022/http://www.virtualschool.edu/mon/Economics/SmartContracts.html>>.

¹⁹⁰ See V. Buterin (*Twitter*, 13 October 2018) <<https://twitter.com/VitalikButerin/status/1051160932699770882>>.

¹⁹¹ E. Felten, ‘Smart Contracts: Neither Smart nor Contracts?’ <<https://freedom-to-tinker.com/2017/02/20/smart-contracts-neither-smart-not-contracts/>>.

¹⁹² Clack, Bakshi and Braine, *supra* n. 187, p. 2.

¹⁹³ For an argument in favour of decoupling SCs from legal language to avoid confusion and their mischaracterisation, see CleanApp, ‘Crypto Legal Theory’ <<https://medium.com/cryptolawreview/crypto-legal-theory-299fa35be21f>>.

part because it suggests an unintended connection with the latter concept, which is concerned with the guiding ‘stories’ designed into an artefact that determine how it might be used – in the case of SCs such inscriptions are particularly strong. Because their design has normative effects, they are still therefore susceptible to the application of ex ante standards of legitimacy developed later in the thesis.

Bearing in mind all of the above, I suggest that to avoid bolstering any incipient confusion about terminology it might be helpful (i) to expressly think of smart contracts as ‘a-legal’ instead of ‘legal’ *per se* (this helps to frame them as simply another species of normative code), and (ii) to use a term other than ‘smart contract’. Clack et al.’s ‘smart contract code’ moves in the right direction but still uses the misleading word ‘contract’, so I propose therefore to use the term ‘blockchain application’ hereinafter. This term seems appropriate because, as with other software applications it does not imply any inherent legal status, and simultaneously it maintains a connection to the underlying blockchain technology that embodies many of the characteristics of computational legalism with which I am concerned.

4.3 The Internet of Things

Compared with blockchain applications, the Internet of Things (IoT) is perhaps a simpler (but no less important) area to conceptualise and analyse. In the early 1990s Mark Weiser, a pioneer of what has variously been termed ‘ambient intelligence’¹⁹⁴ and ‘ubiquitous computing’, spoke of the profundity of technologies that ‘weave themselves into the fabric of everyday life until they are indistinguishable from it’.¹⁹⁵ The US Federal Trade Commission has defined the IoT as ‘devices or sensors – other than computers, smartphones, or tablets – that connect, communicate or transmit

¹⁹⁴ Hildebrandt and Koops, *supra* n. 32, p. 430f.

¹⁹⁵ M. Weiser, ‘The Computer for the 21st Century’ [1991] *Scientific American* 94. Weiser was discussing ‘ubiquitous computing’, a precursor term for the IoT.

information with or between each other through the Internet.¹⁹⁶ This focus on sensors and devices other than traditional platforms (computers and smartphones/tablets) implies the ‘weaving into daily life’ to which Weiser referred. Indeed, IoT devices are designed to do precisely this, both ubiquitously and invisibly, and as such are becoming an increasingly significant proportion of the total number of devices connected to the Internet.¹⁹⁷ This is in part due to a ‘chip-centric mentality’, where manufacturers have bought into commercial hype suggesting that a connected device is better than an unconnected one.¹⁹⁸ The results of this are occasionally absurd.¹⁹⁹

IoT devices both illustrate the design theories mentioned above and exemplify numerous aspects of computational legalism, especially opacity, immutability, and pervasiveness. Because they are intended to be embedded and pervasive, they by nature tend towards both minimal affordances and very strictly-defined inscriptions. The Amazon Dash Button, for example, consists of just a single button and an LED indicator. Its inscription is thus a simple one of ‘press the button’, and its design affords that and little more (‘adhesion’ and ‘throwing’ are perhaps the only alternative action possibilities). As I describe in more detail in Chapter 5,²⁰⁰ behind this apparent simplicity and minimal interface lies a complex series of technical events that are kept hidden from the device’s end-user but which are potentially of great importance to her

¹⁹⁶ ‘Internet of Things: Privacy and Security in a Connected World’ (Federal Trade Commission, 2015) p. 6.

¹⁹⁷ IDC forecasts a 15.4% increase in spending on IoT devices compared with 2018. See M. Torchia and M. Shirer, ‘IDC Forecasts Worldwide Spending on the Internet of Things to Reach \$745 Billion in 2019, Led by the Manufacturing, Consumer, Transportation, and Utilities Sectors’ *IDC* (Framingham, Mass, 3 January 2019) <<https://www.idc.com/getdoc.jsp?containerId=prUS44596319>>.

¹⁹⁸ W. Hartzog and E. Selinger, ‘The Internet of Heirlooms and Disposable Things’ (2016) 17 *North Carolina Journal of Law & Technology* 581.

¹⁹⁹ For a frequently-updated selection of examples that demonstrate this, see the Twitter account @InternetofShit, available at <<https://twitter.com/internetofshit>>.

²⁰⁰ Section 3.2.4 (Discussion: IoT).

(imagine the device being mis-used by a young child or pet).²⁰¹ As with other computing systems, the extent to which complex logic *should* be hidden from the user is one which will vary depending on the system in question. Nevertheless, the central issue of transparency about what lies beneath the physical device's 'tip of the iceberg' is a crucially important one. IoT devices, particularly those that have a single function like the Dash Button, generally combine simple physicality on the part of the object with complex and opaque computation on the 'back-end'. There is therefore significant scope for dissonance between the end-user's understanding of the device's affordances and what it in fact does.²⁰²

In terms of immutability, the poor infrastructural provision made for updates, coupled with a lack of commitment to long-term oversight, have resulted in many examples of IoT devices being used as nodes in bot-nets, being left open to external hacking, and other forms of unintended breach.²⁰³ This lack of flexibility, when coupled with the devices' intended pervasiveness, is potentially deeply problematic.

These problems can even combine with those of blockchain applications – as I mentioned above, the latter are capable of effecting changes in the physical world via IoT devices, such as drones²⁰⁴ and 'smart' devices. The combination implies the concept of 'smart property', where the hybrid IoT-blockchain artefact is autonomous, such that the IoT device's physical functionality is controlled by the logic contained in the blockchain application (for example a smart door lock might refuse to open after a code-based 'lease' expires²⁰⁵).

²⁰¹ 'Internet of Things: Privacy and Security in a Connected World', *supra* n. 197, p. 22.

²⁰² A. Matassa and R. Simeoni, 'Eliciting Affordances for Smart Objects in IoT Era', *Internet of Things. User-Centric IoT* (Springer, Cham, 2015).

²⁰³ Hartzog and Selinger, *supra* n. 198.

²⁰⁴ See references *supra* n. 180.

²⁰⁵ Levy, *supra* n. 181, p. 3.

5. Conclusion and recap

Before moving onto the substantive chapters of the thesis, this section recaps the broad argument and how this connects with the analysis that is to come. Firstly, code can have regulative effects on behaviour that are more pervasive and direct than law is capable of. The various design mechanisms by which it achieves this are discussed in Chapter 2. The regulative effects form a corpus of norms that are separate from (and indeed may conflict with) institutional law. The characteristics of code – what I call computational legalism – mean that in many cases that separate normative order will be enforced whether or not it is compatible with substantive doctrinal law. I describe the elements of computational legalism in Chapter 3.

Secondly, in a democratic society, norms that regulate citizens, of any kind and from any source, ought to be legitimate. In the legal context this can be achieved by the application of *ex ante* theoretical frameworks such as Fuller’s principles of legality or Wintgens’ legisprudence, both of which I discuss in Chapter 4. These legal-theoretical approaches are concerned with the *form* of the legal norm that is created, as distinct from its political content. Legitimacy on this account acts as a kind of design constraint, which through the binding of the sovereign to the inclusion of certain formal characteristics limits to some extent what the substantive content of a norm can be.

Thirdly, given that in the private contexts where code-based norms are designed there are no such formal principles for norm-creation, the question then arises of whether the mechanisms of producing legitimate normativity in the *legal* sphere might be transposed into the *computational* sphere. Some form of *ex ante* legitimation is crucial, because the instrumentality of computational legalism is potentially more problematic than illegitimate legal norms that – because of the hermeneutic gap between text and action – can be ignored, re-interpreted, or contested in court.

Fourthly, any such legitimation of a code rule must, by definition, take place at design time because, unlike the institutional law, there is little or no scope for reinterpreting the code after-the-fact; code and reality are one from the point of distribution onward. I therefore adapt the ex ante frameworks of legitimation from the legal world for their application to code, taking into account design theory and the ways in which code actually regulates. In bringing these various theoretical strands together I set out the framework of digisprudential affordances in Chapter 5, and consider their application to the two case studies. Chapter 6 discusses potential avenues for operationalising the digisprudential framework, including aspects of the design environment, the constitutional role of the programmer of the programmer, and how contestability can be provided for. The thesis then concludes in Chapter 7, summarising the research findings and some opportunities for further work that I have identified in the course of this research.

Chapter 2

A design perspective: code is *more* than law¹

The computer programmer, however, is a creator of universes for which he alone is the lawgiver... [n]o playwright, no stage director, no emperor, however powerful, has ever exercised such absolute authority to arrange a stage or a field of battle and to command such unswervingly dutiful actors or troops.²

The swift effectiveness of a technological code, which cannot, when seen through legal eyes, appear as anything other than uncanny, renders any possible competition between law and computer pointless.³

1. Introduction

This chapter sets out, using the literature on design theory, how code has a concrete and direct effect on the behaviour of end-users. This contribution is then picked up in Chapter 5 where I use these same concepts to formulate the novel framework of digisprudential affordances. The engagement of the legal literature with these theories has until recently been minimal; the tendency has been to treat design only in the abstract, without a concerted engagement with the theory on what things *actually* do, and *how* they do it. Without such engagement the legal view of technology is limited to that of an outside observer, rather than one that can engage with the processes of material production from which the effects of code flow. As I suggested in the introductory chapter, and will consider in greater depth in Chapter 4, a focus on production is crucial if the aspiration of legitimacy is to be realised in the computational paradigm.

¹ Significant parts of this chapter were published in L. Diver, 'Law as a User: Design, Affordance, and the Technological Mediation of Norms' (2018) 15 *SCRIPTed* 4.

² J. Weizenbaum, *Computer Power and Human Reason: From Judgment to Calculation* (San Francisco: Freeman, 1976) p. 115.

³ C. Vismann and M. Krajewski, 'Computer Juridisms' [2007] *Grey Room* 90, p. 93.

Throughout the thesis I refer to the concept of *technological normativity*. Borrowed from Hildebrandt,⁴ I use the term because it usefully implies a contrast between code's normativity and the concept of *legal* normativity that lawyers are more familiar with. Her definition of it is also closely linked to the theory of affordance, which I explore in detail in section 2 below. She defines technological normativity as 'the way a particular technological device or infrastructure actually constrains human actions, inviting or enforcing, inhibiting or prohibiting types of behaviour.'⁵ These effects can be intentionally or unintentionally imposed by the designer, and can be an immediate or emerging characteristic of the code she creates. Hildebrandt also makes a useful distinction between regulative and constitutive normativity, which I discuss below in section 3.5.1, and again in the legal-theoretical context in Chapter 3.⁶

The remaining discussion in this chapter sets out three primary and interconnected theories: affordance, inscription, and the technological mediation of perception and action. Taken together, these theories provide us with the conceptual tools to consider the behavioural effects of a particular artefact's design. From a normative perspective, they also provide a theoretical backdrop for thinking about how consciously to choose and refine those effects in the process of producing code, in order to meet the standards of legitimacy that the thesis proposes.

2. Affordance

The facilitation by an artefact's design of a particular action or behaviour for a particular individual is known as an *affordance*. The concept was originally developed in the late 1960s by the perceptual psychologist James Gibson, who defined affordances

⁴ M. Hildebrandt, 'Legal and Technological Normativity: More (and Less) than Twin Sisters' (2008) 12 *Techné: Research in Philosophy and Technology* 169 *passim*.

⁵ *Ibid.*, p. 173.

⁶ Chapter 3, section 3.2.1 (Constitutive and regulative rules).

collectively as what an artefact ‘offers’ the animal, what it *provides* or *furnishes*, either for good or ill.⁷ The theory of affordance was later developed and introduced into the design sphere by Norman, who defines the concept as ‘a relationship between the properties of an object and the capabilities of the agent that determine just how the object could possibly be used.’⁸ As an example, a common representation of the concept of affordance compares two doors, one with a panel for pushing, and another with a handle that can be pulled:⁹

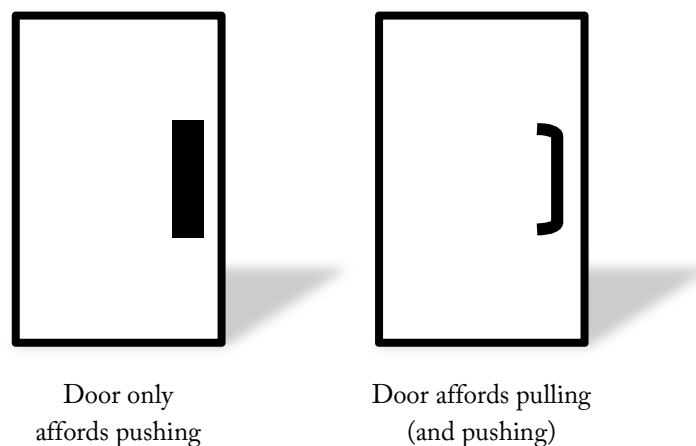


Figure 2. Affordances of pushing and pulling

For an able-bodied person, the door on the left can only be pushed – i.e. it only affords pushing – because there is no part of it that affords pulling (unless one manages to grip the edges of the panel). The handle on the door on the right affords both pulling and pushing – the ability to grasp it readily enables the end-user to pull the door towards

⁷ J.J. Gibson, *The Ecological Approach to Visual Perception* (Classic Edition, New York: Psychology Press, 2015) p. 119 (emphasis supplied). The concept originated in J.J. Gibson, *The Senses Considered as Perceptual Systems* (Boston, MA: Houghton Mifflin, 1966).

⁸ D.A. Norman, *The Design of Everyday Things* (Cambridge, Mass; London, UK: MIT Press, 2013) p. 11.

⁹ Norman discusses this *ibid.*, p. 15. See also W. Hartzog, *Privacy's Blueprint: The Battle to Control the Design of New Technologies* (Cambridge, Mass: Harvard University Press, 2018) p. 13.

her (assuming of course that the door's hinges afford pulling in that direction). The door on the right has at least two affordances (again for an able-bodied person): one of pulling, and another of pushing.

Individual affordances can be both positive and negative, which is to say beneficial and injurious to the end-user, each to varying degrees. Gibson is careful to avoid the value judgements suggested by the terms 'positive' and 'negative', stating instead that such descriptions can be applied objectively if their meanings are 'pinned down to biological and behavioral facts'.¹⁰ So, for example, a fire can afford the warmth that is necessary to life, but it can also afford burning, which can mean injury and potentially death.¹¹ The extent of the benefit or injury will depend on the organism in question. Affordances are therefore not objective physical properties of the artefact, but rather they arise through the relationship between it and a particular individual, as governed by those properties. Gibson illustrates this relationship through the examination of a hypothetical walking surface:

Note that the four properties listed – horizontal, flat, extended, and rigid – would be *physical* properties of a surface if they were measured with the scales and standard units of physics. As an affordance of support for a species of animal, however, they have to be measured *relative to the animal*. They are unique for that animal. They are not just abstract physical properties. They are unity relative to the posture and behaviour of the animal being considered. So an affordance cannot be measured as we measure in physics.¹²

Thus, a surface that *affords* support to a domestic cat (i.e. it is 'stand-on-able'¹³) may or may not afford the same to an adult elephant; the particular mix of physical properties and the size and weight of both animals will determine which capabilities are afforded

¹⁰ Gibson, *supra* n. 7, p. 129.

¹¹ *Ibid.*, pp. 128–129.

¹² *Ibid.*, p. 120 (emphasis supplied).

¹³ *Ibid.*, p. 119.

to each. It can be seen, then, how the concept highlights the inherent and simultaneous objectivity and subjectivity of an artefact's potential effects in the world. As Norman puts it,

[t]he presence of an affordance is jointly determined by the qualities of the object and the abilities of the agent that is interacting... [w]e are used to thinking that properties are associated with objects. But affordance is not a property. An affordance is a relationship. Whether an affordance exists depends on the properties of both the object and the agent.¹⁴

With these definitions in mind, it can be appreciated that designers must include the necessary properties in the artefact in order for the desired relationship between it and the organism (end-user) to arise. This is inevitably a contingent exercise – the designer cannot anticipate the properties of every conceivable end-user; nevertheless, a central aspect of the design enterprise is having an audience in mind whose properties will imply the properties that the code must have in order to bring about the affordance relationships the designer wishes there to be.¹⁵

2.1 Real and perceived affordance

Importantly, an affordance need not be perceived in order to exist; it is an objective fact about how the properties of the artefact and the organism relate to one another.¹⁶ Affordances are potentials that may not be within the organism's awareness and may never be realised, but nevertheless the relationship is always present and ready to be

¹⁴ Norman, *supra* n. 8, p. 11.

¹⁵ J.R. Maier and G.M. Fadel, 'Affordance-Based Methods for Design', *Proceedings of DETC* (Chicago, Illinois: The American Society of Mechanical Engineers, 2003) p. 4.

¹⁶ Norman, *supra* n. 8, p. 13.

acted upon for as long as the properties necessary for it are extant in both the artefact and the organism.¹⁷

This is what Norman refers to in later work as *real*, as opposed to *perceived*, affordance.¹⁸ For example, a particular fruit may afford nutrition to a particular species of animal, but if the animal is unaware of this the relationship will never be acted on, despite its extant potentiality. Perceived affordances are those which the organism ‘picks up on’, which, as the example just given demonstrates do not necessarily represent the full range of relationships that exist between it and the artefact in question. The distinction is important in the digital context, because as Norman puts it, ‘in graphical, screen-based interfaces, the designer primarily can control only perceived affordances [because] the computer system already comes with built-in physical [i.e. real] affordances’.¹⁹ The potential discrepancy between real and perceived affordances is perhaps even more marked in the screen-less devices that are proliferating as part of the Internet of Things (this will be discussed in greater detail in Chapter 5²⁰).

Norman’s comment hints at an important truth about the power of the designer to control end-users’ perceptions through the choices they make in constituting an artefact’s interface. The corollary of this is that in controlling those surface perceptions of what is possible, other underlying (real) affordances can be hidden from sight (for example the ability to view and alter source code, or to submit false details to a registration system to avoid being tracked), or their hypothetical, imagined possibility suppressed altogether (for example where end-users

¹⁷ See P. Nagy and G. Neff, ‘Imagined Affordance: Reconstructing a Keyword for Communication Theory’ (2015) 1 *Social Media + Society* p. 3 and S. Faraj and B. Azad, ‘The Materiality of Technology: An Affordance Perspective’ in P.M. Leonardi, B.A. Nardi and J. Kallinikos (eds.), *Materiality and Organizing: Social Interaction in a Technological World* (Oxford University Press, 2012) pp. 250–251.

¹⁸ D.A. Norman, ‘Affordance, Conventions, and Design’ (1999) 6 *Interactions* 38.

¹⁹ *Ibid.*, p. 39.

²⁰ Chapter 5, section 3 (From characteristics to affordances).

unquestioningly accept the default settings of a system without enquiring as to how the available options might better suit their preferences or interests. This is an issue discussed below, and again in Chapter 3²¹).

2.1.1 Signifiers

The design of the artefact can incorporate signifiers which communicate to the end-user the affordance that is present and how it should be used (this is an important part of the artefact's normativity and is connected to technological intentionality, discussed below in section 3.4).²² For example, returning to Figure 2 above, the panel on the door on the left signifies where to push, while the handle on the door on the right signifies where to grasp (which in turns signifies pulling). Of course, in order to act as a signifier, that element of the artefact must be perceived by the end-user (it can, however, be ambiguous – the hinges of the door with the handle might also afford pushing, despite the handle signifying the affordance of pulling). The presence of signifiers is an important element of communicating to the end-user how the artefact works, but a signifier's utility is also contingent on its accuracy, honesty, and completeness.

The fact that function x is signified to the end-user of course does not entail that function y is also signified – the right functions must be signified at the right moment. The question of *what* to signify and *when* to signify it is therefore extremely important in helping her form an accurate mental model of the system;²³ designs often afford functionalities without signifying them, perhaps to hide complex functionality from novice end-users who may not understand it, or perhaps to provide the functionality required by some external force (regulation, ethics) without actively bringing it to the attention of the end-user because its use might harm the commercial

²¹ *Infra*, section 3.5, and Chapter 3, section 3.3.1 (Default configurations), respectively.

²² Norman, *supra* n. 8, p. 13 *et seq.*

²³ Hartzog, *supra* n. 9, p. 27.

interests of the provider. An example of this is the complex cookie preference notices that have appeared following the coming into force of the GDPR. While these often provide an interface for fine-grained control over which cookies are set on the end-user's computer (i.e. they afford control), the means of accessing this interface (a textual link) is usually much less prominent – less clearly signified – than the option to accept all cookies. If the end-user selects the latter this is of course commercially beneficial because it enables behavioural advertising which is more lucrative for the website's providers. This type of adversarial design is an example of a 'dark pattern', which will be discussed in the next section. In sum, one can appreciate how important signifiers are in assisting the end-user to develop an appropriate mental model of the system she is using.²⁴

3. Infusing code with normativity

As we have seen, affordances are relationships that arise according to the objective characteristics of an end-user and an artefact. In many cases they exist simply by virtue of those properties, as in the example of the surface that can bear the weight of – i.e. afford support to – an elephant. Affordances can of course also be consciously designed, however, making otherwise inert artefacts 'usable' for a certain class of end-user.²⁵ From the perspective of behavioural regulation, the conscious choices about how to make an artefact useful can develop into mechanisms that actively constrain or suggest particular courses of action, thus infusing the design not just with usefulness but also with regulative effect. That is the subject of this section.

²⁴ The appropriate use of signifiers for achieving digisprudential legitimacy is discussed further in Chapter 5, section 3.2 (Opacity).

²⁵ Norman, *supra* n. 8, ch. 6. I discuss the process of code production in more detail in Chapter 6.

3.1 Disaffordance

Gibson's notion of the positivity or negativity of affordances, discussed above, is concerned with the *outcome* occasioned by the affordance (e.g. a fire warming and organism vs. burning it). This should be distinguished from both (i) the objective fact that interaction is prevented and the relationship therefore does not exist – what Norman terms an 'anti-affordance'²⁶ – and (ii) the subjective misapprehension as to the existence of the affordance, where the end-user misinterprets the information she is receiving and believes there to be a particular relationship between herself and the artefact when in fact there is none (or not the one she believes there to be). As an example of the latter, both Gibson and Norman provide the example of a glass pane covering an opening, which gives the end-user the erroneous impression of the affordance of passage. Norman's 'anti-affordance', on the other hand, points simply to the objective fact that there is no such affordance, whether the end-user is aware of this or not (a blind individual, for example, is simply not afforded passage, regardless of her inability to see the glass and interpret what the opening might afford her).²⁷ These two ideas can of course overlap – in the case of the glass pane the end-user misperceives the existence of an affordance of passage which is, in fact, not there.

Drawing on Lessig's discussion of 'architectures of control',²⁸ Lockton takes the notion of 'anti-affordance' further, adding the element of intention that is less evident in Norman's discussion. Lockton defines architectures of control as 'features, structures or methods of operation designed into any planned system with which a user interacts, which are intended to enforce or restrict certain user behaviour'.²⁹ He discusses disaffordance in the context of DRM, including the Sony BMG scandal that

²⁶ *Ibid.*, p. 11.

²⁷ Gibson, *supra* n. 7, pp. 133–134; Norman, *supra* n. 8, pp. 11–12.

²⁸ L. Lessig, *Code: Version 2.0* (New York: Basic Books, 2006) ch. 4.

²⁹ D. Lockton, 'Architectures of Control in Product Design' [2006] *Engineering Designer: The Journal of the Institution of Engineering Designers* 28.

was discussed in Chapter 1.³⁰ Both Gibson's 'negative affordance' (which is concerned with the 'ill' that is offered, provided, or furnished by the artefact) and Norman's 'anti-affordance' (which is concerned with the *bare fact* of the absence of a particular offering, provision, or furnishing by the artefact) are concerned with the ex post outcome of the affordance's operation.

'Positive' for Lockton means the philosophical sense of what is actually in existence, i.e. what is ex ante permitted by the norm-giver, versus what is not – there is a positive affordance of a particular action. The corollary for Lockton is that 'negative' affordance is about the *engineering of obedience*.³¹ He is concerned with the intent of the designer, which connects with my central theme of code's production. Lockton suggests the term 'disaffordance' to describe designs that have 'functionality deliberately removed... or with the functionality deliberately hidden or obscured to reduce users' ability to use the product in certain ways, or a combination of the two.'³² Thus, disaffordances are 'deliberate, intentional, and strategic', as opposed to inadvertent or the result of incompetent design. They therefore embody a conscious value in a way which Gibson explicitly, and Norman implicitly, avoid. The term 'disaffordance' has gained only modest traction,³³ but it is instructive in encapsulating the idea of how an artefact can conceal, discourage, or forbid the possibility of certain behaviours as a result of conscious design decisions. In aggregate, one can appreciate the role played by disaffordances in constraining end-users in their interactions with an artefact. As Longford puts it,

³⁰ Chapter 1, section 4.1 (Digital rights management).

³¹ D. Lockton, 'Disaffordances and Engineering Obedience' <<http://architectures.danlockton.co.uk/2006/10/22/disaffordances-and-engineering-obedience/>>.

³² *Ibid.*

³³ See D.E. Wittkower, 'Principles of Anti-Discriminatory Design', *2016 IEEE International Symposium on Ethics in Engineering, Science and Technology (ETHICS)* (IEEE, 2016) p. 2 (acknowledging this fact in a discussion on how to avoid unethical discrimination in design).

[t]he reconfiguration of the terms of cybercitizenship which these technologies effect is achieved via a gradual process in which new habits, expectations and practices on the part of web users are cultivated and/or inculcated through *subtle mechanisms of inducement, coercion, and reward designed into the very experience* of cyberspace.³⁴

When disaffordances are designed that are contrary to the end-user's interests, they are sometimes termed 'abusive design', and examples that exploit commonly-used design conventions for negative purposes have come to be known as 'dark patterns'.³⁵ Such practices demonstrate the power of the designer to exploit the end-user for purposes which may not be in her interests; Conti and Sobiesk describe the 'intent on the part of the designer to deliberately sacrifice the user experience in an attempt to achieve the designer's goals ahead of those of the user'.³⁶ They set out a taxonomy of approaches used in malicious web interfaces, and provide representative examples that most end-users will be familiar with. These include making form fields mandatory (coercion), use of double or triple negatives in questions (confusion), advertising (distraction), delaying access until an advert is watched (forced work), covering desired text with popups (interruption), hiding access to the free version of an application deep within a website's navigation (manipulating navigation), reducing the contrast of closure buttons on adverts (obfuscation), and designing adverts to appear to be news content (trickery).³⁷

³⁴ G. Longford, 'Pedagogies of Digital Citizenship and the Politics of Code': (2005) 9 *Techné: Research in Philosophy and Technology* 68, p. 77 (my emphasis).

³⁵ 'Deceived by Design: How Tech Companies Use Dark Patterns to Discourage Us from Exercising Our Rights to Privacy' (Consumer Council of Norway (Forbrukerrådet), 2018) <<https://fil.forbrukerradet.no/wp-content/uploads/2018/06/2018-06-27-deceived-by-design-final.pdf>>.

³⁶ G. Conti and E. Sobiesk, 'Malicious Interface Design: Exploiting the User', *Proceedings of the 19th International Conference on the World Wide Web* (Raleigh, North Carolina, USA: ACM Press, 2010) p. 271.

³⁷ *Ibid.*, p. 273.

A recent example that exhibited some of these characteristics was the popup GDPR acceptance screen shown to Facebook users following the Regulation's coming into power in May 2018. The interface showed the well-known friends, messages, and notifications icons in the usual top-right position behind the popup, but two of these had red circles superimposed, implying that there were messages and notifications waiting. Unlike Facebook's usual interface, however, the red circles did not contain a number, which would usually signify how many messages or notifications the end-user had. When the end-user had gone through the GDPR acceptance process, the interface returned to normal, under which the red circles either disappeared if there were no messages or notifications waiting, or were displayed including the actual number. The purpose of displaying the 'blank' red circles during the GDPR interface, therefore, was to incentivise the end-user to accept the terms as quickly as possible in order to gain access to the messages and notifications that were purportedly waiting for her.³⁸ The crucial point is that the circles appeared whether or not the end-user actually had any messages or notifications, strongly implying a manipulation on the part of Facebook. Conti and Sobiesk note how such design practices can increase frustration and even render parts of the Web inaccessible for certain classes of end-user, and that their primary aim is generally to increase revenue for website operators.³⁹

Stepping back from individual artefact-end-user relationships of (dis)affordance, the field of postphenomenology and its theory of technological mediation provides another useful way to conceptualise the broader relationships that arise between code and end-users. This is the subject of the next section.

³⁸ 'Deceived by Design: How Tech Companies Use Dark Patterns to Discourage Us from Exercising Our Rights to Privacy' *supra* n. 35, p. 29. This report gives a detailed account of the desktop interfaces Facebook used to communicate their GDPR update, which included numerous examples of dark patterns and manipulative design.

³⁹ Conti and Sobiesk, *supra* n. 36, pp. 278–279.

3.2 Postphenomenology and code's mediation of reality⁴⁰

Postphenomenology is an area of Science, Technology & Society (STS) studies that explores the relationships between individuals and artefacts, with an emphasis on the material qualities of particular artefacts *per se*.⁴¹ The focus is therefore on specific aspects of an artefact's design and how it creates reality for its user, as opposed to the position of a broadly-conceived 'technology' within a grander social or political narrative.⁴² Peter-Paul Verbeek describes postphenomenology as the analysis of the 'role played by specific technologies in specific contexts',⁴³ which asks what the normative effects are of their materiality on the mediation of the relationship between humans and reality. A central claim of postphenomenology is that technologies are neither wholly neutral nor wholly deterministic, and the ways in which their designs mediate reality place them somewhere between those two poles.⁴⁴ The relationships between humans and artefacts are grouped into those of perception – what the individual *thinks* she can do with the artefact – and those of action – what she can *actually* do with it (recall the discussion above about the distinction between real and perceived affordances). Technological mediation is the ongoing construction and manipulation of these two relationships by and through artefacts, the result of which is the co-constitution of reality for the end-user. The end-user and the artefact, in

⁴⁰ There are various subtleties in the literature in this area that can occasionally be difficult to follow, particularly where similar concepts are referred to using different terms. My aim here is to demonstrate the normativity of code without becoming mired in those nuances. For a very useful clarification of the relations between the various terms used in the literature, see P.-P. Verbeek, 'Materializing Morality: Design Ethics and Technological Mediation' (2006) 31 *Science, Technology, & Human Values* 361, particularly at p. 368.

⁴¹ P.-P. Verbeek, *What Things Do: Philosophical Reflections on Technology, Agency, and Design* (Penn State Press, 2005) p. 3.

⁴² *Ibid.*, ch. 1.

⁴³ *Ibid.*, p. 7.

⁴⁴ Verbeek, *supra* n. 40. On the latter point see Verbeek, *supra* n. 41, p. 11.

bringing together their particular characteristics, constitute a new reality through their relationship. We can conceptualise the relationship as in Figure 3 below:



Figure 3. Artefact ↔ user relationship of technological mediation

One can appreciate the parallels with the theory of affordance; indeed, affordances are the individual building blocks that in aggregate make up the totality of technological mediation between a particular artefact and a particular end-user.⁴⁵ As discussed above, designers play a central role in defining what affordances a given artefact provides, and thus the reality created by the artefact’s mediation of perception and action is to a significant degree determined by choices made by the designer. This connects with the idea of constitutive normativity built into the architecture of an artefact, a topic discussed below in section 3.5.1.

3.3 Code mediating perception

Perception is mediated by technology through the amplification or reduction of aspects of the world.⁴⁶ This relates to the signifiers discussed above: their design can draw the end-user’s attention to the possibility of a particular use, or perhaps ward her off or

⁴⁵ A.H. Kiran and P.-P. Verbeek, ‘Trusting Our Selves to Technology’ (2010) 23 *Knowledge, Technology & Policy* 409.

⁴⁶ For an in-depth discussion of the various forms of perceptual mediation that goes beyond what is necessary for my purposes see D. Ihde, *Technology and the Lifeworld: From Garden to Earth* (Indiana University Press, 1990) p. 72 *et seq.*

distract her from perceiving it. Signifying by itself has no direct coercive effect on the end-user, but by mediating her perception it does play an important role in shaping her comprehension of an artefact and her ability to form an accurate mental model of how it works and what she can and should do with it.⁴⁷ This ability to transform reality as it is experienced by the end-user is one aspect of the power of design, particularly when it goes beyond what she can perceive of reality to include how she can and cannot act. This ability to determine how reality is experienced, both perceptually and in terms of behavioural agency, demonstrates ‘an important aspect of the non-neutrality of technology’,⁴⁸ and points to the significant power that inheres in the designer who exercises control over the nature of those mediations.

3.4 Code mediating action

Whereas the technological mediation of perception amplifies or reduces what can be *comprehended* of reality, the technological mediation of action invites or inhibits specific behaviours. This form of mediation is instrumental, in the sense that it exerts a physically or logically compelling force on the behaviour of the end-user, rather than merely a signal to comply. It is here, then, that the regulative nature of code is most apparent: the coercion of action by code (its ‘moreness’, to reference the title of this chapter) can be contrasted with the mere signal provided by a textually-bound legal norm. This important distinction, between the constitutive nature of code and the regulative nature of textual rules, is discussed further below at section 3.5.1, and again in Chapter 3.⁴⁹

⁴⁷ Norman, *supra* n. 8, pp. 26, 31. See also Hartzog, *supra* n. 9, p. 278.

⁴⁸ Verbeek, *supra* n. 41, p. 131. Verbeek speaks of perception being ‘transformed’, while Latour talks of action being ‘translated’. See B. Latour, ‘Where Are the Missing Masses? The Sociology of a Few Mundane Artifacts’ in W.E. Bijker and J. Law (eds.), *Shaping Technology/Building Society: Studies in Sociotechnical Change* (MIT Press, 1992) p. 174 and *passim*. For my purposes, the difference in terminology is not materially relevant, hence the use of ‘transformations’.

⁴⁹ Chapter 3, section 3.2.1 (Constitutive and regulative rules).

Code embodies a particular idea of how the designer intended the artefact to be used. This is what Latour calls a ‘program of action’,⁵⁰ which like the script of a film or play describes how the designer intends the artefact to be used or what its envisaged effect in the world ought to be. Akrich makes explicit use of this metaphor in her important analysis of *inscription*: ‘like a film script, technical objects define a framework of action together with the actors and the space in which they are supposed to act.’⁵¹ Designers envisage these elements of the ‘script’ when they design the artefact’s (dis)affordances: the framework for behaviour, the actors involved (both human and non-human⁵²), and the space for action.⁵³ The various constituents of the script will be determined according to the envisaged uses of the artefact and the business model the designer aims to uphold.⁵⁴

To give an example, a speed bump in a road has the inscription of ‘slow down when you approach me’,⁵⁵ and its physical properties invite in the strongest terms a particular action – slowing down – on pain of serious damage otherwise being done to the vehicle.⁵⁶ Latour might also say that the imposition of that action is ‘delegated’ to the speed bump (indeed, the description of the latter as ‘sleeping policemen’ implies

⁵⁰ Latour, *supra* n. 48. See also M. Akrich, ‘The De-Description of Technical Objects’ in W.E. Bijker and J. Law (eds.), *Shaping Technology/Building Society: Studies in Sociotechnical Change* (MIT Press, 1992).

⁵¹ Akrich, *supra* n. 50, p. 208. See also Verbeek, *supra* n. 40, p. 362.

⁵² Actor Network Theory (ANT), whose literature Akrich and Latour are central contributors to, explicitly avoids creating a hierarchy between humans and non-humans, instead using the model of a flat web to describe the influences operating between disparate ‘actants’. See generally Latour, *supra* n. 48.

⁵³ Akrich, *supra* n. 50, p. 208. Latour terms this anticipation ‘preinscription’. See Latour, *supra* n. 48, p. 172.

⁵⁴ B. van den Berg and R.E. Leenes, ‘Abort, Retry, Fail: Scoping Techno-Regulation and Other Techno-Effects’ in M. Hildebrandt and J. Gaakeer (eds.), *Human Law and Computer Law: Comparative Perspectives* (Dordrecht: Springer Netherlands, 2013) p. 76.

⁵⁵ Verbeek, *supra* n. 40, p. 366.

⁵⁶ Latour, *supra* n. 48, p. 166. See also Lessig, *supra* n. 28, pp. 128; 135–136.

this re-assignment of the task from a human to a non-human agent).⁵⁷ This coercion of action by the speed bump can be contrasted with the merely signifying effect of a speed limit sign, whose inscription only describes, rather than physically mandates, the required action.⁵⁸

In other work, Latour describes the example of the Berliner lock, whose design means that once its end-user is inside the room, if she wishes to close the door she is forced also to lock it.⁵⁹ The inscription in the lock's design thus limits the possible states that the end-user can leave the door in to one of either (i) open, or (ii) closed and locked; there is no in-between state permitted by the design of the lock (i.e. closed and unlocked). This is a physical example of the binary 'ruleishness' of code, a core element of computational legalism discussed in Chapter 3.⁶⁰

Akrich's and Latour's concepts of inscription, programs of action, and delegation are closely related to the postphenomenological idea of *technological intentionality*. Some examples from the latter field can serve to further illustrate the normative power of the designer to guide the end-user's behaviour. First, Ihde contrasts the technological mediations of a fountain pen and a word processor.⁶¹ The former implies a slower pace of action that inclines the writer towards taking time and considering her sentences before putting pen to paper, while the latter permits something closer to the speed of the spoken word, with additional facilities that allow the text to be edited, moved around, and refactored with ease and speed. Neither the

⁵⁷ Latour, *supra* n. 48, p. 157f and *passim*. See also Verbeek, *supra* n. 41, p. 159f.

⁵⁸ The connection of these concepts to the legal notions of constitutive and regulative norms and the jurisprudential concept of the internal and external perspective of norms is discussed *infra* in section 3.5.1 (Constitutive and regulative normativity).

⁵⁹ B. Latour, 'The Berlin Key or How To Do Words with Things' in P. Graves-Brown (ed.), *Matter, Materiality and Modern Culture* (London: Routledge, 2000). See also Latour, *supra* n. 48, p. 172 *et seq.*

⁶⁰ Chapter 3, section 3.1 (Ruleishness).

⁶¹ Ihde, *supra* n. 46, p. 141f.

pen nor the word processor conclusively *predetermine* the mode of writing – both permit writing that can be anywhere between slow and considered, or fast and careless – but their respective designs do ‘*promote or evoke* a distinct way of writing.’⁶² Second, and towards the more overtly political end of the normativity spectrum, Verbeek describes how the mayor of the Romanian city of Cluj had the municipal gardeners’ rakes shortened so that they could no longer lean against them, thus promoting harder work through the discouragement of ‘laziness’. As Verbeek describes the situation,

Action had to be taken because the rake, en passant, made possible an entirely different practice, one that was not anticipated but that arose only in the practice of raking. The rake *mediates* the relation between the workers and the public gardens; it is not merely a means but plays an active role in the way this relation takes shape.⁶³

Winner’s oft-quoted discussion of Robert Moses’ bridges on New York’s Long Island suggests a similar politicisation of artefacts. In that case, the bridges were reportedly designed intentionally to be too low for public buses to pass beneath them, thus preventing those reliant on public transport (which at the time meant to a disproportionate degree the poor and racial minorities) from accessing the public beaches to which the road that ran under the bridges led.⁶⁴

Each of the examples above demonstrates the first postphenomenological sense of ‘intention’, where through the provision of ‘a framework for action, [artefacts] do form intentionalities and inclinations within which use-patterns take dominant shape’:⁶⁵ the speed bump *intends* to slow drivers down; Moses’ bridges *intend* to prevent

⁶² Verbeek, *supra* n. 41, pp. 114–115 (emphasis supplied).

⁶³ *Ibid.*, p. 115 (emphasis supplied).

⁶⁴ L. Winner, ‘Do Artefacts Have Politics?’ [1980] *Daedalus* 121. Although Winner’s account has since been questioned (see B. Joerges, ‘Do Politics Have Artefacts?’: [2016] *Social Studies of Science*), his example nevertheless demonstrates the concept effectively.

⁶⁵ Ihde, *supra* n. 46, p. 141.

access by the poor and minorities to public beaches; the Berliner lock *intends* that a closed door always be locked; the rake handles *intend* to prevent leaning and to promote work; word processors *intend* the efficient (re)composition of text. This first sense refers to ‘a certain directionality, inclination or trajectory that shapes the ways in which [artefacts] are used’,⁶⁶ and is of course intimately connected with what (dis)affordances the designer embodies in the design.

The second sense of intentionality refers not to a property of the artefact but rather to the *end-user’s* intention, and how the artefact mediates her relationship with the world by dictating what she can and cannot do.⁶⁷ The end-user’s sense of her own agency, and of the possibilities in the world which that agency can interact with, are mediated by the artefact, thus blurring the line between subjectivity and objectivity.⁶⁸ When she sets about to achieve something, her perception of what she can do and what the world permits are mediated by the artefact, and thus so too are her understanding of her self and her world co-constituted through the lens of that mediation.⁶⁹ ‘The world’ here is not an external truth; rather it is constituted by the particular individual who lives in it, as she is by it. The operation is mutual and bi-directional – she makes her world and her world makes her, and that ‘making’ is nudged this way or that by the artefacts’ technological mediation, as comprised by its (dis)affordances.

⁶⁶ Verbeek, *supra* n. 41, p. 114.

⁶⁷ *Ibid.*, p. 116; Ihde, *supra* n. 46, p. 25.

⁶⁸ Shedding the ‘modern’ dichotomy of subject/object is a prime goal of postphenomenology, and of Actor Network Theory. See Verbeek, *supra* n. 41, p. 161 *et seq.* See also Faraj and Azad, *supra* n. 17, pp. 237–238.

⁶⁹ Verbeek, *supra* n. 41, p. 116. This echoes Cohen’s suggestion that ‘as we struggle to shape our technologies and configure our artifacts, they also and quite literally configure us’. See J.E. Cohen, *Configuring the Networked Self: Law, Code, and the Play of Everyday Practice* (Connecticut: Yale University Press, 2012) p. 27.

Contextual changes result in different configurations of mediation, of both perception and action. This is what Ihde terms the *multi-stability* of an artefact: it can facilitate different acts depending on the context of use, the individual using it, and the configuration of the artefact itself. As Verbeek puts it, '[a]rtifacts can only be understood in terms of the relation that human beings have to them.'⁷⁰ Artefacts exist for the purpose of doing something, but that something is not determined entirely by the artefact itself but also by how a particular user approaches it at a particular time and within a particular context.⁷¹ Moses' Long Island bridges demonstrate this contextual dependency: over time their normative effect has lessened as those who were intended to be excluded have become wealthier and less reliant on public transport. Multi-stability refers to how differing contexts can result in different concrete uses of an artefact, the sum of which implies the absence of an 'essential' purpose. Akrich makes a similar argument in relation to inscription: while the designer will envisage some kind of role for the end-users of the code she creates, and it is from this image of the end-user that the inscription of the design is ultimately derived, in practice the domain of action is not absolutely predetermined and will to some extent be adapted 'in the wild'.⁷²

Although the absence of an essential purpose implied by multi-stability means that the mediating effect of an artefact is not entirely within the ex ante control of the designer, she will nevertheless 'inscribe scripts and delegate responsibilities' in and to the artefacts she designs.⁷³ There is also the logical necessity that in creating one particular configuration of normativity – even one that is multi-stable – the designer is

⁷⁰ Verbeek, *supra* n. 41, p. 117.

⁷¹ Verbeek, *supra* n. 40, p. 371.

⁷² Akrich, *supra* n. 50, p. 208. See also T. Swierstra and K. Waelbers, 'Designing a Good Life: A Matrix for the Technological Mediation of Morality' (2012) 18 *Science and Engineering Ethics* 157.

⁷³ Verbeek, *supra* n. 40, p. 372.

making a decision that *a priori* excludes at least some others.⁷⁴ To a greater or lesser degree, the design of an artefact will “groom” the user,⁷⁵ shaping her perception and her scope for action in ways that may not be legitimate according to any external standard. As Akrich notes, while ‘the designer not only fixes the distribution of actors, he or she also provides a “key” that can be used to interpret all subsequent events. Obviously, this key can be called into question – consumer organizations specialize in such skepticism.’⁷⁶ This key is the inscription that is embodied in her design, and I am of course engaging in a kind of scepticism too – one that is informed by a legal-theoretical framing of what constitutes a legitimate exercise of power over end-user behaviour.

An unavoidable part of the designer’s job, then, is in defining the thresholds in the spaces for action created by the artefact’s design between what is strictly inscribed and what can be (re)interpreted by the end-user. Thus, the very existence of an artefact means *a priori* that these choices have been made, inadvertently or otherwise. The extent to which inscription predetermines what behaviour is possible relates to the spectrum of technological normativity, discussed further below in section 3.5. Before turning to that discussion, it is worth summarising the relevance of the relationship between affordance theory and technological mediation.

3.4.1 *Affordance and technological mediation*

Considering the discussion above, we can now comprehend affordances as the underlying building blocks of inscription and technological mediation.⁷⁷ They are a powerful unit of analysis for identifying and critiquing the inscriptions of code which

⁷⁴ This relates to the discussion of defaults and the spectrum of normativity, *infra* at section 3.5.

⁷⁵ Akrich, *supra* n. 50, p. 218.

⁷⁶ *Ibid.*, p. 216.

⁷⁷ See Kiran and Verbeek, *supra* n. 45.

come together in aggregate to mediate the co-constitutive relationship between the end-user and the world.

Both real and perceived affordances are evidence of the second form of technological intentionality, where the artefact mediates the individual's understanding of what she can do in the world as she perceives it.⁷⁸ This connects closely with the concept of multi-stability, where a congruence between the artefact's perceived and real affordances provide a margin of opportunity within which the end-user might adapt her response to the predetermined script of the artefact. Unless the artefact embodies *real* affordances that lie outside the designer's intended inscription, the end-user will by definition be unable to do anything with the artefact that the designer did not intend. In such a case her behaviour vis-à-vis the artefact will be constituted entirely according to the decisions made by the designer. The corollary is that where the designer leaves 'space' for creative interpretation and action – through the conscious (or unconscious) provision of real affordances and their signifiers – the end-user will be able to express her autonomy (within the wider constraints of the artefact's mediation).

Real (dis)affordances are the bread and butter of the first form of technological intentionality: to inscribe a particular programme of action in the artefact, its design must afford that course of action for a particular (class of) end-user; similarly, to *proscribe* a particular course of action, the designer must disafford it for a particular (class of) end-user, as in the cases of the Long Island bridges, the Berliner lock, or the shortened rake handles. The existence of an affordance is an objective fact about the relationship between a particular artefact and individual in a particular context, which when taken in aggregate with any other (dis)affordance results in a particular normative assemblage of technological mediation. And, as discussed above, affordances are not fixed attributes of an artefact, rather they come about as relations

⁷⁸ *Ibid.*, p. 415 *et seq.*

between particular artefacts and particular individuals in particular contexts, albeit that (as we saw above) designers will anticipate what these are likely to be when they are considering the inscriptions they want to embed in the artefact.

3.5 A spectrum of technological normativity

We have seen how inscriptions and affordances as their building blocks exist on a spectrum, from ‘harder’ implementations that admit of no choice to softer ones whose normativity is suggestive rather than coercive.⁷⁹ The former conception of normativity sees the script of the artefact as ‘wired-in’, where the end-user has no choice but to follow a succession of code norms as they are presented to her.⁸⁰ This is the most ‘ruleish’ and immediate aspect of technological normativity: the rule is clearly defined (in code, if not necessarily for the benefit of the end-user) and it is applied immediately at runtime with no opportunity for further consideration. These characteristics of code (ruleishness, opacity, and immediacy) are central elements of the concept of computational legalism that I develop further in the next chapter.

Less strict are code-based suggestions which ‘nudge’ the end-user towards a particular course of action, whilst permitting her to express choice or to ‘disobey’ the default configuration by choosing between two or more options. Despite this notional scope for exercising autonomy, various biasing effects have been shown to operate that render the default setting very ‘sticky’, implicitly discouraging the end-user from making any change.⁸¹ One approach to minimising this effect is to *force* a choice at the

⁷⁹ Van den Berg and Leenes, *supra* n. 54, p. 74f.

⁸⁰ J. Kesan and R. Shah, ‘Setting Software Defaults: Perspectives from Law, Computer Science and Behavioral Economics’ [2006] *University of Illinois Law and Economics Working Papers*. See also Fogg’s discussion of ‘tunneling’ users in *Persuasive Technology: Using Computers to Change What We Think and Do* (Amsterdam; Boston: Morgan Kaufmann Publishers, 2003) p. 34 *et seq.*

⁸¹ See the discussion in Chapter 3, section 3.3.1 (Default configurations).

moment of installation or setup, without any preferred option being suggested.⁸² As we have seen in the section on disaffordance above, however, enterprise will often interpret even strict regulation requiring the protection of user autonomy in ways that subtly (or not so subtly) favour its interests over those of the end-user – post-implementation GDPR privacy notices are a good example of this. This connects with the contemporary evolution of design practices, such as the adversarial interfaces mentioned above, which may provide notional choice but which in reality are targeted at capturing end-users’ attention (often using psychological research to refine the interface’s affordances in order to maximise attention capture through ‘operant conditioning’⁸³). The extent to which such approaches have moved beyond the ‘libertarian paternalism’ of nudging (intended as it was to strike a balance in the civic sphere between the individual’s freedom to choose and better societal outcomes⁸⁴) and into the realm of manipulation and even the cultivation of ‘tech addiction’⁸⁵ is an emerging topic in both the academy and civil society.⁸⁶

⁸² Microsoft were forced to do this when the European Court of Justice found in *Microsoft Corp. v Commission of the European Communities* (Case T-201/04) that the company’s inclusion of its web browser Internet Explorer as the default in the Windows operating system was an abuse of its dominant market position. The agreed solution was to provide end-users with a ‘ballot’ screen asking them to choose from a randomly-ordered range of browsers. See J. Brodtkin, ‘EU Fines Microsoft €561 Million for Not Giving Users a Browser Choice’ (*Ars Technica*, 6 March 2013) <<https://arstechnica.com/tech-policy/2013/03/eu-fines-microsoft-e561-million-for-not-giving-users-a-browser-choice/>>.

⁸³ Fogg, *supra* n. 80, ch. 3 and *passim*. See also van den Berg and Leenes, *supra* n. 54, p. 71f.

⁸⁴ C.R. Sunstein and R.H. Thaler, ‘Libertarian Paternalism Is Not an Oxymoron’ (2003) 70 *University of Chicago Law Review* 1159; van den Berg and Leenes, *supra* n. 54, p. 72f.

⁸⁵ See for example S. Parkin, ‘Has Dopamine Got Us Hooked on Tech?’ *The Observer* (4 March 2018) <<https://www.theguardian.com/technology/2018/mar/04/has-dopamine-got-us-hooked-on-tech-facebook-apps-addiction>>.

⁸⁶ For examples see, respectively, J. Williams, *Stand out of Our Light: Freedom and Resistance in the Attention Economy* (Cambridge, United Kingdom; New York, NY: Cambridge University Press, 2018) and ‘The Problem’ (*Center for Humane Technology*) <<http://humanetech.com/problem/>>.

Towards the more open end of the spectrum of normativity, the code's inscriptions can provide space for interpretation, reinvention, and 'resistance' by the end-user (albeit that once she is using an artefact, such resistance will always be limited to the space left for it within the inherent boundaries of its geography⁸⁷ – this point is discussed further below). In many cases the distinctions here will be on the level of the interface, where the end-user's interactions with the system are guided to varying degrees. There, the role of signifiers is particularly relevant: the end-user cannot avail herself of an affordance if she does not know it is there (recall the discussion of the nutrition-providing fruit in the discussion above). The business model underlying the design of the artefact will determine the extent to which it is multi-stable. For example, if the inscription in a smartphone application such as Instagram is one of 'upload photos and videos to be viewed by other end-users' (which in turn is made possible by a set of affordances for selecting a file, perhaps editing it, transferring it to a remote server, providing a title and tags, and publishing), no amount of 'resistance' can rewrite that inscription to enable the calculation of a tax return. However, there might nonetheless lie within that inscription some scope for reinterpretation, for example using the particular layout that galleries have within the application to imagine new expressive possibilities such as creating a set of images that connect to one another visually, like stills from a film.⁸⁸ In this case, the gallery feature of the application is multi-stable in that it provides both the simple function of attaching multiple images to a single post but also and simultaneously a means for the end-user to reinterpret the structure of the system to express herself in a novel way not anticipated by the designer.

Although this is a rather frivolous example, it does demonstrate the scope of behavioural possibilities that code can provide. We can see how the spectrum of

⁸⁷ Van den Berg and Leenes, *supra* n. 53, p. 77.

⁸⁸ This approach can occasionally be seen on Twitter. See, for example, <<https://mashable.com/2014/04/01/how-to-make-a-twitter-collage/>>.

normativity moves from the most ‘ruleish’ of code norms to the least, with the overall ‘density’ of the constraints on the behaviour of the end-user lessening from one point to the next.⁸⁹ The placing of the thresholds between these represent crucial choices in the process of designing an artefact. Affordances can thus be distinguished according to their normative effect.⁹⁰ Davis and Chouinard, for example, suggest that affordances exist on a six-point spectrum, from ‘request’ to ‘refuse’. They give the example of a set of stairs that might afford easy or difficult climbing depending on the angle of their construction. This is in opposition to the ‘classic’ concept of affordance, which would have it that the stairs either simply do or do not afford climbing for a particular individual.⁹¹ They go on to suggest that affordances can be characterised as one of six mechanisms: request, demand, allow, encourage, discourage, and refuse. Adding one of these modifiers adds useful depth to the bare concept of affordance, enabling a more intuitive understanding of a given individual-artefact relationship. For the example of the stairs above, then, they might *allow* the able-bodied to climb, *discourage* careless climbing (if they are particularly steep), and *refuse* climbing to those who require a wheelchair. Here we get an immediate sense of three normative affordance relationships that exist between the artefact and three hypothetical classes of end-user.

Considered through these affordance mechanisms, it becomes easier to discern the particular makeup of a given artefact’s normativity, and from the preceding discussion we can see how wired-in function tend towards the ‘harder’ mechanisms of request, demand, allow, and refuse, while the mechanisms of encouragement and discouragement are more likely to be found where the artefact’s affordances are designed around nudging and multi-stability. As previously mentioned, it is important

⁸⁹ The concept of ‘normative density’ is one of the intersections between legal and design theory explored in Chapters 4 and 5.

⁹⁰ J.L. Davis and J.B. Chouinard, ‘Theorizing Affordances: From Request to Refuse’ [2017] *Bulletin of Science, Technology & Society* p. 241.

⁹¹ Davis and Chouinard characterise this as a ‘false binary’. See *ibid.*, p. 242.

to note that the design of an artefact will always embody some mix of these characteristics, because as soon as code is laid down choices have been made and *some* configuration of normativity – be it open or closed, strictly ruleish or multi-stable – has come into existence. As Lessig puts it, ‘there is no choice that does not include some kind of building. Code is never found; it is only ever made’.⁹²

3.5.1 *Constitutive and regulative normativity*

This spectrum of normativity connects with the theoretical distinction between constitutive and regulative rules.⁹³ Whereas constitutive rules define how a state of affairs may be brought into being (e.g. a valid game of chess, or a marriage), regulative rules merely *request* action or inaction on the part of an individual or class of individuals (e.g. a speed limit on a road, or a rule asking employees to turn out the lights in an office). If the requirements of a constitutive rule are not met, then the state of affairs does not and cannot obtain; the mere assertion that a couple is married is insufficient to make it so in the eyes of the relevant order from which the concept derives, i.e. the legal system. At the same time, although a regulative rule can request some (in)action from the individual (e.g. do not drive above 70mph) it has no ability directly to impose that requirement – the individual must acquiesce and alter her behaviour accordingly.⁹⁴

A similar distinction applies in the design sphere – both constitutive and regulative normativity can be created by code, and the decision of where to draw the line between the two is within the gift of the designer. As Hildebrandt notes,

it makes sense to discriminate between socio-technical arrangements that are constitutive and those that are regulative of our interactions, if only to make

⁹² Lessig, *supra* n. 28, p. 6.

⁹³ I discuss the relevance of this distinction from a legal-theoretical perspective in Chapter 3, section 3.2.1 (Constitutive and regulative rules).

⁹⁴ For a useful discussion contrasting the legal and technological normativities that surround driving, see Hildebrandt, *supra* n. 4, p. 176.

clear that technology does not necessarily rule out choice in comparison to law.⁹⁵

As we have seen, the (dis)affordances and inscriptions embodied in the artefact's design can be constitutive or regulative of the end-user's behaviour.⁹⁶ Of course, the very existence of the artefact and its functions are the subject of a core of constitutive affordances, e.g. that it can be seen, that it can be touched, opened, executed, etcetera. The boundaries of the artefact represent a set of *a priori* foundational 'rules' which define its very nature *ab initio* – the form of its interface, the platforms it can run on, its physical dimensions, etcetera. Operating above this lower-level sense of constitutive affordance, however, are the specific (dis)affordances and scripts that constitute the behaviour of the end-user when she is interacting with the artefact. As with the example above, she may *wish* to calculate her tax return using Instagram, but the code's constitutive norms do not permit such a use. The possibility is simply not within the 'constitution' of the code. Although the designers of those platforms have (presumably) not consciously made the decision not to include tax calculation functionality, the example underlines the point that design always involves the privileging of one configuration of normativity – one 'technical constitution' – over the near-infinite alternative possibilities that could otherwise have been built in code.⁹⁷ This speaks to the plasticity of code – 'programmers can implement almost any system they can imagine and describe precisely',⁹⁸ but, of course, that very precision will necessarily exclude a huge range of other possibilities.

⁹⁵ *Ibid.*, p. 175.

⁹⁶ M. Hildebrandt, 'A Vision of Ambient Law' in R. Brownsword and K. Yeung (eds.), *Regulating Technologies: legal futures, regulatory frames and technological fixes* (Oxford; Portland, Or: Hart, 2008) p. 177f.

⁹⁷ Weizenbaum, *supra* n. 2, pp. 37f; 113.

⁹⁸ J. Grimmelmann, 'Regulation by Software' (2005) 114 *The Yale Law Journal* 1719, p. 1723.

From the perspective of regulative normativity, some measure of choice is left open to the end-user in how she behaves within the geography set up by the code – for example, she is free to choose from a palette a highlight colour for her social media profile, and to attach up to five photos (or indeed no photo at all) to her social media post. Such regulative latitude, however, always operates within constitutive outer boundaries beyond which choice is unavailable, for example there is no freedom to choose colours that are not included in the provided palette, or to attach a PDF or spreadsheet file to the social media post.

All of these behavioural (dis)affordances are contingent on the choices made by the designer. She can enable a particular functionality or close it off entirely, or perhaps hide it from view. In each case she has exercised her private power to constitute the range of behaviour that the end-user can engage in.⁹⁹ She might also opt for affordances that are merely regulative of behaviour, using one of the less ruleish mechanisms above to allow the end-user to change the configuration (defaults) of the code or reinvent the space it opens up in ways unforeseen by the designer (multi-stability). These possibilities will be returned to in Chapter 5, particularly in the context of the case studies.

3.6 Technological constitutionalism

The discussion above has explored how the normativity embodied in an artefact's design enables and constrains the behaviour of the end-user, focusing on relationship (c) of Figure 1 in Chapter 1.¹⁰⁰ But there is, however, another aspect through which the designer is *herself* made to comply. Further back in the production chain, product designers are made to comply with normativity created by other designers creating

⁹⁹ Van den Berg and Leenes, *supra* n. 53.

¹⁰⁰ Chapter 1, section 1.4.1 (Normative relationships online).

more fundamental elements of architecture. This is relationship (d) in Figure 1.¹⁰¹ Not only are end-users subject to the effects of (dis)affordance, inscription, and mediation, but so too are ‘end-designers’, within the environments that they in turn use to create artefacts intended for end-users. From this perspective, the product designer is herself rendered a type of end-user by those in the position of the *programmer of the programmer* (‘PoP’). Vismann and Krajewski describe the PoP in these terms:

The programmer of the programmer, designing the tools and methods of a coding language (such as the compiler, code syntax, abstract data types, and so on) maintains the ultimate power because he or she, as the constructor of the programming language itself, defines what the “normal” programmer, as a user, will be able to do. Both types of programmers establish the conditions for using the computer, and, as such, they behave like lawmakers or, rather, *code-makers*. Implemented within the CPU and the hierarchy of the file system is the law governing communication with and through the computer. In this respect, code and law maintain a relationship of more than structural homology. The code *is* a law – as Lawrence Lessig pointed out when he described “code” metaphorically as a synonym for the conditions under which the computer runs.¹⁰²

What is fascinating here is the view of the product designer herself as an end-user. in the hierarchy thus developed, she is beholden to the (dis)affordances and inscriptions in the design environment that are themselves chosen and designed by the PoP.

Product designers do not operate in a vacuum, developing their products and services each time as if from a *tabula rasa*. To do so would in many cases involve ‘reinventing the wheel’, which would be costly, inefficient, and even potentially dangerous, because it would invite mistakes and bugs that have otherwise been refined out through decades of the reuse and incremental improvement of fundamental but complex computing mechanisms. An example would be encryption, whose crucially

¹⁰¹ *Ibid.*

¹⁰² Vismann and Krajewski, *supra* n. 3, p. 100 (italics are supplied emphasis; the underline is mine).

important algorithms have been developed and improved across the decades to a point where they can reasonably be relied upon. If these were to be (re)implemented on a project-by-project basis, the result could be disastrously unstable. Designers utilise hardware, programming languages, libraries, and habitual development practices that are in place well before they embark upon the development of new code – this model of incremental improvement and sharing is the cornerstone of the open source movement, from which proprietary business models also draw.¹⁰³ The result is often a bricolage of the old and the new, brought together with the ‘glue’ of a particular designer’s skills, knowledge, and interpretation of the design brief. Designers are situated within a context of decades of this prior art, and their approach to their work will to a greater or lesser extent be guided by all those practices that have gone before, and the technological mediations that bear upon their ability to solve the problem at hand. Before she considers her immediate task, then, she is starting out within an architecture that is itself replete with inscription and (dis)affordance, which mediates how she can and does go about her work.

At this point we can recall again the model of normative relationships in Figure 1 in Chapter 1.¹⁰⁴ We begin to see a parallel between on the one hand the top-down arrangement of constitutive rules that bind the state ‘to the mast’ in its promulgation of legislation,¹⁰⁵ and on the other a bottom-up framework of constitutive normativity that enables and constrains the scope of generativity available to the product designer. The (dis)affordances and inscriptions created by PoPs creates a kind of constitution, delimiting the framework within which the day-to-day ‘parliamentary’ work of the product designer creating technological normativity takes place. The normative power of design is thus deeper than just that of the product designer; it extends into the

¹⁰³ See generally G. Moody, *Rebel Code* (New York: Basic Books, 2002).

¹⁰⁴ Chapter 1, section 1.4.1 (Normative relationships online).

¹⁰⁵ The Odyssean idea of constitutional binding is discussed in greater detail in Chapter 3, section 2 (What is legalism?).

technical ‘constitution’ that makes up the foundation of the design process itself. Like a legal constitution, this technical foundation has implications for the artefacts built upon it in the higher levels of the technical ‘stack’. One of the most important tools through which the PoP exercises this constitutional role is the integrated development environment, or IDE.

3.6.1 *Integrated development environments*

Integrated development environments (IDEs) are software tools ‘intended to assist the software lifecycle processes’.¹⁰⁶ IDEs vary in complexity and sophistication, from applications that are essentially simple text editors with some added programming-specific functions, to large-scale programming suites with in-built compilers/interpreters, build-automation tools, version control, debuggers, code hinting tools, etcetera. According to the Institute of Electrical and Electronics Engineers (IEEE), IDEs assist software lifecycle processes through (i) a reduction in cognitive load, (ii) the automation of repetitive, well-defined actions in order to enable creativity, (iii) a reduction in administrative load through the support of particular software engineering methods (for example object-oriented programming), and (iv) better systematisation of the software development process.¹⁰⁷

The extent to which these goals are achieved is contingent on what the design of the IDE affords the product designer.¹⁰⁸ Kline and Seffah discuss numerous usability issues prevalent in IDE designs, including hiddenness of assistive features, memory overload through overcomplicated interfaces, unhelpful error messages, lack of

¹⁰⁶ A. Abran et al., *Guide to the Software Engineering Body of Knowledge (SWEBOK)* (Los Alamitos, California: IEEE Computer Society and Angela Burgess, 2004) p. 10-1.

¹⁰⁷ *Ibid.*, ch. 10.

¹⁰⁸ R.B. Kline and A. Seffah, ‘Evaluation of Integrated Software Development Environments: Challenges and Results from Three Empirical Studies’ (2005) 63 *International Journal of Human-Computer Studies* 607.

contextual feedback, and insufficient assistance in preventing obvious mistakes.¹⁰⁹ What this demonstrates is how the design environment affects the product designer's work in myriad ways, some subtle and some potentially more profound. As Graham puts it, programming languages and tools are 'not just technologies, but habits of mind.'¹¹⁰ This accords with the idea of inscriptions settling over time;¹¹¹ the more the designer is subject to the (dis)affordances of a particular IDE the more reinforced the guidance of its inscriptions will become for her.¹¹² This is true even though ultimately all mainstream programming languages are Turing-complete and are therefore in principle capable of precisely the same range of computational operations. Despite this, some languages are tailored towards solving particular forms of problem or creating particular forms of application, all within the framework of a particular 'habit of mind'. For example, some languages (e.g. LOGO or Smalltalk) are designed to mirror human mental models of how computation works and so they afford, *ab initio*, a form of transparency¹¹³ that is absent in more complicated languages that require a greater pivot away from conventional ways of thinking. Other approaches, such as Knuth's 'literate programming' paradigm and the WEB language built around it, aim to prioritise human ways of thinking above computational logic, with the latter adapting to the former rather than the converse as is commonly the case with mainstream languages.¹¹⁴ The normative effects of these languages and paradigms are in place before the designer actually writes any code; the design choices of the PoP thus necessarily facilitate a particular set of conditions, from the off, which condition the environment in potentially crucial ways.

¹⁰⁹ *Ibid.*, pp. 613–614.

¹¹⁰ P. Graham, 'Beating the Averages' (2003) <<http://www.paulgraham.com/avg.html>>.

¹¹¹ Van den Berg and Leenes, *supra* n. 53, pp. 77–78.

¹¹² Swierstra and Waelbers, *supra* n. 72, p. 158.

¹¹³ I discuss the affordance of transparency further in Chapter 5.

¹¹⁴ D.E. Knuth, 'Literate Programming' (1984) 27 *The Computer Journal* 97. See also Weizenbaum, *supra* n. 2, p. 109f.

I have here briefly described the power of the PoP in creating an underlying normative framework within which the product designer carries out her work. Later in Chapter 6 I invert this descriptive perspective to consider how the privileged position of the PoP enables the imposition of limits on the product designer that can be leveraged for normatively-desirable purposes, binding the work of the latter so that the code she produces exhibits the qualities of legitimacy that I set out in later chapters. In this way, just as a constitution binds the work of a legislature *ex ante*, so too can the design environment contribute to the creation of legitimate code.

4. Conclusion

The discussion above used the theories of affordance, disaffordance, inscription, and technological mediation to set out the ways in which code regulates the behaviour of end-users. This demonstrates the power that designers have to constitute and regulate end-users' behaviour. I will pick up that analysis again in Chapter 5, where I apply aspects of the theories set out above to shed light on how the subjects of the case studies impose normativity on end-users. Designs that embody more constitutive normativity mean that the balance of power favours the designer and behavioural constraint is potentially more absolute. This is how one might say that code is *more* than law. It is, however, possible for code to embody regulative rather than constitutive normativity, in order to shift (some) power back to the end-user. Striking an appropriate balance between constitutive and regulative normativity is an important part of the digisprudential perspective that I later develop.

This chapter also considered how design environments themselves have a regulating effect on the work of product designers, which paves the way in Chapter 5 for a consideration of how the production of user-facing affordances might be guided by the 'constitutionality' of designer-facing affordances. One can appreciate the parallel

with Hart's jurisprudential concept of primary and secondary rules¹¹⁵ – what I am describing is in effect their equivalent in the design realm, where it is the normativity of code that is constitutionally constrained, rather than the normativity of legal rules. A guiding element of this thesis is the fleshing out of the analogy between the two orders on that basis.

The next chapter builds upon the design theory presented here to deepen the theoretical connection between law and design and to propose the concept of *computational legalism* as a foundation for the thesis' ultimate contribution.

¹¹⁵ H.L.A. Hart, *The Concept of Law* (2nd edn., Oxford: Clarendon Press, 1994) pp. 91–93.

Chapter 3

A legal theory perspective: code is *less* than law

The alternative to legality is not anarchism, it is legalism... ‘not thinking about it’, if left to its own devices, tends to take over the entire social world, or at least cyberspace.¹

1. Introduction

In the previous chapter, I used design theory to describe how the code of artefacts operationalises behavioural regulation. That was the first task in setting out the theoretical grounding of the thesis, demonstrating the directness of code-based regulation. This chapter’s contribution is an analysis of code’s regulative characteristics from a legal theory perspective, from which I develop the concept of *computational legalism*. This idea is borne of the parallel I observe between code’s ruleishness – its reliance on strict, binary logic instead of interpretable standards – and its conceptual equivalent in the jural realm, known as *legalism*. Legalism is a perspective that eschews a holistic interpretation of legal norms, instead requiring that citizens merely follow legal rules as they are presented to them, without enquiring as to their efficacy or their legitimacy beyond the question of where they come from. Code’s characteristics exemplify a particularly strong form of legalism, and therein lies the problem of unlegitimated code-based regulation, and the claim that it is *less* than law. As Wintgens puts it in relation to traditional legislation, ‘[r]ule creation is a matter of choice, and this choice is legitimated because it is based upon the democratic character of the regulating process’.² It is not at all clear that such aspirations are reflected in the production of code, according to which end-users are ‘induced, habituated and, if

¹ Z. Bańkowski and B. Schafer, ‘Double-Click Justice: Legalism in the Computer Age’ (2007) 1 *Legisprudence* 31, pp. 47–48.

² L. Wintgens, *Legisprudence: A New Theoretical Approach to Legislation* (Oxford: Hart, 2002) p. 2.

necessary, compelled, to accept the norms of commercialized cyberspace',³ all of which often takes place outwith democratic debate and legal processes of interpretation, contest, and remediation. As Longford puts it,

[W]hereas the terms and conditions of political citizenship in liberal democratic states are, relatively speaking, subject to free, open and transparent deliberation and negotiation, the codes governing the citizen in the digital era are invisible and opaque, thanks to certain features of the technologies themselves and to the proprietary nature of many of the codes increasingly mediating our lives.⁴

The purpose of this chapter is not to suggest that designers harbour a legalistic ideology. Rather, the intention is to demonstrate how certain characteristics of a legalistic mentality are closely reproduced in the material architectures of code – the ideological *ought* of legalism becomes the technological *is* of code. If we proceed from the starting point that legalism is an undesirable thing in a democracy, then the mechanisms for mitigating it in the traditional legal sphere might also have an ameliorating effect in the analogous context of code-based 'legislation'.

In drawing a parallel between the legalistic outlook in the orthodox legal context on the one hand, and intensified legalism of code on the other, the stage is set for an analysis of the ways in which its mitigation in the former can be imported into the sphere of the latter, strengthening the conceptual relationship between foundational legal theory and the computational paradigm. The aim is to investigate the 'new forms of interaction' that Bańkowski and Schafer suggest are necessary to 'promote the benefits of legality, and to prevent the disadvantages of legalism.'⁵ This chapter explores the latter, setting out how it is characterised in the literature before

³ G. Longford, 'Pedagogies of Digital Citizenship and the Politics of Code': (2005) 9 *Techné: Research in Philosophy and Technology* 68, p. 71.

⁴ *Ibid.*

⁵ Bańkowski and Schafer, *supra* n. 1, p. 46.

demonstrating how it is that code, when it operates as an enforcer, mediator, and generator of legal reality, can be a particularly extreme incarnation of this ideological perspective. As Wintgens notes, ‘long decades of legalism in legal reasoning [have meant that] the dominant views in legal theory... have barred the way for questioning the position of the legislator’.⁶ Questioning the position of the designer, *qua* legislator, is precisely what this thesis aims to do.

2. What is legalism?

There are numerous conceptions of legalism in the literature, with the term occasionally being used interchangeably with different concepts such as legality and the rule of law – Bańkowski and Schafer, for example, note that legalism is ‘often confused with legality, an altogether more reflexive and rational concept’.⁷ MacCormick contended early on that legalism is ‘a prerequisite of free government’,⁸ but this seemed for him to amount essentially to the ex post doctrine of *nulla poena sine lege* (‘acts of government however desirable teleologically must be subordinated to respect for rules and rights’⁹), and is therefore different from the stronger version of legalism with which I am concerned. Indeed, MacCormick explicitly distinguishes between that conception of legalism and the ‘stronger’ conception¹⁰ that comes from the work of scholars including Shklar, Bańkowski and Schafer, and Wintgens. MacCormick’s conception of legalism is akin to Wintgens’ idea of ‘weak’ legalism, which forms a part, rather than the whole, of the jurisprudential conception of legitimacy that I make use of in Chapter 4. At any rate, this intermediate position, later adopted by MacCormick himself in work written alongside Bańkowski, views

⁶ Wintgens, *Legisprudence*, *supra* n. 2, p. 2.

⁷ Bańkowski and Schafer, *supra* n. 1, pp. 31–32.

⁸ N. MacCormick, ‘The Ethics of Legalism’ (1989) 2 *Ratio Juris* 184.

⁹ *Ibid.*, p. 184.

¹⁰ *Ibid.*

some measure of legalistic rule following as a necessary element, but not the whole, of a functioning legal order. This accords with the normative position I develop, whereby rules are the correct basis for regulating behaviour, but the process of their development is constrained so as not to be arbitrary. This idea is embodied in the Greek legend of Odysseus, in which the eponymous captain has himself bound to the mast of his ship so that he cannot succumb to the enticement of the Sirens. The legend's metaphor of a sovereign consciously limiting itself in order to avoid the temptation of iniquity is one that has been considered by various scholars in their discussions of constitutionalism in the computational context.¹¹ Hildebrandt contrasts legality and legalism thus:

Legality, in this sense, refers to justice (proportionality), to legal certainty (the legal ground in positive law, with the necessary safeguards) and purposiveness or expediency (the legitimate aim of the intervention, the requirement of effective remedies). Legalism, instead, reduces all this to the correctly enacted legal ground, which may or may not offer any protection, leaving the subject of government interventions dependent on a rule *by* law instead of the Rule of Law. Even if the sovereign that rules *by* law is the *nation* or the *Parliament*, legalism leaves individual subjects without effective remedies against arbitrary rule.¹²

One can see here the implication in the concept of legality that the rules promulgated must be designed to reflect certain ideals (proportionality, safeguards, the substantive legitimacy of the norm itself). Legalism, by contrast, is concerned only that the rule has been promulgated by a legitimate institution, and cares not what its content or substantive effects are.

¹¹ See for example M. Hildebrandt, *Smart Technologies and the End(s) of Law: Novel Entanglements of Law and Technology* (London: Edward Elgar Publishing, 2015) p. 156; L. Lessig, *Code: Version 2.0* (New York: Basic Books, 2006) pp. 313–314.

¹² M. Hildebrandt, 'Radbruch's Rechtsstaat and Schmitt's Legal Order: Legalism, Legality, and the Institution of Law' (2015) 2 *Critical Analysis of Law* 42, p. 56 (emphasis supplied).

Hildebrandt's characterisation of legalism, where there is an absence of protection against arbitrary rule, matches the stronger variant of the concept. For example, in her seminal text on the subject, Shklar defines legalism as 'an ethical attitude that holds moral conduct to be a matter of rule following, and moral relationships to consist of duties and rights determined by rules.'¹³ For her and other theorists, legalism is a deontological perspective, which is to say it is an ethical perspective built around the duty to follow rules, as opposed to (for example) a consequentialist outlook focused on achieving normatively desirable outcomes, or a virtue ethics perspective aimed at 'living well'.¹⁴ This kind of outlook – termed the 'morality of duty' by Lon Fuller¹⁵ – has a long pedigree in moral philosophy, perhaps most evidently in the work of Kant¹⁶ who, according to Bańkowski and MacCormick, was the 'high priest for a rule based morality'.¹⁷ Without delving into the detail of this seminal pillar of Western philosophy, one can say in brief that such an approach has moral force because it results in a normalisation and systematisation of behaviour across society, which in turn begets the kind of behavioural predictability that has been argued is a desirable goal in the development of a stable (capitalist¹⁸) society.¹⁹

¹³ J.N. Shklar, *Legalism* (Harvard University Press, 1964) p. 1.

¹⁴ S. Blackburn, 'Deontological Ethics', *The Oxford Dictionary of Philosophy* (Oxford University Press, 2008).

¹⁵ L.L. Fuller, *The Morality of Law* (Yale University Press, 1977) ch. 1.

¹⁶ Z. Bańkowski, 'Don't Think About It: Legalism and Legality' in M.M. Karlsson, Ó. Páll Jónsson and E.M. Brynjarsdóttir (eds.), *Rechtstheorie: Zeitschrift für Logik, Methodenlehre, Kybernetik und Soziologie des Rechts* (Berlin: Duncker & Humblot, 1993) p. 45; Bańkowski and Schafer, *supra* n. 1, p. 33.

¹⁷ Z. Bańkowski and N. MacCormick, 'Legality without Legalism' in W. Krawietz et al. (eds.), *The Reasonable as Rational? On Legal Argumentation and Justification; Festschrift for Aulis Aarnio* (Berlin: Duncker & Humblot, 2000) p. 183.

¹⁸ Bańkowski, *supra* n. 16, p. 48. See also L. Wintgens, 'The Rational Legislator Revisited. Bounded Rationality and Legislation', *The Rationality and Justification of Legislation* (Springer, Heidelberg, 2013) p. 4.

¹⁹ Shklar, *supra* n. 13, p. 64.

Heteronomy, the condition of being dominated by an external force (in this case rules), is antithetical to aspirations of reasoned interpretation and action.²⁰ It is exemplified in what Wintgens calls the ‘strong’ variant of legalism, which he characterises as a normative historical strategy²¹ – recall the distinction mentioned in this chapter’s introduction between the *ideological* legalism of the legal realm and the more *descriptive* legalism of the computational realm. On the other hand, it is also often thought that *some* measure of legalism (i.e. respect for rules *qua* rules, or ‘law as law’) is necessary for society to operate, and indeed that legalism should be understood normatively not as being in opposition to legality but rather as a necessary element of it.²² This accords with Wintgens’ concept of ‘weak legalism’, a viewpoint necessary for his theory of legisprudence, according to which rules remain the proper mechanism of behavioural regulation, but the potential for their arbitrary definition is simultaneously constrained. *Strong* legalism undermines legality, whereas *weak* legalism is a necessary (although insufficient) component of it. For Bańkowski and Schafer, legality properly understood requires a complementary combination of adherence to rules (legalism) with thoughtful interpretation of what is being commanded by the rule, with the correct approach varying depending on the circumstances.²³ For them, it is sometimes appropriate to be mindless in following a rule – for individuals to ‘act like automata’ – while at other times it is necessary to act autonomously, considering what the rule asks of us before deciding how to act. Strong legalism implies only the former approach, where weak legalism is the rule-based element of the broader concept of legality.

This strong conception of legalism is very relevant to a descriptive analysis of code, because as we shall see the latter exemplifies its characteristics, and indeed

²⁰ See, for example, Bańkowski and Schafer, *supra* n. 1; Bańkowski and MacCormick, *supra* n. 17, p. 194; Bańkowski, *supra* n. 16, p. 56; MacCormick, *supra* n. 8, p. 192.

²¹ L. Wintgens, *Legisprudence: Practical Reason in Legislation* (Surrey: Routledge, 2012) p. 159.

²² Bańkowski, *supra* n. 16; MacCormick, *supra* n. 8.

²³ Bańkowski and Schafer, *supra* n. 1, p. 48.

amplifies them far beyond what Wintgens and other theorists describe. As Bańkowski and Schafer observe in the computational context, '[code's] unrestricted anarchism in the absence of the state has indeed resulted in the most absolute form of legalism possible.'²⁴

Subsequent references in this chapter to 'legalism' are to its strong variant, unless otherwise specified. The remainder of this section sets out the theory of legalism and its approach to law-making, with a view to setting the stage for a comparison between it and code.

2.1 Solipsism and positivism

Legalism is rooted in a solipsistic view of law as a system of rules and practices that operates separately from the social contexts within which it is embedded. Law is viewed as a 'clean' system, 'self contained and autogenerative',²⁵ that subsumes the outputs of the 'dirty business' of politics²⁶ (i.e. legislation) and applies them according to its own *sui generis* processes, institutions, and vocabulary. Already we have the first glimpse of the parallel with code.

Two important consequences flow from these characteristics. First, legalism seeks to insulate itself from forces which might invite questions as to why a particular rule should be followed in a particular circumstance. Second, in order to maintain its maxim that regulatees act unthinkingly, legalism must avoid any lacunae in the body of rules it provides – there must be a rule to deal with every circumstance that might arise, otherwise the agent will be left without guidance as to which course of action to follow.

²⁴ *Ibid.*

²⁵ Bańkowski, *supra* n. 16, p. 46.

²⁶ *Ibid.*

‘Legislation is a matter of politics, and politics is a matter of choice’, and so the ‘truthiness’ of law requires that it remain separate from anything so contingent.²⁷ Law is viewed as a scientific practice which identifies and works with ‘truths’, which are the product of sovereign legislators. The content of those truths is not to be questioned: from the perspective of the law and legal practice the truth ‘just is’, it is handed down from the political realm where it is the sole province of the legislator to debate the substance of the norm. The lawyer has no valid interest in what goes on there; politics is about choice, and therefore it does not deal in ‘truth’ because competing choices can never be objects of true knowledge.²⁸ Once the legislature chooses between the various possible options and crystallises its preference into a law, it becomes an item of ‘true’ knowledge relevant to the science of law, whose objective (extra-legal) quality is irrelevant to the legal ‘scientists’ who, from that point onward, take it as a datum for application within their field.

In this way, legal thinking becomes ‘fenced off’ from ‘all contact with the rest of historical thought and experience’.²⁹ The result is a positivist view that the law is ‘just there’, and it is not the task of citizens or practitioners to enquire as to how it got ‘there’.³⁰ What matters is whether it is a valid law, and not whether we agree with its substance or not. The ‘truth’ (‘is-ness’) of a given legal norm derives from the validity of its creation vis-à-vis actors and processes, and the question of whether its substance is desirable or not (its ‘ought-ness’) is properly to be viewed as separate from this.³¹ The preoccupation with the ex post examination of what should and should not be

²⁷ L. Wintgens, ‘Legisprudence as a New Theory of Legislation’ (2006) 19 *Ratio Juris* 1, p. 5.

²⁸ *Ibid.*

²⁹ Shklar, *supra* n. 13, p. 3. See also Bańkowski and MacCormick, *supra* n. 17, p. 182, and Fuller, in his response to Hart in their now-classic debate on morality and legal positivism: L.L. Fuller, ‘Positivism and Fidelity to Law: A Reply to Professor Hart’ (1958) 71 *Harvard Law Review* 630, p. 635.

³⁰ Bańkowski, *supra* n. 16.

³¹ Bańkowski and MacCormick, *supra* n. 17, p. 186.

considered ‘law’ is of course a core characteristic of analytical legal positivism. Strong legalism is connected with positivism in its drive to classify rules according to those that are internal to the legal system and those that are external.³²

Thus, from a legalistic (and analytical positivist) perspective, ‘what ought to be done is confined to the knowledge of the rules that contain rights and duties. Following rules is a matter of knowledge, while their enforcement is a matter of application.’³³ Legal practitioners take that knowledge, provided from somewhere ‘out there’, and use it as a tool to achieve a given legal aim. Their practice is ‘neutral’ as to the substance of these materials (rules), and they become technicians whose task it is to manipulate those rules according to the mechanisms of legal reasoning.³⁴

2.2 Legalism according to legisprudence

The preceding discussion set out how strong legalism is concerned with the application rather than the construction of rules.³⁵ In his comprehensive historical discussion of the origins of legalism, Wintgens discusses the theoretical mechanisms of legitimation in both natural law and analytical legal positivism, before identifying a set of specific characteristics of which the phenomenon of strong legalism is a ‘conjugation’, namely representationalism, a-temporality, concealed instrumentalism, *etatism* (the belief that the only true source of law is the state), and the scientific study of law.³⁶

The orthodox source of a rule’s legitimacy differs depending on the source of sovereignty – broadly, natural law or the social contract: respectively, that source is

³² L. Wintgens, ‘Legislation as an Object of Study of Legal Theory: Legisprudence’, *Legisprudence: A New Theoretical Approach to Legislation* (Oxford: Hart, 2002) p. 20.

³³ Wintgens, ‘Legisprudence as a New Theory of Legislation’, *supra* n. 27, p. 5.

³⁴ Bańkowski and Schafer, *supra* n. 1, p. 34.

³⁵ Wintgens, *Legisprudence*, *supra* n. 21, p. 139.

³⁶ Wintgens, ‘Legisprudence as a New Theory of Legislation’, *supra* n. 27, p. 5. For an in-depth philosophical and historical discussion of these characteristics, see Wintgens, *Legisprudence*, *supra* n. 21, ch. 5, especially p. 147 *et seq.*

either a transcendent set of natural law norms, or a social contract which founds a sovereign law-making institution. In the case of a natural law perspective, this is because the source of its substantive content is the ‘background’ knowledge of ‘jusnaturalistic’ (natural law) principles which are inherently true: such representational laws are ‘a concretisation of natural law, or reflect a natural law conception that in its turn legitimises positive law’.³⁷ In the case of the sovereign, this is so because the social contract legitimises such pronouncements as a descendent of some original founding contractual act of the people that set up the institution to represent them³⁸ (perhaps a document with constitutional status, although the social contract can also be a hypothetical moment rather than a real instrument³⁹). That the state defines what is legal is in itself enough to legitimise the substance of the legal norms it chooses to declare; in constituting the field of play (the legal system), the state legitimises *de facto* that which it consequently promulgates as the rules of the game. One can detect in this hierarchical idea of legitimacy Hart’s concept of the ‘rule of recognition’, or Kelsen’s *Grundnorm*.⁴⁰ The outcome in either case is the same, namely a legitimated foundational source from which laws can be promulgated that are themselves *de facto* legitimate as a result of the *a priori* legitimacy of the source, and that therefore ought to be followed.⁴¹

Wintgens describes this as ‘one-shot’ legitimation, operating continually thereafter to validate prospectively any norms promulgated according to the *ex ante*

³⁷ Wintgens, ‘Legislation as an Object of Study of Legal Theory: Legisprudence’, *supra* n. 32, pp. 10–11. See also *Legisprudence*, *supra* n. 21, p. 147 *et seq.*

³⁸ Wintgens, *Legisprudence*, *supra* n. 21, p. 195.

³⁹ *Ibid.*, p. 144.

⁴⁰ Wintgens explicitly identifies the connection between the Rousseauian ‘act of will’ that creates the social contract, and the Hartian system of a founding rule of recognition that is followed by emergence of a combination of primary and secondary rules (*ibid.*, p. 170).

⁴¹ *Ibid.*, p. 196 *et seq.*

framework that it sets up.⁴² Drawing on Hobbes and Rousseau, this is what he describes as the ‘proxy model’ of legitimation,⁴³ according to which the initial legitimation of the external decision maker, described above, permits it to act on behalf of the people (i.e. as a proxy) from that moment onward, despite the inability of either the sovereign itself or the people it represents to foresee all the rules, or ‘limitations on freedom’, that will in the future be imposed.⁴⁴ The citizen is given the imperative ‘not to think about it’; she need only act in accordance with the rule as it is given to her,⁴⁵ since by virtue of those constitutive facts the pronouncement of the sovereign is ‘imputed to [the citizenry], as if they were its author.’⁴⁶ This legalistic idea of minimal interpretation is connected with the positivistic view of textual interpretation, perhaps most famously articulated in Hart’s discussion of the ‘core’ and ‘penumbra’ in the interpretation of the meaning of individual words, the former being deemed to be settled and uncontested, and the latter being where controversies of interpretation arise.⁴⁷ This is discussed below in the section on rules in computational legalism.

In the commercial realm, society essentially gives the designer of code a one-shot ‘legitimation’ of this kind when we (i) endow her with the plasticity of code to create a near-infinite number of conditions which enable and constrain behaviour through technological normativity, (ii) we protect her privatised practices through (legally-sanctioned) commercial secrecy and a general absence of scrutiny, and (iii) we submit to the *sui generis* opacity of code. Each of these characteristics is discussed further in section 3 below.

⁴² *Ibid.*

⁴³ *Ibid.*, ch. 6 generally.

⁴⁴ *Ibid.*, p. 211.

⁴⁵ Bańkowski, *supra* n. 16.

⁴⁶ Wintgens, *Legisprudence*, *supra* n. 21, p. 208.

⁴⁷ H.L.A. Hart, ‘Positivism and the Separation of Law and Morals’ (1958) 71 *Harvard Law Review* 593.

Wintgens' discussion of the metaphysics of strong legalism is densely theoretical and goes into greater historical detail than is necessary for my purposes, which is unsurprising given that his aim is to situate the modern practice of legislation and rule-following within a fully-rendered picture of the historical and philosophical development of the state. However, the parallels between certain elements of his schema and the characteristics of code-as-law are striking. The most salient characteristics of the conjugation (representationalism, a-temporality, and concealed instrumentalism) are set out in the remainder of this section, before section 3 goes on to consider them directly in relation to code.

From a computational perspective, perhaps the most relevant element of strong legalism is *representationalism*. This is the view of law as a representation of reality, either through the latter's reproduction (in the case of natural laws that require to be interpreted by human positive law) or its construction (in the case of laws based on a founding social contract).⁴⁸ Wintgens embarks on a rich discussion of the philosophies of Hobbes and Descartes, covering the relationship between realism and nominalism and how these, despite their seemingly fundamental differences, can both result in the 'naturalisation of positive law', according to which law is deemed to be a representation of objective reality.⁴⁹ What the law states is therefore held to be true, either because natural laws are hypostatically true or because the social contract is true and therefore so too are the rules that are based upon it.

The salient connection with the computational context is that there representationalism is even more concrete than in the ideology of strong legalism:

⁴⁸ Wintgens, 'Legisprudence as a New Theory of Legislation', *supra* n. 27, p. 4. See also C.M. Campbell, 'Legal Thought and Juristic Values' (1974) 1 *British Journal of Law and Society* 13.

⁴⁹ Wintgens, *Legisprudence*, *supra* n. 21, pp. 147–153. This metaphysical discussion centres around the distinction between the view of law as directly reproducing reality (realism), or the view that the creation of law *constructs* reality through the description of what is otherwise 'semantically empty' (nominalism). In either case, the end result is the belief that law is representative of reality, of the world as it is.

whereas proponents of the latter hold the *belief* that the rule presents reality, in the computational context this is much more than mere belief because, as we saw in the previous chapter, code does not just *represent* reality but actively *constitutes* it. I talked about constitutive versus regulative technological normativity in the previous chapter, and will return to that theme in section 3.2.1 below.

The next salient component of strong legalism is *a-temporality*, which flows from the belief in law as a representation of reality. Because either the social contract or natural law represent reality *ex ante*, anything that flows from them is believed also to be true, since they are the genuine and true foundation of political space. That foundation is a-temporal because it is believed to be the universal principled basis for public law, something that is valid separately from man's recognition of it.⁵⁰ Contingent laws built upon this foundation are deemed to 'uncover' the general will, rather than proactively to reflect it (those who disagree with a particular legislative proposition are in error as to what the general will is, rather than in disagreement *per se*). The general will exists at all times, ready to be uncovered and recognised by contingent legislative acts. Thus, 'acts of will then take on the appearance of timelessness'.⁵¹ Admittedly, this is a difficult metaphysical proposition that might appear more relevant to Wintgens' historical analysis of political philosophy than to the current argument regarding code. Despite this, the idea of timelessness does connect with the immutable character of code, discussed further below, and the approach that Wintgens develops to cope with timelessness in law-making – the jurisprudential principle of temporality – does become relevant as a practical consideration (this will be considered in Chapters 4 and 5).

The final relevant aspect of strong legalism is *concealed instrumentalism*. This is the idea of a 'veil of sovereignty', behind which the values or ends of the legislator are

⁵⁰ *Ibid.*, p. 155.

⁵¹ *Ibid.*, pp. 156–157.

hidden. It is evidenced in analytical positivism and textualism, where the policy goal of the legislator is less important than formalistic reasoning from the text. The fiction of timelessness previously discussed combines with this instrumentalism to form a strategy for converting the messy and contingent political into the clean and scientific legal.⁵² This concealment finds its analogue in the legal and economic veils that protect code: enterprise is protected by trade secrets and anti-circumvention laws, while faith is placed in the market to curb any excesses. This is discussed further below in section 3.5.

As previously mentioned, whereas strong legalism and analytical positivism are concerned primarily with the validity of a norm's source, legisprudence suggests that this is a necessary but insufficient condition for legitimacy. Not only must the sovereign be 'bound to the mast', but so too must it actively *legitimate* the norms that it proposes. This is a type of validity that to an extent crosses the line between formal and substantive legitimacy – the substantive content of the norm is constrained according to certain formal qualities that are embodied in the principles of legisprudence. As Wintgens argues:

The basic idea of the rule of law or the *Rechtsstaat*, that both the ruler and the ruled are bound to rules, can be interpreted in two ways. The first interpretation is the path to strong legalism. According to this approach, the ruler's being bound to rules is tantamount to his 'not violating' them. This is both a necessary and a sufficient condition for rules to be valid and legitimate. Under the second interpretation – which is adopted by legisprudence – the idea of following rules by a sovereign counts only as a necessary and not as a sufficient condition for rules to be valid. Legal validity on this view is distinct from legitimacy. Legitimacy for its part can only be obtained through legitimation.⁵³

⁵² Wintgens, 'Legislation as an Object of Study of Legal Theory: Legisprudence', *supra* n. 32, p. 158; Bańkowski, *supra* n. 16, p. 46.

⁵³ Wintgens, *Legisprudence*, *supra* n. 21, p. 145.

For him, then, the achievement of legitimacy requires an additional active step of legitimation, which means the sovereign is not just subject to the same rules as everyone else (i.e. the rule of law, or the Odyssean ‘binding to the mast’), but also that the rules which it seeks to promulgate reflect certain formal characteristics, which in turn logically limit the breadth of possible content that those norms can legitimately have.

Despite its simplicity, then, the strongly legalistic worldview is clearly open to abuse: the prioritisation of heteronomy militates against critical reflection and the application of the underlying principles of legality that are characteristic of democracies. The freedom to interpret is a crucial aspect of legality, without which rules become ‘implements of tyranny’ and legalism a ‘vice of narrow governance’.⁵⁴

By contrast, in addition to a formally-valid source of the rule, legisprudence requires the legitimation of the proposed legislative act through its justification. This justification is achieved according to the principles that legisprudence sets out, which guide the conduct of the ruler regardless of the political content of the norm she is making: ‘though his ruling activity, while following rules, the ruler must supply reasons for his choices.’⁵⁵ One can thus see how legisprudence represents a form of constitutionalism. I discuss the principles of legisprudence in greater detail in Chapter 4. For now, I turn to the parallel between legalism and code, to develop the concept of computational legalism.

3. Computational legalism

Where legalism requires that citizens ‘not think about it’ and simply follow the rule, so too does code when it enables and constrains behaviour. There is a difference of degree, however, since the legalistic mindset is at least something that can be

⁵⁴ Bańkowski and MacCormick, *supra* n. 17, p. 194.

⁵⁵ Wintgens, *Legisprudence*, *supra* n. 21, p. 145.

challenged by a rational individual, and in any event the hermeneutic gap by definition creates a space between the promulgation of the norm and any acquiescence to it. Code, by contrast, admits of no opportunity for challenge: some measure of normativity is inherent in its very existence – the boundaries of the field of play, as well as the rules of the game, are determined from the outset, and there is little or nothing the end-user can do to change them, if she is even aware of what they are to begin with (which as we shall see is far from a given). Thus, not only is she made to ‘not think about it’, she is in many cases not given the opportunity to apprehend what it is that she is not thinking about.

Returning to the characteristics of legalism set out above, we can think about how they apply in the context of code, creating what I term *computational legalism*, the particular species of legalistic obedience that flows from the *sui generis* nature of code. Firstly, we saw how legalism concerns itself with rules that are to be followed as written. Code presents us with an extreme form of ‘ruleishness’, where conditions are hard-edged and admit of no latitude for interpretation. Secondly, legalism views those rules as a representation of reality. We saw in the previous chapter how the code of digital architectures constitutes, not merely represents, the empirical and legal realities that the end-user is presented with (and indeed those she cannot perceive). Thirdly, under legalism rules are seen as a-temporal, or timeless – they reflect background truths, and they ‘just are’. Similarly, code in a sense ‘collapses’ time, through a combination of the immediacy of its execution, its immutability, and the cumulative normativity of its pervasiveness. Fourthly, the source of the sovereign’s power is concealed so that the policy reasons behind the normativity it promulgates are ignored, those norms being treated as simply ‘there’, to be followed without question. The opacity of code, and the privileging of commercial practices and trade secrets, set up a similar concealment in the computational context – the ‘sovereignty’ of those who produce code is concealed by veils of both technical and economic opacity.

The remainder of this section considers these characteristics in more detail, demonstrating how the computational form of legalism is particularly strong, making the application of measures for mitigating it all the more important.

3.1 Ruleishness

Hildebrandt discusses the historical consequences of the embodiment of law in text, where there is a *choice* as to whether to take a legalistic interpretation or not, unlike with code where there is no such latitude. Writing about interpretation, she observes that texts and textual norms have lives of their own:

Absent ostensive reference, the author is never sure how her text will be understood, while the reader cannot take for granted what the author meant to say. This provides for an inevitable latitude in the use of texts and turns law-making (enactment of legal codes as well as their application) into a creative process rather than a mechanical application.⁵⁶

This characterisation of law cleaves to the aspirational view of legality, where the slow iteration of interpretation of norms across heterogeneous circumstances builds towards a body of law that is simultaneously flexible and stable.⁵⁷ As we have seen, this is at odds with strong legalism's binary application of rules *qua* rules.⁵⁸ This 'ruleishness'⁵⁹ is one of the most salient homologies between jural and computational legalism. Taking the orthodox spectrum that at one end has absolute rules that admit of no interpretation, and at the other has more flexible standards that specify broader outcomes but not the detailed means by which they should be achieved, code is very

⁵⁶ M. Hildebrandt, 'Legal and Technological Normativity: More (and Less) than Twin Sisters' (2008) 12 *Techné: Research in Philosophy and Technology* 169, p. 172.

⁵⁷ *Ibid.*, pp. 171–172.

⁵⁸ Bańkowski and Schafer, *supra* n. 1, p. 34.

⁵⁹ J. Grimmelmann, 'Regulation by Software' (2005) 114 *The Yale Law Journal* 1719, *passim*.

much located towards the rule end.⁶⁰ Code execution represents the mechanical application of rules *par excellence*; in Zittrain's phrasing, 'execution is exquisite'.⁶¹ Simultaneously, code's creation (an 'enactment of legal codes') does not admit of the latitude of interpretation referred to in the quote from Hildebrandt above. The rule as laid down by the designer is the rule that will be followed. Whereas traditional laws are purposefully created for 'offline' application, where inefficiency and delay permit consideration of exceptional circumstances or evolving social norms, code 'requires extreme precision and rigor not resident in analog law.'⁶²

3.1.1 *Three consequences*

There are therefore three fundamental consequences that flow from code's ruleishness. First, the rules specified *ex ante* in the code will be applied in *all* instances where the conditions they require are present, regardless of any *ex post* considerations (although of course inputs at run-time, from the end-user or some other source such as a sensor or an oracle, will in many cases be required for the *ex ante* specifications of the rules to be met). In the technical context this is of course a major benefit: even the most complex body of rules can be expected to execute in pre-determined ways under precisely-defined and controlled conditions, giving a notional predictability that has facilitated the rapid innovation that society has enjoyed over the past several decades.

This is connected with the automated nature of code, which means that once it is released into a production environment, it can repeat the same set of operations millions or billions of times with little or no marginal cost and with no human intervention beyond the maintenance of computing and energy infrastructure and any

⁶⁰ *Ibid.*, p. 1723. See also K. Yeung, 'Can We Employ Design-Based Regulation While Avoiding Brave New World?' (2011) 3 *Law, Innovation & Technology* 1.

⁶¹ J. Zittrain, *The Future of the Internet and How To Stop It* (New Haven, Connecticut: Yale University Press, 2008) p. 107.

⁶² L.A. Shay et al., 'Do Robots Dream of Electric Laws? An Experiment in the Law as Algorithm' in R. Calo, A. Froomkin and I. Kerr (eds.), *Robot Law* (Edward Elgar Publishing, 2016) p. 274.

end-user input necessary for its operation.⁶³ Again, provided the conditions specified in the code are met, the code will execute automatically, regardless of any other consideration, provided that it is formally valid. Indeed, this point about formal validity is one of the problems of code's instrumentalism: the machine will execute semantically correct commands faithfully and with no regard to their consequences which, depending on the behaviour and the pervasiveness of the code in question, can be catastrophic.⁶⁴ Back in the legal realm, this is quite evidently undesirable. Even the most 'ruleish' of textual legal norms requires interpretation in order to move from the page to behavioural instantiation, and even where the rule is one of strict liability (for example a speed limit for drivers), enforcement still requires an active process of interpretation, in the course of which justificatory or excusatory reasons may come to light which modulate a strongly legalistic application of the original rule (e.g. the driver was rushing to get her injured passenger to hospital).⁶⁵

Secondly, as long as the precise conditions specified in the rule do *not* exist, it will *never* be executed. No matter how closely the code-based rule matches the circumstances that arise in operation, if the two do not match then the code will sit inertly, doing nothing. Taken together with the first characteristic, there is therefore in code an emphatic absence – and indeed impossibility – of Hart's concept of the 'penumbra of doubt':⁶⁶ there can only be the core of meaning, except that the core

⁶³ Grimmelmann, *supra* n. 59, p. 1729. See also Yeung, *supra* n. 60.

⁶⁴ H. Surden, 'Values Embedded in Legal Artificial Intelligence' (Rochester, NY: Social Science Research Network, 2017) SSRN Scholarly Paper ID 2932333 p. 5 <<https://papers.ssrn.com/abstract=2932333>>. For a recent example that continues to affect both hardware and software at the most fundamental level of execution, see 'Meltdown and Spectre' (2018) <<https://meltdownattack.com/>>.

⁶⁵ Shay et al. use the example of speed cameras in their discussion of the practical difficulties inherent in transposing a textual norm into code. See *supra* n. 62.

⁶⁶ Hart, *supra* n. 47. Connected is Dworkin's idea of integrity, and the background influence on interpretation of guiding principles behind the enterprise of law. See R. Dworkin, 'Law as Interpretation' (1982) 9 *Critical Inquiry* 179.

expressed in the code reflects the subjective understanding of the designer, and not necessarily the settled meaning understood by the legislature, courts, or society more generally.⁶⁷ As Grimmelmann notes, the ‘hard edges’ of software rules are not susceptible to blurring, no matter how complex the set of rules is that is being applied⁶⁸ (which, in the modern computing context, will invariably be staggeringly complex). Whereas a human’s ability to apply simultaneous rules might result in less precision as the set increases, there is no such limitation for code (save perhaps that speed of execution might decrease).

Thirdly, and as a corollary of the above two characteristics, code’s ruleishness limits by definition the conditions that it will respond to. This limiting of possibilities is put in place by the designer, who of course is interested in solving a particular problem by a particular set of technical means, each of which is considered from the perspective of the underlying business model. In so doing, however, she may fail to consider the other possibilities that were relevant to the situation, thus reducing the world to a limited set of conditions and responses. From her perspective, she intends that in the operation of the system conditions *A*, *B*, or *C* will arise, and the system should respond with one or a combination of *X*, *Y*, or *Z*. These conditions comprise the *ontology* of the code,⁶⁹ which once it is defined is rigid and cannot (without the code being altered⁷⁰) be made to be sensitive to, for example, conditions *D* or *G*, or

⁶⁷ Work on ‘fuzzy logic’ aims to map the indeterminacy of language onto the determinacy of numbers; see L. Philipps and G. Sartor, ‘Introduction: From Legal Theories to Neural Networks and Fuzzy Reasoning’ (1999) 7 *Artificial Intelligence and Law* 115, p. 122 *et seq.* Such approaches are experimental and by no means the norm in design, however.

⁶⁸ Grimmelmann, *supra* n. 59, p. 1733.

⁶⁹ D.C. Dennett, *From Bacteria to Bach and Back: The Evolution of Minds* (London: Penguin Books, 2018) pp. 60–70, and particularly p. 68.

⁷⁰ Research in defeasible and non-monotonic reasoning in legal AI could feasibly help to mitigate this rigidity, although further research would be required to render these approaches practically useful in the design context. See for example G. Governatori and S. Sadiq, ‘The Journey to Business Process

responses *W* or *Q*. What this means is that the designer's predetermined view of the code's operation (conditions *A*, *B*, and *C*, and potential responses *X*, *Y*, and *Z*) is reified, and while that reification may not reflect the empirical reality of the world, or the requirements of substantive law or some other relevant normative value such as legitimacy or the rule of law, this fact will pose no barrier whatsoever to the execution of the code on the basis of the ontology predetermined by the designer.

As Grimmelmann puts it, '[w]hen a programmer creates a program, she predetermines its responses to every possible input – to every possible "case" it may adjudicate. The algorithm is the rule.'⁷¹ This connects with Hart's discussion of the open texture of language, where he argues against attempts to regulate unambiguously in advance:

If the world in which we live were characterized by only a finite number of features, and these together with all the modes in which they could combine were known to us, then provision could be made in advance for every possibility... Everything could be known, and for everything, since it could be known, something could be done and specified in advance by rule. This would be a world fit for 'mechanical' jurisprudence. Plainly this world is not our world.⁷²

Code makes this vision a reality, although not in the way Hart imagined. It goes further by *imposing* such 'mechanical jurisprudence' on a contingent and complicated world, reducing its complexities to the set of features that the designer chose to include, whether or not those features are sufficient or even appropriate for the context in which the code will operate.

Compliance' in J. Cardoso and W. Van der Aalst (eds.), *Handbook of Research on Business Process Modeling* (IGI Global, 2009).

⁷¹ Grimmelmann, *supra* n. 59, p. 1732.

⁷² H.L.A. Hart, *The Concept of Law* (2nd edn., Oxford: Clarendon Press, 1994) p. 128.

Bańkowski makes an early connection between legalism and code-based regulation in his description of a hypothetical system for controlling the borrowing of e-books from a university library.⁷³ Although he does not use the term, what he describes is what we would now call a DRM system. His description shows how the transition from an ‘offline’ manual library system to an automated system affects the rules that governed the former. Those rules state that:

- (1) borrowers must complete a separate form for each volume borrowed,
- (2) books should be returned before the due date,
- (3) borrowers have a limited number of loans that must not be exceeded,
- (4) no further books will be loaned to borrowers who have overdue loans.⁷⁴

The above regulations are automated by their transposition into code, which governed the release of an ebook that would ‘self-destruct’ after the appropriate borrowing period.⁷⁵ This is an archetypal DRM system, where *ex ante* constraints on media use define its availability and are embedded in and enforced by the artefact itself. The computational legalism of such a system becomes evident when we consider what happens to the textual rules listed above, which previously under the manual system were interpreted and applied by the human librarian. The rules are bright lines that admit of no interpretation – once the borrowing limit is reached, if no appeal process is built into the code then no further books can be borrowed, regardless of any extenuating circumstances (recall the third consequence of code’s ruleishness discussed above). Once the borrowing period is reached, the book ‘self-destructs’, again regardless of any extenuating circumstance that might have moved a human librarian to make an exception (e.g. a combination of illness and exams).

⁷³ Bańkowski, *supra* n. 16, p. 54 (‘Norms and Machines’), *et seq.*

⁷⁴ *Ibid.*, p. 55.

⁷⁵ *Ibid.*

In sum, then, there is no room for discretion, ambiguity, or subversion,⁷⁶ and nothing outside the realm of rules can be implemented internally within the machine. Digital systems are thus ‘crude and inflexible, often brutal and not open to critical reason.’⁷⁷ Returning to the example above, the speed camera detects two conditions (the car is travelling below the speed limit; the car is travelling above the speed limit) and it has two responses (do not take photo; take and process photo). In spite of the legalism of such a system, there is nevertheless an ex post buffer to enable some interpretation – a human may interpret the photo and, upon realising the vehicle was an ambulance, override the automated decision. The decision to include such oversight (i.e. a ‘human in the loop’⁷⁸) is a design choice and is by no means a given – there is no technological barrier preventing a speed camera and penalty system being fully automated with no ex post adjudication.⁷⁹ Such scenarios demonstrate how the three elements of ruleishness can come together to amplify legalism in the computational context, especially when they are combined with the automation and immediacy of code.

3.2 Representationalism

Representationalism is a key aspect of strong legalism – in both natural law and positivist accounts, law is held to represent reality. As discussed above, for the former its validity comes from the underlying truth of nature, while for the latter it comes from the founding social contract, rule of recognition, etcetera. If the underlying

⁷⁶ Grimmelman, *supra* n. 59, p. 1723.

⁷⁷ Bańkowski and Schafer, *supra* n. 1, p. 46.

⁷⁸ W. Hartzog et al., ‘Inefficiently Automated Law Enforcement’ [2015] *Michigan State Law Review* 1763, p. 1780 *et seq.*

⁷⁹ L.A. Shay et al., ‘Confronting Automated Law Enforcement’ in R. Calo, A. Froomkin and I. Kerr (eds.), *Robot Law* (Edward Elgar Publishing, 2016).

natural norms or the founding political act are true, then the rules which flow from them must also be true.⁸⁰

The way in which code constructs a rule-based normativity is not the same as law's approach. Law is limited by its instantiation in the technology of the script (text), which creates a hermeneutic gap between what the text requests and how the addressee(s) of that text interpret its terms and choose to reflect them in their behaviour.⁸¹ The instantiation of code rules, on the other hand, is not so limited – technological normativity, as we saw in the last chapter, can have a direct effect in a way that legal normativity that is constrained by its textual embodiment does not.⁸²

One crucial way in which code is less than law is that it does not promulgate rules in the basic sense that it provides citizens with a set of guides that they can find, interpret, and follow (such promulgation is Fuller's second principle of legality⁸³). Nor does an appreciation of code's representationalism require any metaphysical wrangling to connect its rules with the underlying empirical reality. The rules in code work in a different way – they are not commands to be followed, rather they are instrumental tools that crystallise behavioural possibilities from the outset, instead of requesting particular behaviour from the citizen. The result is that the hermeneutic gap that exists between the text of a legal rule and its effect on behaviour in the physical world collapses. Vismann and Krajewski note that while traditionally the law was the ultimate arbiter of reality via its use of rhetorical fictions, it now has to compete with 'digital virtuality' in the constitution of reality.⁸⁴ More and more, this competition is

⁸⁰ Wintgens, 'Legisprudence as a New Theory of Legislation', *supra* n. 27, p. 4.

⁸¹ Hildebrandt, 'Legal and Technological Normativity', *supra* n. 56, p. 175.

⁸² *Ibid.*, p. 176.

⁸³ Fuller, *supra* n. 15, p. 49 *et seq.*

⁸⁴ C. Vismann and M. Krajewski, 'Computer Juridisms' [2007] *Grey Room* 90, p. 92.

being lost: 'the virtual is a mode of reality that evades the space-time categories of the law.'⁸⁵

The extent to which the hermeneutic gap collapses is profound; its effacement is not only easy to do but is standard practice, not necessarily through malice or intentional obfuscation, but simply by the very nature of the technology. The hermeneutic gap can so easily be collapsed because the 'text' of the rule (the code) constitutes directly the geography of the artefact: they are not just *isomorphic*, they are one and the same. Unlike law, whose 'carrier' has hitherto been the inherently passive medium of text, software code allows us to, in Bruno Latour's words, 'conceive of a text (a programming language) that is at once words and actions'.⁸⁶

However, even within our analysis of code we can think at varying levels of abstraction. Asscher draws a distinction between rules on the 'conceptual level' and the 'technical commands within a certain computer language'.⁸⁷ While this is perhaps to a degree necessary when we are dealing with higher-order rules with legal relevance, we must not discount the concrete materiality of the technical commands that are the building blocks of the normativity that makes up those rules operating at that higher level of abstraction. For efficiency's sake it might not always make sense to enquire into the minutiae of a code's behaviour, it is important to find an appropriate abstraction threshold between individual commands and the technological normativities that they collectively bring into being.

⁸⁵ *Ibid.*

⁸⁶ B. Latour, 'Where Are the Missing Masses? The Sociology of a Few Mundane Artifacts' in W.E. Bijker and J. Law (eds.), *Shaping Technology/Building Society: Studies in Sociotechnical Change* (MIT Press, 1992) n. 1.

⁸⁷ L. Asscher, "'Code' as Law: Using Fuller to Assess Code Rules' in E. Dommering and L. Asscher (eds.), *Coding Regulation: Essays on the Normative Role of Information Technology* (The Hague: TMC Asser Press, 2006) p. 83.

Leenes and Koops argue that the negligent production of privacy-eroding code can be viewed as akin to a rule stating that privacy is not important or is less important than other values. They write that

[a]lthough this is perhaps stretching the term ‘rule’ rather far, we are inclined to think that the development and application of code that negligently fails to take privacy effects into account can indeed be seen as the embedding of a ‘rule’ in the technology, namely that privacy is unimportant and secondary to other values that the code primarily serves. Such technology does indeed serve to guide or control (what is perceived as) proper and acceptable behaviour, since privacy-infringement is considered an acceptable outcome of its use.⁸⁸

One can appreciate how abstracted this view is from the idea of viewing individual code commands as rules. In this case, the broader functionality of the code is interpreted as a rule. Here we can see a connection with the design theories discussed in the previous chapter. Leenes’ and Koops’ analysis might have been framed in such terms, whereby the (dis)affordance of a particular privacy protecting functionality is taken to be a *de facto* rule embodied in that particular code. From this level of abstraction,⁸⁹ we can appreciate the normativity the code imposes without having to look directly at the underlying commands, providing a useful bridge between the idea of a rule and the normativity of the code. I discuss this topic further in Chapter 4.⁹⁰

In any event, in code we find the collision of rules (at whatever level of abstraction) and reality, where *ought* becomes *is*. In this way, representationalism finds its apex: no appeal to metaphysical thought or belief is required to see how computer

⁸⁸ R. Leenes and B.-J. Koops, “‘Code’ and Privacy or How Technology Is Slowly Eroding Privacy’ in E. Dommering and L. Asscher (eds.), *Coding Regulation: Essays on the Normative Role of Information Technology* (The Hague: TMC Asser Press, 2006) p. 191.

⁸⁹ For a relevant discussion of levels of abstraction in information theory, see L. Floridi, *The Ethics of Information* (Oxford: Oxford University Press, 2013) ch. 3.

⁹⁰ Chapter 4, section 3.3.2.3 (Does code contain rules *per se*?).

code not just *represents* reality, but by definition *constitutes* it. Behavioural possibilities are thus constituted by the rules, not merely regulated by them.

Returning to the library borrowing system described by Bańkowski, we can appreciate how in the transition to code the substance of the rules becomes merely descriptive rather than regulative. The tracking of library users and their loans is obviated by means of swipe-card authentication (rule 1 collapses); the end of loans and the ‘return’ of ebooks is automated by code, thus rendering rule 2 a description; and rule 3 merely describes the state of the system that rule 4 enforces (again, automatically). The simple if-then instantiation of rule 4’s logic simply does not allow the subroutine of ‘issue book’ ever to take place. As Bańkowski describes the system:

What we see then, is how *the normative has become the descriptive*. This gives us an example of rule following which has the machine-like quality of heteronomy: we ‘don’t think about it’.⁹¹

The library’s rules, which were once normative, have become descriptive – they are simply how the system *is*. One can appreciate how this representation sees the code rule constituting reality, with no gap or space between the rule and its imposition. This connects deeply with the theory of constitutive and regulative rules, discussed next.

3.2.1 *Constitutive and regulative rules*

We saw in Chapter 2 the distinction between constitutive and regulative technological normativity.⁹² Hildebrandt speaks of the difference between legal rules that are constitutive of other (institutional) facts or rules, and those that aim to regulate behaviours which can take place independent of the rule’s existence.⁹³ For example, the institutional fact of marriage cannot exist independently of a constitutive rule which

⁹¹ Bańkowski, *supra* n. 16, p. 56 (my emphasis).

⁹² Chapter 2, section 3.5 (A spectrum of technological normativity).

⁹³ Hildebrandt, ‘Legal and Technological Normativity’, *supra* n. 56, p. 172 *et seq.*

creates it, while it is possible to drive at 100 miles per hour even though there is a regulative rule which prohibits it.⁹⁴ Regulative rules are therefore aimed at regulating existing activities, while constitutive rules ‘create the very *possibility* of certain activities.’⁹⁵ In a sense, then, constitutive rules are creative, or generative,⁹⁶ while regulative rules are limiting. Searle gives the example of chess: the rules of the game do not *regulate* what was already happening (i.e. it is not a common activity to idly push around miniature carved figurines representing kings, queens, knights, etcetera on a chequered board); rather, the rules in fact *constitute* the game. The game of chess does not exist outside of its constitutive rules – if people ignore them, they may be playing *something*, but it is not chess. The constitutive rules are thus creative in their bringing about (i) the general institution of ‘chess’, and (ii) the contingent institutional fact of any given particular game of chess.⁹⁷

This idea of an ‘institutional fact’ stems from the distinction between facts that are socially constructed, and ‘brute facts’ which exist ‘out there’.⁹⁸ Examples of institutional facts include ‘money’ (specially-designed pieces of paper that have exchange value) or a ‘doctoral thesis’ (a detailed and novel analysis of a specific topic usually somewhere between 60,000 and 100,000 words in length). Examples of brute facts include the distance between the earth and the sun at this very moment, or the amount of force I am currently exerting on my laptop’s keys. ‘Money’ and ‘doctoral thesis’ are institutional facts, while the distance and the force are brute facts. An ‘institution’ in this sense does not refer to the more common usage akin to the concept

⁹⁴ *Ibid.*

⁹⁵ J.R. Searle, *The Construction of Social Reality* (New York: Free Press, 1995) p. 27 (my emphasis).

⁹⁶ Zittrain discusses how the fundamental characteristics of the Internet as an open platform make it ‘generative’ of myriad other practices. See Zittrain, *supra* n. 61 *passim*.

⁹⁷ Searle, *supra* n. 95, pp. 27–28. One can also appreciate here the contrast drawn between ‘broad’ and ‘narrow’ normativity in M. Piekarski and W. Wachowski, ‘Artefacts as Social Things: Design-Based Approach to Normativity’ (2018) 22 *Techné: Research in Philosophy and Technology* 400.

⁹⁸ Searle, *supra* n. 95, *et seq.*

of an ‘organisation’, but rather to the less common meaning of ‘an established law, custom, usage, practice, organization, or other element in the political and social life of a people’,⁹⁹ as in the game of chess above.

Because institutional facts are social, which is to say they are created by humans and do not simply exist ‘out there’ in the physical world, they must be brought into being. This is the creative work done by constitutive rules. One can appreciate the tension here between legalism on the one hand, which *does* hold that the constitutive rules of law (a quintessential system of institutional facts) are ‘out there’, and less positivist viewpoints which seek to question the design of those constitutive rules (as this thesis does).

From a legalistic perspective, constitutive rules can be arranged in a hierarchy which creates the underlying framework (itself an institutional fact or set of facts, sometimes termed a ‘constitution’) within which other rules can be made. This is Wintgens’ proxy model, discussed above, where the legitimacy of a given legal rule flows from some founding act which operates in the background to give validity to those subsequently promulgated norms.¹⁰⁰ As previously stated, the idea is connected with various accounts of law that culminate in a notional foundational legal rule.

As with the institution of chess, in law constitutive rules are necessary to create valid legal institutions, and the institutional facts that are created as instances of them. This is the basis of MacCormick’s theory of law as institutional fact.¹⁰¹ For example, ‘marriage’ is a legal institution defined by the requirements laid down in certain constitutive rules, and *a* marriage is a legal institutional fact that has come into being by following those rules.¹⁰² To speak of a couple being ‘married’ outside the

⁹⁹ ‘Institution, n.’ (Oxford English Dictionary online).

¹⁰⁰ See Section 2.2 *supra*.

¹⁰¹ N. MacCormick, *Law as Institutional Fact* (Edinburgh: University of Edinburgh, 1973).

¹⁰² *Ibid.*, p. 5. MacCormick sub-categorises constitutive rules into ‘institutive’, ‘consequential’, and ‘terminative’ rules; in this case the rule that provides the conditions necessary for the institution of

institutional framework (in both senses of ‘institution’) does not make sense, or at least does not point to a legally-recognised institutional fact.¹⁰³

How does this relate to code? Whereas the institutional facts created by legal constitutive rules are always only ‘real’ within the law’s own ‘regime of veridiction’¹⁰⁴ (i.e. they are not brute facts about the physical world¹⁰⁵), code-based rules are in a great many cases ‘brute’ in the sense that they are ‘just there’ and part of the ‘physical’ fabric of the system. As Hildebrandt puts it, they are ‘constitutive of our behaviour’.¹⁰⁶ In including, excluding, and defining the limits of behaviour within the ambit of the system, they limit end-user freedom in material ways. The qualitative difference is that, with code, these limits are not merely regulative, where the end-user, already engaging in some activity online, is ‘requested’ to amend her behaviour this way or that (this is the equivalent of the speed limit which can only ‘request’ that the driver travel at less than 70 miles per hour). Rather, the limits are constitutive, in that the salient features of the behaviour are themselves defined (i.e. constituted) by code. Whereas the chess player can use the architecture of the artefact (the board and its pieces) in ways that are outside the rules of the game, for example to play a simpler game like checkers with a young child, the architecture of code sets the rules in place *ex ante*, and does not by default allow them to be so adapted according to the discretion of the end-user.¹⁰⁷ To

marriage to exist are institutive (they define the conditions for a given institution to exist). Consequential rules are those which define what legal rights and duties arise upon the creation of a given institution (e.g. spousal succession rights), while terminative rules define how the institution comes to an end (e.g. divorce). See *ibid.*, pp. 5–6.

¹⁰³ Hildebrandt, *Smart Technologies*, *supra* n. 11, p. 145.

¹⁰⁴ *Ibid.*

¹⁰⁵ Although from some viewpoints they might be treated as brute facts: as Hildebrandt notes, ‘depending on one’s perspective, any brute fact can be rearticulated as an institutional fact, while institutional facts can be “used” as brute facts to be regulated.’ See Hildebrandt, ‘Legal and Technological Normativity’, *supra* n. 56, p. 172.

¹⁰⁶ *Ibid.*, p. 174.

¹⁰⁷ The topic of what code’s architecture affords the end-user was covered in Chapter 2.

quote Lessig in one of his earliest works on the subject, ‘one obeys these laws as code not because one should; one obeys these laws as code because one can do nothing else.’¹⁰⁸ The constraints and enablements of code are ‘like laws of nature, like the world as we find it’;¹⁰⁹ they are simply ‘there’, with the crucial difference between code and law being that rather than suggesting an ‘ought’, as law always does, code can simply impose an ‘is’.¹¹⁰ The three traditionally-conceived of phases of regulation (direction, detection, and correction)¹¹¹ are thus susceptible to being collapsed into a single step.¹¹²

It is, as we shall later see, possible for code rules to be regulative so that the end-user is in fact *invited* to behave in certain ways, and of course many digital systems do allow a range of behaviours in end-users’ interactions with them. Social networks, for example, give end-users fairly wide latitude on the volume and content of the text, images, and videos that they can upload. But even within this seemingly unlimited freedom to upload there are nevertheless constitutive limits – the code will accept only text, images, or video files (not PDFs or Word documents, or images/videos in formats the platform does not recognise), and only a certain volume of data can be uploaded. For most usage these boundaries will never be approached, and so the end-user is unaware of their existence, but they are nevertheless there. This connects with Brownsword’s concept of levels of technological control: in one sense the end-user is

¹⁰⁸ L. Lessig, ‘The Zones of Cyberspace’ (1995) 48 *Stan. L. Rev.* 1403, p. 1408.

¹⁰⁹ Grimmelman, *supra* n. 59, p. 1740. Similarly, Bamberger notes that ‘the fact that architectural technology embodies normative choices at all can escape notice, as the perfect constraints code places on behavioral possibility can seem as natural, immutable, and invisible as the laws of physics.’ See K.A. Bamberger, ‘Technologies of Compliance: Risk and Regulation in a Digital Age’ (2010) 88 *Texas Law Review* 669, pp. 724–725.

¹¹⁰ R. Brownsword, ‘Lost in Translation: Legality, Regulatory Margins, and Technological Management’ (2011) 26 *Berkeley Technology Law Journal* 1321.

¹¹¹ C. Scott, ‘Regulation in the Age of Governance: The Rise of the Post-Regulatory State’ in J. Jordana and D. Levi-Faur (eds.), *The politics of regulation: Institutions and regulatory reforms for the age of governance* (Cheltenham, UK; Northampton, MA: Edward Elgar Publishing, 2004) p. 147.

¹¹² Brownsword, *supra* n. 110, p. 1344.

free to submit whatever she likes (level 0), but simultaneously she is constrained both by the ‘physical’ limitations of the technology (only plain text or image/video files can be submitted; level 2) and by normative signals as to what content is acceptable (i.e. graphic images or hate speech are liable to be reported and removed by a moderator; level 1).¹¹³

In many cases, then, the normativity embodied in code is more constitutive than regulative of the practice in question.¹¹⁴ The more constitutive a digital system is of a practice, the more control the designer has in defining that practice, which in turn means that it will in many cases be commercially attractive to limit the ‘regulative latitude’ given to the end-user in her interactions with the system. Even on a purely practical level, building in or enlarging the contingent regulative space, at the expense of constitutive certainty, requires anticipation of more conditions, which in turn requires more code and therefore more expense, lending further commercial impetus to take a constitutive approach to design.

3.3 Immediacy

Code’s immediacy refers to the temporal aspect of execution. As discussed above, the hermeneutic gap between text and behaviour is collapsed, but not only does text constitute both rule and reality, but the application of it is arranged prior to, and imposed immediately at the point of, execution. The conditions specified in that prior arrangement are imposed without delay and without consideration of alternative actions that might have been appropriate. There is no scope to ‘hesitate well’, as in Latour’s description of how judges work;¹¹⁵ linear time becomes compressed, further

¹¹³ *Ibid.*, p. 1346.

¹¹⁴ Hildebrandt, ‘Legal and Technological Normativity’, *supra* n. 56, p. 175.

¹¹⁵ B. Latour, *The Making of Law: An Ethnography of the Conseil d’Etat* (Revised edition, Cambridge, UK; Malden, MA: Polity, 2009) pp. 193–194.

distancing the character of code's operation from the role that law's more measured pace plays in the stabilisation of societal expectations.¹¹⁶

Grimmelmann refers to this characteristic as code's *immediacy*, where the ex ante nature of code means that '[s]oftware cannot – as law can – adapt its response in light of later-available information or a later determination that such information is relevant.'¹¹⁷ As we have seen, whereas law as a prospective regulator is inert in the absence of the will to reflect its terms in real-world behaviour, the enablements and constraints of code, put in place by the designer, have latent efficacy even before the system is operational. Recalling the discussion of the library system above, the end-user is simply presented with the de facto 'ruling' of the code: she may not borrow any more ebooks, the ebook on her system will cease to be accessible at a specific time, etcetera. The code's swiftness is brutal and entirely impervious to external reason.¹¹⁸

3.3.1 *Default configurations*

Immediacy is embodied in code even where the design includes the possibility of altering its configuration. End-users tend to accept as a 'natural and immutable fact'¹¹⁹ code's pre-defined configuration when it is first supplied to them. They are also susceptible to 'automation bias', whereby the configuration and responses of a machine are given greater credence than a human equivalent.¹²⁰ For these reasons, the designer

¹¹⁶ M. Hildebrandt, 'A Vision of Ambient Law' in R. Brownsword and K. Yeung (eds.), *Regulating Technologies: legal futures, regulatory frames and technological fixes* (Oxford; Portland, Or: Hart, 2008) pp. 186–187. See also N. Luhmann, *Law As A Social System* (Fatima Kastner et al. eds, Klaus A Ziegert tr, Oxford: Oxford University Press, 2004) p. 205 *et seq.*

¹¹⁷ Grimmelmann, *supra* n. 59, p. 1730.

¹¹⁸ Bańkowski and Schafer, *supra* n. 1, p. 46.

¹¹⁹ M. Goldoni, 'The Politics of Code as Law: Toward Input Reasons' in J. Reichel and A.S. Lind (eds.), *Freedom of Expression, the Internet and Democracy* (Leiden: Brill, 2015) p. 128. See also J.P. Kesan and R.C. Shah, 'Setting Software Defaults: Perspectives from Law, Computer Science and Behavioral Economics' (2006) 82 *Notre Dame Law Review* 583, p. 591 *et seq.*

¹²⁰ D.K. Citron, 'Technological Due Process' (2008) 85 *Washington University Law Review* 1249, pp. 1271–1272.

has significant power in choosing one pre-configuration over another. As Tien notes, ‘default settings may seem “normal” because the equipment is common, or have become “legitimate” as people have grown accustomed to the situation presented by the equipment.’¹²¹ Simultaneously, the ‘is-ness’ of default settings militates against enquiring as to whether other configurations might be more suitable for the end-user – instead the defaults are accepted as immutable facts, and alternatives (should they even imagine them) as impossible or even unreasonable.¹²² For many end-users, the assumption will in many cases be that the designer knows best (known as the ‘legitimizing effect’¹²³). Furthermore, they might lack the technical sophistication to investigate all the possible customisation options,¹²⁴ or they might not have the time to do so,¹²⁵ much less to think critically about what intentions might lie behind a particular choice of defaults and how these may have evolved over time according to the technical or commercial imperatives motivating the designer’s choices.¹²⁶ As Tien notes, the behaviour-guiding quality of default configurations is often stripped away by the perception that they are ‘mere design features’¹²⁷ – in other words, as with legalism, that they are part of what is ‘just there’. Even where end-users care about the underlying values that are impacted by the default configuration of the design – and presumably a value such as legitimacy would qualify, for many if not most end-users –

¹²¹ L. Tien, ‘Architectural Regulation and the Evolution of Social Norms’ (2004) 7 *Yale Journal of Law & Technology* 1, p. 16.

¹²² Kesan and Shah, *supra* n. 119, pp. 596, 601; L. Diver, ‘Law as a User: Design, Affordance, and the Technological Mediation of Norms’ (2018) 15 *SCRIPTed* 4, p. 11.

¹²³ Kesan and Shah, *supra* n. 119, p. 603.

¹²⁴ *Ibid.*, pp. 611–612.

¹²⁵ *Ibid.*, p. 598 *et seq.*

¹²⁶ Tien, *supra* n. 121, p. 16. An example of this is Facebook’s then-controversial move to the ‘news feed’ layout as the default for all end-users. See J. Leyden, ‘Users Protest over “creepy” Facebook Update’ *The Register* (9 July 2006) <https://www.theregister.co.uk/2006/09/07/facebook_update_controversy/>.

¹²⁷ Tien, *supra* n. 121, p. 12.

unless they are aware of the possibility of choice they will accept the default, even if it is injurious to the value.¹²⁸

Furthermore, the designer must make a choice as to how to balance the number of defaults (that is, options which the end-user can change) against the amount of ‘pre-wired’ functionality: too many options or a confusing interface can confuse the end-user, undermining the value of the choice.¹²⁹ Indeed, providing configurable options within interfaces that are antagonistic to exercising that choice is one means by which some unscrupulous (or perhaps just negligent) enterprises can argue that they are respecting end-users’ autonomy whilst simultaneously undermining their interests in favour of commercial expediency. A recent example is the design change in Google’s Chrome browser that obfuscates the circumstances in which end-users are logged into Google’s services, even when they have enabled the ‘block third-party cookies’ preference that would normally prevent this kind of behaviour (and indeed still does in other browsers).¹³⁰ Such practices, sometimes termed ‘dark patterns’, were discussed in the design context in Chapter 2. It is worth noting, too, that the ‘block third-party cookies’ preference itself is *not* the default setting on any mainstream browser, and thus its privacy-enhancing mechanism is something end-users must manually enable, which requires first that they are aware of the option and what it means.¹³¹

Kesan and Shah describe three concepts from behavioural economics which are relevant to the impact of defaults: bounded rationality, cognitive biases, and the

¹²⁸ Kesan and Shah, *supra* n. 119, pp. 601–602.

¹²⁹ *Ibid.*, p. 627. See also Brownsword’s discussion of ‘prudential choice’, *supra* n. 110, p. 1345.

¹³⁰ L. Olejnik, ‘Am I Logged in or Not? GDPR Case Study on the Example of Chrome Browser Change’ <<http://blog.lukaszolejnik.com/am-i-logged-in-or-not-gdpr-case-study-on-the-example-of-chrome-browser-change/>>.

¹³¹ This is in line with the evolution of the relevant IETF standards, which have changed from requiring a higher standard of privacy (RFCs 2109 (1997) and 2965 (2000) suggest that the default should be to reject persistent (cross-session) cookies) to permitting browsers to implement whatever default standard they wish, albeit while noting the ‘worrisome’ nature of third-party cookies (RFC 6265 (2011), section 7.1).

legitimizing effect. If the end-user is not made aware of the underlying code's behaviour and the possibility of exercising agency to configure it, she will effectively see the system as fixed, even if she would otherwise be interested in making a change, and is to that extent disempowered from exercising her rationality to choose.¹³² Cognitive biases operate in three ways.¹³³ Under the 'status quo bias', the end-user favours 'inertia over action', meaning she accepts *de facto* the 'is-ness' of the system. The 'omission bias' operates to discourage end-users from actively making a change, because it is believed that worse effects (i.e. breaking the system) are likely to result in doing something (i.e. changing a setting) rather than nothing. The last bias, the 'endowment effect', sees individuals ascribe higher value to settings that they perceive to favour them, thus biasing them against switching to an alternative. Lastly, the third concept from behaviour economics which Kesan and Shah discuss is the legitimizing effect. This sees end-users interpret defaults as normative signals as to what they *should* do; they assume defaults to be 'reasonable, ordinary, and sensible' because if this was not the case the designer would have made another setting the default choice.¹³⁴

From this discussion we can see how easily the design of defaults can contribute to a minimisation of end-users' comprehension of code and, conversely, the increase of its regulative effect on their behaviour.

3.3.2 Pervasiveness

The pervasiveness of code is not difficult to appreciate. Digitalisation is all around us, and increasing at ever-greater speed. Cisco estimates that by 2020 there will be 250

¹³² Kesan and Shah, *supra* n. 119, pp. 601–602.

¹³³ *Ibid.*, pp. 602–603.

¹³⁴ *Ibid.*, pp. 603–604. For empirical evidence of the legitimizing effect, see R.C. Shah and C. Sandvig, 'Software Defaults as De Facto Regulation: The Case of the Wireless Internet' (2008) 11 *Information, Communication & Society* 25.

'things' connecting to the Internet every second, up from 80 per second in 2013.¹³⁵ They suggest further that the total volume of devices is expected to reach 1.8tn in 2020, with an annual growth rate of 3%.¹³⁶ IoT devices are being integrated further into daily life through the development of more sophisticated low-power infrastructure¹³⁷ and the diversification of connected applications.¹³⁸ Similar trends are also in evidence for the blockchain artefacts that increasingly govern and make use of the data that flows from IoT devices.

3.3.3 Immutability

Surden notes how the subjective value judgements of designers, and the resulting effects of the rules embodied in their code, can be magnified when the systems are distributed and adopted widely.¹³⁹ The choice of rules becomes fixed in the code, enabling its exponential magnification as its effects compound with successive execution.¹⁴⁰ Bamberger notes similar risks in his discussion of 'systemic effects' – whereas at production time the designer has great freedom to choose how the code should behave, this plasticity is to a great extent 'locked down' once production has ceased.¹⁴¹ This is what Winner terms the 'initial commitments'. For him,

¹³⁵ K. Tillman, 'How Many Internet Connections Are in the World? Right. Now.' <<https://blogs.cisco.com/news/cisco-connections-counter>>.

¹³⁶ 'Connections Counter: The Internet of Everything in Motion' (*Cisco: The Network*, 29 July 2013) <<https://newsroom.cisco.com/feature-content?type=webcontent&articleId=1208342>>.

¹³⁷ J. Twentyman, 'IoT Drives Progress towards Low-Power Technology' *Financial Times* (8 January 2018) <<https://www.ft.com/content/f2b4de5a-d8ee-11e7-9504-59efdb70e12f>>.

¹³⁸ '14 Predictions For The Future Of Smart Home Technology' (*Forbes*) <<https://www.forbes.com/sites/forbestechcouncil/2018/01/12/14-predictions-for-the-future-of-smart-home-technology/>>.

¹³⁹ Surden, *supra* n. 64, p. 5.

¹⁴⁰ *Ibid.*

¹⁴¹ Bamberger, *supra* n. 109, pp. 710–711.

...technological innovations are similar to legislative acts or political foundations that establish a framework for public order that will endure over many generations. For that reason, the same careful attention one would give to the rules, roles, and relationships of politics must also be given to such things as the... tailoring of seemingly insignificant features on new machines.¹⁴²

These observations strengthen the impetus to focus on ex ante production of code rather than ex post assessments of it. Goldoni summarises why this is so:

Given the nature and the logic of architectural regulation, the emphasis on output legitimacy is misplaced for several reasons. First, since technology is often irreversible – once it is developed and applied in society, it is difficult to change it or remove it from society in those applications – the process which develops code as law becomes a key concern when normativity is at stake. In fact, it may well be too late when a particular version of a technology appears or is adopted... [t]he difficulty of reversing embedded code is often evident and makes it fundamental to focus on the procedure and the actors involved in [its] development.¹⁴³

Without some ready means of altering the code after it is produced, the normative import of the ‘initial commitments’ is all the greater. Even where the code can be updated, its immediacy means its normative effect is in place before this happens, and so it is important that the design is produced in a legitimate fashion from the outset.

¹⁴² L. Winner, ‘Do Artifacts Have Politics?’ [1980] *Daedalus* 121, p. 128.

¹⁴³ Goldoni, *supra* n. 119, p. 128. Koops makes a similar argument, although he focuses nevertheless on ex post assessment. See B.-J. Koops, ‘Criteria for Normative Technology: The Acceptability of “Code as Law” in Light of Democratic and Constitutional Values’ in R. Brownsword and K. Yeung (eds.), *Regulating Technologies: Legal Futures, Regulatory Frames and Technological Fixes* (Oxford; Portland, Or: Hart, 2008) p. 166. His contribution is discussed in detail in Chapter 4.

3.4 Opacity

We saw in the previous chapter how design operates in ways which are not always within the conscious apprehension of the end-user. This opacity forms another cornerstone element of computational legalism and the requirement that end-users ‘not think about it’: if the end-user cannot comprehend the rules to which her behaviour is subject, she cannot possibly consider whether and how to respond to them. This foundational issue is problematic in traditional processes of democratic law-making. Waldron, for example, notes that ‘those interested in democracy will have a direct interest also in this opacity itself – that is, in the sheep-like ignorance of the nature of the law one is ruled by’.¹⁴⁴ So too in the computational context, except that there the extent to which end-users-as-citizens are rendered ‘sheep-like’ is qualitatively and quantitatively greater.¹⁴⁵ As Goldoni notes, ‘given the opacity of architectural regulation, to be aware of how technology is directly or indirectly impacting upon agents’ behaviours may prove to be too difficult in many cases.’¹⁴⁶ Although focused on source code and the web, Longford’s observations are apposite:

A central feature of new media design, in fact, is that the source code for any particular application or program which structures an end-user’s experience is hidden from them... HTML, IP addresses, and web browser software are exemplary of *code’s self-concealing character*. HTML conceals the textual information which is ultimately responsible for the graphical web pages presented to surfers.

Whereas most browsers have a ‘view source’ option that makes HTML relatively accessible (if not necessarily comprehensible, despite its human-readability¹⁴⁷) to the

¹⁴⁴ J. Waldron, ‘Can There Be a Democratic Jurisprudence?’ (2009) 58 *Emory Law Journal* 675, p. 696 *et seq.*, discussing the problematic nature of analytical positivism from a democratic perspective.

¹⁴⁵ Citron, *supra* n. 120, p. 1254f.

¹⁴⁶ Goldoni, *supra* n. 119, p. 128.

¹⁴⁷ This is the heart of the transparency fallacy, discussed in Chapter 5.

end-user, the compiled code that implements specific rules in other digital artefacts is generally both inaccessible and inscrutable because of its translation into machine-readable ‘object code’. Regardless of the programming language used, however, end-users face difficulty in comprehending the totality of the system before them. The basic point is as Winner put it, as far back as 1977: end-users ‘are unable to give an adequate explanation of man-made phenomena in their immediate experience. They are unable to form a coherent, rational picture of the whole.’¹⁴⁸ As a result, ‘all persons do and, indeed, must accept a great number of things on faith.’¹⁴⁹ The user interfaces of digital artefacts are inherently limited in their communication of the myriad operations taking place behind-the-scenes. Even the apparently simplest of operations, for example clicking a hyperlink on a webpage, involves a host of unseen technical processes.¹⁵⁰ Most of the time obscuring all of this is beneficial, because the cognitive load of having to comprehend all that is *really* going on would quickly overwhelm. As mentioned above, it would be undesirable to enquire into the detail of every rule being followed in every computational operation, since the burden of comprehension is too great. The task then is to give information and control ‘over the right things at the right time’.¹⁵¹ This obfuscation of *actual* behaviour can however be used both for good and bad; the

¹⁴⁸ L. Winner, *Autonomous Technology: Technics-out-of-Control as a Theme in Political Thought* (2nd edn., Cambridge, Mass: MIT Press, 1977) p. 284.

¹⁴⁹ *Ibid.*

¹⁵⁰ See for example Floridi’s discussion of ‘handshaking’ between digital systems, a mundane but complex element of the behaviour of all online devices that is entirely removed from the end-user’s experience: L. Floridi, *The Fourth Revolution: How the Infosphere Is Reshaping Human Reality* (OUP Oxford, 2014) pp. 36–37.

¹⁵¹ B. Friedman, ‘Value-Sensitive Design’ (1996) 3 *interactions* 16, p. 18. See also A. Pols and A. Spahn, ‘Designing for the Values of Democracy and Justice’ in J. Van Den Hoven, P.E. Vermaas and I. van de Poel (eds.), *Handbook of Ethics, Values, and Technological Design: Sources, Theory, Values and Application Domains* (Dordrecht: Springer, 2015).

ability to hide the complexity of standard technical behaviours for the sake of the end-user can also be used to obfuscate technical behaviours that are not in her interests.¹⁵²

We have seen on the one hand how end-users tend to show deference to defaults, while on the other the immutable aspects of a system's architecture situate the end-user within an assemblage of behaviour-constraining rules that might admit of only minimal, if any, interpretation – a *fait accompli*, 'achieving compliance by default rather than through active enforcement'.¹⁵³ As Hildebrandt notes, whereas

[l]egal norms do not rule out disobedience, contestation of the technological defaults that regulate our lives may be impossible because *they are often invisible* and because most of the time there is no jurisdiction and no court.¹⁵⁴

The normativity of code is in no way contingent on it being intelligible to those whose behaviour is regulated by it (or indeed even those who created it). There is no requirement that its rules be made public or available, much less in a form that can be understood by humans. Indeed, the complexity of code rules is such that they rapidly become unintelligible, even to the experts who created them. As discussed in the previous chapter, it thus becomes extremely difficult (and in most cases impossible) for the end-user to scrutinise the rules to which her behaviour is subject. We therefore have the blindest of legalistic 'blind rule following'.¹⁵⁵

3.4.1 Code as legal positivism

The discussion above aims to demonstrate the sheer 'isness' of code; its 'positivism' is of a form that is deeply challenging to a vision of law as a reasoned enterprise that reflects the democratically-expressed outlook of a society, from and according to which

¹⁵² See generally F. Pasquale, *The Black Box Society: The Secret Algorithms That Control Money and Information* (Cambridge: Harvard University Press, 2015).

¹⁵³ Tien, *supra* n. 121, p. 12.

¹⁵⁴ Hildebrandt, *Smart Technologies*, *supra* n. 11, p. 12.

¹⁵⁵ Bańkowski and Schafer, *supra* n. 1, p. 31.

its norms are developed and applied. As Vismann and Krajewski note, code's 'virtuality challenges the law's core concepts: corporeality, finitude, and authentication, concepts that are fundamental to any claim of territorial sovereignty as well as to imputations and rules of evidence'.¹⁵⁶

This 'positivism' removes the possibility of deliberation on the part of the end-user, resulting in a kind of instrumentalism that strips individuals of their ability to take part in the moral community, even where they disagree with the substance of the rules. End-users have no choice but to obey the rules, but simultaneously they have no standing in their formulation.¹⁵⁷ As Brownsword – a 'card-carrying natural lawyer'¹⁵⁸ – notes, design simply 'by-passes practical reason to eliminate all options other than the desired pattern of behaviour.'¹⁵⁹ One of the effects of this is to 'demoralise' citizens, blunting their sensitivity to social norms and thus their capacity for self-control and for doing good.¹⁶⁰ This latter point connects with Fuller's discussion of the morality of aspiration, and how it conflicts with a legalistic morality of duty where the rule constitutes the entirety of what is required of regulatees, with no further expectations.¹⁶¹ For Bańkowski, this minimal expectation is not enough; legalism and positivism collapse the aspirational aspect of legality and reduce the guiding normative value of social practices, because the presence of a 'critical morality' that invites self-scrutiny becomes displaced by a rote rule-bound approach to the direction of citizens'

¹⁵⁶ Vismann and Krajewski, *supra* n. 84, p. 92.

¹⁵⁷ Bańkowski and Schafer, *supra* n. 1, p. 48. See also Longford, *supra* n. 3.

¹⁵⁸ R. Brownsword, 'Code, Control, and Choice: Why East Is East and West Is West' (2005) 25 *Legal Studies* 1, p. 10.

¹⁵⁹ *Ibid.*, p. 4.

¹⁶⁰ *Ibid.*, p. 19, quoting criminologist D.J. Smith, 'Changing Situations and Changing People' in A. von Hirsch, D. Garland and A. Wakefield (eds.), *Ethical and Social Perspectives on Situational Crime Prevention* (Bloomsbury Publishing, 2004). See also Brownsword, *supra* n. 110 *et passim* and especially pp. 1355-1356.

¹⁶¹ Fuller, *supra* n. 15, ch. 1.

conduct.¹⁶² In other words, by not needing to consider what a proper course of action would be, our ability or habit of raising such questions becomes atrophied. Brownsword suggests that a community fully reliant on such forms of (state) regulation which obviate the possibility of moral deliberation is ‘no longer an operative moral community.’¹⁶³ In other words, we require the opportunity to *choose* to do good, in the face of at least the possibility of doing otherwise, if we are to continue to exercise practical reason as moral actors. Waldron makes a similar point about the ‘dignitarian’ aspect of legality: we aspire to a law which ‘conceives of the people who live under it as bearers of reason and intelligence’, even if the price of this is a ‘diminution in law’s certainty’.¹⁶⁴ Computational legalism demonstrates the most certain of certainties, but simultaneously hides it under the veil of opacity from the end-user’s comprehension. As we will see in Chapter 5, the provision of spaces for the exercise of such reason and choice is not just about providing more choice, but is about providing the right quality of choice within the geography of the code.¹⁶⁵

3.5 The veiling of code’s production

Wintgens discusses the ‘veil of sovereignty’ that shrouds the work of the legislator, shielding it and her from the gaze of the legal philosopher and the legalistic citizen.¹⁶⁶ Legislative sovereignty is perceived as a black box,¹⁶⁷ a concept that is apposite in this context given its staple use in Science, Technology and Society studies (STS)

¹⁶² Bańkowski, *supra* n. 16, pp. 49–50.

¹⁶³ Brownsword, *supra* n. 158, p. 19. See also Brownsword, *supra* n. 110, pp. 1355–1356.

¹⁶⁴ J. Waldron, ‘The Rule of Law and the Importance of Procedure’ (2011) 50 *Nomos* 3, p. 18 *et seq.*

¹⁶⁵ Chapter 5, section 3.1.1 (Affordance: choice).

¹⁶⁶ Wintgens, *Legisprudence*, *supra* n. 21, p. 2; Wintgens, ‘Legislation as an Object of Study of Legal Theory: Legisprudence’, *supra* n. 32, p. 2.

¹⁶⁷ Wintgens, *Legisprudence*, *supra* n. 21, p. 212 *et seq.* Wintgens quotes I. Kant, ‘Metaphysical First Principles of the Doctrine of Right’ in M. Gregor (tr.), *The Metaphysics of Morals* (Cambridge University Press, 1996).

analyses.¹⁶⁸ The ultimate source of sovereignty is not in question and the mechanisms by which its outputs are arrived at are not to be questioned by jurists. Boyle noted this obscuring function early on when he suggested that “[t]he technology appears to be “just the way things are”; its origins are concealed, whether those origins lie in state-sponsored scheme or market-structured order, and its effects are obscured because it is hard to imagine the alternative’.¹⁶⁹

In the computational context, the ‘sovereignty’ of the designer is protected by two ‘veils’ – one technical and the other legal. The first is the code-based opacity discussed in the previous section: this veil is one of technical inscrutability, which the ordinary end-user is usually ill-equipped to lift, if that is even a possibility. The second veil protects enterprise by, for example, trade secrecy and anti-circumvention laws, which limit the scrutiny their development practices are put under, strengthening their quasi-sovereignty.¹⁷⁰ The prevailing neoliberal economic outlook sustains this second veil by shifting sovereignty away from the state and onto the market,¹⁷¹ while simultaneously unfettered technological innovation is prioritised as a good in itself.¹⁷² Thus the private ‘sovereignty’ of the profit-seeking enterprise is black-boxed unless and until a real-world harm is detected, which might never happen, owing to the technical veil. It is the task of the market and not the state to respond to the enterprise’s designs

¹⁶⁸ See for example M. Akrich, ‘The De-Description of Technical Objects’ in W.E. Bijker and J. Law (eds.), *Shaping Technology/Building Society: Studies in Sociotechnical Change* (MIT Press, 1992). In the technology law sphere, see Pasquale, *supra* n. 152.

¹⁶⁹ J. Boyle, ‘Foucault in Cyberspace: Surveillance, Sovereignty, and Hardwired Censors’ (1997) 66 *University of Cincinnati Law Review* 177, p. 205.

¹⁷⁰ Pasquale, *supra* n. 152, p. 15.

¹⁷¹ B.H. Bratton, *The Stack: On Software and Sovereignty* (Cambridge, Massachusetts: MIT Press, 2016) p. 21; M. Foucault, *The Birth of Biopolitics: Lectures at the Collège de France, 1978–79* (Basingstoke; New York: Palgrave Macmillan, 2008) p. 280.

¹⁷² J.E. Cohen, ‘The Regulatory State in the Information Age’ (2016) 17 *Theoretical Inquiries in Law* 369, p. 387f. See also D. Harvey, *A Brief History of Neoliberalism* (Oxford; New York: Oxford University Press, 2005) p. 157 *et seq.*

in order to ascertain their correct value and desirability. The commercial entity is free to exercise its imperative for profit, while the market is trusted to curb any excesses. As Schulz and Dankert put it, ‘code is essentially a resource through which the ones designing the code can pursue their interests’.¹⁷³ Similarly, Bańkowski and Schafer note that

[f]or the individual, more often than not, the absence of government is not experienced as liberating, but as subjugation to commercial interests which effortlessly [sic] project, and indeed magnify, their offline powers into cyberspace.¹⁷⁴

Herein lies a paradox of commercial computational legalism: as discussed above, legalism is ideologically attractive in part because it helps establish a baseline of certainty which is advantageous to capitalist enterprise.¹⁷⁵ As those enterprises have developed into promulgators of code-based rules, however, their need for certainty has circumscribed the liberty of the ordinary citizen through the development of an imbalance of regulative power between government and commerce,¹⁷⁶ and by the lack of incentives to ensure that their design processes and the products of them embody values of legitimacy and legal protection that are intrinsic to that liberty. Hildebrandt and Koops suggest that cost, convenience, the difficulty of controlling risk, and the power imbalance between government and commerce all combine in the context of privacy to ‘favour privacy-threatening technology far more than privacy-friendly

¹⁷³ W. Schulz and K. Dankert, “Governance by Things” as a Challenge to Regulation by Law’ (2016) 5 *Internet Policy Review* p. 5. Hildebrandt and Koops term this a ‘provider-centric scenario’, see their ‘The Challenges of Ambient Law and Legal Protection in the Profiling Era’ (2010) 73 *The Modern Law Review* 428, p. 457.

¹⁷⁴ Bańkowski and Schafer, *supra* n. 1, p. 47.

¹⁷⁵ Wintgens, ‘The Rational Legislator Revisited. Bounded Rationality and Legisprudence’, *supra* n. 18, p. 4; Bańkowski, *supra* n. 16, p. 48.

¹⁷⁶ Hildebrandt and Koops, *supra* n. 173, p. 444.

“code”¹⁷⁷ In the absence of incentives to create the latter, code which is supportive of commercial interests but detrimental to end-users’ interests is likely to win out; the features of computational legalism make it easy (and ‘rational’, from a neoliberal perspective) to take that route. The same is true for the more fundamental form of user-antagonistic code I am concerned with. Looking more broadly than just substantive privacy, this refers to code which constrains behaviour according to commercial imperative but has not been legitimately produced, that is its *ex ante* design neither takes into account its legitimacy at runtime nor provides sufficient *ex post* mechanisms for redress.

4. Conclusion

To summarise, we have seen how the characteristics of code come together to demonstrate a form of legalism that is significantly stronger than even that envisaged by Wintgens. With code we have the apex of legalism: from the end-user’s perspective, code’s architecture is simply ‘just there’, while simultaneously its constitutive nature defines what practices are possible, by definition ruling out all the possible alternatives that the plasticity of the code might otherwise have allowed. We therefore have not just representationalism, but realism: code does not just represent reality, it actively constitutes it. The behaviour of the end-user is to a great extent bound *ex ante*, and since she will in most cases not be aware of that binding she is, through no oversight or mistake on her part, forced to acquiesce blindly to the rules that are inscribed in the code:¹⁷⁸ she is deprived of even the possibility of choosing whether or not to, in Bańkowski’s terms, ‘think about it’. The code does not just *represent* reality, it *constitutes* it, and there is little the end-user can do to query that constitution.

¹⁷⁷ *Ibid.*

¹⁷⁸ Hildebrandt, ‘Legal and Technological Normativity’, *supra* n. 56, p. 174.

The hard edges of code rules admit of no interpretation or latitude beyond what the designer has had the foresight (and incentive) to implement. These are strengthened by the immediacy of code: it executes without delay, imposing those potentially harmful rules without deliberation. While the end-user may in some cases have the opportunity to alter the default configuration of the rules, the literature shows that they tend not to do this, deferring instead to the perceived knowledge and expertise of the designer. In any event, the provision of an option is contingent on both the designer deciding to do so and the interface making it clear what the options are and what they mean. Here are a-temporality and concealed instrumentalism in evidence.

Finally, the veil of sovereignty is widened, covering veils of both technical inscrutability and neoliberal market protection. The opacity of digital architectures militates almost fatally against end-users understanding the incentives and rules to which their behaviour is subject and the possible functionalities that were left aside in the designer's choice to render the system as it is. This takes place in the context of an economy which hitherto has promoted unfettered innovation over regulation, leaving any negative effects of the fruits of the process to be dealt with according to market signals.

Taken together with the design analysis in the previous chapter, the discussion here has set up the concept of computational legalism, strengthening the theoretical parallel between code and law as normative orders whose rules can be created in ways that are legalistic to varying degrees. By developing this parallel, I set up the foundation for an analysis that imports the measures designed to reduce or avoid legalism in the legal sphere into the design sphere. Before I embark on that synthesis, the next chapter sets out the two primary legal theories from which I develop the digisprudential framework, and reviews the existing legal literature on criteria for the use of code as a regulator.

Chapter 4

Normative criteria for producing {law, code}

Translating the paradox of the ‘Rechtsstaat’ into digital code – using a technology to protect us against undesired consequences while regulating its use – would thus require two things. First, the use of code must be legitimized in democratic procedure and second, the implications of automatic application must be faced and mitigated.¹

...it is perfectly fair to ask regulators to justify both their purposes and the instruments (the rules or the designs) that they have adopted.²

1. Introduction

The previous chapter looked at the characteristics of strong legalism, before considering how they are evidenced by code, in order to posit the concept of *computational legalism*, a species of ‘legalism’ that exemplifies and amplifies those characteristics. This chapter takes a similar approach in its structure, looking first at the traditional legal sphere before moving into the computational realm. The first part discusses various scholars’ understanding of legitimacy, before going on to set out two normative frameworks of criteria for the creation of traditional legal norms, namely Fuller’s internal morality of law, and Wintgens’ legisprudence. The chapter then shifts focus to review the existing literature on normative criteria for code-making. From this discussion we gain a sense of the kinds of concerns around computational legitimacy that are displayed in the literature, and what considerations are missing or less fully analysed: the most obvious limitations in the current literature concern private code

¹ M. Hildebrandt, ‘Legal and Technological Normativity: More (and Less) than Twin Sisters’ (2008) 12 *Techné: Research in Philosophy and Technology* 169, p. 178.

² R. Brownsword, ‘In the Year 2061: From Law to Technological Management’ (2015) 7 *Law, Innovation and Technology* 1, p. 29.

production, and the unintended creation of technological normativity. These gaps are then considered in the development of the digisprudence framework in the next chapter.

2. Normative criteria for law-making: the aspirations of legality

The strong legalism we saw in Chapter 3 might be described as the extreme end of a spectrum, at the opposite of which lies the open-ended particularism of certain strands of Critical Legal Studies scholarship.³ Legality, on the other hand, is viewed by some as a more aspirational concept, of fundamental importance to constitutional democracy: according to Bańkowski, it is a crucial element in what makes a free and democratic society – it is ‘something worth living for; something worth dying for’.⁴

While the precise meaning and components of legality are contested, there are some overarching themes. Bańkowski talks of the operation of individual will ameliorating the blanket heteronomy of legalism, with legality representing the appropriate interplay, or placing of the threshold, between the moralities of duty and aspiration (i.e. between those rules which citizens should follow ‘mindlessly’ like automata, and those which they should consider more deeply before acting, perhaps with the help of appropriate institutions).⁵ Brownsword views legality as concerning a ‘legal approach’ that embeds ‘participation, transparency, due process, and the like.’⁶

³ Z. Bańkowski and N. MacCormick, ‘Legality without Legalism’ in W. Krawietz et al. (eds.), *The Reasonable as Rational? On Legal Argumentation and Justification; Festschrift for Aulis Aarnio* (Berlin: Duncker & Humblot, 2000) p. 191; Z. Bańkowski, ‘Don’t Think About It: Legalism and Legality’ in M.M. Karlsson, Ó. Páll Jónsson and E.M. Brynjarsdóttir (eds.), *Rechtstheorie: Zeitschrift für Logik, Methodenlehre, Kybernetik und Soziologie des Rechts* (Berlin: Duncker & Humblot, 1993) p. 51.

⁴ Bańkowski, *supra* n. 3, p. 45.

⁵ *Ibid.*, pp. 56–57. See also Z. Bańkowski and B. Schafer, ‘Double-Click Justice: Legalism in the Computer Age’ (2007) 1 *Legisprudence* 31, pp. 36–37.

⁶ R. Brownsword, ‘Lost in Translation: Legality, Regulatory Margins, and Technological Management’ (2011) 26 *Berkeley Technology Law Journal* 1321, p. 1363.

He also speaks of ‘processual public law values of transparency, accountability, inclusive participation, reason-giving and the like together with the controls exerted by background fundamental values (such as compatibility with respect for human rights and human dignity).’⁷ For Hildebrandt, legality is about purpose binding, where the exercise of power by the sovereign is limited according to pre-existing principles, ‘tying the state to its own legal rules, but also [instantiating] a system of checks and balances that safeguards against the Sirens of tyranny or those of market fundamentalism’.⁸ It is also concerned with the provision of ‘fundamental rights that prevent [law] from turning into legalism.’⁹ She invokes the Greek legend of Odysseus as an illustration of the purpose binding principle – like its protagonist, in a constitutional democracy the government is bound by its own pre-defined rules, despite its notional power to redraw those rules and the evident temptation to do so.¹⁰ Ultimately this results in a ‘modern law’ that consists of ‘self-rule, disobedience and contestability’: (i) laws are constituted by a democratic legislator and are made visible and intelligible to those they seek to govern, (ii) the governed have the ability to violate those laws, and so the decision to comply is an exercise of autonomy, and (iii) the substance of the legal norms, and the consequences of their violation, can be contested in a court of law.¹¹

These various conceptions of legality have a certain liminal quality; they seem to point towards a kind of dignified space for contemplation and choice that lies somewhere between the two extremes of heteronomy and anarchism. Acting

⁷ Brownsword, ‘In the Year 2061’, *supra* n. 2, p. 48.

⁸ M. Hildebrandt, *Smart Technologies and the End(s) of Law: Novel Entanglements of Law and Technology* (London: Edward Elgar Publishing, 2015) p. 157f.

⁹ *Ibid.*, p. 157.

¹⁰ Lessig invokes the same legend to highlight the difference between what he calls a ‘codifying constitution’ (the codification of existing norms to maintain stability – Odysseus being tied to the mast) and a ‘transformative constitution’, in which new significant societal changes are sought to be implemented (as in the French revolution). See L. Lessig, *Code: Version 2.0* (New York: Basic Books, 2006) pp. 313–314.

¹¹ Hildebrandt, *Smart Technologies*, *supra* n. 8, p. 10.

appropriately involves more than simply falling in line with legalism's rote morality of duty, but also something more structured than a chaotically subjective choice based on the particulars of each and every circumstance. Somewhere between these extremes lies a balance of autonomy and duty, provided by legal and social frameworks whose guiding force and institutions create spaces that allow for that consideration to take place. The 'intimate justice' of particularity cannot be *true* justice because of its lack of even-handedness, while simultaneously an 'aloof' and objective justice will at times feel harsh and unforgiving.¹² The ideal of legality aims to tread a difficult line between these poles, providing a measure of institutional guidance and rule-bound certainty whilst also maintaining some freedom of choice and reflection. In that respect, then, it encompasses aspects of legalism; the latter is a necessary element of legality, providing a level of predictability that is necessary to avoid the need to enquire into the particularities of every circumstance.

For Bańkowski and MacCormick, strong legalism is a 'negative ideology'. Their definition of legality, which maps onto Wintgens' idea of 'weak legalism' (discussed below), makes use of the legalistic outlook insofar as formalism and certainty are required in the ordering of 'durable social organizations',¹³ but it rejects heteronomy untempered by rational and critical reflection:

It remains true, however, that rules without underlying principles of a kind that could be assented to by a rational autonomous being are rules that can be implements of tyranny. It is also true that every application of a rule is also an interpretation of it. Approaches to interpretation that ignore or undervalue the need for attention to principles, and to the consequences of decision [sic] judged against implicit values and principles of law, are undesirable on the

¹² L.L. Fuller, *The Morality of Law* (Yale University Press, 1977) p. 72.

¹³ Bańkowski and MacCormick, *supra* n. 3, p. 194.

same ground if to a lesser degree. *Legalism as vice is indeed the vice of this narrow governance of rules, unleavened by the principled approach of interpretation.*¹⁴

This view of legality thus encompasses legalism as far as is necessary to create a predictable and reliable institutional order, but leaves a space in which the individual can deliberate about what her course of action ought to be. Unlike the *strong* legalism set out in the previous chapter, legality on this view does not confuse the rules and heuristics with the entirety of the law – it allows a dignified space for reflexive exercise of reason, intelligence, and freedom,¹⁵ in contrast to the ‘one shot’ at autonomy allowed in the social contract-based proxy model of strong legalism.¹⁶

Drawing on Fuller, Bańkowski argues that whereas the legalistic attitude cares only to meet the threshold of the morality of duty and no more, legality expands this to include the idea of a morality of aspiration. Here the question is not simply of what is ‘owed’, but rather what comes into play are less strictly limned values, like authenticity.¹⁷ It is, he argues, not sufficient simply to meet the minimum standard represented by a bald interpretation of the rules; instead, these exist on an ‘aspirational scale’¹⁸ and there are times when it is appropriate to expect more of an actor (or indeed perhaps ‘less’, in the consequentialist sense of disobeying the rule being morally desirable if it means a better outcome, *contra* the Kantian categorical imperative). The morality of duty is one point on the aspirational scale and represents (*ceteris paribus*) the minimum action that is required. But the scale goes further; it is possible to do more. Aspiration goes beyond the morality of duty’s ‘rules of grammar’, aiming instead

¹⁴ *Ibid.* (my emphasis). This discussion of background principles operating to fashion individual interpretations connects with Dworkin’s theory of law as integrity. See R. Dworkin, *Law’s Empire* (Cambridge, Mass: Belknap Press, 1986).

¹⁵ J. Waldron, ‘The Rule of Law and the Importance of Procedure’ (2011) 50 *Nomos* 3, pp. 19–20.

¹⁶ L. Wintgens, *Legisprudence: Practical Reason in Legislation* (Surrey: Routledge, 2012) p. 206.

¹⁷ Bańkowski and Schafer, *supra* n. 5, p. 33; Bańkowski and MacCormick, *supra* n. 3, p. 183; Bańkowski, *supra* n. 3, p. 45.

¹⁸ Bańkowski, *supra* n. 3, p. 51.

for what Smith called ‘what is sublime and elegant in composition.’¹⁹ This related to Floridi’s discussion of ‘post-compliance ethics’ and the line between compliance with the bare law and the legitimate expectation that more is done.²⁰ As we shall see, the question of where this threshold should lie arises in the context of code: to what extent should the concrete behaviour of the technical design be legalistic (fixed configuration; the heteronomous end-user), and to what extent should it build in the aspirations of legality (flexible configuration; the autonomous end-user)? It also connects with the goals of responsible research and innovation (RRI), where the ‘substance’ of research is not constrained (unlawfulness aside) but it is nevertheless guided by pre-defined ‘normative anchor points’ towards desirable outcomes.²¹

2.1 Input and output legitimacy in law

We can see from the above scholars’ views of legality that it is a somewhat amorphous ideal, with conceptions of it focusing sometimes on the ex ante criteria that dictate the process of norm creation and the formal qualities of the resulting rules, sometimes on the other ideas of ex post criteria that provide for due process and non-arbitrariness in administration (this latter perspective is what I understand by the rule of law²²), and sometimes both.²³ Waldron makes a similar distinction between what law *is* (the ‘concept of law’, i.e. what constitutes law), and how it is administered and applied (the

¹⁹ Fuller, *supra* n. 12, p. 6, quoting A. Smith, *The Theory of Moral Sentiments* (2nd edn., London: Millar, 1761) p. 257.

²⁰ L. Floridi, ‘Soft Ethics, the Governance of the Digital and the General Data Protection Regulation’ (2018) 376 *Phil. Trans. R. Soc. A* 20180081.

²¹ R. von Schomberg, ‘A Vision of Responsible Research and Innovation’ in R. Owen, J. Bessant and M. Heintz (eds.), *Responsible Innovation* (Chichester, UK: John Wiley & Sons, Ltd, 2013) p. 57, *et passim*.

²² Cf. J. Raz, ‘The Rule of Law and Its Virtue’, *The Authority of Law: Essays on Law and Morality* (Oxford University Press, 1979).

²³ This is true both of Raz’s principles (*ibid.*, p. 218) and Fuller’s, the latter of which I discuss in detail *infra*.

'rule of law').²⁴ Legality on this account speaks to the formal qualities of the rules, while the rule of law speaks to how those rules are applied in practice. Neither of these ideas concerns the substantive content of the rules directly – the question is one of process, both before promulgation (i.e. in the design of the norm), and after (in its application).²⁵

Scharpf and Waldron each contrast the concepts of input (process) and output (result) legitimacy, within the traditional contexts of legislation and judicial review. Input legitimacy is about the process being followed, which in the traditional sphere requires participation or representation of some form.²⁶ Scharpf calls this 'government *by* the people';²⁷ Waldron gives the obvious examples of political equality and enfranchisement.²⁸ Output legitimacy, on the other hand, is about the 'proof being in the pudding' – legitimacy is established through an assessment of the results of a rule's operation. Scharpf calls this 'government *for* the people', where a result is deemed legitimate because it solves the problem it was aimed at.²⁹

Disagreement about the desirability of a norm's substantive content (its output) can exist alongside an agreement that the norm was arrived at and formulated in a proper way (its input legitimacy) – this is how the norm, despite its divisiveness, attains (political) legitimacy.³⁰ This distinction between input and output reasons chimes with the Fullerian ideas of the inner and external morality of law, the former of which is constituted by his principles of legality (these are discussed below). Whereas input

²⁴ J. Waldron, 'The Concept and the Rule of Law' (2008) 43 *Georgia Law Review* 1.

²⁵ Waldron appears to consider 'procedure' to relate only to ex post due process, arguing that Fuller's use of the word is misplaced and that the latter is in fact referring to formal validity. See Waldron, 'The Rule of Law and the Importance of Procedure', *supra* n. 15, p. 8.

²⁶ F. Scharpf, *Governing in Europe: Effective and Democratic?* (Oxford University Press, 1999) p. 7; J. Waldron, 'The Core of the Case against Judicial Review' (2006) 115 *Yale Law Journal* 1346, p. 1372.

²⁷ Scharpf, *supra* n. 26, p. 6 (my emphasis).

²⁸ Waldron, 'The Core of the Case against Judicial Review', *supra* n. 26, p. 1373.

²⁹ Scharpf, *supra* n. 26, p. 11 (my emphasis).

³⁰ Waldron, 'The Core of the Case against Judicial Review', *supra* n. 26, p. 1387.

criteria speak to the procedural aspects of the creation of a given rule or judicial decision, output criteria speak to its efficacy or desirability in the world. Goldoni describes the distinction in the following terms:

Input reasons are those reasons that apply to the procedural aspects of decisions, that is, to how a decision is reached. As a measure for legitimacy, input reasons take into account the fairness of the adopted procedure. Output reasons concern the content of decisions and they represent a moral yardstick for judging the legitimacy of technologies. What counts as legitimate, according to the output-based perspective, is the end result of a decision and its normative content, not how the decision was reached.³¹

Very crudely, then, a focus solely on input criteria is deontological, while a focus on only output criteria is consequentialist. As with Fuller, the two perspectives interact – the quality of the rule in action (i.e. its consequences, or output) is shaped by the conditions that channel how it was made; those conditions can tend towards normatively undesirable as well as desirable substantive rules³² (although of course Fuller argued in his debate with Hart that the principles of legality tended to minimise the possibility of substantive iniquity³³). The inner, input, morality constrains the substantive content of its outer, or output, morality; form circumscribes substance.³⁴ Wintgens' theory of legisprudence makes a similar claim: whether a given proposed legislative rule is legitimate or not is contingent on it being justified according to the principles of legisprudence, whose formal qualities dictate, apart from any substantive political content, the base level required in order for legitimacy to obtain. Floridi too suggests that the scope for 'soft ethics' to operate is determined by the space left after

³¹ M. Goldoni, 'The Politics of Code as Law: Toward Input Reasons' in J. Reichel and A.S. Lind (eds.), *Freedom of Expression, the Internet and Democracy* (Leiden: Brill, 2015) p. 127.

³² Cf. Waldron, 'The Core of the Case against Judicial Review', *supra* n. 26, p. 1374.

³³ L.L. Fuller, 'Positivism and Fidelity to Law: A Reply to Professor Hart' (1958) 71 *Harvard Law Review* 630, p. 636.

³⁴ *Ibid.*

'hard ethics' (and their legal instantiations) are defined, and so too is the quality of the possible forms that soft ethics can take predetermined to an extent by the morality of the constitutive laws that define *ab initio* that 'ethical geography'.³⁵

We can perhaps sense from the preceding discussion a roughly four-part classification of the various types of criteria, according to their target and temporal position. In terms of input (*ex ante*) criteria, these can be split into (i) procedural criteria that govern the process of deliberation that leads to a given norm being created,³⁶ and (ii) criteria that constrain what formal qualities the norm should have, assessed separately from its substantive content.³⁷ In terms of output (*ex post*) criteria we have (iii) mechanisms of due process, transparency, and accountability to enable the detection and remedy of wrongs in operation,³⁸ and (iv) assessments of the moral or political content of the norm itself.³⁹

As we shall see, most theorists' frameworks include criteria from more than one of these categories. Of course, as I have so far argued, computational legalism requires greater focus on the first two of these classifications, although in the private sector, at least, *ex ante* procedural criteria are less likely to be readily applicable than *ex ante* formal criteria, given the lack of incentives to facilitate participation. The digisprudential theory I am developing is primarily concerned with the second classification (formal *ex ante*, or input, criteria), part of which is intended to facilitate the *ex post* criterion of contestability (the third classification). In this way, the former

³⁵ Floridi, *supra* n. 20, pp. 5–6, where the author notes that 'soft digital ethics can be more easily exercised the more digital regulation is considered to be on the good side of the moral versus immoral divide'.

³⁶ Broadly, this view requires democratic participation, and roughly encompasses the analyses of Brownsword, Hildebrandt, and Goldoni.

³⁷ Roughly, this would encompass the analyses of Fuller, Wintgens, and this thesis.

³⁸ This is the *ex post* conception of procedure that Waldron refers to and includes Hildebrandt's requirement of contestability. I too include the latter in the digisprudential framework, as Chapter 5 will demonstrate.

³⁹ This would broadly encompass Brownsword's and Koops' analyses.

operate both to constrain iniquity and to facilitate the latter. I return to the question of criteria for code below in section 3, but for now the remainder of this section sets out more fully Fuller's and Wintgens' normative frameworks for the design of legal norms.

2.2 Fuller's internal morality of law

Perhaps the most prominent and influential discussion of normative principles for law-making can be found in Fuller's *The Morality of Law*.⁴⁰ Explicitly aspirational, Fuller's eight 'principles of legality' are intended to appeal 'to a sense of trusteeship and to the pride of the craftsman'.⁴¹ They are about trying to achieve excellence (not to say perfection) in the business of law-making and application – a primary consideration arising from his principles is how best to design a law, as distinct from what its political content is or ought to be. Indeed, Fuller uses the language of design on various occasions, referring to law-making as a 'craft',⁴² and to the eight principles as 'those laws respected by a carpenter who wants the house he builds to remain standing and serve the purpose of those living in it'.⁴³ As can be appreciated from the principles, listed below, they are not just about making good law from the perspective of the conscientious law-maker; they can also be viewed as *constraining* the unconscientious law-maker to prevent the possibility of (excessive) iniquity.⁴⁴ Fuller's eight principles are as follows:

- (1) **The generality of law.** In order for conduct to be regulated, rules must be laid down that display 'reasoned generality' rather than the 'patternless exercise of

⁴⁰ Fuller, *supra* n. 12.

⁴¹ *Ibid.*, p. 43.

⁴² *Ibid.*, pp. 43, 156.

⁴³ *Ibid.*, p. 96.

⁴⁴ Fuller, *supra* n. 33, p. 636.

political power'.⁴⁵ Arbitrariness is to be avoided. (Recall the discussion of legality above, and the threshold between duty and aspiration).

- (2) **Promulgation.** The rules must be made available to those who will be governed by them, who are thus empowered to interpret and criticise them, and observe how they are applied and enforced.⁴⁶
- (3) **Retroactive laws.** This principle overlaps with the others: if laws are promulgated which render conduct unlawful that was not prohibited at the time it took place, the possibility of citizens knowing and obeying the law becomes fatally undermined. This principle is not absolute, however; Fuller notes that while retrospective law-making is *prima facie* 'a monstrosity', in some cases a holistic view of the principles will require it in order to cure some other 'shipwreck' in the enterprise of legality.⁴⁷ This hints at the non-absolute, deliberative nature of legality.
- (4) **Clarity of laws.** Fuller describes this as one of the most essential principles. Whereas legalism deems that what looks like law *is* law (i.e. formal validity begets law, regardless of its substantive merits), this principle requires the legislator to do more. As Fuller puts it, 'it is obvious that obscure and incoherent legislation can make legality unattainable by anyone, or at least unattainable without an unauthorized revision which itself impairs legality.' If a rule is so unclear that its interpretation distorts either its original expression or the intention behind it, then recursively we hit the buffers of legality again, whereby the law as practiced is not the law as promulgated.⁴⁸
- (5) **Contradictions in the laws.** Fuller suggests that dealing with contradictions in legal norms is not simply a case of logic, that is to say it is not enough to observe that norm *A* cannot be the same as *not-A* ($\neg A$); clearly such a statement does not on its own assist in resolving the contradiction. Fuller makes the point that

⁴⁵ Fuller, *supra* n. 12, pp. 46–49.

⁴⁶ *Ibid.*, pp. 49–51.

⁴⁷ *Ibid.*, pp. 51–62.

⁴⁸ *Ibid.*, pp. 63–65.

in determining whether two laws are contradictory something more is required: an appeal to extra-legal factors is necessary to determine the state of the world, what he calls the ‘whole institutional setting of the problem – legal, moral, political, economic, and sociological.’⁴⁹

- (6) **Laws requiring the impossible.** Fuller’s treatment of this principle is complex and includes a discussion of criminal and delictual liability, unjust enrichment, and economic (taxation) law that is not of relevance here. The essential concept is simple, however: a law should not compel the impossible, for example that ‘one should become ten feet tall’.⁵⁰ The promulgation of laws which are impossible to follow risks ‘doing serious injustice or [...] diluting respect for law’. Whereas in other contexts (such as the classroom) the exhortation to impossible ends can be an incentive to aspire to better results, in the legal context the stakes are different, and higher.⁵¹
- (7) **Constancy of the law through time.** This requirement is interesting from the perspective of the dignified pace of law. Fuller observes that both retrospective laws and constantly changing laws are apt to create injustice. If one of the aims of law is to normalise expectations, this can only be achieved if norms have a chance to settle in to the society in which they are promulgated.⁵²
- (8) **Congruence between declared rule and official action.** Fuller describes this as the most complex principle, such congruence being potentially undermined in various ways, including ‘mistaken interpretation, inaccessibility of the law, lack of insight into what is required to maintain the integrity of a legal system, bribery, prejudice, indifference, stupidity, and the drive toward personal power.’⁵³ This is in a sense a catch-all requirement, under which institutions

⁴⁹ *Ibid.*, pp. 65–70. This idea of ‘peering in’ to the legal system from some outside vantage point, to garner from that external perspective necessary information, is echoed strongly in Wintgens’ principle of coherence, discussed in section 2.3.3 *infra*.

⁵⁰ *Ibid.*, p. 70, n. 29.

⁵¹ *Ibid.*, pp. 70–79.

⁵² *Ibid.*, pp. 79–81.

⁵³ *Ibid.*, p. 81.

such as procedural due process, contest, and judicial review operate to identify and address such problems. It relates also to the constancy and retroactivity principles; the settling of arrangements over time may be necessary for latent incongruences to emerge as circumstances evolve.

The purpose of the principles of legality is to ensure the presence of what Fuller calls the ‘inner morality of law’.⁵⁴ This inner morality is distinct from its ‘external’ morality, which represents the substantive content of legal norms. A legal norm can, theoretically, be simultaneously internally *moral* and externally *immoral* – the assessment of external morality will differ according to each individual’s moral outlook whereas, Fuller argues, the internal morality of law requires certain standards which are universal in a democratic polity. Internal morality should not be optional, whatever one’s political persuasion might be.⁵⁵

Fuller’s principles are a mix of input (ex ante formal) and output (ex post procedural) criteria. Principles three to six govern the form of a proposed rule, constraining ex ante what its substance can possibly be: the rule cannot be retroactive (with exceptions); it must be reasonably clear in order to enable interpretation by regulatees; it cannot contradict extant rules without amending or repealing them; and it cannot require the impossible. Principles two and eight are examples of ex post procedural criteria: the former requires that the rules once made are publicised, while the latter binds the implementing authority to operate according to a reasonable interpretation of the substantive content of the rule (subject to the overarching contestability requirement).

As will become apparent, there are significant overarching parallels between Fuller’s concept of legality and Wintgens’ legisprudence, although Wintgens’ formal prescriptions constrain even more tightly the substantive content a norm can possibly

⁵⁴ *Ibid.*, p. 4 *et passim*.

⁵⁵ I follow this same idea for the digisprudential affordances, as we shall see in the next chapter.

have. The same concern, adverted to above, applies however, whereby the formal characteristics ought to exist in *any* norm in order for it to be legitimate, regardless of the substantive political content (external morality) that it instrumentalises.

2.3 Wintgens' legisprudence

Legisprudence combines criteria for ex ante formal validity with additional criteria that constrain the possible substance of a given rule, rejecting a legalistic/positivist perspective in order actively to peer behind the 'veil of sovereignty'⁵⁶ to find additional legitimisation of the proposed norm. Given the sovereignty of the legislator, Wintgens says that there is no limit to the theoretical foundations for a rule: the rationale could for example be economic, sociological, or technical. The important point, however, is that there must always be *justification* from some theoretical basis other than just bare sovereign whim. Unlike strong legalism, which as we have seen is content not to lift the veil of sovereignty to reveal the reasoning that motivated a particular exercise of power, the legisprudence framework requires both formal qualities in the norms that are promulgated, and enquiries as to the other, extra-legal, theoretical bases that provide the necessary additional justification.

This provides some useful, and I would suggest necessary, theoretical foundations for a critique of computational legalism and for the guidance of the production of technological normativity. The remainder of this section sets out the main parts of the theory of legisprudence, before moving on to discuss each of its principles.

⁵⁶ Wintgens, *Legisprudence*, *supra* n. 16, p. 2. Recall the discussion in Chapter 3, section 3.5 (The veiling of code's production).

2.3.1 *What is legisprudence?*

Legisprudence aims to shift jurisprudence away from a focus on only the ex post reasoning of the judiciary and legal professionals towards a greater consideration of the ex ante reasoning of legislators. The intention is to provide an explicitly legal-theoretical (as opposed to ethical or political) framework for the creation of legal rules, instead of ignoring the latter as an aspect of the ‘dirty business’ of politics. Shklar’s fences of legalism – those boundaries that surround and aim to separate law from other human endeavours – are thus broken down. Wintgens summarises legisprudence like this:

A different position is to study legislative problems from the angle of legal theory. This approach I propose to call *legisprudence*. Legisprudence has as its object legislation and regulation, making use of the theoretical tools and insights of legal theory. The latter predominantly deals with the question of the *application* of law by the *judge*. Legisprudence enlarges the field of study to include the *creation* of law by the *legislator*.⁵⁷

Legisprudence is a practical approach through which those who legislate can avoid succumbing to a strongly-legalistic ideology. It adopts a position of ‘weak’ legalism, whereby fidelity to rules is accepted as necessary, but on the condition that the form of those rules meets certain criteria and the rules are subject to ongoing justification. As Wintgens puts it,

[L]egisprudence can therefore be taken to be a meta-theory of morality, in that it allows for the formulation of principles that justify external limitations [of freedom, i.e. legal norms]. It is the latter that make morality possible, without enforcing any substantive moral principle whatsoever.⁵⁸

⁵⁷ L. Wintgens, *Legisprudence: A New Theoretical Approach to Legislation* (Oxford: Hart, 2002) p. 2 (emphasis supplied, references omitted).

⁵⁸ Wintgens, *Legisprudence*, *supra* n. 16, p. 297.

The framework thus constrains the substance of the rules to which it is applied regardless of the subject area they are concerned with. If we consider that the principles of legality in a constitutional democracy are broadly about fairness and accountability, legisprudence can be viewed as a tool both to achieve legality at the outset of the legislative process, and as an on-going means of upholding it as time passes and circumstances change.⁵⁹ It views the law-making process in a holistic fashion that seeks to achieve not just formal validity but also a broader rationality in the enterprise of making new norms.⁶⁰

2.3.2 *The requirement of justification*

Wintgens describes the distinction between strong legalism, the type described in Chapter 3, and what he terms weak legalism, which is legalism tempered by what he calls the *justification test*,⁶¹ which lies at the centre of the theory – Wintgens says justification is ‘the core of legisprudence: limitations of freedom must be justified.’⁶² Whereas strong legalism is satisfied by the presence of an authorised sovereign and does not enquire (indeed, prohibits enquiry) as to how a particular rule was made, weak legalism permits the lifting of the ‘veil of sovereignty’ in order to enquire as to the reasons behind a particular rule.⁶³ The justification for a new rule cannot simply be the ‘bare sovereign power’ of the legislator,⁶⁴ nor the natural laws or social contract she might purport to instrumentalise.

Whereas under strong legalism the hierarchy of powers means the subordinate may not (in most cases) question the superior, under weak legalism the requirement of

⁵⁹ L. Wintgens, ‘The Rational Legislator Revisited. Bounded Rationality and Legisprudence’, *The Rationality and Justification of Legislation* (Springer, Heidelberg, 2013).

⁶⁰ Wintgens, *Legisprudence*, *supra* n. 16, pp. 234–235.

⁶¹ L. Wintgens, ‘Legisprudence as a New Theory of Legislation’ (2006) 19 *Ratio Juris* 1.

⁶² Wintgens, *Legisprudence*, *supra* n. 16, p. 220.

⁶³ *Ibid.*, p. 2.

⁶⁴ Hildebrandt, *Smart Technologies*, *supra* n. 8, p. 155.

justification enables precisely this – the hierarchy can be, for specific normative purposes and only to a certain extent, reversed. This is the work of legisprudence, which provides a framework for this temporary reversal, to legitimate the work of the legislator which the strongly legalistic perspective requires must be ignored. In this way the appropriate level of justification for a particular limitation on freedom (i.e. a legislative norm) can be ascertained, both in advance of its promulgation (*ex ante*), and as an on-going test of its efficacy in the world (*ex post*).

2.3.2.1 Individual freedom and the trade-off model

For Wintgens, there is a *principium*, or foundational principle, of individual freedom. This has two elements. The first is descriptive, akin to the ‘state of nature’ in political philosophy: ‘[i]n the absence of any norm, anyone is free. In the beginning that is, there is freedom. From this perspective, freedom is at the origin of our philosophical inquiry’.⁶⁵ The second is normative, in that the *principium* should be a *leitmotif*, or guide, for both politics and law.⁶⁶ By definition, legislative rules constrain that foundational freedom for some individual or group in some place at some time. According to legisprudence, the fact that a proposed norm constrains foundational freedom means it should be rejected *a priori*, unless its imposition is sufficiently justified; requiring citizens to acquiescence to rules simply because they are ‘there’, as strong legalism does, is not a legitimate exercise of power according to this conception of freedom as *principium* – it is merely the arbitrary exercise of power.⁶⁷ Individuals’ ‘conceptions of freedom’ are their subjective senses of what freedom is (and is not), and as such they should not be interfered with lightly by the political project of the legislator’s ‘conceptions about freedom’. Conceptions of freedom are the individual’s

⁶⁵ Wintgens, *Legisprudence*, *supra* n. 16, pp. 124, 207 and see generally *ibid.*, ch. 4 (‘Freedom in Context’).

⁶⁶ Wintgens, *Legisprudence*, *supra* n. 16, p. 207.

⁶⁷ Wintgens, ‘Legisprudence as a New Theory of Legislation’, *supra* n. 61, p. 10.

internalised and subjective understanding of the idea, whereas conceptions *about* freedom are concretisations that come from an external source, such as the state.⁶⁸ Under legisprudential theory, with the primacy it places on individual freedom, the individual's idea of substantive freedom therefore takes precedence over the state's external view of it.⁶⁹ The distinction seems somewhat similar to Hart's discussion of the internal and external points of view of rules, where the individual has an internalised view of the concept and the state is by definition 'on the outside', looking in.⁷⁰

As Chapter 3 discussed,⁷¹ under strong legalism the sovereign is given a 'general proxy' to promulgate rules, and the resulting legislative acts are thus *de facto* legitimated – the veil is not lifted to consider whether or not they are justified; they are deemed *a priori* to be so. Under this model, the individual circumscribes her absolute freedom *ab initio* through the 'outsourcing' of its limitation to the sovereign, thereafter accepting whatever limits the latter promulgates under that arrangement. The proxy model provides the sovereign with generalised justification for imposing limitations on the freedom of the individual.

By contrast, Wintgens' alternative 'trade off' model requires that such limitations be justified in each case, in order that the *principium* of maximising individual freedom (or, expressed another way, the minimising external limitations on individuals' conceptions of freedom) be honoured each time its scope is under threat of being circumscribed.⁷² Under weak legalism, then, there is no proxy that 'takes control' of the individual's conception of freedom and is able unilaterally to limit it. There is instead a 'trade-off', in which the sovereign's desire to impose regulation based

⁶⁸ Wintgens, *Legisprudence*, *supra* n. 16, pp. 126, 254.

⁶⁹ *Ibid.*, pp. 254–257.

⁷⁰ H.L.A. Hart, *The Concept of Law* (2nd edn., Oxford: Clarendon Press, 1994) pp. 89–91.

⁷¹ Chapter 3, section 2.2 (Legalism according to legisprudence).

⁷² Wintgens, *Legisprudence*, *supra* n. 16, p. 229.

on its ‘conceptions *about* freedom’ must be balanced with the individual’s subjective conceptions *of* freedom:

Legislative ruling on the trade-off theory is not a priori legitimate as it was in the proxy theory. Legitimation, according to the trade-off model, consists of a justification as to why acting on a conception about freedom is preferable to acting on a conception of freedom. The legitimation of law under the trade-off theory, in short, consists in a justification of each external limitation of freedom that is a priori presumed to be legitimate or justified under the proxy model.⁷³

This balance between the moral (individual) conception *of* freedom, and the political (sovereign) conception *about* freedom, prioritises the former over the latter, and puts in place a requirement of justification, the standards for which must be met in order for the balance legitimately to be tipped in favour of the limitation on freedom. This process of justification is one of ‘weighing and balancing the moral and political limitations of freedom.’⁷⁴ The justification test then is a challenge to the impact of a new legislative provision (i.e. an external limitation) on individual freedom. Moral justification is achieved when the principles of legisprudence are taken properly into account. Wintgens summarises this as follows:

It is on the basis of freedom as *principium* that a norm giver is to justify why freedom is limited, that is, because (1) social interaction is failing, and (2) weaker alternatives are insufficient. In addition, freedom as principle requires that (3) the norm giver justify why he is issuing an external limitation at a certain time, in addition to an upholding of the limitation of freedom over time, and (4) a justification of its relation to the legal system as a whole.⁷⁵

⁷³ *Ibid.*, p. 220.

⁷⁴ Wintgens, ‘Legisprudence as a New Theory of Legislation’, *supra* n. 61, p. 10.

⁷⁵ Wintgens, *Legisprudence*, *supra* n. 16, pp. 283–284 (emphasis supplied).

Those requirements map onto the four principles of legisprudence, which in turn become duties the legislator must consider in the course of making a new norm. The following discussion sets out each of these principles.

2.3.3 *The principle of coherence (PC)*⁷⁶

Wintgens views the legal system as a complex system of dynamic and intertwined rules, which has grown exponentially (rules beget rules, in order to facilitate the ‘operative closure’ of legalism⁷⁷). Within this he identifies four levels of coherence which apply to ex ante legislative as well as ex post judicial reasoning. The principle of coherence is cumulatively normative: its levels are stepped through, and in order to be properly justified on the basis of the PC, a legislative act should attain coherence at each level. Wintgens argues that ‘[c]oherent legislation as the upshot of freedom as *principium* takes citizens morally seriously in legislative and not only in judicial decision making.’⁷⁸

Level of coherence 0 (‘LoC₀’; ‘internal or synchronic coherence’). The basic vocabulary and grammar of a discourse, this level of coherence is about the building blocks of intelligibility, without which the substance of the concepts that make up the system cannot be communicated.⁷⁹ ‘Coherence₀ is a necessary condition for any discourse to make sense.’⁸⁰ This level is concerned with the basic elements of communication (grammar, semantics, the logic of individual norms), and their compatibility with one another. In an earlier paper Wintgens labels this first level of coherence ‘simultaneous consistency’, which has the slightly different meaning that no inconsistencies or contradictions be permitted within a particular decision or

⁷⁶ Wintgens, ‘Legisprudence as a New Theory of Legislation’, *supra* n. 61, pp. 15–22; Wintgens, *Legisprudence*, *supra* n. 16, pp. 235–257.

⁷⁷ N. Luhmann, ‘Self-Organization and Autopoiesis’ in B. Clarke et al. (eds.), *Emergence and Embodiment: New Essays on Second-Order Systems Theory* (Duke University Press, 2009).

⁷⁸ Wintgens, *Legisprudence*, *supra* n. 16, p. 256.

⁷⁹ *Ibid.*, p. 242.

⁸⁰ Wintgens, ‘Legisprudence as a New Theory of Legislation’, *supra* n. 61, p. 16.

instrument.⁸¹ These two elements – the alignment of individuals’ understanding of the intension of a concept and the absence of logical contradiction between those understandings – can be read together to make up LoC₀. As Fuller notes, difficulties in avoiding contradictions can arise both within and between legislative instruments, and it is in part the ‘legislative carelessness about the jibe of statutes with one another’⁸² that is ultimately ‘very hurtful to legality’.⁸³ The aim of the coherence principle is to mitigate this kind of carelessness.

This level relates to Hildebrandt’s discussion of the printing press as essential to the affordance of modern legality. Without an agreed vocabulary of settled meanings, facilitated by the communicative affordances of writing and later the printing press, this level of coherence will always be contingent on the accuracy and consistency of verbal communications between practitioners and between generations.⁸⁴

The idea of epistemic continuity is closely related to the second level of coherence, which begins to look at the relations between the elements of the discourse rather than their individual intelligibility.

LoC₁ (‘diachronic or rule coherence’). This level considers consistency over time – it is the requirement that similar cases should attract similar judgements. This is in large part the consistency required by the rule of law: everyone is equal before the law, and the external limitations on freedom should be uniform across every individual who is addressed by them (*ceteris paribus*). From the perspective of the legislator, who as the sovereign is not bound by *stare decisis*, this translates into the principle that the

⁸¹ L. Wintgens, ‘Legislation as an Object of Study of Legal Theory: Legisprudence’, *Legisprudence: A New Theoretical Approach to Legislation* (Oxford: Hart, 2002) pp. 36–37, citing Fuller, *supra* n. 12, pp. 65–70.

⁸² Fuller, *supra* n. 12, p. 69.

⁸³ *Ibid.*

⁸⁴ Hildebrandt, *Smart Technologies*, *supra* n. 8, p. 47 *et seq.* (‘3.3.1 Affordances of Information and Communication Infrastructures (ICIs)’).

rules should not be changed too frequently, but when they are, good reasons should be given.⁸⁵ If an element of justice is the modulation of expectations over time, it follows that injustice arises from the excessive promulgation of new rules that override arbitrarily what has gone before.⁸⁶

LoC₂ ('compossibility or system coherence'). Despite LoC₁, circumstances can and do change over time, and so can (and should) norms/external limitations of freedom, provided that the change is otherwise legitimated by the justification principle. The arguments that warrant this departure from precedent are provided by LoC₂; they take into account not just the individual facts of a particular case, but the legal system as a whole. Through 'systematic interpretation', which views the legal system holistically, it may be that another part of it invites, permits, or justifies a different interpretation and thus a different judgement from that which came before.

A paradigm example given by Wintgens is the question of whether to view a lease through the lens of contract or of property law.⁸⁷ Either choice is *prima facie* legitimate, but the legal implications differ significantly. Stopping at LoC₁ would require continuity with past similar decisions, so no change in approach would be mandated. In this situation, however, a departure could be justified under LoC₂: the judge views the legal system as a systematic whole and observes that there are parts of it other than those elements used in preceding cases that can legitimately influence her ruling. Wintgens provides a real-life example from Belgium that is perhaps more illustrative.⁸⁸ There, a judge ruled that a husband was eligible for a 'spousal premium', despite judicial precedent in that jurisdiction never previously having ruled in favour of a male spouse receiving it. The justification for departure on LoC₂ (in direct contradiction of LoC₁) was demonstrated by other instruments that expressed a

⁸⁵ Wintgens, 'Legislation as an Object of Study of Legal Theory: Legisprudence', *supra* n. 81, p. 38.

⁸⁶ Fuller terms this 'legislative inconstancy'. See Fuller, *supra* n. 12, pp. 79–80.

⁸⁷ Wintgens, 'Legisprudence as a New Theory of Legislation', *supra* n. 61, p. 19.

⁸⁸ *Ibid.*, n. 6.

general principle of spousal equality, including domestic legislation such as Belgium's Matrimonial Act 1976, and international treaties such as the European Convention on Human Rights and the Universal Declaration on Human Rights. In this instance, the judge took a holistic view of the system, rather than just the precedents relevant to the instant case, and found external, but justifying, reasons to rule differently. Indeed, when viewing the system as a whole, coherence is improved by such a ruling because it brings judicial precedent with respect to the spousal premium into line with the principles expressed in those various legislative instruments.⁸⁹ A departure from LoC₁, *justified by LoC₂*, resulted in a more coherent legal system; judicial interpretation was re-aligned to fit legislative principle.

Whereas the judge *assumes* the possibility of viewing the legal system as a coherent whole, it is the legislator's duty to facilitate that systematic 'wholeness', *contra* the infinite latitude that her sovereignty gives her. Of course, this is a difficult task, replete with possibilities for carelessness and oversight.⁹⁰ The point is that the legislator 'has to justify his external limitations so that they allow the judge to make coherence₂ arguments'.⁹¹ The unbridled sovereignty of the legislator means that he or she is not constrained in the way that the judge is by the assumption that the legal system is coherent. The onus on her is all the greater, then, to justify her legislative 'activism', in order that its effects cohere with the rest of the system, including ex post adjudication.⁹² One can see how this contrasts with strong legalism, according to which the legislator's promulgated rules are law, to be grappled with by the adjudicator regardless of how incoherent they may be. Whereas LoC₀ was concerned with mere logical coherence,

⁸⁹ Again, here we see shades of Dworkin's theory of law as integrity.

⁹⁰ Fuller, *supra* n. 12, pp. 65–70.

⁹¹ Wintgens, 'Legisprudence as a New Theory of Legislation', *supra* n. 61, p. 20.

⁹² *Ibid.*

LoC₂ is about ‘compossibility’, the requirement that norms do not contradict one another’s substantive effect.⁹³

LoC₃ (‘environmental coherence’). According to this level of coherence, attaining a holistic view of the system is not possible from a standpoint within that system – an external perspective is required to make sense of it.⁹⁴ Wintgens suggests that to get such a perspective is possible only by ‘leaning over the edges of what is considered the whole’.⁹⁵ Whereas LoC₁ and LoC₂ are concerned with the internal rationality of the legal system, LoC₃ places that coherence within a wider, non-legal, context. This is where Shklar’s fences of legalism are broken down; not only do we observe that law does not operate in a vacuum, but we require that sensitivity to this fact be embodied in it through its justification according to the broader societal context within which it operates.⁹⁶

This required sensitivity is what Wintgens calls ‘theory dependence’; at LoC₃ the legitimacy of the legislator’s proposed rule is dependent on some extra-legal theory that can justify it – it is not enough to look for justification ‘inside’ the legal system. Unlike strong legalism, where the perspective of the sovereign legislator is held to be a direct conduit to reality and so her pronouncements are isomorphic with that reality, under LoC₃ there is a requirement for an external mediating theoretical framework which justifies, according to the requirements of its own ‘regime of veridiction’,⁹⁷ the

⁹³ Wintgens, *Legisprudence*, *supra* n. 16, p. 252.

⁹⁴ Wintgens, ‘Legisprudence as a New Theory of Legislation’, *supra* n. 61, p. 21.

⁹⁵ *Ibid.*

⁹⁶ On the importance of viewing law as an integral part of a broader social good, see Z. Bańkowski, ‘Bringing the Outside In: The Ethical Life of Legal Institutions’ in T. Gizbert-Studnicki and J. Stelmach (eds.), *Law and Legal Cultures in the 21st Century* (Warsaw: Wolters Kluwer, 2007).

⁹⁷ This is Hildebrandt’s terminology, borrowing from Latour. She makes a related argument that ‘[t]he ends of law – though deeply entwined with their internal validation – are thus co-determined by the needs of the society it serves and co-constitutes’. The first part of this quote maps onto LOCs 1 and 2; the latter part onto LOC₃. See Hildebrandt, *Smart Technologies*, *supra* n. 8, pp. 144–145 citing B.

legislative rule that is to be made. From the perspective of that external theoretical framework, the law will take on a shape that is different from that of the sovereign who operates in a strong legalistic bubble, or indeed even that of the enlightened legal practitioner whose perspective is nonetheless circumscribed by her professional background. Fuller makes a related argument about identifying contradictions between rules as part of his fifth principle, noting that it is not ‘merely or even chiefly technological’ incompatibilities that must be taken into account but the ‘whole institutional setting of the problem – legal, moral, political, economic, and sociological.’⁹⁸ Wintgens gives the law and economics school as one example of this – the study of law from the external viewpoint of a non-legal field, i.e. economics.⁹⁹

On LoC₃, then, the legal system must be viewed holistically and *in context*. It is not simply a question of applying the rules according to the internal logics of LoCs 1 or 2 – something more is required (this hints at Fuller’s distinction between the moralities of duty and aspiration in the search for legality, cf. legalism¹⁰⁰). There is a connection here with MacCormick’s hermeneutic perspective on rules, whereby the legislator has both an internal view of the legal system and an external view of its coherence vis-à-vis the social context. Quoting MacCormick, Wintgens says:

Since law or a legal system refers to a ‘form of life’, as MacCormick and Aarnio rightly puts [sic] it, coherence, then, is not a matter of logic alone, but a matter of ‘making sense as a whole’. This ‘making sense as a whole’ refers to ‘the whole corpus of the normative system’, and thus brings MacCormick to state: ‘To

Latour, ‘Biography of an Inquiry: On a Book about Modes of Existence’ (2013) 43 *Social Studies of Science* 287.

⁹⁸ Fuller, *supra* n. 12, p. 70.

⁹⁹ Wintgens, ‘Legislation as an Object of Study of Legal Theory: Legisprudence’, *supra* n. 81, p. 22 and n. 47.

¹⁰⁰ See *ibid.*, p. 26 *et seq.* and Fuller, *supra* n. 12, p. 5 *et seq.*

put it crudely, legal decisions must make sense *in the world* and they must also make sense in *the context of a legal system*.¹⁰¹

From the perspective of rule-making, the proposed norm can be observed then from both the legal (internal) perspective and its broader social (external) perspective. Viewing it only from the ‘inside’ begets legalism, while the addition of extra-legal justification consolidates its legitimacy – its ability to make sense ‘in the world’. The norm must be justified therefore according to both the internal logic of the legal system (LoCs 1 and 2; the ‘cognitive internal’ aspect in MacCormick’s language¹⁰²), and the external reality of society (LoC₃), it being reflected in whatever external theoretical framework provides the extra justification that is required under legisprudence.¹⁰³ Rationality in the project of legislating arises from the legislator taking a hermeneutic perspective: the rational quality of a rule consists not just in (legal) formal validity, i.e. the internal perspective, but also validity flowing from an investigation into the ‘external social data’ that have been produced and rendered as knowledge by other scholarly fields.¹⁰⁴ Legal reality is thus made to relate to social reality.¹⁰⁵

2.3.4 *The principle of alternativity (PA)*¹⁰⁶

The PA requires that the creation of an external limitation on freedom (i.e. a legislative rule) must be preferable to the absence of that rule. Creating a rule that prohibits

¹⁰¹ Wintgens, ‘Legislation as an Object of Study of Legal Theory: Legisprudence’, *supra* n. 81, p. 35, citing N. MacCormick, *Legal Reasoning and Legal Theory* (Oxford University Press, 1994) p. 103 (Wintgens’ emphasis).

¹⁰² MacCormick, *supra* n. 101, p. 290 *et seq.*

¹⁰³ Wintgens, ‘Legislation as an Object of Study of Legal Theory: Legisprudence’, *supra* n. 81, p. 17 *et seq.*

¹⁰⁴ *Ibid.*, p. 31.

¹⁰⁵ *Ibid.*, p. 38.

¹⁰⁶ Wintgens, ‘Legisprudence as a New Theory of Legislation’, *supra* n. 61, pp. 10–11.

certain conduct removes the possibility of agonistic conflict,¹⁰⁷ which thereby contracts to that extent ‘social space’ while simultaneously expanding ‘political space’ to fill the gap that is created. Social space involves practices whose dimensions are discernible partly by the observation and resolution of conflict according to the practice’s internal rules, and if that possibility is removed, which an external rule threatens to do, the practice itself might also cease to exist. By imposing a rule externally, i.e. a legislative rule, the ability of individuals to choose is removed, thus potentially reducing their scope to exercise moral autonomy. The imposition of a rule can only be justified, then, to correct a dysfunction that the practice cannot resolve according to its own internal processes.¹⁰⁸ This relates to contestability as an inherent part of legitimacy, and legislators should be loath to promulgate rules without first considering whether an alternative scheme might have the desired effect. The PA is concerned not with the substantive content of the proposed rule, but with whether it is justified to have a *rule* at all – because freedom is notionally infinite prior to the imposition of a limitation (i.e. a rule) and, as we have seen, freedom is the principium under legisprudence, the proposal to impose a limitation must be *a priori* justified, regardless of its actual content.¹⁰⁹ The PA is therefore a threshold requirement, which once passed is connected particularly intimately with the principle of normative density with respect to the normative impact of the design mechanism that is chosen (this principle is discussed below). This idea of a threshold is related to the discussion in Chapter 2 of the spectrum of technological normativity.¹¹⁰

¹⁰⁷ *Ibid.*, p. 11. See also M. Hildebrandt, ‘Algorithmic Regulation and the Rule of Law’ (2018) 376 *Philosophical Transactions of the Royal Society A* 20170355.

¹⁰⁸ Wintgens, *Legisprudence*, *supra* n. 16, p. 257.

¹⁰⁹ *Ibid.*, p. 297.

¹¹⁰ Chapter 3, section 3.5 (A spectrum of technological normativity).

2.3.5 *The principle of temporality (PT)*¹¹¹

The principle of temporality signals a significant departure from the ‘single moment’ focus of strong legalism. Whereas legalism wants to ‘switch time off’,¹¹² the PT requires a recognition that external limitations (legislative instruments) exist in a historical context. Unlike natural (physical) laws, which are constant, the contexts within which legislation must operate are evolving, and so the process of enacting a legislative provision must take account of, and be responsive to, such contingency.

Over time the justification for a legislative norm may change. Whereas strong legalism takes no account of this (the law is the law until the legislator changes it; the morality of duty requires obedience to the rule as-is), the weak legalism of legisprudence requires that time be considered. In terms of equality, distinctions that obtained at the time of the rule’s promulgation may no longer hold, leading to discrimination.¹¹³ More broadly, justificatory reasons that held true at the time of promulgation may no longer apply. The legislator’s focus is the future, but because her rationality is, in economic terms, bounded, she cannot possibly foresee all the possible circumstances that in the future might undermine the justification of the norm she proposes in the present.¹¹⁴ Justifying the promulgated rule is thus an on-going process, and circumstances must be continually observed to ensure that the legislative norm continues to be an appropriate response to the circumstances at which it was targeted.¹¹⁵ Failure to do so is to fall back into legalism, where the rule is viewed as ‘just there’, to be followed without further consideration of its legitimacy. Legitimacy under legisprudence is therefore an ongoing process that requires continual renewal in

¹¹¹ Wintgens, ‘Legisprudence as a New Theory of Legislation’, *supra* n. 61, pp. 13–15.

¹¹² Wintgens, *Legisprudence*, *supra* n. 16, p. 268.

¹¹³ *Ibid.*, p. 269.

¹¹⁴ Wintgens, ‘The Rational Legislator Revisited. Bounded Rationality and Legisprudence’, *supra* n. 59; Wintgens, *Legisprudence*, *supra* n. 16, pp. 268–269; 281.

¹¹⁵ Wintgens, *Legisprudence*, *supra* n. 16, pp. 269–270.

response to the requirements of each of the principles.¹¹⁶ The PT requires consideration of the prospective effects of the rule, but because some effects are likely to be unintended, ongoing assessment and (if necessary) rectification are also required.¹¹⁷

2.3.6 *The principle of normative density (PN)*¹¹⁸

The principle of normative density is related to, but more nuanced than, the concept of proportionality. The extent of the limitation on freedom that a legislative norm imposes must be in proportion to its justification. In other words, the stronger (more ‘dense’ or ‘intense’, in Wintgens’ language¹¹⁹) the regulation, the greater the level of justification that is required to legitimate it. For him, normative density exists on a spectrum – sanctions represent the highest density, while other options include ‘regulatory techniques such as information, incentives such as tax relief, self-regulation based on codes of conduct or agreements, labelling and the like.’¹²⁰ The PN expects there to be a proportionate connection between a policy aim and the means by which it is achieved; the impact on freedom should be as close to the minimum required to achieve the policy aim as is possible, in order not to over-regulate.¹²¹ The use of a technique with a particular normative impact must therefore be justified against any techniques that would have a lesser impact.¹²² Again, one can appreciate the connection to the spectrum of normativity discussed in Chapter 2, from wired-in configuration to greater openness of behavioural possibilities.¹²³

¹¹⁶ *Ibid.*, pp. 270–271; 300–303.

¹¹⁷ *Ibid.*, pp. 301–304.

¹¹⁸ Wintgens, ‘Legisprudence as a New Theory of Legislation’, *supra* n. 61, pp. 11–13.

¹¹⁹ Wintgens, *Legisprudence*, *supra* n. 16, p. 271.

¹²⁰ *Ibid.*, pp. 299–300; Wintgens, ‘Legisprudence as a New Theory of Legislation’, *supra* n. 61, pp. 12–13.

¹²¹ Wintgens, *Legisprudence*, *supra* n. 16, pp. 276, 279.

¹²² Wintgens, ‘Legisprudence as a New Theory of Legislation’, *supra* n. 61, p. 13.

¹²³ Chapter 2, section 3.5 (A spectrum of technological normativity). This topic is returned to in Chapter 5.

2.4 Conclusion

We have seen then that the legisprudential principles are about legitimising an incursion on freedom, and without sufficient justification such incursion is *a priori* illegitimate. The principle of coherence speaks to the proposed norm's fit within the existing legal and social system. The principle of alternativity asks whether the imposition of an external *rule* is justified in order to ameliorate a dysfunction in a given social practice. The principle of temporality requires the rule-giver to maintain sensitivity to changing circumstances, and therefore to reappraise regularly the justification for the rule and adapt it if the conditions change such that it is no longer justified. Finally, the principle of normative density queries whether the instrumental technique of the rule is proportionate to its aim, in order to ensure that those to whom the norm applies are not unnecessarily over-regulated (i.e. their freedom is not unjustifiably limited).

Although the principles have equal weight, Wintgens notes that, like Fuller's principles, they do not apply equally in every case. The justification under each principle can therefore operate more or less strongly depending on the circumstances¹²⁴ (I consider this again in Chapter 5). As with Fuller's principles, these are aspirational and might never be fully embodied in the terms of a proposed norm, but the idea is to reach for the best possible laws, rather than to achieve a perfection that is unattainable due to the contingencies of time and the limits inherent to the legislator herself.¹²⁵ Observing the bare constitutional rules is a necessary but insufficient element of achieving legitimacy under legisprudence; the rule that is proposed must be rendered 'good' by its observation of the four principles¹²⁶ or, put another way, a proposed rule

¹²⁴ Wintgens, *Legisprudence*, *supra* n. 16, p. 280.

¹²⁵ *Ibid.*, pp. 282; 305–307.

¹²⁶ *Ibid.*, pp. 287–288.

is reasonable when it has been sufficiently justified, and unreasonable when it has not.¹²⁷

Returning to the discussion of input and output legitimacy above, we can see how the principles of legisprudence expand upon the categories. The PC is an ex ante formal standard for assessing the intelligibility of the proposed norm vis-à-vis both existing legal norms and one or more potentially legitimating extra-legal theories. The PA is ex ante procedural – it speaks to the decision of whether or not to institute the proposed norm, and ex ante formal in that it asks whether a *rule* is the correct format for achieving the desired outcome. The PT is both ex ante and ex post procedural, in that it requires justification both that at the time of original promulgation that that was the correct thing to do, and ongoing (ex post) legitimation of the norm. The PN is unusual in that it requires an ex ante assessment of the substance of the rule and the extent of the constraint that it imposes on the individual. To that extent it therefore does not fit within the criteria discussed above, but is nevertheless important.

Fuller's principles are to an extent more hands-off than Wintgens'; the latter constrain more forcefully what the substantive content of a rule can possibly be. We can, however, detect some overlaps between the two theories. Fuller's first principle requiring the use of rules connects with the legisprudential principle of alternativity – whether or not to use a rule in the first place. The second principle (promulgation) also connects with the principle of alternativity – can the mechanism chosen (if not a rule) be promulgated such that regulatees are able to understand how they are being regulated? It also connects with the principle of normativity, where the extent of normative force may be such, and in so many forms, that promulgation in the usual sense becomes impossible. The third, fifth, and seventh principles of legality (respectively, against retroactivity, against contradiction, and in favour of constancy) speak to the levels within the principle of coherence.

¹²⁷ *Ibid.*, p. 306.

The next chapter maps these criteria for law-making alongside the criteria for code-making below, in order to identify how a code artefact's design can afford mechanisms that embody the criteria, thereby ameliorating computational legalism. For now, the next section considers the literature on normative code-making.

3. Normative criteria for code-making

The literature on the regulation of and by technology is of course significant, but the subset that deals specifically with what normative criteria should justify and/or guide the production of code is very small.¹²⁸ According to Goldoni, this is due to a scepticism developing in the decade following Reidenberg and Lessig's 'code is law' thesis, which he suggests limited the 'flourishing of a debate on the normative criteria for assessing code as a lawmaking procedure'.¹²⁹ Ohm and Frankle have very recently made a similar argument:

Too many scholars have interpreted Lessig as doing little more than issuing a license to imagine that anything is possible online, falling into a 'science fiction trap'. Too rarely do they consider the process of *how* code ends up the way it does (let alone how regulators can make use of this process), leaving a significant void in the utility of this body of work.¹³⁰

The scepticism in the literature referred to is somewhat ironic from the perspective of the current analysis, because it brings to mind aspects of the legalistic ideology discussed in Chapter 3, whereby an unwillingness to consider extra-legal sources of normativity leads lawyers to retreat to their intellectual bunkers, from where they can continue to view law as a separate enterprise that is 'fenced off' (in Shklar's evocative phrase) from those other concerns. This strengthens the instinctive belief that code is

¹²⁸ Goldoni, *supra* n. 31, p. 123.

¹²⁹ *Ibid.*, p. 117.

¹³⁰ P. Ohm and J. Frankle, 'Desirable Inefficiency' (2019) 70 *Florida Law Review* 1, p. 23.

not, and should not be seen as, law, and that legal thinking should therefore not concern itself with it. Brownsword noted this tendency recently when he suggested that the domain of jurisprudence should be ‘redrawn’ to sensitise it to a ‘bigger regulatory picture’, which can include forms of non-legal normativity that are ‘at least as important as legal norms in the daily lives of people’.¹³¹

As Chapter 1 discussed,¹³² the purpose of this thesis is not to validate private enterprises as producers of law *per se*. Rather, its aim is that code which has normative effects is legitimate, which is to say it embodies effects, features, or affordances alongside its commercially-purposive functionality that ameliorate the negative effects of computational legalism. The issue then is not one of the ‘legalness’ of code rules *per se*, i.e. of viewing them as a source of law, but rather the question of how the ‘non-law’ of code can, in spite of those negative effects, be produced in ways that are legitimate from the perspective of the law and constitutional democracy.¹³³ A failure to do so leaves a significant and serious deficit in our understanding of how citizens, as end-users, have their behaviour enabled and constrained by unelected private enterprises.

The remainder of this chapter considers the existing literature on normative criteria for the production of code. Following Goldoni, it is possible to discern a broad separation between those arguments which focus on input criteria and those which focus on output criteria. First, I discuss briefly what these classifications mean in this context and why input criteria deserve much greater focus within the context of computational legalism, before summarising each scholar’s contribution. We can then take stock before moving onto the next chapter, where I build on the criteria discussed here to propose a framework of affordances whose presence in a design can legitimise privately-produced normative code.

¹³¹ Brownsword, ‘In the Year 2061’, *supra* n. 2, pp. 10–14, 30.

¹³² Chapter 1, section 1.2 (Highlighting a tension).

¹³³ Cf. Waldron, ‘The Concept and the Rule of Law’, *supra* n. 24, p. 12, where the author discusses how in characterising something as ‘law’, we ‘dignify it with a certain character’.

3.1 Input and output legitimacy in code

We saw above how a norm's legitimacy can be considered by focusing on its production and/or the effects it has in operation. Chapter 2 set out why in the computational context the deontology of input legitimacy is necessary: the *ex ante* characteristics of computational legalism demonstrate that an *ex post* consequentialist perspective is not, on its own, sufficient to ameliorate those negative characteristics. When we move from the traditional legislative sphere into the computational context, though, things pivot somewhat. The focus on process is not just one of participation – indeed, the participatory aspect will in a great many cases be minimal, owing to the private spheres within which technologies are developed. Rather, the 'input' aspect shifts to have more of a temporal focus, where more granular design decisions about particular aspects of the code's functionality are the focus of legitimation according to binding criteria – because of their private production they might not be the product of a participatory democratic process *per se*, but they are 'input' in the sense that they are crucial constituents of the products (outputs) of the design process that ultimately are responsible for the code's effects in the world.

The distinction is a subtle but crucial one in the context of computational legalism: if we only assess a system according to its operation in the real world (i.e. we apply only criteria that assess output) then the production ship has already sailed, and the opportunity to amend the design to remedy any defects we discover may be limited or impossible. Furthermore, this assumes that assessments of output are capable of detecting all salient negative effects which is, of course, far from guaranteed, especially owing to the opacity of code. The shift toward input criteria puts the focus on the design process, to ensure that certain design characteristics are in place *ab initio* that allow for better output assessments but simultaneously lessen to a degree the need for them, because the initial configuration of the system is more legitimate.

Within the sphere of privacy by design, Hartzog argues in favour of focusing on processual standards because ‘even certain risky designs can be tolerated so long as companies take the right steps to mitigate potential harm and ensure that debatable design decisions were justified.’¹³⁴ Thus, mandating certain processes can potentially mitigate risk through the requirement to consider, during the process of design, the extent to which the proposed code embodies the standards we wish to see in a legitimate normative order. This approach also has practical appeal in terms of reducing the expense and delay of having to reconfigure a design once post hoc assessment uncovers that it does not meet one or other requirement. Because of the integrated nature of software development processes, such ex post ‘patches’ are often less effective than approaches that took matters into consideration *ab initio*. As Luger and Golembewski note in the privacy by design sphere,

[a]ddressing these concerns at the end of a design cycle leaves the creators of the system with little time or agency to manoeuvre, and leads to a situation where potential privacy problems are addressed – if at all – as afterthoughts, with inelegant solutions and imperfect implementations bolted on to a mostly-complete design.¹³⁵

Importing the distinction between input and output reasons into the computational sphere, Koops observes that

Input legitimacy implies legitimacy through rules-of-the-game and the procedure followed, output legitimacy means that the result establishes legitimacy... *in the context of normative technology input legitimacy is a primary*

¹³⁴ W. Hartzog, *Privacy's Blueprint: The Battle to Control the Design of New Technologies* (Cambridge, Mass: Harvard University Press, 2018) p. 179.

¹³⁵ E. Luger and M. Golembewski, ‘Towards Fostering Compliance by Design; Drawing Designers into the Regulatory Frame’ in M. Taddeo and L. Floridi (eds.), *The Responsibilities of Online Service Providers* (Springer International Publishing, 2017) p. 296. See also L. Diver and B. Schafer, ‘Opening the Black Box: Petri Nets and Privacy by Design’ (2017) 31 *International Review of Law, Computers & Technology* 68, pp. 74–75.

concern. Because technology is often irreversible – once it is developed and applied in society, it is hard to fundamentally remove it from society in those applications – the process of developing technology is a key focus when normativity is at stake. After all, it may well be too late when technology simply appears in society to ask whether it is acceptable to use this technology; quite often, the genie may then be out of the bottle never to be put back in... *criteria addressing the process of technology development – ‘rules of the game’ – should be a key part of our acceptability criteria.*¹³⁶

Input criteria are important because the characteristics of code (i.e. computational legalism) militate against the effectiveness of ex post assessments of effects in the world (i.e. outputs). Goldoni argues along similar lines:

Given the nature and logic of architectural regulation, the emphasis on output legitimacy is misplaced for several reasons... The difficulty of reversing embedded code is often evident and makes it *fundamental to focus on the procedure and the actors involved in the development of the technology*. Second, given the opacity of architectural regulation, *to be aware of how technology is directly or indirectly impacting upon agents’ behaviours may prove to be too difficult* in many cases. Last but not least... the importance of default technology cannot be underestimated. What appears to be default in code is often taken as a natural and immutable fact.¹³⁷

He argues therefore that ‘input-based legitimacy should become the primary concern in choosing normative criteria.’¹³⁸ Moving from a focus on output (ex post) to input (ex ante) legitimacy is necessary if the public dimension involved in traditional rule-making is to be imported into the computational sphere, particularly when so much of the latter is privatised. Crucially, however, the latter does not replace the former – ex

¹³⁶ B.-J. Koops, ‘Criteria for Normative Technology: The Acceptability of “Code as Law” in Light of Democratic and Constitutional Values’ in R. Brownsword and K. Yeung (eds.), *Regulating Technologies: Legal Futures, Regulatory Frames and Technological Fixes* (Oxford; Portland, Or: Hart, 2008) p. 166 (my emphases).

¹³⁷ Goldoni, *supra* n. 31, p. 128 (my emphases).

¹³⁸ *Ibid.*

post measures remain crucially important in order to maintain a connection with institutional legal processes. Goldoni thus advocates for a shift from a ‘descriptive to a normative approach’ to code-as-law (recall that this is a reversal of the effects of computational legalism, where the normative becomes the descriptive¹³⁹).

In his brief survey of the literature, Goldoni categorises it between authors who have focused on input and output criteria, noting a tendency towards the latter. This is perhaps to be expected, because observations of the discernible real-world effects of code (i.e. its output) can be more easily subjected to a traditional legal critique based on substantive doctrine (i.e. a critique built around compliance). The problem with this view is that it does not address directly those who *produce* the very code that is in question – it sustains the ‘fencing off’ of jurisprudential analysis from the object of that analysis, which as I have emphasised is unsatisfactory given the realities of computational legalism. Lawyers continue to be viewed as *ex post* assessors of code without acknowledging designers as the *ex ante* producers of it.

Ultimately Goldoni suggests that two principles should govern code production: transparency and ‘publicness’.¹⁴⁰ The first suggests that rules embodied in code must be knowable in order that they can be observed and their creators held accountable, and the second that there must be opportunity for those subject to the rules to have a say in their creation.

In the remainder of this section I consider the literature on the question of normative criteria for code, following Goldoni in separating the works broadly into those who focus on substantive output criteria and those who focus on procedural input criteria. As I have argued, the latter is the more appropriate sphere for the construction of criteria that can assist in guiding the design (production) of digital artefacts. While

¹³⁹ See Chapter 3 *passim*. Bańkowski uses this exact formulation: ‘What we see then, is how the normative has become the descriptive. This gives us an example of rule following which has the machine-like quality of heteronomy’. See Bańkowski, *supra* n. 3, p. 56.

¹⁴⁰ Goldoni, *supra* n. 31, pp. 128–129.

ex post assessments are also crucially important, we have seen how computational legalism tends towards obfuscation, and thus the ability to carry out the assessments is itself contingent on ex ante design decisions that ensure they are possible. The complementarity of input and output criteria is therefore a part of the framework I develop in the next chapter.

3.2 Output legitimacy

3.2.1 Brownsword's 'technological management'

Writing from a natural law perspective,¹⁴¹ Brownsword's primary criterion for assessing techno-regulation is that of justification, which he characterises as a judgment on 'whether we are over-regulating or under-regulating'.¹⁴² This has a bearing on the central theme of Brownsword's work more generally: the acceptability of techno-regulation assessed from the perspective of human rights and human dignity. He views the latter as a question of 'empowerment', which consists of three elements: 'that one's capacity for making one's own choices should be recognised; that the choices one freely makes should be respected; and that the need for a supportive context for autonomous decision-making (and action) should be appreciated and acted upon'.¹⁴³ This conception of dignity leads ultimately to the suggestion, in the computational context, that individuals always retain the choice not to follow the rule as inscribed in the artefact.

¹⁴¹ R. Brownsword, 'Code, Control, and Choice: Why East Is East and West Is West' (2005) 25 *Legal Studies* 1, p. 10.

¹⁴² R. Brownsword, 'What the World Needs Now: Techno-Regulation, Human Rights and Human Dignity' in R. Brownsword (ed.), *Global governance and the quest for justice*, vol. 4 (Oxford: Hart Publishing, 2004) p. 205.

¹⁴³ *Ibid.*, p. 211.

To encourage the development of ‘moral community’,¹⁴⁴ the individual should where appropriate be positioned to take moral rather than merely ‘prudential’ choices (i.e. choices that are in one’s own interest).¹⁴⁵ Technological management is problematic not because it naturally favours a particular form of (a)moral reasoning, but rather because it has the capability of bypassing practical reason altogether,¹⁴⁶ effacing opportunities for both moral and prudential choice.¹⁴⁷ Without the opportunity to exercise such choice, the possibility of moral community begins to falter, with individuals becoming ‘de-moralised’, that is having their capacity for moral judgement corroded through the elision, by technological architecture, of opportunities to exercise a decision between two or more courses of (moral) action.¹⁴⁸ The result is a blunting of sensitivity to social norms, and a break-down in moral community.¹⁴⁹ Indeed, the very concept of morality might disappear altogether if the very possibility of infringing rights (i.e. doing harm) is removed by techno-regulation.¹⁵⁰

¹⁴⁴ A concept characterised as a community that is built on publicly-proclaimed principles that are open to review according to processes that are inclusive of its members. See Brownsword, ‘Lost in Translation’, *supra* n. 6, p. 1335 *et seq.*

¹⁴⁵ Brownsword, ‘Lost in Translation’, *supra* n. 6. See also Brownsword, ‘In the Year 2061’, *supra* n. 2, pp. 32–33.

¹⁴⁶ Brownsword, ‘Code, Control, and Choice’, *supra* n. 141, p. 13.

¹⁴⁷ Brownsword, ‘In the Year 2061’, *supra* n. 2, pp. 34–35.

¹⁴⁸ Brownsword, ‘Code, Control, and Choice’, *supra* n. 141, p. 4. See also A. Le Sueur, ‘Robot Government: Automated Decision-Making and Its Implications for Parliament’ in A. Horne and A. Le Sueur (eds.), *Parliament - Legislation and Accountability* (Oxford; Portland, Oregon: Hart, 2016) p. 192f, discussing the passage of the Social Security Act 1998 and the potential effects of the increased use of automation in public administrative decision making.

¹⁴⁹ Brownsword, ‘Code, Control, and Choice’, *supra* n. 141, p. 19, quoting criminologist D.J. Smith, ‘Changing Situations and Changing People’ in A. von Hirsch, D. Garland and A. Wakefield (eds.), *Ethical and Social Perspectives on Situational Crime Prevention* (Bloomsbury Publishing, 2004). See also Brownsword, ‘Lost in Translation’, *supra* n. 6 *et passim* and especially pp. 1355–1356.

¹⁵⁰ Brownsword, ‘What the World Needs Now’, *supra* n. 142, p. 231.

In his earlier work on ‘techno-regulation’ Brownsword maintains a focus on state regulators as its source, suggesting that any movements from what he terms normative regulation (that is, measures which invite compliance) towards non-normative regulation (measures that do not permit scope for choice) should be ventilated by means of a ‘regulatory margin’ that can ‘facilitate deliberation about, and review of, changes to the complexion of the regulatory environment’.¹⁵¹ In later work he suggests that this must take place *ex ante* in order to ratify the use of technological management before it is rolled out. Failure to do so will result in the potentially unlegitimated use of code which, because of the efficiency with which it enforces rules (i.e. its immediacy), closes the gaps in enforcement that previously permitted civil disobedience and the resulting friction and conflict that can be a driver for positive social change.¹⁵²

This overarching goal is sensible from a democratic perspective but does not provide much beyond the policy level in terms of how the production of code ought to be guided. The essential concern is that we ought to be wary of decisions that might lead to unfettered use of code for regulation. As an overarching goal, Brownsword’s requirement of respect for human dignity, embodied in the ability to reason practically and to exercise choice, is insightful and useful. In later work he expands beyond the focus on dignity and moral community to consider more legal-theoretical ideas, for example Fuller’s principles of legality (his discussion of legality is considered alongside Asscher’s in section 3.3.2 below). For him, Fuller’s characterisation of legality as involving a reciprocal relationship between the end-user and the state is key to the latter’s use of code, and therein lies his prescription for the ‘regulatory margin’ that can facilitate the participatory mechanisms that will legitimise such regulation.¹⁵³

¹⁵¹ Brownsword, ‘Lost in Translation’, *supra* n. 6, p. 1351.

¹⁵² Brownsword, ‘In the Year 2061’, *supra* n. 2, pp. 36–37.

¹⁵³ Brownsword, ‘Lost in Translation’, *supra* n. 6, pp. 1363–1364.

Brownsword's earlier focus on the *public* regulation of citizens means that his analyses do not venture far beyond relationship (a) in the model set out in Chapter 1 (depicting the normative relationship between the state and the citizen/end-user¹⁵⁴).

I return to Brownsword's more recent work in section 3.3.2 below, where he shifts towards a more legal-theoretical perspective that focuses specifically on the question of legitimacy with which I am concerned. For now, from his earlier work three criteria can be identified for the proper application of 'techno-regulation', namely (i) respect for individual dignity through the preservation of choice (and more choice is better), (ii) reciprocity between the regulator and the regulatee in the designing of norms, and (iii) the need for a delaying 'regulatory margin' that can facilitate this reciprocity.

3.2.2 *Leenes' 'techno-regulation'*

Leenes expands Brownsword's conception of techno-regulation to include private actors as well as the state.¹⁵⁵ His explicit sensitivity to the role of private regulators in creating technological normativity is useful, and so too are his analyses of normativity more generally. Despite his evident sensitivity to design theory, he nevertheless maintains a focus on techno-regulation as regulation borne of identifiable legal sources, namely state legislation or private contracts. Although relevant, then, this focus sets his analysis apart from my own, although he does obliquely reference the kinds of extra-legal normativity I am concerned with:

In the case of techno-norms implementing contractual terms or deriving legal status from the law... the legal status of the norms embedded in the artifact and the legal effects of breaching the norms are clear. In other cases the norms may be legally null and void and hence not legally bind individuals, yet as long

¹⁵⁴ Chapter 1, section 1.4.1 (Normative relationships online).

¹⁵⁵ R. Leenes, 'Framing Techno-Regulation: An Exploration of State and Non-State Regulation by Technology' (2011) 5 *Legisprudence* 143.

as the norms remain embedded in the technology they in fact do regulate behaviour: *legitimacy and effectiveness may be disjoint in practice*.¹⁵⁶

The latter class of norms that Leenes refers to is of course the focus of the thesis, although he does not say much more about it.¹⁵⁷ Ultimately, for Leenes the key factor is transparency of the ‘techno-norms’ and the process by which they are arrived at. For him, in an ideal situation regulatees consider the norms promulgated by privately-produced code to be legitimate, the latter being achieved by ‘engaging this community in deliberate discourse’ which ‘requires a free flow of unhindered vital information.’¹⁵⁸ This would appear, then, to overlap with Brownsword’s regulatory margin and Goldoni’s transparency and ‘publicness’ requirements. Again, the limitations of participation in the private design process are something that I consider again below.

3.2.3 *Koops’ ‘criteria for normative technology’*

Koops provides an overview of criteria to be considered when assessing what he calls ‘normative technology’ (i.e. code).¹⁵⁹ He notes many of the concerns we have already seen in the discussion of computational legalism, around for example the ability of code to establish new norms,¹⁶⁰ the effect of translating textual norms into code,¹⁶¹ and the applicability of democratic and constitutional values even in the context of private sector code production.¹⁶²

While his survey of other scholars’ criteria provides a useful overview of the literature, his classification perhaps adds rather than reduces complexity, as well as conflating concepts that I believe should be kept apart. For example, he classifies due

¹⁵⁶ *Ibid.*, p. 168 (my emphasis).

¹⁵⁷ Later work with van den Berg, discussed below, does consider this aspect.

¹⁵⁸ Leenes, *supra* n. 155, p. 167.

¹⁵⁹ Koops, *supra* n. 136.

¹⁶⁰ *Ibid.*, p. 159.

¹⁶¹ *Ibid.*, pp. 160–161.

¹⁶² *Ibid.*, p. 161.

process, legality, and ‘checks and balances’ all under the umbrella of the ‘rule of law’, and refers to them as substantive (as opposed to procedural or formal) criteria.¹⁶³ Similarly ‘transparency of rule-making’, ‘transparency of rules’, and accountability are listed as ‘secondary criteria’, while ‘rule of law’ and ‘democracy’ are listed as primary.¹⁶⁴ It is not clear that these concepts are quite so easily distinguished or prioritised.¹⁶⁵ Of course, these concepts are all contested and open to subjective interpretation, but without a clearer picture Koops’ criteria are a useful starting point but do not go much further (a point he acknowledges, and something he suggests be left to further research).

Koops takes what he describes as a ‘pragmatic, bottom-up’ approach that identifies the criteria already applied by other scholars, rather than the alternative ‘top-down... theory-based interpretation of law’.¹⁶⁶ Here we have another point of departure between his work and the present analysis. He is consciously concerned primarily with ‘outcome justice’ rather than the procedural or input justice discussed above. Although he does, as we saw in the quote above, acknowledge the fundamental importance of procedural (input) legitimacy, he nevertheless explicitly frames his analysis in terms of the ex post operation of specific technologies. This necessarily limits the extent to which it can engage with the design stage of a technology and separates it from the present analysis and those others discussed below. The set of criteria he develops is thus intended as a heuristic for structuring the process, rather than a means of performing it.¹⁶⁷ His fourth level of abstraction begins to push towards concrete practices, particularly in his class of ‘secondary criteria’, where he includes for

¹⁶³ *Ibid.*, p. 168.

¹⁶⁴ *Ibid.*, p. 169.

¹⁶⁵ For a useful overview that separates out various of these concepts, see J. Tasioulas, ‘The Rule of Law’ in J. Tasioulas (ed.), *The Cambridge Companion to the Philosophy of Law* (Cambridge University Press, 2019).

¹⁶⁶ Koops, *supra* n. 136, p. 162.

¹⁶⁷ *Ibid.*, p. 170.

example review, audit, the possibility of choice, optimal default settings, and context-adaptability.¹⁶⁸ He notes that the ‘proof of the pudding is in the eating’,¹⁶⁹ suggesting that what matters is testing of the criteria against concrete technologies. This will never, he says,

be a straightforward or uncontested exercise. For one thing, several of the criteria are culture-dependent, in their interpretation (e.g., moral values, democracy) or in their importance (e.g., human rights, choice).¹⁷⁰

Like other authors, this is the ex post ‘output’ legitimacy that represents a kind of ‘thick’ version of legitimacy. As a result, the substantive aspects of, for example, human rights, become part of the assessment, contributing to the difficulties that Koops refers to and the complexity of his criteria. Whether such assessments of substantive law can ever be expected to be carried out by designers all across the private sector remains to be seen, but it would seem doubtful in light of the complexity and nuance of the law and the limited resources of companies (especially SMEs) without dedicated legal departments who can investigate it and come to an informed conclusion as to what its implications are.¹⁷¹

Koops’ perspective seems, as mentioned above, to privilege the position of the lawyer as code assessor, thus maintaining an inbuilt bias towards legalistic ex post assessment. Goldoni criticises Koops on this ground: ‘In a rather typical legalistic and formalist fashion, Koops would also have lawyers testing the set of normative criteria.’¹⁷² The gaps between lawyers and designers and between a product’s design

¹⁶⁸ *Ibid.*, p. 168.

¹⁶⁹ *Ibid.*, p. 171.

¹⁷⁰ *Ibid.*, p. 170.

¹⁷¹ For a relevant discussion of some of the practical challenges involved in balancing ethics in design, see N. Manders-Huits, ‘What Values in Design? The Challenge of Incorporating Moral Values into Design’ (2011) 17 *Science and Engineering Ethics* 271.

¹⁷² Goldoni, *supra* n. 31, pp. 127–128.

and runtime phases seem therefore unfortunately to be maintained rather than bridged by Koops' analysis.

Koops also suggests that the list of criteria itself will require periodic reassessment,¹⁷³ but again this is because of his focus on substantive rather than formal or procedural legitimacy. The latter should be able to stand the test of time, as in Fuller's internal morality of law, because a procedure that follows legitimising formal principles ought to underpin the making of all code-based norms regardless of their substantive content.¹⁷⁴ Indeed, in a constitutional democracy we might say that it is a prerequisite of those rules being legitimate.¹⁷⁵ Furthermore, focusing on procedure also potentially simplifies the criteria that need to be applied, since (as we shall see) there are fewer of them, and they are more-or-less constant.

Admittedly, as previously mentioned Koops sees his set of criteria as a starting point that is open to further refinement, and to that extent his contribution is indeed useful. He finishes with an enjoinder to consider the question of 'ambient law', or the incorporation of legal norms and values of legality into technological infrastructure itself. This notion, developed alongside Hildebrandt,¹⁷⁶ is the precursor to the latter's concept of 'Legal Protection by Design', which the next section discusses.

¹⁷³ Koops, *supra* n. 136, p. 171.

¹⁷⁴ Although the legisprudential principle of temporality also requires periodic reassessment, this is of the substance of the rule and not the affordance of reassessment itself. The latter (i.e. providing the ability to reassess) is timeless, while the justification for a rule may change over time or indeed disappear – it is the principle of temporality (and its concordant affordance) that allows for this to be determined.

¹⁷⁵ Cf. M. Hildebrandt and B.-J. Koops, 'The Challenges of Ambient Law and Legal Protection in the Profiling Era' (2010) 73 *The Modern Law Review* 428, p. 454.

¹⁷⁶ Hildebrandt and Koops, *supra* n. 175.

3.3 Input legitimacy

In contrast to Koops' explicit focus on pragmatism rather than legal-theoretical approaches to assessing legitimacy, Hildebrandt, Brownsword, and Asscher each take the latter approach, and in so doing they push the focus away from output legitimacy towards input, or production, legitimacy.

3.3.1 Hildebrandt's 'Legal Protection by Design'

The concept of Legal Protection by Design, or LPbD, is central to this thesis – indeed its contribution can be viewed as an interpretation of LPbD as well as a framework for achieving it. In earlier work Hildebrandt used the term 'ambient law',¹⁷⁷ suggesting that we must 'find ways to articulate the legal framework of democracy and the rule of law into the technological architecture it aims to regulate, creating what has been called "Ambient Law"'.¹⁷⁸

Chapter 1 discussed the use of the term 'by design', and how it can perhaps confuse matters between substantive compliance with particular fields of law (most commonly privacy, as in privacy by design) and the more general – and indeed fundamental¹⁷⁹ – goal of achieving 'legal protection', or the multifaceted protection afforded by traditional legal processes. It is therefore more of a philosophical project about the operation of law and traditional legal institutions *within* the computational context, rather than merely about the application of substantive doctrine *to* the computational context.¹⁸⁰ Hildebrandt suggests that LPbD as an umbrella concept is

¹⁷⁷ M. Hildebrandt, 'A Vision of Ambient Law' in R. Brownsword and K. Yeung (eds.), *Regulating Technologies: legal futures, regulatory frames and technological fixes* (Oxford; Portland, Or: Hart, 2008).

¹⁷⁸ *Ibid.*, p. 176.

¹⁷⁹ Hildebrandt and Koops suggest that 'the challenge of Ambient Law is altogether far more fundamental than transposing "legal" norms into "technical" architectures.' See Hildebrandt and Koops, *supra* n. 175, p. 460.

¹⁸⁰ M. Hildebrandt, 'Legal Protection by Design: Objections and Refutations' (2011) 5 *Legisprudence* 223, pp. 238–239.

concerned with both aspects – on the one hand, technological normativity should comply with substantive law, and on the other it should be both *resist-able* and *contestable* in a traditional court of law.

As I have previously discussed, the first requirement (substantive compliance) is not the focus of the present thesis. The second and third requirements point to the design of an artefact, and what it enables the end-user to do: can she exercise choice, and can she contest the design in a court? For Hildebrandt, ‘[t]he “resistability” requirement rules out deterministic environments, and the contestability requirement rules out invisible regulation.’¹⁸¹ The goal is that ‘the exercise of [...] rights should not be obstructed by the intended or unintended effects of new technologies’.¹⁸² In essence, then, there are for her two criteria for the non-doctrinal – i.e. input – aspects of LPbD, and these are choice and transparency.¹⁸³ I have already set out in detail in Chapters 2 and 3 how computational legalism creates the conditions she is arguing against. The challenge now is to move beyond them to suggest solutions. Hildebrandt does not provide concrete suggestions on how these can be achieved, instead setting out the overarching goals of LPbD. She sounds a warning, too:

[d]eveloping a methodology for LPbD entails a vertiginous challenge to traditional doctrinal research methods within legal scholarship and to the scientific methods of computer science, requirements engineering and electronics.¹⁸⁴

This challenge is precisely what this thesis begins to grapple with, particularly in the next chapter. Crucially, I build on the design theory set out in Chapter 2 to suggest

¹⁸¹ Hildebrandt, *Smart Technologies*, *supra* n. 8, p. 218.

¹⁸² Hildebrandt, ‘Legal Protection by Design’, *supra* n. 180, p. 240.

¹⁸³ See also Hildebrandt and Koops, *supra* n. 175, p. 456.

¹⁸⁴ Hildebrandt, *Smart Technologies*, *supra* n. 8, p. 218.

ways that the second aspect of LPbD might be achieved, answering Hildebrandt's second enjoiner that we

should always include attention to the 'resistability' and contestability of the ensuing normativity, and should always involve *testing how the configuration or design of the affordances can best serve the goals of justice, legal certainty and purposiveness*.¹⁸⁵

These latter three elements – justice, legal certainty, and purposiveness – are central to Hildebrandt's understanding of the ends, or purposes, of law. The implication here is that the focus is on the design stage, where the affordances of the product are developed and where it can be considered whether they not only meet the product's commercial requirements, but also those of legitimacy so conceived. As she mentions in relation to the data protection by design provisions of the GDPR, '[data protection by design] will force existing technology developers to include a new set of requirements at the starting point of their design process, while at the same time creating a market for new technologies that help to render data processing systems compatible with the [GDPR]'.¹⁸⁶ More generally, LPbD requires that the design of a product's 'commercial' affordances (i.e. the things that make it attractive or useful to the end-user) must take account of its *legal* affordances; in (dis)affording particular behaviours for the end-user, the code must at all times permit the operation of the ideals of legality, which means the possibility of the end-user (i) observing the rules to which the system is subjecting her, (ii) exercising choice as to which rules apply, and (iii) contesting those rules in a court of law.¹⁸⁷ Hildebrandt's analysis thus concerns input criteria, even though the focus is about the end-user having the ability to exercise

¹⁸⁵ *Ibid.* (my emphasis).

¹⁸⁶ *Ibid.*, p. 221.

¹⁸⁷ M. Hildebrandt, 'Law As an Affordance: The Devil Is in the Vanishing Point(s)' (2017) 4 *Critical Analysis of Law* 116, p. 122.

her rights ex post. Her discussion of affordance and the ‘designing in’ of mechanisms to facilitate LPbD is inherently concerned with input criteria and the requirement that the design process reflect those ex ante requirements; if that is achieved then the ex post operation of the system will by definition embody the procedural, if not necessarily the substantive, aspects of output legitimacy (*ceteris paribus*).

3.3.2 *The Fullerian principles applied to code*

Both Brownsword and Asscher have considered Fuller’s principles of legality in the context of code, discussing how they might be adapted to that context.

3.3.2.1 Brownsword

In more recent work, Brownsword moves towards a more conventionally legal-theoretical perspective that is pluralist and sensitive to the private production of code and its capability to ‘compete with or complement, or simply supersede Hartian legal norms.’¹⁸⁸ For him, the principles of legality are an example of ‘cosmopolitan values’ that normatively bind all regulators, regardless of the substantive content of the regulations they promulgate.¹⁸⁹

Brownsword appears throughout to maintain an ontological separation between the ‘rule’ or decision which animates the use of a particular code measure, and the substantive effect of the measure itself (this is made explicit in his treatment of the second principle, as we shall see). This creates a distance between his analysis and the substantive materiality of code, and therefore the design questions – what an artefact (dis)affords, and how it mediates reality for the end-user – that, as we shall see in the next chapter, some of the Fullerian principles can usefully point towards. Nevertheless, Brownsword’s analysis is one of the very few that have considered code from the

¹⁸⁸ Brownsword, ‘In the Year 2061’, *supra* n. 2, pp. 10-14; 19.

¹⁸⁹ R. Brownsword, ‘Technological Management and the Rule of Law’ (2016) 8 *Law, Innovation and Technology* 100, p. 113.

perspective of Fuller's principles, and so it is a very useful source for the present analysis. In the discussion below, I follow Brownsword's ordering of the principles, which does not match that in the original text.¹⁹⁰

Promulgation of rules (second Fullerian principle). Brownsword claims that code environments are not governed by rules *per se* (and therefore there can be no operation of the first Fullerian principle, which states that there must be rules). Thus, for him, the second principle is converted into a requirement of transparency vis-à-vis the proposed use of technological management (i.e. code).¹⁹¹ The result is that regulatees should be given 'a fair opportunity to participate in the processes that will determine whether such a use is authorised',¹⁹² the idea being that 'the purpose of promulgation is to invite public debate about the use of [code] measures.'¹⁹³ As mentioned above, whether this is workable in commercial contexts where digital artefacts are designed is at the very least questionable, particularly given the lack of incentives designers have to consult their end-users. Brownsword seems to aim at transparency of *intent* rather than actual technical transparency; the gap between the two is problematic as I shall discuss later. At any rate, while the role of participation in the design process may be a desirable one, it is at most complementary to the digisprudential theory I am developing because it speaks to either (i) organisational processes (Brownsword's focus on the decisions to use code, rather than the code itself), or (ii) the substantive functionality that makes the code attractive to a given class of end-user, at which point the question has moved beyond the 'constitutional' design principles that should be present in all digital artefacts regardless of their commercial purpose. This is a theme I return to below.

¹⁹⁰ Set out in Fuller, *supra* n. 12, ch. 2 and summarised on p. 39.

¹⁹¹ Brownsword, 'Technological Management and the Rule of Law', *supra* n. 189, p. 117.

¹⁹² *Ibid.*

¹⁹³ *Ibid.*, p. 118.

Rules should be prospective, not retrospective (third principle). Although it is possible for retrospective acts in technologically-managed environments (Brownsword gives the examples of database records being deleted, or contractual provisions being altered) he suggests that in general changes to the environment are prospective and therefore technological management does not introduce any new risk of ‘unfair retrospective penalisation of conduct.’¹⁹⁴

Rules should not require the impossible (sixth principle). Brownsword’s discussion here focuses on the notional mental state of the regulatee, and how various legal systems deal with (criminal) attempts that are frustrated because of their impossibility.¹⁹⁵ Here, though, his focus shifts to the subjective position of the regulatee, rather than the legitimacy of the technological management measure, and so his analysis of this principle is not relevant here.

Rules should be clear (fourth principle). In the context of technological management, Brownsword suggests that the channelling of regulatees’ behaviour ‘should be done with less friction and confusion where the regulatory signal is clearly and decisively transmitted.’¹⁹⁶ Regulatees ought to have it communicated clearly that their conduct will be limited in some way by a technological measure.

Rules should be relatively constant (seventh principle). Brownsword suggests that frequent changes in what an application of technological management permits and denies, either because of malfunction or a deliberate change to the ‘regulatory coding’, might invite the uncertainty in regulatees that the principle aims to guard against. He warns against causing confusion to regulatees, caused by frequent code changes, resulting in their contravention of the ‘terms’ of the system and thus the levying of what are therefore unfair penalties because of a lack of constancy.¹⁹⁷

¹⁹⁴ *Ibid.*, p. 120.

¹⁹⁵ *Ibid.*, pp. 120–122.

¹⁹⁶ *Ibid.*, p. 122.

¹⁹⁷ *Ibid.*, p. 123.

Rules should not be contradictory (fifth principle). In the technological management context this principle should be ‘consistent in allowing or disallowing a certain “act”’.¹⁹⁸ This would appear more or less to match his proposal for the seventh principle, discussed above. He suggests further that where the system permits a particular act, i.e. renders it possible, the regulatee should be given the benefit of an assumption against levying a penalty where it was the ‘fault’ of the system that what should have been a prohibited act (presumably owing to some other legal requirement) was in fact made possible.¹⁹⁹

The practical administration of rules must match their content (eighth principle). Again, Brownsword maintains the ontological separation between code and the ‘offline’ rules that sit ‘behind’ the technological measure and animate its use. This is perhaps necessary for this particular principle. Here, his focus is on the translation of rules into code, and it is there where his concern over congruence arises: whether the rule as stated (or written) is properly reflected in the technological management measure.²⁰⁰ This is the well-known problem of compliance by design,²⁰¹ or the translation of “law in the books” to “law in other technologies”.²⁰²

Rules should be general (first principle). Here Brownsword shifts focus onto the question of algorithmic profiling, whereby the technological management system can feasibly have as many bespoke rules as there are regulatees subject to it.²⁰³ To that extent, this focus on ‘data-driven’ law is, as explained in Chapter 1, outwith the scope of the thesis.²⁰⁴ Nevertheless, the concept of generality is still relevant in the ‘code-driven’ context on which I am focused. For example, end-users can believe themselves

¹⁹⁸ *Ibid.*, p. 124.

¹⁹⁹ *Ibid.*

²⁰⁰ *Ibid.*, pp. 124–125.

²⁰¹ Discussed in Chapter 1, section 1.4 (Legitimacy vs. ‘compliance by design’).

²⁰² Hildebrandt and Koops, *supra* n. 175, p. 452, *et seq.*

²⁰³ Brownsword, ‘Technological Management and the Rule of Law’, *supra* n. 189, pp. 125–126.

²⁰⁴ See the discussion in Chapter 1, section 1.1 (Code).

to be having the same experience as one another, when in fact this is not the case (an evocative example is the Facebook emotional contagion experiment, an example of so-called ‘A/B testing’²⁰⁵). Another example is the use of software ‘alpha’ and ‘beta’ programmes, where end-users elect to access new features in a system before they are fully completed and ready for widespread distribution. In some cases, updates are released that fragment the uniformity of the code across the userbase.

Ultimately Brownsword summarises his understanding of Fuller’s principles in the technological management context as requiring ‘openness, or transparency, in authorising the use of measures of technological management for particular regulatory purposes, supported by ideals of fairness and due process.’²⁰⁶ As I previously mentioned, this focus on authorisation maintains an ontological separation between the policy animating the use of technological management and the code that actually implements the normativity. The focus too on ex ante deliberation, the ‘regulatory margin’, is sensible from the perspective of large public regulatory bodies but is, I suggest, less plausible in the context of small commercial enterprises creating low-cost digital artefacts. The suggestion that such firms submit to ‘special procedures possibly akin to applications for planning permission’²⁰⁷ seems unlikely to receive much purchase given the fecundity of the Internet as a generative platform and the ease with which almost anyone can get started creating code that has normative effect.²⁰⁸ Where Brownsword does discuss the private sector, he sets up a dichotomy between ex ante ‘approval and authorisation’ on the one hand, and ex post ‘challenge and review’ on the other. Where the ex ante measures are not present, the ex post measures are thus

²⁰⁵ R. Booth, ‘Facebook Reveals News Feed Experiment to Control Emotions’ *The Guardian* (29 June 2014) <<https://www.theguardian.com/technology/2014/jun/29/facebook-users-emotions-news-feeds>>.

²⁰⁶ Brownsword, ‘Technological Management and the Rule of Law’, *supra* n. 189, p. 127.

²⁰⁷ *Ibid.*

²⁰⁸ J. Zittrain, *The Future of the Internet and How To Stop It* (New Haven, Connecticut: Yale University Press, 2008).

necessary.²⁰⁹ While this is a good starting point for thinking about the responsibilities of software designers and their employers, I suggest that on its own this does not go far enough. As I have argued throughout this thesis, and as we saw from the discussion of computational legalism in Chapter 3, an either/or approach is insufficient; if we rely only on ex ante measures we cannot account for emergent and/or unexpected regulatory effects, while if we rely only on ex post measures there may be significant harm being done that will not be detected in order for those processes to be invoked.

In summary, then, Brownsword's application of Fuller boils down to the need for openness, transparency, and due process in the authorisation of the use of technological management, together with the requirement – a longstanding part of his work – that the conditions for moral community be maintained. As I discussed in section 3.2.1 above, for him the latter conditions are provided where there is preservation of individual choice and the ability to make a moral decision.²¹⁰ Brownsword does not engage with the concrete materiality of design, beyond a passing reference to transparency 'about how the particular technologies work.'²¹¹ For my purposes, this observation is particularly apposite: 'while it is certainly a *necessary* condition for the acceptability of a particular use of technological management that the underlying rule or policy is compatible with the Rule of Law, it might not be *sufficient*.'²¹² My contention is that it is indeed not sufficient, because it does not engage with the materiality of the design that actually implements the normativity that ought to be legitimate. This will be discussed in the next chapter, but for now I turn to Asscher's application of Fuller's principles to code.

²⁰⁹ Brownsword, 'Technological Management and the Rule of Law', *supra* n. 189, p. 117.

²¹⁰ *Ibid.*, pp. 129–131.

²¹¹ *Ibid.*, p. 139.

²¹² *Ibid.* (emphasis supplied).

3.3.2.2 Asscher

Asscher's analysis is more closely focused on the idea of code *per se*, as opposed to Brownsword's focus on the legitimacy of the rules operating *behind* the code. Because of this sensitivity to the code *per se* Asscher's analysis is therefore closer to my own. His approach in adapting Fuller's principles is to pose numerous questions for the assessment of code.²¹³ First, is it transparent: can citizens discern the rules they are subject to, or, in computational terms, can we be sure of what the code is doing, and is this what we expect to happen?²¹⁴ Second, is the code consistent, both in the temporal sense (i.e. it is not updated arbitrarily), and in the sense of congruence both with other code rules and with legal rules? This speaks to the trust that end-users can have in the system. Third, is its provenance clear, i.e. can end-users identify who is responsible for its production ('can a sovereign be distinguished that can also be held accountable for the influence of the software?'²¹⁵). Fourth, is autonomy respected through the preservation of the choice of whether or not to obey?²¹⁶ He distils these adaptations of Fuller's principles into the following 'checklist':

1. Can *rules* be distinguished in the code?
2. Can they be *understood*, i.e., is it understandable how code works and what it does? If so, are those rules transparent, are they accessible to the general public?
3. Can the rules be *trusted*, is there any guarantee that rules are not changed during the game? Are code rules *reliable* in the sense that they are predictable?
4. Is there a sovereign? An *authority* who makes the code rules?

²¹³ L. Asscher, "Code" as Law: Using Fuller to Assess Code Rules' in E. Dommering and L. Asscher (eds.), *Coding Regulation: Essays on the Normative Role of Information Technology* (The Hague: TMC Asser Press, 2006).

²¹⁴ *Ibid.*, p. 84.

²¹⁵ *Ibid.*

²¹⁶ *Ibid.*, pp. 84–85.

5. Is there a choice? Can consumers/citizens choose not to obey the rules?
Can consumers/citizens freely *choose* another system of law/code?²¹⁷

If the answer to the first question is negative, the rest can be ignored. Questions two and three are connected, while the fourth is a practical concern. Interestingly, for Asscher the fifth question, of whether the end-user retains choice, is connected with the issue of competition (cf. Brownsword's interest in choice as a foundation for moral reasoning and community).²¹⁸ For him, the questions are about restoring balance between code and law. This is connected with the traditional process of legislation and law-application, one element of which is the practice of balancing competing interests. Asscher suggests that his Fullerian analysis of code is apt to assess whether the balance of power has tipped away from institutional law, in favour of the 'code world', and thereby whether some kind of state intervention is required to restore it.²¹⁹

Asscher's criteria can be summarised as follows. When there are rules enforced by code, (i) the code must be transparent (understandable to those regulated by it – question 2), (ii) the code must be trustworthy and reliable (it operates as expected, and is not changed arbitrarily), (iii) the producers of the code must be identifiable, and (iv) end-users must have the choice of whether or not to obey its rules. We will see in the next chapter how these considerations are taken into account in the suggested digisprudential affordances.

3.3.2.3 Does code contain rules *per se*?

We have seen from Brownsword's analysis a reticence to approach substantive code as rules *per se*. He prefers to focus on the policies that operate in the background to animate particular applications of code as regulation. Similarly, scholars like Benoliel focus on technological standards as the 'background' rules from which individual

²¹⁷ *Ibid.*, p. 85.

²¹⁸ *Ibid.*

²¹⁹ *Ibid.*, pp. 85–86.

digital artefacts are derived.²²⁰ These are more readily compared to legal norms *per se*, because they are indeed couched in normative terms that set out the necessary conditions, or protocols, for interfacing with a particular technology²²¹ (in this regard they are like the constitutive rules of a game of chess, discussed in Chapter 3²²²). The point that technical standards are rules is accepted; protocols such as TCP/IP and the standards processes²²³ from which they arose do set out the ‘rules of the game’ which must be followed in order to be a player. While those rules do to an extent curtail what is possible with the systems that are built upon them, they are in general constitutive in an infrastructural sense, operating at a level far beneath the technological normativities that I am concerned with. They are less about the individual artefact’s design, and more about the underlying foundations of the network. This is in part because of the ‘end to end’ principle, where the network provides only the most basic and generic of infrastructure, and the applications that provide bespoke functionality are properly ‘at the edges’.²²⁴ A great deal has been written about the regulation of the Internet and how its architecture facilitates particular behaviours – indeed, most early ‘cyberlaw’ scholarship focused on this level of the technology.²²⁵ The focus of this thesis, however, is on the individual artefact as the site at which technological normativity is imposed, and so the ‘rules’ in question are not the underlying infrastructural standards with which all networked devices must comply in order to communicate, but rather the *sui generis* rules that constrain or enable behaviour in the ‘immediate’ and material

²²⁰ D. Benoliel, ‘Technological Standards, Inc.: Rethinking Cyberspace Regulatory Epistemology’ (2004) 92 *Cal. L. Rev.* 1069.

²²¹ For a critical account of the latter topic see A.R. Galloway, *Protocol: How Control Exists after Decentralization* (Cambridge, Mass: MIT Press, 2004).

²²² Chapter 3, section 3.2.1 (Constitutive and regulative rules).

²²³ For example, the Request for Comments (RFC) documents that since the 1970s have set out the workings of all the major technologies underlying the Internet. See Galloway, *supra* n. 221, p. 6 *et seq.*

²²⁴ Zittrain, *supra* n. 208, p. 164.

²²⁵ I discussed this in Chapter 1, section 1.2 (Highlighting a tension).

design of the artefact, as identified according to the theories set out in Chapter 2. The sovereignty involved here is that of the designer, developer or enterprise that is responsible for the affordances of a given artefact's design, and not those who develop the underlying infrastructure of the network that the artefact in turn relies upon.

In this regard, Asscher suggests also that 'rules should not be confused with the technical commands within a certain computer language but must be understood on the conceptual level'.²²⁶ I am not convinced that this is so, or at least that the issue of code-as-rules *per se* can be quite so easily dismissed. While it may be too much to focus on the individual commands in source code, the materiality of the system in operation is precisely where the action happens, and thus is it not only appropriate but is indeed necessary to focus upon it (I made the same argument against Brownsword in section 3.3.2.1 above). It may be that Asscher is simply implying that code does not present us with rules in the conventional Austinian sense of a law-maker issuing a command. He is possibly adverting to this when he refers to 'the subtle examples of the intertwined effects of legal policy and software effects',²²⁷ the latter perhaps pointing to the concrete, 'a-legal' effects of technological normativity. The 'rule' in this sense comes about in its inducement or enforcement of behavioural patterns that become like habits,²²⁸ but it is nevertheless the case that it is the design *per se* that makes this happen.

We saw in Chapter 3 Leenes' and Koops' suggestion that private enterprises' negligent production of privacy-eroding code can be viewed as akin to a rule stating that privacy is not important, or is less important than other values.²²⁹

²²⁶ Asscher, *supra* n. 213, p. 83.

²²⁷ *Ibid.*, p. 87.

²²⁸ Hildebrandt, 'Legal Protection by Design', *supra* n. 180, p. 239 and n. 55.

²²⁹ R. Leenes and B.-J. Koops, "Code" and Privacy or How Technology Is Slowly Eroding Privacy' in E. Dommering and L. Asscher (eds.), *Coding Regulation: Essays on the Normative Role of Information Technology* (The Hague: TMC Asser Press, 2006) p. 191. See also the discussion in Chapter 3, section 3.2 (Representationalism).

[w]e are inclined to think that the development and application of code that negligently fails to take privacy effects into account can indeed be seen as the embedding of a ‘rule’ in the technology, namely that privacy is unimportant and secondary to other values that the code primarily serves.²³⁰

This sensitivity to the totality of the code’s effects is important, shifting us away from the narrow focus on just purposive effects, adopted in much of the literature.²³¹ We saw in Chapter 1 how the scope of our enquiry ought to consider not just the intended regulatory effects of code, but also those wider ‘techno-effects’ that may not have a legal underpinning, whether public or private.²³² It is important to consider not just the *intended* normative effects of a system but also its potential unintended effects, and if those are deleterious to the values of legitimacy we have been discussing then this is a cause for concern.²³³ Again, as has been emphasised throughout, this is where the issue of ex ante design standards comes in, including ensuring that *unexpected* effects are capable of being dealt with post hoc.

Adapting Leenes’ and Koops’ formulation from the quote above, we can say that code rules which (negligently) undermine the standards of legitimacy that normative orders ought to reflect in effect represent ‘rules’ stating that those elements are not important and need not be valued. The (dis)affordances and inscriptions that the code embody can be seen as rules of this sort, and the code ought to be compatible with those standards by providing certain (digisprudential) affordances. In so doing, the likelihood of the artefact’s commercial normativity being illegitimate is accordingly reduced.

²³⁰ *Ibid.*

²³¹ Cf. B. van den Berg and R.E. Leenes, ‘Abort, Retry, Fail: Scoping Techno-Regulation and Other Techno-Effects’ in M. Hildebrandt and J. Gaakeer (eds.), *Human Law and Computer Law: Comparative Perspectives* (Dordrecht: Springer Netherlands, 2013).

²³² *Ibid.*, p. 81.

²³³ *Ibid.*, p. 84.

3.4 Conclusion

To conclude, we can summarise the various contributions from the work surveyed in the previous sections. Brownsword and Koops are interested mostly in substantive assessments of code's real-world effects. Brownsword's earlier contributions focused on public regulation, arguing quite abstractly that 'techno-regulation' must maintain scope for individual choice in order that moral decision-making, as a foundational element of moral community, is retained as a possibility. Koops, apart from providing an interpretation of other scholars' criteria, focuses on *ex post* assessments of specific technologies as they operate in the world. The criteria he identifies are mixed, although he prioritises the substantive (output) criteria of human rights and 'other moral values' before the rule of law and democracy.

Hildebrandt, Asscher, and Brownsword (in his later work) include a focus on input criteria, with the latter two scholars considering the application of Fuller to code. Hildebrandt's concept of 'legal protection by design' can be summarised as requiring transparency in the rules which technological normativity embodies, the ability of the end-user to exercise choice (i.e. to resist the default configuration), and lastly the possibility of contesting the rules in a court of law. Brownsword surveys Fuller from a rule of law perspective, but his ontological separation between code itself and the policy rule which animates its use means that his target of assessment is not the design of the code *per se*, but rather the decision of whether or not to use it. Although he does show some sensitivity to private regulation, this focus on the rules-behind-the-code perhaps belies Brownsword's general focus on public (state) regulation. At any rate, he requires transparency and due process in the authorisation of the use of code regulation, as well as the original requirement of the retention of the possibility of (moral) choice on the part of the individual. Unlike Brownsword, Asscher maintains a closer focus on code *per se*. His distillation of the Fullerian principles requires that code is transparent, that it works as expected and is not changed arbitrarily, that its producers can be identified,

and that end-users retain choice as to whether or not to obey its rules. I adopt various of these requirements in the framework of affordances I develop in the next chapter.

4. Conclusion

This chapter has considered two influential normative frameworks aimed at the creation of legitimate legal rules. For Fuller, achieving the latter is about respecting the ‘internal morality’ of law, which in turn minimises the potential for iniquity in the substance of the norms that can subsequently be promulgated from within that framework. For Wintgens, legitimate rules respect as far as possible individual autonomy, requiring justification of incursions on that autonomy only when in accordance with the legisprudential principles which, like Fuller’s theory, limit what the content of the resulting rules can possibly be. This analysis gives us a sense of what is expected of legislators when they are trying to make good laws, regardless of the political content of those laws.

The chapter then surveyed the literature on normative criteria for code. The conclusion here is that there are certain gaps in existing analyses, particularly with respect to the private production of code (as opposed to state-sanctioned use of code as a regulator), to the production of unforeseen normativity, and in the focus on ex post assessments of codes’ operation instead of the ex ante design decisions which lead to those effects. What this analysis provides is both a legal-theoretical view on legitimate normativity, and the state of the art in the literature of scholars whose work focuses on code-as-law.

As we have previously seen, with code there is an inevitability about the initial configuration – once the decision to build something has been made, that something by definition embodies some set of initial commitments which by necessity involve the privileging of one configuration of normativity over all the other possibilities. And as the discussion of input and output legitimacy showed, in the code realm we cannot

simply wait to see how a particular (legal) issue will be settled by the courts; the decision is by definition *ex ante*. To restate the central concern of the thesis, the configuration of the system by the designer is an inevitability, and the plasticity of software means that the range of ways in which she can regulate the end-user's behaviour is near-infinite. We must therefore interrogate critically that initial configuration in order to ensure its legitimacy. The next chapter uses the understanding gleaned here to develop a framework of design strategies that can help to ameliorate computational legalism.

Chapter 5

Affording computational legitimacy

To promote the benefits of legality, and to prevent the disadvantages of legalism, we will require new forms of interaction with these systems.¹

...regulatory schemes should seek to ensure that the design of consumer technologies is presenting the most optimal and sustainable choices, affordances, and constraints for users.²

1. Introduction

This chapter will now synthesise the ideas from the previous three chapters to suggest a framework of digisprudential affordances. It can be appreciated from the review of the literature in the previous chapter that although work in this area has considered criteria for the use of code as a regulator, this scholarship has not engaged in depth with the theories of design that I set out in Chapter 2, and has therefore not considered the desirability, efficacy, or indeed legitimacy of code from that perspective. This is the novel contribution of the present chapter. I synthesise the legal-theoretical perspective of legitimacy with design theory in order to suggest the affordances that code ought to exhibit in order to be deemed legitimate.

I begin by mapping the characteristics of computational legalism onto the Fullerian and legisprudential principles to which they are related. From this mapping I identify relationships between those principles and the affordances that can help to embody their aims within the design of code. The affordances can be viewed as simultaneously positive and normative; they provide a way of asking what a given

¹ Z. Bańkowski and B. Schafer, 'Double-Click Justice: Legalism in the Computer Age' (2007) 1 *Legisprudence* 31, p. 46.

² W. Hartzog, *Privacy's Blueprint: The Battle to Control the Design of New Technologies* (Cambridge, Mass: Harvard University Press, 2018) p. 57.

design affords, and a set of goals for what *should* be afforded in order to achieve legitimacy. Designers should not think only about what their code is intended to do from a commercial perspective; they ought also to make informed assessments of whether it is, and if not how it might become, legitimate. The digisprudential framework is a mechanism for guiding those anticipations from a legal-theoretical perspective. This is similar to Verbeek's discussion of anticipating a design's technological mediation from a moral perspective,³ which requires anticipation of the design's operation and, ultimately, *ex ante* restrictions on what forms it can possibly take. It also chimes with Hoepman's work on privacy design strategies and the idea of defining concrete design goals at the outset of a project.⁴

This chapter further bridges the gap between design and a legal-theoretical perspective of what those restrictions ought to be, importing insights gained from the latter to suggest how they might be concretely reified in the former. Each of the affordances is discussed from the perspective of the two case studies – blockchain applications and the Internet of Things (IoT) – to give them a grounding in concrete application.

1.1 Testing *decisions* or testing *design*?

We saw in Chapter 4 how Brownsword and others have focused on background decisions as the target for tests of legitimacy, and have (implicitly) viewed the code that embodies them as a separate product of those decisions. By contrast, I am more concerned with the legitimacy of the resulting design itself. As previously discussed, I

³ P.-P. Verbeek, 'Materializing Morality: Design Ethics and Technological Mediation' (2006) 31 *Science, Technology, & Human Values* 361, pp. 370, 372.

⁴ J.-H. Hoepman, 'Privacy Design Strategies', *ICT Systems Security and Privacy Protection* (Springer, 2014). The specifics of Hoepman's strategies are implemented by concrete 'design patterns' (*ibid.*, p. 3 *et seq.*); for reasons of space I do not discuss practical implementation at that level of granularity, although Chapter 6 does discuss some novel means of practically operationalising the digisprudential framework.

do not therefore maintain an ontological separation between the normativity of the code, which I frame as a *de facto* ‘rule’,⁵ and the preceding decision of the designer or enterprise to use code to achieve a particular aim. Or, alternatively, if it is thought that such a separation ought to be maintained, then it is the concrete instantiations in code of those background decisions that I am concerned with. We saw in Chapter 2 how design has a direct influence on end-user behaviour. We also saw in Chapter 3 how representationalism means that the text of the ‘rule’ (i.e. source code) and its operation are effectively one and the same:⁶ isomorphism between source code and the materiality of the artefact is to a great extent a given in the computational context. Rather than querying the motivations behind the design of a particular code (as important a concern as this may be), the task then is to query what the code *actually does*, and whether the normativity that it imposes is itself legitimate, separately from those background motivations. The distinction is nuanced but important, for if we focus on only the motivation behind a design but fail to look critically at how that motivation is in fact instrumentalised in code, we risk not only failing to observe what the artefact *actually* does, but also – and potentially worse – *sanctioning* it, in the erroneous belief that because the decision to use code was sound the implementation must also have been.

The aim therefore is to guide production of the ‘moreness’ of code (its instrumentality; Chapter 2) in ways that reduce its ‘lessness’ (its computational legalism; Chapter 3). The question is ultimately one of what the design affords the end-user (contestability, choice, transparency, and delay), legal institutions (evidential standards), and its own designer or manufacturer (oversight). The ultimate goal is fidelity to the input criteria of quasi-legality (Chapter 4) that ensure that its

⁵ On the topic of code-as-rules, recall the discussion in Chapter 4, section 3.3.2.3 (Does code contain rules *per se*?).

⁶ Chapter 3, section 3.2 (Representationalism).

technological normativity, whatever its substantive output functionality, includes mechanisms that ameliorate the path dependencies of computational legalism.

2. Mapping the criteria

In Table 1 below I map the Fullerian and legisprudential principles discussed in Chapter 4 onto the relevant characteristics of computational legalism, showing how they apply across the separate normative orders of institutional law and code. I then consider how the digisprudential affordances reflect the purposive goals of the principles. Legitimacy under the digisprudential framework requires that, at a minimum, code ought to provide the end-user with those affordances. It will be appreciated that there are overlaps between the characteristics and the suggested affordances – I am not suggesting that a given affordance applies only to ameliorate the specific characteristic indicated; the idea rather is to consider them in a holistic fashion, with the goal of achieving more legitimate technological normativity through a concurrent sensitivity to each of the issues raised. Wintgens represents a similar claim in the legisprudential context using the following model, where each X represents a proposed norm:⁷

⁷ L. Wintgens, *Legisprudence: Practical Reason in Legislation* (Surrey: Routledge, 2012) p. 282.

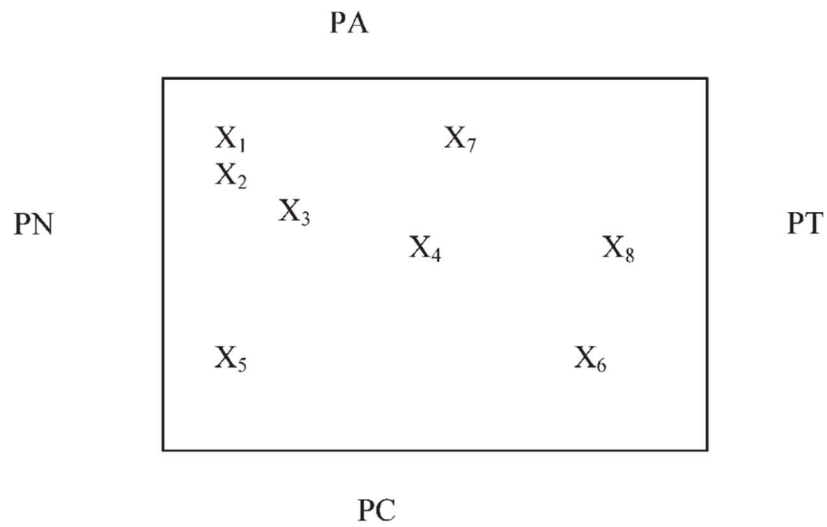


Figure 4. Differing norm justifications (reproduced from Wintgens 2012)

A given norm will be more or less justified according to each of the four jurisprudential principles; the point is that the particular circumstances surrounding the proposed norm will dictate the extent to which it must be justified by each of them. For example, emergency legislation following a natural disaster might be justified despite a lack of in-depth fact-finding or foresight, because the alternatives would not implement the powers and duties that are necessary to deal with the crisis quickly enough.⁸ This might mean for example that there is strong justification under the PA, but less justification under the PC, but that on balance the former is sufficient to justify the norm. Proposed norms that fall outside the box, i.e. that are not sufficiently justified by any of the principles, are not legitimate(d) at all.

In a similar fashion, the digisprudential affordances are not intended to be viewed in isolation, but rather to be seen as a set of elements that work in concert to achieve more legitimate configurations of technological normativity. Depending on the purpose and intended use of a particular code, the relevance of each principle will

⁸ *Ibid.*, p. 307.

vary, and therefore so too will the justificatory contribution of each affordance. A code that includes none of the affordances is on balance unlikely to be legitimate.

The intention is to contribute a set of ‘normative anchor points’⁹ that are explicitly oriented toward legal-theoretical concerns.¹⁰ The exercise is necessarily qualitative to an extent, and therefore it requires human judgement and the willingness to do the right thing on the part of the designer. Table 1 below maps the characteristics of computational legalism onto the Fullerian and legisprudential principles, and suggests the digisprudential affordances that can embody the purposive aims of those principles within code.

⁹ E.g. R. von Schomberg, ‘A Vision of Responsible Research and Innovation’ in R. Owen, J. Bessant and M. Heintz (eds.), *Responsible Innovation* (Chichester, UK: John Wiley & Sons, Ltd, 2013) p. 63 *et seq.*

¹⁰ Cf. other frameworks for computer ethics. See for example ‘ACM Code of Ethics and Professional Conduct (2018)’ <<https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct>>; B.C. Stahl et al., ‘From Computer Ethics to Responsible Research and Innovation in ICT’ (2014) 51 *Information & Management* 810.

Table 1. Mapping legitimacy across domains

<i>Characteristic of computational legalism</i>	<i>Fullerian principle</i>	<i>Legisprudential principle</i>	<i>Digisprudential affordances</i>
Ruleishness	Contradictory (v) or impossible (vi) rules	Alternativity (PA); Normative density (PN)	Transparency (provenance & purpose); Choice
Opacity	Promulgation (ii); Intelligibility (iv)	Alternativity (PA); Normative density (PN); Coherence (PC) LoC ₃	Transparency (provenance & purpose); Transparency (operation)
Immediacy	Contradictory (v) or impossible (vi) rules; Frequency of change (vii)	Normative density (PN); Temporality (PT)	Delay; Choice
Immutability	[in]Frequency of change (vii)	Temporality (PT); Coherence (PC) LoC ₃	Oversight; Choice
Pervasiveness	-	Normative density (PN)	-

2.1 Contestability as an overarching affordance

In addition to the ordering of code’s normativity suggested by the digisprudential framework, it is important also to bear in mind the importance of facilitating contestability as an overarching concern. Contestability is the keeping open of the possibility of questioning the code in, and taking its designer/manufacturer to, a court of law. As we saw in the previous chapter, contestability is an important criterion for the maintenance of the rule of law in the computational context; Hildebrandt in particular sets it apart as a fundamental requirement of the maintenance of legal protection in the computational paradigm.¹¹ From a strictly legal point of view, the

¹¹ Chapter 4, section 3.3.1 (Hildebrandt’s ‘Legal Protection by Design’).

ability to ‘return’ from the normative order of code to that of institutional law, and especially the courts, is an important part of retaining the role and the rule of law, even within the a-legal realm of code. Computational legalism militates against contestability: end-users must be in a position to understand the normativities they are being subject to in order to mount any kind of legal challenge to them. As the discussion below will consider, resistance and transparency are aspects of this that are concerned with the ability of the end-user to ‘see’ and question the norms to which she is subjected. This is the user-centric side of the contestability coin, but there is also an institution-centric side which ought to be considered at the point of production.

2.1.1 Affording scrutiny to legal institutions

In order for contestability to be fully embodied in code, legal institutions must be afforded proper evidential scrutiny; if the end-user contests the code in a court it is axiomatic to say that she must be able to bring evidence of her complaint, and that evidence must be intelligible to the trier of fact (cf. the opacity of computational legalism). Here we maintain the connection between the realm of code and the rule of law, ensuring that whatever happens in the code-based order the judicial process can still be the ultimate arbiter of any dispute. The affordance of contestability, then, is necessary not just to enable the end-user to understand the code’s normativities sufficiently well that she can choose to contest them, but is necessary also to enable legal institutions to grapple with the code from an evidential perspective. This implicates further questions of due process vis-à-vis evidential quality and propriety, and how these might interact with the design process. From an evidential perspective, certain standards must be met in order for an action to succeed; from a computational perspective this means that affordance of those standards must be considered *ex ante* during the design process if valid contestability – and therefore legitimacy – is to be achieved. Without advertence to the need for contestability at the stage of production,

it will not be possible (or will be that much more difficult) at the stage of operation. I discuss in the next chapter some approaches that can assist in implementing this institutional aspect of contestability at design time.

In sum, then, contestability operates as an overarching concern, suffusing digisprudence as an ultimate backstop; no matter the merits or demerits of the design from a digisprudential perspective, it must in the end always be possible for the end-user to resort to a court action to determine illegality (of whatever substantive form). This ensures the continuing role of the rule of law, notwithstanding code's existence as a separate a-legal normative order. In that regard, while the digisprudential affordances do not aim to render code legal *per se* – as the table above sets out, and as I argued in Chapter 1, they are concerned with the mirroring of legal notions of legitimacy within the a-legal order of code – they do nevertheless contribute towards the design of code that reflects the quintessentially legal characteristic of contestability, in particular via the affordance of transparency (discussed below).

3. From characteristics to affordances

We come now to an important contribution of the thesis, where the analysis of computational legalism and its negative effects meets the affordances that can ameliorate them. Following the design theory set out in Chapter 2, the idea here is to sensitise designers to the issues exposed by the legal-theoretical analysis of the previous two chapters, in order to shape or augment the affordances of the code they design to reflect the goal of legitimacy I have been considering. This is aspirational; the idea is to aim for legitimacy in privately-designed code, in the knowledge that on the one hand attaining absolute perfection is unlikely, and on the other the characteristics of computational legalism make the attempt all the more important.

3.1 Ruleishness

Code rules are, as we saw in Chapter 3, extreme in their precision; they by nature do not admit of flexible application. While they might exhibit exquisite non-discrimination, in that they apply to every end-user completely regardless of their characteristics, this is only a virtue if the rule is legitimately designed.¹² As I discussed in Chapter 3,¹³ code rules execute in *every* situation where their precise requirements are met, they execute in *no* situations where those requirements are *not* met (no matter how close the circumstances are), and the precise consequences specified within the rule are all that will or can flow from its execution. They are thus, by default, brittle in the extreme. This aspect of ruleishness – in the sense of strict and brittle application, rather than merely the laying down of rules, cf. Fuller’s principle of generality – is at the heart of computational legalism.¹⁴

This connects with the previous discussion of constitutive norms and the threshold between a design’s ‘constitution’, or the behavioural constraints which are ‘wired in’, and its merely ‘regulative’ aspects,¹⁵ which provide the end-user with the latitude to decide whether or not to acquiesce to a suggested limitation on freedom – in other words avoiding architectural determinism through the retention of ‘resistability’.¹⁶

3.1.1 *Affordance: choice*

We saw in Chapter 3 how the default configurations of code contribute to shaping an end-user’s understanding of the behavioural possibilities it affords her.¹⁷ End-users

¹² Bańkowski and Schafer, *supra* n. 1, p. 46.

¹³ Chapter 3, section 3.1 (Rules).

¹⁴ Chapter 3, section 3.1 (Ruleishness).

¹⁵ Chapter 3, section 3.2.1 (Constitutive and regulative rules).

¹⁶ M. Hildebrandt, *Smart Technologies and the End(s) of Law: Novel Entanglements of Law and Technology* (London: Edward Elgar Publishing, 2015) p. 218.

¹⁷ Chapter 3, section 3.3.1 (Default configurations).

tend to trust that the designer has made the ‘right choice’ for them, even where the code permits alternative choices to be made – the situation presented to the end-user is perceived to be normal, and even legitimate in systems that are pervasive. Once the artefact is operating, the outputs of its processes also tend to be trusted by end-users, due to automation bias.¹⁸

The effects of immediacy and immutability can be countered through the affordance of appropriate configurability. For the latter characteristic, the ability to change the configuration of the system is by definition in opposition to the state of immutability. But the mere provision of choice is not enough on its own – if it is to ameliorate ruleishness and empower the end-user, choice must be between *appropriate* options and must come at the appropriate time,¹⁹ otherwise we might come full circle to the code equivalent of long terms of use documents which notionally inform the end-user but which in practice leave her bewildered. Configurability *per se* can thus potentially become a counterproductive burden, particularly for naïve end-users²⁰ who may be ‘confused and intimidated by the number of choices’.²¹ At any rate, customisation is viewed by many end-users as time-consuming, and so they avoid it even where objectively it could benefit them by enabling them to choose options that reflect their interests and/or preferences.²² Thus, the ways in which code affords

¹⁸ D.K. Citron, ‘Technological Due Process’ (2008) 85 *Washington University Law Review* 1249, p. 1271f.

¹⁹ J.P. Kesan and R.C. Shah, ‘Setting Software Defaults: Perspectives from Law, Computer Science and Behavioral Economics’ (2006) 82 *Notre Dame Law Review* 583, p. 601.

²⁰ D.D. Clark et al., ‘Tussle in Cyberspace: Defining Tomorrow’s Internet’ (2005) 13 *IEEE/ACM Transactions on Networking (ToN)* 462, p. 467.

²¹ Kesan and Shah, *supra* n. 19, p. 627.

²² *Ibid.*, p. 598. For an early and influential study on this point, see W.E. Mackay, ‘Triggers and Barriers to Customizing Software’, *Proceedings of the SIGCHI conference on Human factors in computing systems Reaching through technology - CHI '91* (New Orleans, Louisiana, United States: ACM Press, 1991).

configurability must be considered in advance if the appropriate audience is to be appropriately empowered by them.

3.1.1.1 ‘Tussles’ and designing for choice

One theory from computer science that touches on the issue of choice and the role of design in responding to different parties’ interests is known as ‘tussle’.²³ ‘Tussle points’ are sites of conflict between those with conflicting interests that have some bearing on the design of code. Conflicts can be technical, legal, social, or economic, and it is for the designer to anticipate them and consider how their code ought to respond:

Our position is that the laws of men and the so-called whims of bureaucrats are part of the fabric of society, like it or not. They are some of the building blocks of tussle, and must be accepted as such. We, as technical designers, should not try to deny the reality of the tussle, but instead recognize our power to shape it.²⁴

Unusually for a computer science paper, the authors’ analysis demonstrates a sensitivity to STS concepts in its discussion of how technologists might identify the conflicting actors and interests that contribute to tussle.

One can appreciate how the commercial benefits of computational legalism are in conflict with the goals of legitimacy, thus creating a ‘tussle space’: enterprise makes use of the former to advance their commercial interests – ruleishness and immutability provide predictability; opacity provides protection of commercial secrets and hides profitable but dubious design/normativity; and immediacy gives feedback and marketable results. We saw in Chapters 2 to 4 how those characteristics are, at least potentially, antagonistic to end-users’ interests. There is thus a tussle between the interests of the end-user and the enterprise that wishes to channel her behaviour in

²³ Clark et al., *supra* n. 20.

²⁴ *Ibid.*, p. 473.

predictable (and profitable) ways. The tussle space arises because digisprudential legitimacy looks to uphold basic ‘constitutional’ safeguards against illegitimate behavioural regulation, but these are by nature absent in the default condition of computational legalism.

Anticipating points of tussle during the production phase is important to avoid problems during operation. One of the principles that Clark et al. suggest for dealing with tussle spaces is designing to permit choice.

3.1.1.1.1 Designing for choice

The premise here is that the design ought to anticipate, or least allow for, different possibilities: ‘[r]igid designs will be broken; designs that permit variation will flex under pressure and survive.’²⁵ While Clark et al. are thinking from the perspective of infrastructural design, from a digisprudential perspective we can think of designs which afford spaces in which users can exercise their autonomy, where what the design affords them is not limited so as to impose heteronomy (all things being equal – the notional scope for exercising autonomy depends of course on the fundamental purpose of the design). This accords with Brownsword’s requirement, discussed in the previous chapter, that code permit choices to be made by individuals in order to preserve their capacity for moral action.²⁶

Designing for choice also accords with the legisprudential principle of alternativity (PA),²⁷ which as we saw concerns the legitimate use of ‘rigid’ rules that admit of no latitude, versus alternatives that incentivise or suggest courses of action, thereby implying the possibility of choice. Leaving the tussle space open to allow for

²⁵ *Ibid.*, p. 466.

²⁶ Chapter 4, section 3.2.1. Koops, Hildebrandt, and Asscher also argue that choice is necessary (*ibid.*, sections 3.2.3, 3.3.1, and 3.3.2.2, respectively).

²⁷ Chapter 4, section 2.3.4 (The principle of alternativity (PA)).

interpretation and end-user choice shifts power away from the designer and on to her.²⁸ The PA in this context requires that (i) the use of a rule in the code be more desirable than not, and (ii) the use of a *rule* rather than some less ruleish mechanism be necessary, for example a nudged default, or the option for the end-user to choose. The first speaks to the enterprise's business model and how this is articulated in code, raising some potentially existential questions as to the fundamental desirability of a given approach or product. In the context of privacy regulation, Hartzog warns us about the reluctance to consider design: 'too often industry wants the freedom to experiment on the public without accepting the responsibility for the harm they cause.'²⁹ We might then think of a kind of Hippocratic oath for code – 'first, do no harm' – which might ultimately lead a responsible designer/enterprise to conclude that the feature or product should not be developed at all.³⁰ Thus, shifted into the present context, a computational PA would assess first whether a given (dis)affordance/inscription is necessary for the operation of the artefact and the business model being pursued. If it is not, then *a priori* it should not be included in the design because it represents an unnecessary and unjustified limitation of the end-user's freedom.

If the element of technological normativity *is* necessary, the question then becomes one of the ruleishness of the implementation – how 'wired in' does the functionality need to be to achieve the designer's goal? Should the code require the end-user to exercise a choice, or perhaps provide a passive configurable option? Or does the purpose of the code imply the need for nudging/inscription, or even the requirement (wiring-in) of one of the possible options to the exclusion of all others?

²⁸ Clark et al., *supra* n. 20, p. 473.

²⁹ Hartzog, *supra* n. 2, p. 85.

³⁰ A.D. Selbst et al., 'Fairness and Abstraction in Sociotechnical Systems' (Rochester, NY: Social Science Research Network, 2018) SSRN Scholarly Paper ID 3265913 pp. 13–14 <<https://papers.ssrn.com/abstract=3265913>>.

The latter is the most ‘ruleish’ form of technological normativity, while the former approaches are progressively less constraining.³¹ As with the legisprudential PA, the decision to choose a more ‘ruleish’ (and therefore less choice-oriented) design approach must be justified because of the correspondingly larger limitation it places on freedom. I consider this further in the discussion of default choices in section 3.1.1.2 below.

3.1.1.1.2 Taking choice further – agonism in design

The concept of tussle and the anticipation of conflicting interests is connected with Hildebrandt’s discussion of agonism in constitutional democracy.³² Agonism is a political theory that views adversarial debate to be fruitful where it enables contrasting points of view to be ventilated and compromise thereby to be achieved. ‘Inviting dissent’ in this way, which can be consciously facilitated by design,³³ is ultimately at ‘the core of both democracy and the rule of law.’³⁴ For Hildebrandt, this is reflected in participatory design processes like Constructive Technology Assessment (CTA)³⁵ which aim to achieve a ‘settlement’ during the design process that takes into account the views of those with a stake in the outcome.³⁶ I have previously suggested that such processes are less likely to be used in some design contexts, particularly small and medium enterprises.³⁷ Furthermore, as Pols and Spahn have noted, design *for* all need not be the same as design *with* all.³⁸ Initiatives like CTA seek to legitimise a design by

³¹ Chapter 2, section 3.5 (The spectrum of technological normativity).

³² M. Hildebrandt, ‘Algorithmic Regulation and the Rule of Law’ (2018) 376 *Philosophical Transactions of the Royal Society A* 20170355, pp. 7–8.

³³ See generally C. DiSalvo, *Adversarial Design* (Cambridge, Mass: MIT Press, 2012).

³⁴ Hildebrandt, ‘Algorithmic Regulation and the Rule of Law’, *supra* n. 32, p. 7.

³⁵ J.W. Schot, ‘Constructive Technology Assessment and Technology Dynamics: The Case of Clean Technologies’ (1992) 17 *Science, Technology, & Human Values* 36.

³⁶ Recall the discussion in the previous chapter contrasting input and output assessments of code.

³⁷ Chapter 1, section 1.4 (Legitimacy vs. ‘compliance by design’).

³⁸ A. Pols and A. Spahn, ‘Designing for the Values of Democracy and Justice’ in J. Van Den Hoven, P.E. Vermaas and I. van de Poel (eds.), *Handbook of Ethics, Values, and Technological Design: Sources, Theory, Values and Application Domains* (Dordrecht: Springer, 2015) p. 351.

the fact of having involved stakeholders in decisions as to its substantive characteristics. This is a separate concern from my own, since such processes are built around the value of participation, rather than an underlying theory of what provides legitimacy.³⁹ Recalling the discussion of input and output legitimacy in the previous chapter, these approaches follow a ‘thick’ conception of legitimacy based on particulars rather than a ‘thinner’ formal conception that is separate from the substantive content of participants’ views on the merits of a particular design. The ‘constitutional’ standards of digisprudence are sited at an earlier point than the question of whether stakeholders had their views taken into account. The two do not conflict and a design process can certainly involve both, but if the design is not otherwise digisprudentially-legitimate then the fact that a participatory process was used to shape it will not by itself provide legitimacy of the kind I am proposing.

Nevertheless, the preservation of agonistic space can stand as a constitutional principle for code design, along the lines of tussle previously discussed: by anticipating in advance the points at which tussle is likely to arise in the operation of the design, it is possible to avoid imposing one path or outcome for that tussle ahead of time, thereby preserving the space for both choice and agonism. This extended view of designing for choice enjoins the designer consciously to ‘retreat’ from the impulse to impose a pre-determined outcome, thus preserving a space for agonism, for tussle, within the operating space of the design itself. The domain of the morality of duty (computational legalism; external limitation on freedom) is reduced, and the domain of aspiration (‘legality’; individual conceptions of freedom) accordingly expanded. This twist on agonism operates at runtime but is facilitated by decisions made at design time, hence the continued relevance of input legitimacy. Unlike Hildebrandt’s discussion, the agonism in this case is anticipated in the *operation* of the artefact rather than in the design process; of course, both may be present (i.e. participation in the design process,

³⁹ *Ibid.*, p. 353.

and the design affording space for agonism during operation). The extent to which implementation of this extended affordance of choice will be possible or plausible depends on the intended use of the artefact. The design of a single-function IoT device, for example, is less likely to admit of space for agonism than is the design of a social network platform. This is an example of how different artefacts reflect the digisprudential affordances to differing degrees, as in Wintgens' model in Figure 4 above.

Returning to the analysis by Clark et al., they suggest another principle relevant to dealing with tussles in the production process, namely modularisation. Modularisation is also relevant to the affordance of choice, as we shall see.

3.1.1.1.3 Modularisation

Breaking down a system into (re-usable) modules of functionality is standard in the modern practice of code production.⁴⁰ Modularity requires that 'functions that are within a tussle space should be logically separated from functions outside of that space.'⁴¹ This idea connects with the legisprudential principle of normative density (PN).⁴² In terms of the goals of that principle, the code should avoid bundling together norms that are not conceptually related, forcing the end-user to acquiesce to multiple heterogenous normativities simultaneously when she might wish to accept some but resist others. This idea is reflected in the GDPR's provisions requiring consent to be a genuine and free choice, requiring separate consents for separate operations, and preventing the bundling of consent along with performance where the former is not necessary for the latter.⁴³

⁴⁰ S. Gürses and J. van Hoboken, 'Privacy after the Agile Turn' in E. Selinger, J. Polonetsky and O. Tene (eds.), *The Cambridge Handbook of Consumer Privacy* (1st edn., Cambridge University Press, 2018) p. 582. I discuss this further in Chapter 6.

⁴¹ Clark et al., *supra* n. 20, p. 466.

⁴² Chapter 4, section 2.3.6 (The principle of normative density (PN)).

⁴³ GDPR, Arts. 7(2) and 7(4); Recitals 32, 42, and 43.

By modularising code normativity in this way, other digisprudential affordances are also made more readily achievable, for example transparency of the code's operation. The aggregated normativities of a system, when they exhibit the characteristics of computational legalism, can lead to exponential negative effects as the characteristics each amplify the impact of the others. Modularisation as a design approach should seek to isolate discrete elements of normativity, perhaps along the boundaries of specific features or functions, thereby enhancing the end-user's ability to model accurately the system's effects by avoiding the conflation of what should be conceptually isolated issues. As Clark et al. recommend, designers should 'modularise the design along tussle boundaries, so that one tussle does not spill over and distort unrelated issues'.⁴⁴ For example, in a smart thermostat, an example of an Internet of Things device, functionality that involves profiling the end-user implicates a tussle involving different interests (i.e. the end-user's right to data protection versus the enterprise's desire to extract commercially-valuable insights from her habits) from functionality that merely gets on with controlling her heating system (i.e. the end-user's desire for domestic warmth versus the excessive consumption of energy).⁴⁵ Modularising these discrete functionalities enables the end-user to understand them separately and to respond to them in different ways. The connection with choice and the exercise of autonomy can be readily appreciated.

For now, the next section picks up the discussions of technological normativity in Chapters 2 and 3,⁴⁶ considering the connection between the design of digital systems' default configurations and the affordance of choice.

⁴⁴ Clark et al., *supra* n. 20, p. 466.

⁴⁵ For a critical discussion of the problems with IoT thermostats, see Brian Krebs, 'IoT Reality: Smart Devices, Dumb Defaults' (*Krebs on Security*, 16 February 2008) <<https://krebsonsecurity.com/2016/02/iot-reality-smart-devices-dumb-defaults/>>.

⁴⁶ Chapter 2, section 3.5 (A spectrum of technological normativity) and Chapter 3, section 3.3.1 (Default configurations), respectively.

3.1.1.2 Default choices

As Sunstein and Thaler note about the ‘offline’ world, default configuration is all around us – *some* initial configuration of rules is an inevitability,⁴⁷ which in turn implies the inherent non-neutrality of technologies.⁴⁸ We have seen at various points throughout this thesis how this observation applies even more strongly in the computational context. It is not possible for a designer to leave open to interpretation what the design of the artefact should be in the way that it is possible for the legislator deliberately to leave the meaning of a textual norm somewhat open. *Some* choice must be made by the designer that constrains the infinite possibilities of the *tabula rasa*, and therefore intervention to shape those initial decisions is all the more necessary.

Kesan and Sandvig note that to rely on default settings (i.e. settings that can be changed) is in many cases to outsource decision-making to designers, moving it away from both government and end-users. Consequently, it becomes necessary to ‘push and prod developers to set default settings that comport with established societal concerns’.⁴⁹ If we accept that one of those concerns ought to be the legitimacy of code, then the aspects of the code that are made ‘chooseable’ must accord with that spirit. The number of choices and their quality (i.e. what substantive functionality they enable the end-user to alter) is thus an important design question with respect to the scope of autonomy that the code provides the end-user, and so too is the way in which these affordances of choice are communicated (signified) to her through the design. The provision of choice for choice’s sake does not beget legitimacy if those choices do not provide real autonomy.

⁴⁷ C.R. Sunstein and R.H. Thaler, ‘Libertarian Paternalism Is Not an Oxymoron’ (2003) 70 *University of Chicago Law Review* 1159, p. 1174 *et seq.*

⁴⁸ M. Hildebrandt, ‘Legal Protection by Design: Objections and Refutations’ (2011) 5 *Legisprudence* 223, p. 238f.

⁴⁹ R.C. Shah and C. Sandvig, ‘Software Defaults as De Facto Regulation: The Case of the Wireless Internet’ (2008) 11 *Information, Communication & Society* 25, p. 42.

Kesan's analysis (with Shah) of the power of default settings in code is particularly relevant in this context.⁵⁰ They identify a spectrum of design mutability, from 'wired-in' functionality that cannot be changed, through to default settings that can be changed, and on to the notional 'free choice' of full customisation⁵¹ (it must be noted that even this level of configurability can never be completely free, because as I noted above the initial commitments of the design by definition circumscribe possibilities, which in turn limits the area within which the end-user can exercise autonomy). The extent to which end-users are aware of the control they have over configuration is a core concern,⁵² and is entirely contingent upon the affordance of choice being perceived – it is not enough if the affordance is merely real⁵³ but unknown (or so complex as in practice to disafford⁵⁴). In their survey of real-world approaches to setting defaults, the authors identify two core principles that drive design decisions, namely the consideration of novice end-users and efficiency.⁵⁵ Kesan and Shah note the vagueness of the two principles, particularly with regard to who might be considered a 'novice', and by whose standards 'efficiency' is to be judged, particularly since the effect of a default will often impact on 'fuzzy' values that are difficult to calculate in such terms.⁵⁶ The latter demonstrates a value often held by commercial designers; we will see below in section 3.3 how the goal of increasing efficiency –

⁵⁰ Kesan and Shah, *supra* n. 19.

⁵¹ *Ibid.*, p. 591 *et seq.* See also L.F. Cranor and R.N. Wright, 'Influencing Software Usage' [1998] *arXiv:cs/9809018* pp. 6–7 <<http://arxiv.org/abs/cs/9809018>>.

⁵² Kesan and Shah, *supra* n. 19, p. 597.

⁵³ These distinctions were discussed in Chapter 2, section 2.1 (Real and perceived affordance).

⁵⁴ This is a common criticism levied at open source projects, whose configurability gives the end-user great notional power but whose ease-of-use disaffords that configurability to those without the necessary expertise. See K. Noyes, 'Is Linux Really Harder to Use?' (*PCWorld*, 2 August 2010) <https://www.pcworld.com/article/202364/is_linux_really_harder_to_use.html>.

⁵⁵ Kesan and Shah, *supra* n. 19, p. 600.

⁵⁶ *Ibid.*

pervasive among technologists – is not necessarily desirable in every case, even where there is technical scope for it.

In terms of guiding design, the authors draw on legal notions of default rules to consider the threshold between what configuration is ‘wired in’ (immutable) and what is set merely as a default.⁵⁷ Adopting analysis by Radin,⁵⁸ they suggest that where the code is ‘wired in’ it must both notify the user and permit a judicial remedy (i.e. it must be contestable). This accords with this discussion throughout this chapter on the affordances of contestability, transparency, and delay.

The system ought to provide ‘an easy-to-use interface that allows users to configure the software according to their preferences’⁵⁹ – Kesan and Shah do not refer to the theory, but this is of course about what affordances the interface provides, which should follow common design and usability conventions that end-users are familiar with.⁶⁰ Added to this requirement, the authors set out a framework of principles for the setting, by designers, of the initial defaults in code. The starting point is what they call the ‘would have wanted standard’. This requires anticipation of what the parties (the enterprise and end-user) would likely have negotiated, had that been a possibility.⁶¹ This principle applies in situations where the setting does not materially affect a fundamental societal concern, such as (they suggest) privacy or online security. To those I might add normative legitimacy and contestability, as herein described.

The next requirement is that where there is an information imbalance between designer and end-user, the default must be set to protect the latter, with appropriate information or guidance provided to her should she wish to change it. This of course

⁵⁷ *Ibid.*, p. 614 *et seq.*

⁵⁸ M. Radin, ‘Machine Rule: The Latest Challenge to Law’ (2005) (unpublished manuscript, on file).

⁵⁹ J. Kesan and R. Shah, ‘Setting Software Defaults: Perspectives from Law, Computer Science and Behavioral Economics’ [2006] *University of Illinois Law and Economics Working Papers* p. 615f.

⁶⁰ Norman calls these ‘conventions’. See D.A. Norman, ‘Affordance, Conventions, and Design’ (1999) 6 *Interactions* 38, p. 40 *et seq.*

⁶¹ Kesan and Shah, *supra* n. 19, p. 618.

is inconvenient if the functionality in question is the very purpose of the device. Kesan and Shah give the example of cookie settings in a web browser, where because of the informational imbalance (end-users do not readily understand what cookies are and what they are used for⁶²) the default setting should be to reject them.⁶³ Should web companies wish them to be turned on (presumably to facilitate the business model of online behavioural advertising), they then must explain to the end-user their purpose and how to enable them⁶⁴ (this is redolent of the concept of affording transparency of purpose, discussed below). The idea with such ‘penalty defaults’ is that before the end-user can choose the non-default setting, the burden is on the designer to explain its effect. The default is therefore what the party with the greater understanding of the code (i.e. the designer or enterprise) would *not* have wanted, as a delaying mechanism that allows for the end-user to be informed (this is connected with the discussion of friction in section 3.3.1.2 below).

The next principle suggested by Kesan and Shah also justifies this ‘would *not* have wanted’ approach and is based on the economic concept of ‘externalities’, or the broader (potentially negative) effects of the system on third parties. The default setting should reduce externalities or, if the stakes are particularly high, there should be no choice and the beneficial option should be ‘wired in’.⁶⁵ An example might be an IoT webcam that is configured by default to stream whatever it captures, with either no authentication mechanism enabled by default or with a standard and obvious default password (such as ‘admin’). The negative effects of such designs, especially given the

⁶² L. Edwards, ‘Data Protection and E-Privacy: From Spam and Cookies to Big Data, Machine Learning and Profiling’ in L. Edwards (ed.), *Law, Policy and the Internet* (Oxford, UK; Portland, OR: Hart, 2018) p. 126 *et seq.*

⁶³ This is at odds with current practice. See the discussion *ibid.*, and n. 128 of Chapter 3.

⁶⁴ Kesan and Shah, *supra* n. 19, p. 621.

⁶⁵ *Ibid.*, pp. 621–622.

potential pervasiveness of such technology,⁶⁶ can be significant.⁶⁷ The idea, then, is that while the end-user might wish the camera to work immediately upon connection (and the enterprise might wish this too, from the perspective of given the end-user what she wants), the ‘would not have wanted’ standard might require that streaming is not enabled by default and that a (strong) password must be set before the device will connect. Anecdotally, this has been the direction of travel in the design of domestic routers, where instead of merely suggesting that end-users change the administration password or WiFi code, the device comes pre-configured with strong, unique options already set.

In terms of design, the cognitive biases mentioned in Chapter 3⁶⁸ can strengthen the ‘stickiness’⁶⁹ of a default setting, militating against the end-user exercising choice; this implies an even greater responsibility to design the initial configuration wisely.⁷⁰ Furthermore, the prominence of the setting in an interface can affect end-users’ awareness of it, and indeed the designer can explicitly draw attention to defaults that require special attention but do not cross the threshold to merit being ‘wired in’ (for example the ‘would not have wanted’ defaults just discussed – making the administration password for a domestic router ‘wired in’ would be an odd design choice). Attention can be called by, for example, alerts which require the end-user to confirm a choice, or to make it *ab initio* at the first point of use, with no pre-selected option and no opportunity to bypass the configuration request. Such measures can build in an element of the delay and friction described below in section 3.3.

⁶⁶ See Chapter 3, section 3.3.2 (Pervasiveness).

⁶⁷ For chilling real-world examples of precisely this, see J.M. Porup, ‘How to Search the Internet of Things for Photos of Sleeping Babies’ (*Ars Technica UK*, 19 January 2016) <<http://arstechnica.co.uk/security/2016/01/how-to-search-the-internet-of-things-for-photos-of-sleeping-babies/>>.

⁶⁸ Chapter 3, section 3.3.1 (Default configurations).

⁶⁹ Sunstein and Thaler, *supra* n. 47, p. 1175.

⁷⁰ Kesan and Shah, *supra* n. 59, p. 633.

Importantly, the design of these affordances of choice must take into account what Sunstein and Thaler call ‘framing effects’,⁷¹ or the way in which wording affects comprehension of one or other option (the classic analogue in the legal world being the leading question). The design should not promote the enterprise’s aim at the expense of digisprudential legitimacy, and of course the use of the adversarial design approaches that we saw in Chapter 2⁷² is *de facto* illegitimate.

This analysis of choice *qua* configurability gives added texture to the bare requirement suggested by Brownsword and others that the technological normativity of code preserve the possibility of choice, and also the idea that more choice is *de facto* better. The affordances of the artefact ought to reflect the spirit of digisprudential legitimacy at each point of the end-user’s ‘journey’ through the inscriptions designed into the code. More choice *per se* is not sufficient to legitimate code if it is not the right *kind* of choice; the design must afford meaningful spaces for the exercise of autonomy and not simply more options. As mentioned above, this might raise difficult existential questions as to the desirability of a given artefact or business model, but such questions are precisely what digisprudential legitimacy is intended to provoke.

3.1.2 Discussion: blockchain applications

Blockchain applications pose potentially significant problems from the perspective of affording choice. We saw in Chapter 1 how one of their primary attractions is that the code of the application, once written and added to the chain, is essentially immutable, owing both to how it is stored and the decentralisation of the network. If the possibility of choice has not been anticipated in advance, the central benefit to blockchains of *tamper*-resistance becomes a hindrance to its exercise at runtime. To afford choice, then, it is particularly necessary in the case of blockchain applications to choose

⁷¹ Sunstein and Thaler, *supra* n. 47, pp. 1179–1180; 1182–1183 (discussing default employee enrolment into a pension scheme).

⁷² Chapter 2, section 3.1 (Disaffordance).

carefully in advance how much of the code is ruleish and how much relies on external contingencies, including end-user input. This is intimately connected to, and overlaps with, the issue of immutability, discussed in section 3.4 below: the fixity of the code that flows from its storage on, and execution by, a blockchain (recall the discussion of Sony BMG's DRM and the immutability of the CD medium) means that the design of the threshold between wired-in and configurable code is important, particularly given the additional complication of blockchain applications' execution of logic that is (i) automatic, and (ii) potentially legally-relevant.⁷³

If blockchain applications are to be used to implement legally-relevant operations (cf. the discussion in Chapter 1) such as transfers of assets, one approach to ameliorating ruleishness is to reconceive of them as mere 'custodians' of the mutability and flexibility of language, and thus to change expectations of what can and should be implemented computationally. An example is the Ricardian Contract, an approach that aims not to automate the purposive elements of a contract-like agreement (which, as we saw in Chapter 1, is the aim of some smart contract maximalists) but rather to maintain the flexibility of textual agreement and to augment it with a limited amount of coded functionality that complements, rather than replaces, the latter.⁷⁴ The (natural language) text of the agreement is 'wrapped' in a minimal code semantics that enables basic code-based features such as tamper-resistance and provenance checking, through the use of hashing and public-key cryptography. Because the actual text of the agreement retains all the possibilities of nuance that contract law and indeed language in general have evolved to accommodate, the apparent immutability of the agreement

⁷³ K.E.C. Levy, 'Book-Smart, Not Street-Smart: Blockchain-Based Smart Contracts and The Social Workings of Law' (2017) 3 *Engaging Science, Technology, and Society* 1, p. 3.

⁷⁴ I. Grigg, 'The Ricardian Contract', *Proceedings of the First IEEE International Workshop on Electronic Contracting* (IEEE, 2004). This accords with Wagner's argument in favour of 'more law and less software'. See R.P. Wagner, 'On Software Regulation' (2005) 78 *Southern California Law Review* 457 *passim*.

can nevertheless be combined with the inherent flexibility of expression. In other words, the execution of the agreement remains ‘human’, any choice possibilities are reified in the immutable text, and the contribution of code is in providing the benefits of immutability and provenance checking that are ancillary to the substance of the agreement. The exercise of choice, then, takes place outside the code. The extent to which this is practically useful from the perspective of the technology remains to be seen; limiting the ruleishness of blockchain code to such ancillary benefits in this way might in practice undermine their perceived value in the first place.

Alternative approaches that maintain the full(er) utility of code will need to involve designing the application’s inscriptions to afford the end-user choice at key moments. Those inscriptions ought to be designed with sensitivity to the implications of blockchains as a platform (i.e. its automated instrumentality and immutability). The greater the normative impact of the code’s logic (its normative density; the PN), the greater the need for choice to be preserved; in practice the implementation of this will vary between applications and will involve a mixture of the appropriate selection of oracles, notification to the end-user, appropriately-defined choices, and logic that can deal with the outcomes. Given the peculiar characteristics of blockchains, whether it can reasonably be expected of designers to anticipate all the relevant points where choice is necessary is at the very least questionable. These requirements may fundamentally undermine the premise of blockchain applications, such as distributed autonomous organisations, that are predicated on their ability to execute operations without human input, some of which are of quintessential legal relevance, such as purchasing goods or services. This again raises the existential question of whether such applications can be legitimate – if it is accepted that the affordance of choice is necessary to achieve digisprudential legitimacy, it may well be that a failure to provide it renders such applications *a priori* illegitimate.

3.1.3 Discussion: IoT

As we saw in Chapter 1,⁷⁵ one of the common characteristics of IoT devices is their minimal interfaces. One way of affording choice is to provide better, more sophisticated interfaces, perhaps through the connection of the IoT artefact itself with another device that affords more complex interactions. This could be a smartphone or a television, whose affordances in turn facilitate the presentation and signifying of choice affordances to the end-user. This is a difficult balance to strike, because of course many IoT devices are intended to have a minimal number of functions. In the next section I discuss the Amazon Dash Button, which consists of a single button but whose background functionality is extremely complex; in that case the affordance of choice at the point of using the device is dramatically curtailed – its only real use is the pressing of a button, but the number of configurable variables that are relevant to the process that is put in train by doing so is significant, as we shall see.

3.2 Opacity

In the computational context opacity is connected most closely with the Fullerian principles of promulgation and intelligibility, and the legisprudential principles of alternativity (PA)⁷⁶ and normative density (PN).⁷⁷ In terms of promulgation, Fuller was concerned that citizens should know (or be in a position to find out) the rules by which they were being governed, partly as a check against them being disregarded by the authorities administering them⁷⁸ (this is related also to his eighth principle, requiring congruence between the declared rule and the official action that flows from it⁷⁹). This of course enables citizens to observe their operation, which is a prerequisite

⁷⁵ Chapter 1, section 4.3 (The Internet of Things).

⁷⁶ Chapter 4, section 2.3.4 (The principle of alternativity (PA)).

⁷⁷ *Ibid.*, section 2.3.6 (The principle of normative density (PN)).

⁷⁸ *Ibid.*, section 2.2 (Fuller's principles of legality).

⁷⁹ *Ibid.*

for contesting it. In order to be valid, the rules should also be intelligible – obscurity and incoherence can make legality difficult or impossible to attain.

For the legisprudential principles, opacity is targeted again by the PA and the PN. Under the former, the inherent opacity of code again imbues the decision to implement a rule with extra normative impact versus a less ruleish measure. The inability of the end-user to see the rule to which she is subject is emphasised in the computational context, and its use is therefore subject to a higher standard of justification. As before, the threshold of justification is lowered when a less ‘ruleish’ design measure is employed, but at all times the fact of code’s opacity must be borne in mind. This in turn relates to the application of the PN: the more opaque the code, the more difficult it may be for the end-user to appreciate the aggregate ‘density’ of technological normativity that she is being subjected to. The PN expects there to be a proportionate connection between the policy aim and the means by which it is achieved, with threats of sanctions at the ‘denser’ end of the scale and mere suggestions that can be easily ignored towards the ‘lighter’ end.⁸⁰ In terms of justification, the use of a particular design technique must be justified in the context and in light of the other principles, particularly if there are alternative mechanisms that might have achieved the same end. In the computational context, we saw how the geography of code is often taken by the end-user to be a ‘natural fact’, rather than merely one possibility among infinite others.⁸¹ This opacity of normative impact is particularly strong where the configuration of (dis)affordances and inscriptions strongly guides the end-user’s behaviour; the necessity for justification is therefore all the stronger in such situations.

⁸⁰ Chapter 4, section 2.3.6 (The principle of normative density (PN)).

⁸¹ *Supra*, section 3.1.1 (Affordance: choice); Chapter 3, section 3.3.1 (Default configurations).

3.2.1 *Affordance: transparency of provenance & purpose*

An important aspect of affording transparency, which is also connected to the affordance of contestability, is that of provenance. This links to Asscher's adaptation of Fuller's principles, where he states that the source of the code-based norm must be identifiable.⁸² This can be problematic when complex digital systems are made up of multiple components, often from different suppliers.⁸³ Designers ought to afford reasonable notice of the sources of the code in their systems (or at least provide a means to find out⁸⁴) so as to inform and to afford contestability. As Gürses and van Hoboken suggest, while many websites appear to end-users to be published by a single commercial entity, they are in fact 'a mix of a Frankenstein and a Matryoshka doll concealing dozens of services.'⁸⁵ Recent scrutiny of online behavioural advertising has shown just how large the network of unseen third parties operating in the background can be, including situations where the design of a website's interface might suggest to the end-user that there is only one provider involved.⁸⁶ The same is often true of other digital artefacts whose back-end processing relies on a host of services (and third parties) the end-user is unlikely to have cognisance of.

The question of the code's *purpose* relates to the LoC₃ element of the legisprudential principle of coherence (PC).⁸⁷ Transparency under this rubric will require information as to the reason for a given piece of functionality, where this is not self-evidently the artefact's *raison d'être*. For example, a smart fridge might be marketed

⁸² Chapter 4, section 3.3.2.2 (Asscher).

⁸³ Gürses and van Hoboken, *supra* n. 40; D. Oberle et al., 'Engineering Compliant Software: Advising Developers by Automating Legal Reasoning' (2012) 9 *SCRIPTed* 280.

⁸⁴ I discuss the contrasting ideas of the 'monitoring citizen' vs. the 'well-informed citizen' *infra*.

⁸⁵ Gürses and van Hoboken, *supra* n. 40, p. 584.

⁸⁶ See for example Z. Yu et al., 'Tracking the Trackers', *Proceedings of the 25th International Conference on World Wide Web - WWW '16* (Montreal, Quebec, Canada: ACM Press, 2016). The authors of the study found that 95% of websites accessed by its German participants contained potential third-party trackers.

⁸⁷ Chapter 4, section 2.3.3 (The principle of coherence (PC)).

as including features that provide information on the goods that are stored within it (nutritional information, expiry dates, recipes, etcetera), but this does not imply that data about the goods stored in the fridge will be used to profile the end-user vis-à-vis her diet, income, and preferences, information which can be used for purposes far outside the basic functionality of a fridge.⁸⁸ The normativity of such functionality ought to be justified via the affordance of transparency, and where this cannot be done the function should not be included in the design. Another example is the inclusion of a geolocator in a smartphone torch application – transparency ought to be afforded on a similar basis, because it is not a part of the functionality commonly required of a torch.⁸⁹ The designer must then consider from where this unorthodox function of geolocation is justified, given what affordances are commonly expected of torches. This kind of transparency might be termed *transparency of purpose*.⁹⁰

With each form of transparency, however, the designer must not succumb to the transparency fallacy, where essentially any otherwise legal normativity can be included in the code provided it is described in a terms document that gives the end-user notice and the notional opportunity to exercise her judgment.⁹¹ Pols and Spahn discuss the concept of the ‘monitoring citizen’ as a better normative ideal than the ‘well-informed citizen’ that such documents envisage.⁹² In an ideal world a fully-

⁸⁸ This is of course what the purpose limitation principle in data protection law is aimed at. See, for example, Art. 5(1)(b) of the GDPR. This is also related to the discussion of delay in section 3.3.3, *infra*.

⁸⁹ Hartzog, *supra* n. 2, p. 24. See also ‘Android Flashlight App Developer Settles FTC Charges It Deceived Consumers’ (*Federal Trade Commission*, 5 December 2013) <<https://www.ftc.gov/news-events/press-releases/2013/12/android-flashlight-app-developer-settles-ftc-charges-it-deceived>>.

⁹⁰ Here we begin to touch the edges of the fields of computer ethics and responsible research and innovation (RRI), which lie outside of the scope of this thesis. Further research could fruitfully explore this connection. See, for example, S. Vallor, *Technology and the Virtues: A Philosophical Guide to a Future Worth Wanting* (Oxford, New York: Oxford University Press, 2016); von Schomberg, *supra* n. 9; L. Floridi, *The Ethics of Information* (Oxford: Oxford University Press, 2013).

⁹¹ Hartzog, *supra* n. 2, p. 68 *et seq.*

⁹² Pols and Spahn, *supra* n. 38, p. 348.

informed end-user might be a possible and desirable goal, but with the complexity and pervasiveness of code it seems likely only ever to be a mirage. The ‘monitoring citizen’, on the other hand, is empowered to find out information *when she needs it*. As Pols and Spahn put it, ‘[t]he Monitoring Citizen does not *know* everything that is going on but can *monitor* it successfully and can investigate and contest policy when needed.’⁹³ This ideal is more plausible than aiming for ‘real’ or ‘full’ transparency, and it provides a guiding principle for design: the aim is to empower the end-user by affording her, at an appropriate level of abstraction, information about the operation and purposes of the code she is interacting with. One can think here of an analogy with legislative procedure: in addition to the citizen being able to access directly the legislative rule in the statutory document (cf. Fuller’s first principle), it is also possible for her access explanatory notes, impact assessments, Hansard, and other ancillary material in order to glean a deeper understanding of the purpose of a piece of legislation.

3.2.2 *Affordance: transparency of operation*

The most obvious mechanism here is transparency in the imposition of normativity, in the form of documenting the use of a design that lies at a particular point on the normativity spectrum or actively informing the end-user of this fact. As we saw above, however, transparency is a contested concept. In the context of machine learning systems where it has been criticised as a means by which engineers can ‘whitewash’ decisions that are antagonistic to end-users’ interests;⁹⁴ similar concerns can apply in the code-driven context.⁹⁵ Transparency as a goal can be framed such that including descriptions of functionality in lengthy terms documents that notionally inform the

⁹³ *Ibid.* (emphasis supplied, references omitted).

⁹⁴ See for example L. Edwards and M. Veale, ‘Slave to the Algorithm: Why a Right to an Explanation Is Probably Not the Remedy You Are Looking For’ (2017) 16 *Duke Law & Technology Review* 18.

⁹⁵ Recall the distinction between code-driven and data-driven systems discussed in Chapter 1, section 1.1 (Code).

end-user is legitimate, when in practice it does little to illuminate for her what is actually going on.⁹⁶ The (ideological) belief here is that simply providing more information will enable end-users to make informed choices about which products can and cannot fulfil their preferences, thus leading to greater competition and better products by dint of the operation of market forces⁹⁷ (recall the normative distinction between the ‘monitoring citizen’ and the ‘well-informed citizen’ in the previous section).

Other work argues for solutions that do not involve the end-user directly. For example, the source code that underpins regulatory software systems could be required to be open, i.e. viewable, in order that it might be audited by third parties to detect unethical or unlawful functionality that might not otherwise be apparent.⁹⁸ This proposal may be plausible for public sector regulators (the European Commission, for example, follows such a mandate⁹⁹), but as the ‘code wars’ in the late 90s to mid-2000s showed, commercial enterprise has been reticent if not actively hostile to the idea of opening up the proprietary code at the core of its products and services.¹⁰⁰ Others have

⁹⁶ Hartzog, *supra* n. 2, ch. 2; M. Piekarski and W. Wachowski, ‘Artefacts as Social Things: Design-Based Approach to Normativity’ (2018) 22 *Techné: Research in Philosophy and Technology* 400, pp. 414–415.

⁹⁷ O. Ben-Shahar and C.E. Schneider, ‘The Failure of Mandated Disclosure’ [2011] *University of Pennsylvania Law Review* 647.

⁹⁸ D.K. Citron, ‘Open Code Governance’ [2008] *University of Chicago Legal Forum* 355; L. Lessig, *Code: Version 2.0* (New York: Basic Books, 2006) ch. 8. Both authors are specifically concerned with abuse by public institutions.

⁹⁹ ‘Open Source Software Strategy’ (European Commission, 2016) <https://ec.europa.eu/info/departments/informatics/open-source-software-strategy_en>.

¹⁰⁰ Even if, in recent years, some of the largest contributors to open source projects have turned out to be those who were previously most hostile to them. See for example E. Angelova, ‘Microsoft Embraces Open Source’ [2018] *Fordham Intellectual Property, Media & Entertainment Law Journal* <<http://www.fordhamiplj.org/2018/11/28/microsoft-embraces-open-source/>>.

suggested an escrow system, where an artefact's source code is held by a trusted third party who can be required to release it by a court if litigation should arise.¹⁰¹

In neither of these suggested cases are the full context and texture of the code's materiality taken into account. (Dis)affordance and inscription speak to more than just the bare logic of the code, so while it is true that a great deal can be gleaned from code just by studying it (assuming the relevant expertise, of course), a broader sensitivity to qualitative design concepts is required in order fully to appreciate its effects in operation, particularly those on the end-user. Perhaps more importantly, such approaches are not based on input legitimacy, because they operate as a kind of insurance policy to be invoked only once malfeasance has been suspected or detected. By definition, such assessments are *ex post*, and so they do nothing to avoid the production of illegitimate code in the first place. Should the effects of the code not be suspected or detected then it might continue indefinitely to operate illegitimately.

The goal of transparency in this context should not, therefore, be limited to the literal openness of source code. As we saw in Chapter 2, design can signify to the end-user what the functionality of the system is and what it allows her to do.¹⁰² What matters is comprehension rather than bare notification, and thus it is incumbent on the designer to ensure as far as possible that the end-user's mental model of the system matches what it actually does.¹⁰³ As Hartzog notes, '[g]ood design means that a user's mental map of how a technology works matches reality.'¹⁰⁴ The designer must ensure that she has taken into account the 'conceptual model' (the combination of information about the product from advertisements, sales people, instruction manuals, and the

¹⁰¹ Cf. J.L. Mezrich, 'Source Code Escrow: An Exercise in Futility' (2001) 5 *Marquette Intellectual Property Law Review* 117.

¹⁰² Chapter 2, section 2.1.1 (Signifiers). See also Hartzog, *supra* n. 2, p. 27.

¹⁰³ D.A. Norman, *The Design of Everyday Things* (Cambridge, Mass; London, UK: MIT Press, 2013) pp. 26, 31.

¹⁰⁴ Hartzog, *supra* n. 2, p. 278.

interface of the artefact itself) that the end-user is likely to construct, acknowledging that her own model of the system, with her intimate knowledge of its operation, is likely to differ significantly from that of the much less informed end-user.¹⁰⁵

Similar considerations arise in relation to the legisprudential principle of coherence (PC).¹⁰⁶ In terms of internal coherence (LoCs 0 and 1), the code should be consistent in its design language (cf. the grammar and basic meaning of words under the PC) and it is the designer's role to ensure the artefact is 'understandable and usable.'¹⁰⁷ End-users should not be confronted with conflicting or inconsistent design in order to avoid causing confusion and misapprehension (recall the discussion in Chapter 2 of malicious design practices and how these are used for nefarious – i.e. illegitimate – ends¹⁰⁸). In terms of LoC₁, this includes arbitrary changes that can confuse or trick the end-user, especially after she has developed a familiarity with how the code works.¹⁰⁹

Affording this kind of transparency might be termed *transparency of operation* – the requirement is to design in such a way that the end-user can understand what the code is doing as it operates. This form of transparency is also an on-going concern (and linked with the strategy of oversight, discussed below). Because systems are frequently updated with features added and removed, it is incumbent on the designer to appropriately communicate such changes where they 'reconfigure' the relationship between end-user and enterprise.¹¹⁰

¹⁰⁵ Norman, *supra* n. 103, p. 31.

¹⁰⁶ Chapter 4, section 2.3.3 (The principle of coherence (PC)).

¹⁰⁷ Norman, *supra* n. 103, p. 32.

¹⁰⁸ Chapter 2, section 3.1 (Disaffordance).

¹⁰⁹ This is connected with the concept of vendor lock-in, where the features that attracted end-users to a product or service are changed after the point where they are willing to look elsewhere.

¹¹⁰ Gürses and van Hoboken, *supra* n. 40, p. 594.

3.2.3 Discussion: blockchain applications

Blockchain applications are usually published on a public chain where they can be viewed by anyone, or at least by those who are subject to their terms. This however simply repeats the problem of source code transparency described above – having access to the application’s code does not automatically render it intelligible to those end-users who are or might be affected by its operation. Certain initiatives in the cryptocurrency community are seeking to ameliorate this problem. For example, the community behind the Ethereum platform is developing the ‘Ethereum Natural Specification Format’ (‘ENSF’),¹¹¹ designed to be used alongside Solidity, the programming language for Ethereum blockchain applications. ENSF is a form of code commentary that enables blockchain application designers to tag the individual elements of an Ethereum application with descriptions, from which a natural language explanation of the application’s purpose can be automatically generated using templates into which the contingent values of the code are inserted (e.g. party and parameter names, and notices of what a particular function does). The result is a commentary of the blockchain application, a simple example of which for a single function might look like this:

Send 1.125 BTC from the account of ABC to an account accessible only by
XYZ¹¹²

This message might be derived from the following tags which immediately precede the actual application code which performs the action:

¹¹¹ ‘Ethereum Natural Specification Format’, , *The Ethereum Wiki* (Ethereum Foundation, 2018) <<https://github.com/ethereum/wiki/wiki/Ethereum-Natural-Specification-Format>>.

¹¹² This and the next example are simplified versions of those found at *ibid*.

```
/// @notice Send `(valueBTC / 1000).fixed(0,3)` BTC from the
account of `message.caller.address()` to an account accessible
only by `to.address()`
```

One can see how the elements between the backticks (`) are placeholders for the actual values generated within the application's logic, and how the tagging thereby results in the commentary above.

This provides some useful measure of transparency of operation, in that the logic of the blockchain application can theoretically be explained to the end-user. A problem with such approaches, however, is that they inevitably rely on the designer's subjective understanding of the code. For the approach to work she must accurately model, in a combination of natural language and code placeholders, the logic of the application. If she fails to do this, intentionally or mistakenly, the end-user might end up with both an erroneous understanding of the system *and* a misplaced confidence in that understanding, an outcome which is arguably less desirable than if there were no explanation at all. This is akin to Brownsword's focus on the rules which led to the use of code, rather than the normativity that it actually implements.¹¹³ Relying on descriptions of this sort adds an additional layer of abstraction between the end-user and the instrumentality of the code, increasing the likelihood of errors and misinterpretations on the part of the designer, the end-user, or both. What we have, then, is the resurgence of the hermeneutic gap between text on the page (in this case the tagging/commentaries) and the instrumentality of the code.

This may be a necessary compromise if transparency is to be achieved, however, since as we saw merely publishing raw source code with no commentary by itself does little to advance end-user understanding. There is also the possibility of leveraging the programmer of the programmer, in the form of the software design environment, to detect instances during development where such tags might be included in the code

¹¹³ This point was previously made *supra* in section 1.1.

and to suggest that or force the designer to add them. This approach is discussed further in Chapter 6.

3.2.4 Discussion: IoT

Affording transparency in the IoT is a complex challenge for several reasons. IoT devices are often intended to be embedded and pervasive, creating a network of interconnected devices that communicate with one another to create ‘ambient intelligence’ or ‘ubiquitous computing’.¹¹⁴ As a consequence of the invisibility that arises from such devices ‘receding’ into the background of everyday life, they often have either minimal or no interface with which the end-user can interact in order to observe what the code is actually doing. Many IoT devices will offer few or even no perceptual affordances, and with such minimal means of communicating their presence and/or purposes to the end-user, the opacity of the normativity is all the more impenetrable. As Matassa and Simeoni warn,

...the existing affordances in connected and technologically augmented objects are becoming unable to immediately communicate to people their actual values and meanings... [t]he impossibility of establishing a clear connection between objects and functionalities could become a threat for humans, since they are missing their innate ability to understand what they can do only based on their knowledge and perception of the surrounding context.¹¹⁵

This connects with the discussion of real and perceived affordances in Chapter 2.¹¹⁶ The invisibility of IoT devices, and/or their minimal interfaces, mean that the communication of even perceived (dis)affordance is already limited, much less the *real*

¹¹⁴ M. Weiser, ‘The Computer for the 21st Century’ [1991] *Scientific American* 94.

¹¹⁵ A. Matassa and R. Simeoni, ‘Eliciting Affordances for Smart Objects in IoT Era’, *Internet of Things. User-Centric IoT* (Springer, Cham, 2015) pp. 77–78.

¹¹⁶ Chapter 2, section 2.1 (Real and perceived affordances).

(dis)affordances that the design embodies. The scope then is all the greater for the end-user to experience a dissonance between what she thinks is happening, what her possibilities for action are, and what is actually taking place.¹¹⁷

It may be, then, that to achieve legitimacy an IoT device must be consciously designed to facilitate understanding on the part of the end-user, even where this is not necessary for the product's purpose to be achieved. The normativity of the artefact should thus be made apparent to the end-user. IoT devices are hybrids – they combine the up-front physicality of a tangible object with the background processing of (networked) code, the latter being made even less tractable by the absence of an interface.

We saw above that one approach to facilitating intelligibility is through the use of a traditional device such as a smartphone or connected television to provide the interface through which the user can interact with and monitor the IoT device. Devices such as 'smart' thermostats and doorbells use this approach. The degree to which the obscured (dis)affordances/inscriptions embodied in the device are communicated to the end-user will vary according to the complexity of the device's functions. For example, after it is configured via the end-user's smartphone, the Amazon Dash Button provides the most minimal of interfaces – a simple adhesive push-button that can be stuck at various points around the home, which when pressed re-orders the product indicated by a logo on its surface.¹¹⁸ The apparent simplicity of the single-button interface belies the significant complexity of the operations that take place when it is pressed: these involve multiple networking processes (WiFi, TCP/IP handshakes, Amazon account authentication), a financial transaction (communication between Amazon's servers and the provider of the end-user's bank account) and, perhaps most

¹¹⁷ Matassa and Simeoni, *supra* n. 115, p. 78.

¹¹⁸ 'Amazon Help: Set Up Your Dash Button' <https://www.amazon.co.uk/gp/help/customer/display.html/ref=amb_link_1?nodeId=201746340>

importantly, the generating of data that deepens Amazon's profile of the end-user's domestic preferences and shopping habits. The gap between the simplicity of the device and what actually goes on demonstrates the potential dissonance Matassa and Simeoni refer to (albeit that the end-user will presumably have at least some awareness of what the button does given its purpose).

3.3 Immediacy

The immediacy of code is especially problematic when combined with the contradictory or impossible rules warned of in Fuller's fifth and sixth principles. We saw above in relation to the legisprudential principle of coherence how contradiction can apply within the computational sphere – at the level of the interface, design language can be confusing to the end-user if it lacks consistency. Impossible rules can guide end-users into situations where there is no logical way out. For example, website cookie notices often give the illusion of providing the choice of whether or not to consent but in reality require acquiescence in order to gain access.¹¹⁹ Frequent changes to the code are also problematic – end-users can become accustomed to one way of working with an artefact then find this being changed or reversed by a software update. Depending on the kind of artefact the scope for such changes can vary – consider the interface changes that online platforms occasionally roll out which can disorient end-users to the point of backlash.¹²⁰ Beneath the surface of code, alterations to functionality can also have important effects: the periodic tweaks to Google's search

¹¹⁹ An example of this is the Oath advertising network (a merger of Yahoo! and AOL, owned by Verizon) whose cookie notice opens into a 'privacy dashboard' with a byzantine set of notifications about tracking, rather than options to disable it. For an animation vividly depicting this situation, see @gael, 'How Did We Get Here?' (*Twitter*, 8 January 2019) <https://twitter.com/_gael_/status/1082702284746883072>.

¹²⁰ Consider Facebook's move to a 'news feed' layout, which resulted in significant resistance from end-users. See J. Leyden, 'Users Protest over "creepy" Facebook Update' *The Register* (9 July 2006) <https://www.theregister.co.uk/2006/09/07/facebook_update_controversy/>.

algorithm can significantly alter what material is found on the web for a given search term.¹²¹

From a legisprudential perspective, immediacy speaks to the principle of normative density (PN).¹²² The fact of the immediate imposition of a given configuration of (dis)affordance/inscription means that its impact is of particular importance, making *ex ante* consideration all the more necessary. We saw above in the discussion of opacity how the density of technological normativity is a topic of crucial concern; the immediacy of its application heightens this further. Immediacy also concerns the legisprudential principle of temporality (PT),¹²³ requiring sensitivity to the moment of the imposition of normativity and ongoing justification to ensure the mechanism for achieving the aim of the norm continues to be appropriate in light of the other principles.

3.3.1 *Affordance: delay*

We have seen how speed and immediacy of execution are quintessential elements of computational legalism. Floridi suggests that in the modern computational context the lack of ‘informational friction’ contrasts with previous eras of human society where the inherent makeup of the social fabric meant that information could not travel above a certain speed or beyond a certain geographical radius.¹²⁴ For him, information privacy is facilitated in part by the ‘ontological friction’ within a system, which operates to oppose the flow of information and increase the effort required to gain access to it.¹²⁵

¹²¹ See ‘Google Algorithm Change History’ (Moz, 2018) <<https://moz.com/google-algorithm-change>>.

¹²² Chapter 4, section 2.3.6 (The principle of normative density (PN)).

¹²³ *Ibid.*, section 2.3.5 (The principle of temporality (PT)).

¹²⁴ L. Floridi, *The Fourth Revolution: How the Infosphere Is Reshaping Human Reality* (OUP Oxford, 2014) ch. 5. See also W. McGeeveran, ‘The Law of Friction’ (2013) 2013 *University of Chicago Legal Forum* 15.

¹²⁵ Floridi, *supra* n. 90, p. 231 *et seq.*

Similarly, in the legal context Hildebrandt argues that it is precisely the inherent ‘friction’ of text that has resulted in the existence and character of the institutions and processes of law that we now take for granted.¹²⁶ The affordances of text and the printing press have in turn afforded society a certain kind of law, one that is based around text-based norms whose meanings are under-determined but whose textual expressions are stable enough to facilitate a baseline of inter-geographical and inter-generational consensus that can, through incremental evolution through democratic processes, adapt to societal change.¹²⁷

The possibilities for friction provided by a given design are affordances of that design, and are therefore susceptible to conscious implementation by the designer. Floridi suggests (rather optimistically) that ‘contrary to old ICTs [information and communications technologies], new ICTs empower users in both directions, as they can both increase and decrease informational friction’.¹²⁸ Whether or not this is true for a particular system is contingent in large part on its designer’s willingness to make it so – Floridi acknowledges this reality when he warns that ‘[s]olutions will not develop by themselves without some effort on our part.’¹²⁹

3.3.1.1 Desirable inefficiency

In a recent paper Ohm and Frankle posit the concept of ‘desirable inefficiency’,¹³⁰ where the efficiency of code (embodied as we have seen in the computationally-legalistic characteristics of ruleishness and immediacy) is consciously tempered in order

¹²⁶ Hildebrandt, *Smart Technologies*, *supra* n. 16 *passim*.

¹²⁷ *Ibid.*, ch. 3. I contrast Hildebrandt’s conception of (orthodox) law as an affordance of text and the printing press with the idea of code’s affordances being compatible with the substantive requirements of the law in L. Diver, ‘Law as a User: Design, Affordance, and the Technological Mediation of Norms’ (2018) 15 *SCRIPTed* 4, p. 30 *et seq.*

¹²⁸ Floridi, *supra* n. 124, p. 113.

¹²⁹ *Ibid.*, p. 115.

¹³⁰ P. Ohm and J. Frankle, ‘Desirable Inefficiency’ (2019) 70 *Florida Law Review* 1.

to permit or protect some value that it might otherwise undermine. They define efficiency in the computer science context as ‘the extent to which [a system] minimizes the consumption of time, energy, space, or cost in satisfying a specification of correctness for a given problem.’¹³¹ A desirably inefficient approach is one that

fails to minimize the consumption of time, energy, or space in satisfying a specification of correctness for a given basic problem in order to address a different, related enhanced problem.¹³²

The ‘basic problem’ is the underlying technical outcome that the designer wishes to achieve, while the ‘enhanced problem’ is one that requires ‘human judgment, values, or discretion in the definition of success or failure’.¹³³ The authors argue in favour of desirable inefficiency as a set of design patterns, part of their call for a ‘new interdisciplinary research agenda investigating how values can be embedded into code.’¹³⁴ They give the example of a passcode screen on a smartphone, which locks the phone for a progressively longer period of time when incorrect attempts are registered. Designs like this balance the inconvenience that a forgetful end-user experiences with the security of the device that might otherwise be compromised in the hands of a thief;¹³⁵ the technical ‘basic problem’ is providing the end-user with secure access to her smartphone, while the societal ‘enhanced problem’ is the prevention of access for thieves.¹³⁶ Another example is proof-of-work in blockchain applications,¹³⁷ where the technical processes involved that would otherwise be capable of near-instant computation (i.e. recording in a database the outcome of a transaction) are, through

¹³¹ *Ibid.*, p. 28.

¹³² *Ibid.*, p. 29.

¹³³ *Ibid.*, p. 32.

¹³⁴ *Ibid.*, p. 5.

¹³⁵ *Ibid.*, p. 15.

¹³⁶ *Ibid.*, pp. 29–30.

¹³⁷ This is the ‘mathematical challenge’ discussed in Chapter 1, section 4.2.1 (Blockchain design).

proof-of-work, made sufficiently 'inefficient' to re-introduce the human values of trust and 'clock time'.¹³⁸ The basic problem is tamper-resistant validation of transactions, while the enhanced problem is their *fair* validation.¹³⁹

Ohm and Frankle's analysis focuses strongly on the underlying logics of computational systems, explicitly excluding from their discussion designs that 'do no more than slow down the operation of a computer to match the speed of human processing systems'.¹⁴⁰ There seems, however, to be no principled reason why the concept of desirable inefficiency cannot be a conscious part of the design of end-user-facing code, especially where doing so can help to facilitate another human value such as comprehension of the code's behaviour or the opportunity to exercise autonomy. As Clark et al. suggest, even where greater efficiency is possible from a technical perspective, in some cases it will be better to opt for a less efficient design where doing so makes it possible to separate points at which the artefact's design implicates diverging or conflicting interests (this is the idea of tussle that I discussed above).¹⁴¹

When framed in terms of (dis)affordances and inscriptions, the concept of desirable inefficiency can be applied fruitfully to end-user-facing systems where that inefficiency operates to throttle computational performance in service of end-user comprehension and empowerment, in opposition to the immediacy of code. Fraser and Kitchin discuss 'slow computing',¹⁴² the idea of consciously reducing 'time compression, fragmentation, densification and stresses' in end-user interactions with

¹³⁸ Ohm and Frankle, *supra* n. 130, pp. 19–22. That proof-of-work blockchains are in most contemporary cases extremely inefficient in terms of energy consumption is a separate concern – that form of *undesirable* inefficiency is categorically different from the type of inefficiency Ohm and Frankle are considering.

¹³⁹ *Ibid.*, pp. 29–30.

¹⁴⁰ *Ibid.*, pp. 35–36.

¹⁴¹ Clark et al., *supra* n. 20, p. 467.

¹⁴² A. Fraser and R. Kitchin, 'Slow Computing' (Maynooth University, 2017) The Programmable City Working Paper Series 36.

digital artefacts.¹⁴³ Pols and Spahn connect this outlook with critical theories of technology that view technology as a threat to democracy and justice.¹⁴⁴ According to such theories, social spheres in which democratic values ought to be given time and space to operate are in danger of being limited by a ‘technological rationality that centers on efficiency and strategic rationality’.¹⁴⁵ Democracy and justice depend on ‘communicative’ rather than ‘strategic’ rationality, and thus the space for the former must be left open. This can be read to imply the affordance of inefficiency to circumscribe ‘technological rationality’ (speed, efficiency, certainty) in favour of the space for communicative possibilities. We can also see here a connection to Julie Cohen’s concept of the ‘play of everyday practice’.¹⁴⁶ For Cohen, play is a necessary part of how individuals are empowered to exercise their autonomy within architectural and institutional structures, exploiting the space between predictability and contingency. This is akin to the space between the imposition of heteronomy by core constitutive functionality – the ‘morality of duty’ – and the provision of ‘open’ spaces within the code’s geography that can invite the exercise of freedom and autonomy – the ‘morality of aspiration’. This relates to the counterintuitive concept of consciously encouraging ambiguity in a design’s affordances so as not to constrain end-users’ responses to the artefact to only those possibilities predetermined by the designer.¹⁴⁷ We can see how these ideas relate to the concept of multi-stability and the possibility of the end-user re-interpreting a design’s purpose, discussed in Chapter 2.¹⁴⁸ The extent

¹⁴³ *Ibid.*, p. 8f.

¹⁴⁴ Pols and Spahn, *supra* n. 38, p. 342 *et seq.*

¹⁴⁵ *Ibid.*, p. 345.

¹⁴⁶ J.E. Cohen, *Configuring the Networked Self: Law, Code, and the Play of Everyday Practice* (Connecticut: Yale University Press, 2012) ch. 2.

¹⁴⁷ W.W. Gaver, J. Beaver and S. Benford, ‘Ambiguity As a Resource for Design’, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA: ACM, 2003) <<http://doi.acm.org/10.1145/642611.642653>>.

¹⁴⁸ Chapter 2, section 3.4 (Code mediating action). See also *Ibid.*, p. 236f.

to which this can happen is contingent, at least to some extent, on the latitude for such reinterpretation left open by the designer. What can be taken from these various perspectives is that technological inefficiency, when consciously and appropriately designed into an artefact, can be an important bulwark against the limitations on freedom that technological normativity can impose.

3.3.1.2 Friction

The concept of friction has been used in recent literature to refer to its opposite, particularly in the context of information sharing on social networks. The presumption has been, as with inefficiency in the computer science context, that less is always better.¹⁴⁹ ‘Frictionless sharing’ refers to the ease and speed with which the code of such platforms affords sharing,¹⁵⁰ through for example the use of metadata standards (e.g. Open Graph tags that describe content in a machine-readable way, enabling previews and analytics) and front- and back-end functionalities that reduce the cognitive burden involved in sharing (e.g. share buttons embedded on third party websites and liking/re-sharing functionalities that combine with end-user interaction metrics to enable ‘virality’). The reduction in friction can be taken even further, to the point where everyday events such as visiting a particular retailer or going for a run are automatically shared by the code on social media platforms without the end-user’s input.¹⁵¹ Before these affordances existed, the act of sharing content online was subject to the ‘friction’ of having to go through various manual steps: copying the URL of the item into an email or instant message, choosing the recipient(s), and perhaps composing a short note describing what the link referred to (before the introduction of metadata

¹⁴⁹ McGeeveran, *supra* n. 124, p. 51; Ohm and Frankle, *supra* n. 130, pp. 10–13. Calo also argues in favour of a reduction in friction, which he pits in opposition to the ‘facilitation’ of the end-user’s aims. See R. Calo, ‘Code, Nudge, or Notice’ (2013) 99 *Iowa Law Review* 773.

¹⁵⁰ McGeeveran, *supra* n. 124. See also C. Reed and A. Murray, *Rethinking the Jurisprudence of Cyberspace* (Cheltenham, UK: Edward Elgar Publishing, 2018) p. 120.

¹⁵¹ T. Bucher, ‘A Technicity of Attention: How Software “Makes Sense”’ (2012) 13 *Culture Machine*.

previews, all that would be included was the URL itself, which on its own might make little sense to the recipient). All of this requires thought and conscious decision-making on the part of the sharer, in contrast to the single-click, one-to-many forms of sharing described above.

Designs that include especially efficient affordances can have unforeseen consequences if they are not accompanied by appropriate instruction or information; they remove friction from the action in question while hiding the potentially undesirable and complex consequences of that action (recall the discussion of IoT devices in section 3.2.4 above). One of the problems of Facebook's frictionless sharing is the extent to which some end-users were unaware of the precise audience they were sharing their intimate posts with.¹⁵² In that respect, McGeeveran connects the idea of friction to design:

...the amount of friction is a complex design choice, which inherently helps some users and burdens others. We cannot avoid making some choice, whether through code or law; there is no "natural" state of online friction.¹⁵³

This is ultimately a question of affordance; the option to share is an affordance of the interface and back-end code that is made available to the end-user at a time chosen by the designer. In the case of sharing, McGeeveran suggests a design principle, according to which the affordance (though he does not use the language of design theory) of sharing should not be made available before the act itself has taken place. His 'law of friction' states that 'it should not be easier to "share" an action online than to do it.'¹⁵⁴ A similar principle can apply in relation to any computational operation that has

¹⁵² McGeeveran calls this a 'misclosure': McGeeveran, *supra* n. 124, p. 39 *et seq*, citing K.E. Caine, 'Supporting Privacy by Preventing Misclosure', *CHI09 Extended Abstracts on Human Factors in Computing Systems* (ACM, 2009). See also S. Sengupta, 'Private Posts on Facebook Revealed' <<https://bits.blogs.nytimes.com/2013/01/18/private-posts-on-facebook-revealed/>>.

¹⁵³ McGeeveran, *supra* n. 124, pp. 53–54.

¹⁵⁴ *Ibid.*, p. 63.

normative effect: the end-user should be afforded the opportunity to consider the operation before it takes place. The intention here, then, is to *consciously* design friction into the appropriate parts of the artefact, in order that end-users are given an opportunity to take stock before the code moves on to the next step in its logic.¹⁵⁵ The end-user can comprehend a system only through the relatively small keyhole of its interface; the depth and breadth of the mass of code steps that are *actually* being executed is akin to the vast bulk of an iceberg hidden beneath surface waters – as Vismann and Krajewski put it, ‘[b]eyond the interface, users have no access whatsoever.’¹⁵⁶ The challenge then is to design interfaces that afford the appropriate pacing of computation, alongside an appropriate level of technical feedback, in order to facilitate a sufficiently detailed mental model for the end-user to enable her to make an accurate prediction of what will happen next.¹⁵⁷ This idea connects with the concept of the ‘human in the loop’, discussed next.

3.3.1.3 Human in the loop

We saw in Chapter 3 how the automatic application of code-base rules is in very many cases perfectly possible from a purely technological perspective.¹⁵⁸ The characteristics of computational legalism are attractive from a commercial perspective, particularly when channelling end-user behaviour can be made more efficient and pervasive while at the same time reducing cost. A primary mechanism for forcing delay into code-mediated processes is the ‘human in the loop’ (‘HitL’), where those elements of the process that are appropriate for the code to execute are left to the machine, and

¹⁵⁵ Ohm and Frankle, *supra* n. 130, pp. 51–52; McGeeveran, *supra* n. 124.

¹⁵⁶ C. Vismann and M. Krajewski, ‘Computer Juridisms’ [2007] *Grey Room* 90, p. 100.

¹⁵⁷ Hartzog, *supra* n. 2, p. 278. For a recent argument in favour of making end-user experiences less simple in certain circumstances, see K. Roose, ‘Is Tech Too Easy to Use?’ *The New York Times* (13 December 2018) <<https://www.nytimes.com/2018/12/12/technology/tech-friction-frictionless.html>>.

¹⁵⁸ Chapter 3, section 3.1 (Ruleishness).

decisions that have social, ethical, or legal import are made by a human (or the machine's suggestion ratified by a human). The classic application of the HitL principle is in lethal autonomous weapon systems, where the elements of an engagement are automated up to the point of the final decision on whether or not to strike, which must be taken by a human controller.¹⁵⁹ Discussing the use of code in the law enforcement context specifically, Hartzog et al. argue for a 'conservation principle' that would require inefficiency and indeterminacy to be conserved, primarily by retaining a role for humans to exercise judgment at specific points within the criminal justice process.¹⁶⁰ For them, HitL is a necessary bulwark against the determinism of inflexible code, and they suggest that where one of the three elements of law enforcement that they consider is automated in code (surveillance, analysis, and detection), the (desirable) inefficiency and indeterminacy of the other two should be increased proportionately.¹⁶¹

In the context of consumer products, the role of the HitL will be played by the end-user herself. In order to retain the desirable delaying effect of the HitL's 'inefficiency', then, interfaces ought to afford end-users notification and choice before normatively-significant computational events take place. As discussed above, information about these scenarios should not be front-loaded in terms documents that are not read, but rather should be delivered piecemeal at appropriate moments in the end-user's journey through the code's inscriptions. This might be achieved by, for example, employing 'just in time' notifications, akin to those used in the Android operating system to allow the end-user to give or deny permissions to an application requesting access to an operating system feature at the moment the request is made,

¹⁵⁹ N. Sharkey, 'Grounds for Discrimination: Autonomous Robot Weapons' (2008) 11 *RUSI Defence Systems* 86.

¹⁶⁰ W. Hartzog et al., 'Inefficiently Automated Law Enforcement' [2015] *Michigan State Law Review* 1763.

¹⁶¹ *Ibid.*, p. 1778.

instead of in bulk at the time of installation when she might not foresee all the relevant implications.¹⁶² The idea is to granularize permission-giving at the appropriate level, and to request it only at the relevant time, such that the end-user can make a choice that is informed by context.

HitL is also a necessary element of retaining indeterminacy, the quality of a circumstance not being adequately reflected in the code or data which come to represent it (recall the discussion on representationalism in legalism, and how code's representations are reductive of the world).¹⁶³ Whereas code *imposes* such (reductive) interpretations, indeterminacy (or 'underdeterminacy', in Hildebrandt's terms¹⁶⁴) should actively be preserved in order to allow for responses that are sensitive to the texture of the real world. Thus, the HitL has a role in 'completing the narrative' in such scenarios, filling in the contextual gaps which computational representations are incapable of showing sensitivity to but which are nevertheless important in the pursuit of justice or of end-user autonomy.¹⁶⁵ The goal, then, is to ensure that the design affords HitL input at all appropriate points in its inscription, in order that the aspirations of freedom and rational interpretation are not effaced by the 'duty' of wired-in code.

3.3.2 Discussion: blockchain applications

Many of the considerations of ruleishness discussed above also apply here in the context of immediacy. Levy notes that '[b]ecause they are based on code', blockchain applications 'can be *immediately and automatically* effectuated, without reliance on

¹⁶² 'Android Developers Guide - Permissions Overview' <<https://developer.android.com/guide/topics/permissions/overview>>.

¹⁶³ F.A. Pasquale, 'A Rule of Persons, Not Machines: The Limits of Legal Automation' [2019] *George Washington Law Review* (forthcoming) p. 53 *et seq.*

¹⁶⁴ M. Hildebrandt, 'Legal and Technological Normativity: More (and Less) than Twin Sisters' (2008) 12 *Techné: Research in Philosophy and Technology* 169, p. 177.

¹⁶⁵ Hartzog et al., *supra* n. 160, p. 1785 *et seq.*

manual transfer, or the intervention of institutions like courts.¹⁶⁶ One of the putative benefits of blockchain applications (promoted in particular by smart contract enthusiasts¹⁶⁷) is their removal of the perceived inefficiency of ambiguity and processual costs.¹⁶⁸ This efficiency is potentially deeply problematic, especially if the code has been poorly designed. When combined with the immutability of blockchains, the consequences can be serious indeed. As de Filippi and Wright suggest,

[t]he automated nature of smart contracts, combined with the inability to readily alter their underlying code, could further lead to situations where a faultly [sic] piece of code would repeatedly run, to the detriment of all parties involved.¹⁶⁹

This implies the need for *ex ante* consideration of the implications of automated and immediate execution: assets or funds could be transferred or goods and services ordered according to the predetermined logic of the blockchain application, without any human intervention or oversight. This could happen near-instantaneously if the conditions in the code were met – the good is ordered, the funds transferred, and the drone commandeered (recall the discussion in Chapter 1¹⁷⁰). As with the affordance of choice, providing delay may therefore require the design of appropriate moments in which the end-user is afforded the opportunity to consider the situation before execution of the code continues. Given that anticipation of every conceived outcome

¹⁶⁶ Levy, *supra* n. 73, p. 2 (emphasis supplied).

¹⁶⁷ Pasquale, *supra* n. 163, *passim*. For an example see ‘Mattereum Protocol: Turning Code into Law’ (Mattereum Project, 2018) <https://www.mattereum.com/upload/iblock/784/mattereum-summary_white_paper.pdf>.

¹⁶⁸ Levy, *supra* n. 73, p. 2.

¹⁶⁹ P. De Filippi and A. Wright, *Blockchain and the Law: The Rule of Code* (Cambridge, Massachusetts: Harvard University Press, 2018) p. 201.

¹⁷⁰ Chapter 1, sections 4.2.2 (‘Smart contracts?’) and 4.3 (The Internet of Things), respectively.

is impossible,¹⁷¹ contingency ought not to be the province of code, and any attempt to enclose it is perhaps likely to set up unforeseen and undesirable results. Simultaneously, however, imposing friction in blockchain applications is arguably anathema to their ethos. It may be necessary, however, given their exemplification of computational legalism, if they are to achieve digisprudential legitimacy.

3.3.3 Discussion: IoT

I alluded above in the discussion of default choices to IoT webcams that have ‘out-of-the-box’ configurations that are potentially problematic, such as insecure default passwords.¹⁷² We saw how end-users often trust that designers know better than they do, and so they assume that the default configuration is the most sensible one. Such configurations are especially problematic in the IoT, because the object itself might be ‘plug and play’, which is to say it starts operating according to its default configuration as soon as it is switched on. This single action may be enough by itself to set off various undesirable path dependencies, for example joining an open wireless network, and connecting to a remote server to register its existence. Designing in delay in this context, then, might involve ensuring that IoT devices have all defaults set initially to prevent any functionality that is not immediately signified by the physical characteristics of the device. This relates to the discussion of the ‘would not have wanted’ standard in defaults, discussed above.

To return to the example of the smart fridge, when the device is first switched on it would immediately begin cooling because that is its inherent purpose (i.e. its intuitive physical affordance of cooling would be immediately available), but its ‘smart’ (Internet-connected) functionalities would remain disabled until the end-user takes the active step of configuring and enabling them (i.e. they are intentionally disafforded

¹⁷¹ Cf. C.D. Clack, V.A. Bakshi and L. Braine, ‘Smart Contract Templates: Foundations, Design Landscape and Research Directions’ [2017] *arXiv:1608.00771 [cs]* p. 4.

¹⁷² Brian Krebs, *supra* n. 45.

until the end-user has taken the conscious decision to enable and configure them). Building in this kind of delay before the normativity of the code gets a chance to execute can enable space for the other affordances above to be facilitated: the end-user can consider the implications of the device's provenance and purpose *before* the code that imposes its normativity is executed, giving her a chance to respond to any misgivings she may have before (potentially opaque) harm is done. If she chooses to go ahead, she can then think about which configurable choices best fit her interests. This would accord with the aim expressed by Hartzog in the quote at the beginning of this chapter, where the design of code reflects the 'most optimal and sustainable choices, affordances, and constraints for users.'¹⁷³ Much like the suggestion discussed in below in the section on oversight that manufacturers include a 'floor' of security in even their disposable IoT products, one might imagine a minimum delay where no functionality that is not signified by the physical properties of the artefact can be enabled prior to the end-user taking the active choice to do so, even (and particularly) where this is in opposition to the commercial interests of the manufacturer. In addition to a 'floor' of security, one can think of a 'ceiling' of affordance that can only be extended by the conscious choice of the end-user.

3.4 Immutability

The issues with immutability overlap with those demonstrated by ruleishness and immediacy. The Fullerian principle regarding frequency of change applies, but in the opposite sense: the fact that certain media are resistant to being updated must be borne in mind in their design; the threshold between unchangeable duty and the space allowed for the end-user's 'aspiration' to operate must be set with this concern in mind, lest path dependencies arise which can result in users being locked into the constraints

¹⁷³ Hartzog, *supra* n. 2, p. 57.

of a particular design.¹⁷⁴ This relates to the legisprudential principle of temporality (PT)¹⁷⁵ which, as previously mentioned, requires sensitivity to the concreteness of the imposition of an external limitation, particularly so where there is less scope for future alteration – in such cases the requirement of ex ante justification is all the stronger. Furthermore, because of the potential for unexpected future harm, the necessity for ongoing vigilance to ensure the crystallisation of normativity continues to be justified is also particularly important in this context.

This connects too with the principle of coherence (PC),¹⁷⁶ which as we saw requires a broader societal justification (under LoC₃) and not just coherence according to the system's own internal rationale. On this footing, the digisprudential strategy of oversight means that a change in the external justification must be capable of being reflected in a change of code rules; a failure to afford this would mean the continued operation of illegitimate code, regardless of whether or not it was legitimate at the original time of release.

3.4.1 *Affordance: oversight*

The principle here is that the manufacturer ought not release code to market unless the necessary conditions are in place for them to maintain oversight of it and to correct any (unforeseen) negative consequences. This is similar to the concept of revocability in the HCI-Security literature, where user-centric design principles suggest that the end-user must be afforded the possibility of revoking any permissions she has granted within the system.¹⁷⁷ In this context, the concept of revocability requires that the creator of the code is capable of maintaining some control over it. This brings to mind

¹⁷⁴ *Ibid.*, pp. 76–77.

¹⁷⁵ Chapter 4, section 2.3.5 (The principle of temporality (PT)).

¹⁷⁶ *Ibid.*, section 2.3.3 (The principle of coherence (PC)).

¹⁷⁷ K.-P. Yee, 'User Interaction Design for Secure Systems' in R. Deng et al. (eds.), *Information and Communications Security* (Berlin, Heidelberg: Springer Berlin Heidelberg, 2002).

one of the central themes of Shelley's *Frankenstein* and Winner's discussion of its contemporary relevance: 'men release powerful changes into the world with cavalier disregard for consequences'.¹⁷⁸ In order to maintain legitimacy, therefore, the design must anticipate *ex ante* the potential need to make changes *ex post*. Consideration of this principle requires that the design permits it, and any design that does not is *prima facie* illegitimate.¹⁷⁹

Consider, for example, the Sony BMG DRM scandal discussed in Chapter 1.¹⁸⁰ The problematic effects of the design were amplified by the fact it was stored on an inherently immutable medium, namely the compact disc. Although the system *was* ultimately revoked, this was only as a result of the significant public relations impact of the scandal, and that revocation took the form of a laborious and expensive physical recall of over seven million CDs.¹⁸¹ Similar issues arise in relation to the IoT, where the market provides slim margins on inexpensive devices and thus the incentive to invest in long-term updates and support is diminished. Devices are therefore rushed to market without either the necessary capacity for post hoc software updates or the ongoing commitment to providing bug and security fixes.¹⁸² The design must therefore afford oversight by the designer or enterprise so that necessary changes can be made.

¹⁷⁸ L. Winner, *Autonomous Technology: Technics-out-of-Control as a Theme in Political Thought* (2nd edn., Cambridge, Mass: MIT Press, 1977) p. 314. For an edition of the novel that includes annotations which illuminate Shelley's cautionary themes from an engineering perspective, see M. Shelley, *Frankenstein: A New Edition for Scientists and Engineers* (Ed Finn, David Guston and Jason Scott Robert eds, Cambridge, Mass: MIT Press, 2017).

¹⁷⁹ This brings to mind recent developments in the EU's Ecodesign initiative, mandating that product designs include the facility (affordance) of repairability in order to reduce waste and extend the lifespan of consumer electronics. See M. Anastasio, 'EU Governments Support First Set of Laws for More Repairable Products' (*EEB - The European Environmental Bureau*) <<https://eeb.org/eu-governments-support-first-set-of-laws-for-more-repairable-products/>>.

¹⁸⁰ Chapter 1, section 4.1 (Digital rights management).

¹⁸¹ *Ibid.*

¹⁸² W. Hartzog and E. Selinger, 'The Internet of Heirlooms and Disposable Things' (2016) 17 *North Carolina Journal of Law & Technology* 581.

This might involve anticipation of software updates, now a fairly standard feature in modern networked devices.

This is in line with the design standard suggested by Hartzog and Selinger, according to which IoT device manufacturers must provide a ‘minimum expectation for servicing’ and a ‘floor of data security for even disposable items.’¹⁸³ The implication here is that, as elsewhere within the digisprudential framework, if the enterprise cannot commit to such standards of oversight then the legitimacy of the design has *de facto* not been demonstrated. Similarly, where the design will not permit updates by its very nature (for example due to limited connectivity or processing power) then the scope of wired-in functionality should be to that extent limited to ensure that the unchangeable code will not be responsible for any future negative effects. The design must therefore anticipate the possibility of external change, either by the facilitation of remote updates or by restricting the scope of its normativity *ab initio*. Where it proves too difficult to anticipate such eventualities, ex post remedial measures of the sort envisaged by Hartzog and Selinger (e.g. third-party maintenance or insurance) must be put in place. If none of this is possible, the inevitable conclusion is that the design is *a priori* illegitimate.

3.4.1.1 Sunsetting and ‘lobotomy switches’

In their discussion of the Sony BMG DRM scandal Halderman and Felten suggest the inclusion of a ‘sunsetting’ feature that renders the system inert after a specified date or period.¹⁸⁴ Depending on the business model being pursued this might avoid some of the problems of code operating far into the future, particularly if it is especially difficult to alter it post hoc (as with the physical CD media discussed above). They suggest that the Sony BMG DRM system could have been designed to run only until

¹⁸³ *Ibid.*, p. 597.

¹⁸⁴ J.A. Halderman and E.W. Felten, ‘Lessons from the Sony CD DRM Episode’, *15th USENIX Security Symposium* (USENIX Association, 2006) p. 89.

a point beyond which the commercial benefit of preventing copying through code was negligible enough as to be outweighed by the other (computationally-legalistic) effects of the system. They propose a period of three years, during which they expect that nearly all possible revenue from disc sales would by that point have been raised and collected. An approach of this sort is contingent on the business model that is adopted – the economics of CD sales have of course changed significantly in the years since the Sony BMG scandal. Nevertheless the principle is still valid: designers ought to consider the medium- and long-term effects of the technological normativity they embody in their systems, and where oversight over such a period is anticipated to be difficult or impossible, sunseting may be an appropriate safety mechanism to limit the possible effects of the code operating blindly in unforeseen contexts.

A related approach is what Hartzog calls a ‘lobotomy switch’ which, following the discovery of a serious bug, reduces the system to a set of core functions while disabling any optional affordances (and in particular network access).¹⁸⁵ This is the mirror image to the discussion of core affordances above in the section on delay: basic functionality is retained, but optional ‘smartness’ is disabled. Hartzog gives the example of a child’s Internet-enabled doll: once the lobotomy switch is flipped, the doll can still be played with, but its potentially security- and privacy-harming connectivity is disabled. The efficacy of such an approach of course depends on the type of device; if networking is a central element of its purpose (e.g. the Amazon Dash Button, which relies on network connectivity) then disabling it through a lobotomy switch might render the device effectively useless. Furthermore, this sort of design proposal is also complicated by questions of who should control such a switch, and under which precise conditions it should be activated. The determination of such

¹⁸⁵ Hartzog, *supra* n. 2, p. 272.

questions might represent a salient point at which the institutional law can step in through traditional regulation.¹⁸⁶

In any event, the overarching question of legitimacy operates, raising the thorny question of whether the device should have been designed in such a way in the first place: if the manufacturer cannot commit to (i) supporting the device with updates and maintenance for a reasonable period, (ii) ‘sunsetting’ the device after a specified period, or (iii) retaining sufficient control to permit a ‘lobotomy’ to be performed should this turn out to be necessary, then it can be argued that the design is not digisprudentially legitimate, because it does not afford the necessary level of oversight.

3.4.2 Discussion: blockchain applications

One of the defining characteristics (and selling points) of blockchains is that data stored on them is tamper-resistant.¹⁸⁷ From the perspective of traditional contracting this is especially problematic, since the ex ante interpretation that is reified in the code of the blockchain application when it is stored on the chain will be executed when the relevant conditions arise, regardless of any intervening factors which might suggest a change is desirable.¹⁸⁸ The technical necessity for consensus to be reached in order to make changes, coupled with the inability unilaterally to break the contract (and face the (legal) consequences), makes these artefacts particularly problematic from the perspective of oversight. Observing the fact of the application’s *execution* may be possible, because the storage of the output of a blockchain application’s execution is generally stored on the underlying chain, thus enabling audit ex post. This is, however, a different form of oversight to that with which I am here concerned. What matters from a digisprudential perspective is ongoing maintainability and revocability, in order

¹⁸⁶ *Ibid.*, p. 273; B. Schneier, ‘I’ve Seen the Future, and It Has a Kill Switch’ [2008] *Wired* <<https://www.wired.com/2008/06/securitymatters-0626/>>.

¹⁸⁷ De Filippi and Wright, *supra* n. 169, pp. 35–37.

¹⁸⁸ K. O’Hara, ‘Smart Contracts - Dumb Idea’ (2017) 21 *IEEE Internet Computing* 97, p. 98.

to ensure that the code's normativity can be accounted for; both are, however, undermined by the immutability of the blockchain. If one end-user party to the application changes her mind, or is incapacitated, the code will remain on the blockchain and will execute as stored, regardless of such contingencies.

This goes to the very heart of the kind of ex ante anticipation that digisprudence is concerned with. Designers of blockchain applications must be aware of contingencies (far) in advance and must, because of the immediacy of the code, limit its normative scope to those facts they can be reasonably certain of. The number of variables can quickly become extremely complex, however, but such conditions will not prevent a blockchain application from operating unless its code is designed to include some external check of such facts.¹⁸⁹ It is questionable whether it is feasible properly to predetermine all the relevant contingencies that might arise, and even those that are foreseen will require reliable sources that the code can interpret. The ability of blockchain applications to react to external conditions through oracles – external sources of contingent data¹⁹⁰ – may not provide the necessary information, or that information may be inaccurate, incomplete, or not provided in a format the code is equipped to 'understand'. Even where these issues are not present, the assumption is that the third-party oracle will continue to operate as it did at the time when the application was designed, but this may well not be the case if that third party alters their systems or shuts down altogether. Furthermore, in terms of contestability, even where a judicial process might in theory be invoked to attempt to address any conflict that arises, it may be difficult to identify the parties from the application's code in order to demonstrate standing to launch an action or seek a decree, because identification on the blockchain is achieved by means of public keys rather than names. In any event,

¹⁸⁹ R.H. Weber, "Rose Is a Rose Is a Rose Is a Rose" – What about Code and Law? (2018) 34 *Computer Law & Security Review* 701, p. 5.

¹⁹⁰ Cardozo Blockchain Project, "Smart Contracts" & Legal Enforceability' (New York: Benjamin N Cardozo School of Law, 2018) p. 8.

even were it possible such an appeal to judicial process would in many cases take place after the code has executed and its negative effects have been felt.

From the perspective of affording oversight, then, two factors may ultimately militate against the use of blockchain applications. First, if the designer cannot be sure that certain crucial facts will obtain at the point of execution, she must limit the ‘wired-in’ elements of the code to exclude these. Difficulties arise in identifying the threshold between what Clack et al. call the ‘operational aspects’ of the blockchain application, namely those that are automatable, and the ‘non-operational aspects’, that is those that cannot or should not be automated.¹⁹¹ Too much automation and many or all of the effects of computational legalism are amplified; too little and what logic of the blockchain application remains automated may be so simplified that it is essentially ‘dumb’, rendering the application a mere tool (or a ‘mechanism’, in Felten’s terminology¹⁹²) for the execution of a real-world agreement, the latter retaining responsibility for managing human ‘messiness’ and contingency.¹⁹³ Thus, the social aspects of traditional agreement (including institutional legal contracting) continue to deal with contingent ‘real world’ parts of human arrangements, while the role of the blockchain application is constrained to those limited factors that are susceptible to reliable and predictable code-based representation and enforcement.¹⁹⁴ This is, in a sense, to flip the ‘lobotomy switch’ ex ante, limiting the design of the application from the beginning in the knowledge that it might otherwise harbour too much normative power, power that is exacerbated by the other characteristics of computational legalism that blockchain applications exhibit. The result of this may be that while the code is

¹⁹¹ Clack, Bakshi and Braine, *supra* n. 171, p. 5.

¹⁹² E. Felten, ‘Smart Contracts: Neither Smart nor Contracts?’ <<https://freedom-to-tinker.com/2017/02/20/smart-contracts-neither-smart-not-contracts/>>.

¹⁹³ De Filippi and Wright, *supra* n. 169, p. 199f; Cardozo Blockchain Project, *supra* n. 190, p. 4.

¹⁹⁴ Levy, *supra* n. 73.

legitimate from an oversight perspective, from a commercial perspective the lack of ‘smartness’ undermines its attractiveness or utility.

The second factor militating against the use of blockchain applications in this context relates to the code’s ability to respond to contingent facts. If the designer is unwilling to forego the ‘smartness’ of the application in the manner just described, the external contingent facts that it relies upon must be verifiable at the point of execution. This implies the use of oracles that are themselves trustworthy and accurate, which is problematic from an oversight perspective because it devolves the determination of a crucial element of the artefact’s logic away from the designer, thus undermining her ability to oversee her own design. It may be possible to design ‘meta-contingency’, somewhat akin to sunseting, whereby if the blockchain application will simply lie inert if, at the point of execution, it cannot confirm a given fact to the requisite degree of certainty. Of course, any such safety valve requires to be consciously built into the logic of the blockchain application, which of course is not a given. Whether the precise set of facts that would come within this bracket can be identified by a designer in advance (rather than by a court *ex post*, with all the benefits of expert evidence and time to deliberate), and whether they can be provided by an oracle in a form that is susceptible to computational representation, are questions that are themselves contingent on many external conditions being in place (e.g. a facility providing information that the relevant end-users are still alive, or that the property which the application relates to still exists and is in the possession of the relevant party who retains the right to dispose of it). It may be that the complexity and the variety of factors that ought to be taken into consideration means that these standards of oversight cannot be met, which may call into question the legitimacy of such applications *a priori*.¹⁹⁵

¹⁹⁵ Pasquale, *supra* n. 163, p. 24 *et seq.*

3.4.3 Discussion: IoT

We have already seen in the discussion above some suggestions from the literature that relate specifically to IoT devices. Because they tend to be low-cost, manufacturers have in some cases underinvested in the ongoing maintenance of their devices.¹⁹⁶ The resources required to track and fix device bugs, and to provide infrastructure for delivering updates to the devices, can mean that in a competitive environment resources are redirected instead to researching and developing new products that can meet the demands of a febrile market. Some manufacturers have even resorted to altering *legal* terms in order to contract out of the responsibility for the technological normativity of their designs, for example Vtech, who, in the wake of a serious personal data breach, simply changed their terms document to shift responsibility onto the end-user instead of altering the design of their product, an Internet-connected child's toy.¹⁹⁷ This is clearly an illegitimate approach by digisprudential standards, given that it *prima facie* does nothing to make legitimate the *technological* normativity of the device.

For IoT devices, then, oversight must be designed into the system itself, including the ability to update its software should this be required in future (and, by implication, a commitment to support such updates). As Hartzog and Selinger suggest,

[i]magine a system where companies told users how long they think a wired object will last and how long the company will commit to providing security patches. In the event that a company goes bankrupt before then, companies

¹⁹⁶ L. Edwards, D. McAuley and L. Diver, 'From Privacy Impact Assessment to Social Impact Assessment', *2016 IEEE Security and Privacy Workshops (SPW)* (San Jose, CA: IEEE, 2016).

¹⁹⁷ L. Franceschi-Bicchierai, 'Hacked Toy Company VTech's TOS Now Says It's Not Liable for Hacks' [2016] *Motherboard* <<http://motherboard.vice.com/read/hacked-toy-company-vtech-tos-now-says-its-not-liable-for-hacks>>.

would work quickly to either notify users of its impending shut down or facilitate the [transfer of] responsibility for security patches to a third party.¹⁹⁸

This might be combined with sunseting facilities that either warn the end-user that the device has an expected operating life of a specified period or, if the supporting infrastructure becomes unavailable (e.g. due to insolvency of the manufacturer), that there will be either a third-party support mechanism or the system will either gracefully degrade (sunseting/the lobotomy switch) or be disabled altogether. What such measures might mean for consumer protection or contract law remains to be seen; of course, as with all the other digisprudential affordances, if the designer or manufacturer of the device cannot commit to producing a design that embodies a sufficient level of legitimacy then the conclusion must always remain open that the design is *a priori* illegitimate and should not be released.

3.5 Pervasiveness

The pervasiveness of code connects with the idea of ‘juridification’ and the legalistic proliferation of ‘ever more refined and rigid systems of formal definitions.’¹⁹⁹ This is an implied aspect of what the legisprudential principle of normative density (PN)²⁰⁰ aims to reduce, through the increase in justification required proportionate to the imposition of a limitation on freedom, with a criminal sanction being the ‘densest’ example.²⁰¹ The concept of juridification takes this wider to consider not just the ‘density’ of a given norm’s limitation on freedom, but the aggregate impact on freedom of the proliferation of (legal) normativity more generally.²⁰²

¹⁹⁸ Hartzog and Selinger, *supra* n. 182, p. 597.

¹⁹⁹ J.N. Shklar, *Legalism* (Harvard University Press, 1964) p. 2.

²⁰⁰ Chapter 4, section 2.3.6 (The principle of normative density (PN)).

²⁰¹ *Ibid.*

²⁰² L.C. Blichner and A. Molander, ‘What Is Juridification?’ (Centre for European Studies, University of Oslo, 2005) p. 12 *et seq.*

In the legal sphere the effects of juridification are limited by human cognitive capability and institutional resources – beyond a certain threshold, regulatees cannot comprehend the body of norms they are subject to, and there are limited resources for legal enforcement. There is thus a natural limit to the impact of pervasiveness within the legal sphere. In the computational realm, however, such limits do not exist (or their threshold is much higher); the number of norms or aggregate normativity that can be applied by and through code is effectively unlimited, and the self-enforcing nature of computational norms removes any issue around resource allocation. Pervasiveness under computational legalism thus exemplifies these two aspects of juridification – normative density of individual norms, and complex aggregations of normativity embodied in the inscriptions of both individual devices and networked collections of devices. We have already seen how technological normativity can have an immediate regulatory effect in a way that law cannot; whereas traditional legal norms can be directed at whole populations (or even large classes of individual), their text-bound character limits the direct and real-time imposition of their normativity. The way that code differs in this respect is made all the stronger when the artefact has widespread adoption – large numbers of individuals can be subject simultaneously to the regulatory effect of even a single design decision.²⁰³ The categorical difference here means that the Fullerian and legisprudential principles do not apply quite as readily in this context as in the others. However, one can adapt the legisprudential principle of normative density (PN)²⁰⁴ to take account of the collective normative effect of a design, questioning whether the aggregate imposition of normativity can be justified when factors such as the product’s target market and likely ‘market penetration’ are considered. When combined with the other digisprudential affordances, the question

²⁰³ A. Huldtgren, ‘Design for Values in ICT’ in J. van den Hoven, P.E. Vermaas and I. van de Poel (eds.), *Handbook of Ethics, Values, and Technological Design* (Dordrecht: Springer Netherlands, 2014) p. 741.

²⁰⁴ Chapter 4, section 2.3.6 (The principle of normative density (PN)).

of aggregate technological normativity becomes extremely salient; pervasiveness takes the qualitative aspects of computational legalism and adds a quantitative element to the balance of justification. Put simply, the problems of computational legalism are amplified by code's pervasiveness.

Practically, the very idea of 'ubiquitous computing' as the precursor to the IoT implies very directly to the idea of pervasiveness. Such devices are intended to be embedded in other 'things', and/or to be innocuous in operation. Statistics on their proliferation suggest that pervasiveness is the trend,²⁰⁵ especially given advances in battery and cellular technology. The issue of pervasiveness seems therefore only to heighten the need for the digisprudential affordances set out above.

4. Conclusion

This chapter has sought to strengthen the relationship between legal-theoretical notions of legitimacy and the practical question of what legitimate code ought to afford the end-user (contestability, choice, transparency, and delay), legal institutions (evidential standards), or its own manufacturer (oversight). It sets out a digisprudential framework, couched in terms of affordances, that can guide the design of code towards legitimacy. It thus contributes to a bridging of the divide between legal-theoretical notions of normative legitimacy and their practical instantiations, discussing the issues from the perspective of the case studies to provide a sense of real-world application.

The next chapter considers some of these issues even more practically, focusing on the code development cycle and consider how the programmer of the programmer

²⁰⁵ Gartner predicts that there will be 14.2bn 'connected things' in use in 2019, rising to 25bn by 2021. See Gloria Omale, 'Gartner Identifies Top 10 Strategic IoT Technologies and Trends' (<https://www.gartner.com/en/newsroom>, 7 November 2018) <<https://www.gartner.com/en/newsroom/press-releases/2018-11-07-gartner-identifies-top-10-strategic-iot-technologies-and-trends>>.

can be employed as a 'constitutional actor' able to guide the production of code toward legitimacy.

Chapter 6

Operationalisation: towards agile *digisprudence*

An insurmountable barrier between users and system programmers safeguards the computer's inalterable functions. Beyond this barrier, as in Kafka's story, a new barrier appears between the programmer and the programmer of the programming language who decides how the basic set of elements is to be designed, which rights and properties will be granted to whom, and which will be denied.¹

1. Introduction

The discussion in the previous chapter set out the digisprudential affordances and their relevance to both the legal-theoretical principles of legality and legisprudence, and the normative criteria for code, discussed in Chapter 4. This final substantive chapter discusses some practical ways in which the digisprudence framework might be operationalised. It is not intended to be an exhaustive survey of coding practices, but rather to identify some points in the production process where operationalisation of the framework is particularly relevant, and to draw from the literature some practical approaches to that could prove effective.

We have seen that the intent of the framework is to bind in a constitutional way the design of code to underlying principles that are reflected in that design, regardless of its commercial purpose (and whose presence may therefore logically prevent illegitimate commercial purposes from being pursued). Like Odysseus, the product designer is 'bound to the mast', and the design of her code along with her. We saw in Chapter 1 how this idea of the product designer herself being constrained by a prior set of 'constitutional' design choices can be conceptualised in the position of what

¹ C. Vismann and M. Krajewski, 'Computer Juridisms' [2007] *Grey Room* 90, p. 101.

Vismann and Krajewski call the *programmer of the programmer* ('PoP').² This is an under-studied area in the literature; the contribution of this chapter is in part to strengthen the practical connection between the PoP and its legal-theoretical analogues.

2. The programmer of the programmer

I referred in Chapters 1 and 2 to Vismann and Krajewski's discussion of the 'structural homologies' between computers and law.³ The vertical model of normative relationships (Figure 1 in Chapter 1⁴) hints at the analogy described there: the constitution binds the legislature, which promulgates norms that regulate the citizen, those norms being legitimated by the democratic process and the formal requirements of legality and jurisprudential legitimation. This is the top-down aspect of the vertical model. By analogy, from a bottom up perspective we have the PoP building into the design environment (i.e. the software and tools that designers use) 'constitutional' limits that bind the product designer's activities when she uses the software and tools of her trade to produce code. This is particularly true of 'integrated development environments', discussed below, which are the software environments that lie at the heart of coding practice. The 'parliamentary' or 'legislative' work of the product designer is thus constrained according to the (dis)affordances and inscriptions contained in the design environment, which if defined according to the digisprudential perspective can in turn mean that the normativities embodied in the code she produces are also legitimate.

² *Ibid.*, p. 100.

³ Chapter 1, section 2 (Code is law(-like)); Chapter 2, section 3.6 (Technological constitutionalism).

⁴ Chapter 1, section 1.4.1 (Normative relationships online), Figure 1 (Legal and technological normativity relationships).

As an aside, with respect to democratic participation in the design process, I suggested in Chapter 1 that this might be a separate element of the process of design, but because it is concerned more with substance than with form it falls outside the scope of my enquiry. Rational citizens would presumably never decline the benefits of legality, and therefore, by analogy, neither would end-users decline digisprudential legitimation.⁵ Participatory design approaches are concerned with substantive functionality of the code (its output, or ‘external morality’ in Lon Fuller’s terminology⁶), and not necessarily with whether it provides principled foundational features of the kind I am arguing for (derived from ‘input criteria’ or ‘internal morality’⁷); we might therefore say that digisprudence is to participatory design practices as legisprudence is to the democratic process. Whatever the political outcome of a democratic process we expect certain formal features to have been present; so too with code – whatever its substantive purposes, we expect it to embody certain basic underlying design features. For that reason, I expressly do not consider the role of participatory design, although I acknowledge that it is not incompatible with digisprudence and, indeed, if the two approaches were to be *combined*, the resulting code would undoubtedly have a very strong claim to legitimacy.

Returning to the normative (rather than merely descriptive⁸) perspective on the PoP, the product designer is in a sense merely a user, because despite the vast freedom she enjoys in defining her code’s normativity, she is herself constrained by prior design decisions made by those PoPs who create the tools of her trade: hardware, programming languages, and the software tools used to design new code. The parallel even runs to the fundamental level of the CPU’s architecture, where Vismann and Krajewski characterise the chip as a ‘sovereign’ and Intel (one of the world’s largest

⁵ Chapter 1, section 1.4 (Legitimacy vs. ‘compliance by design’).

⁶ Recall the discussion in Chapter 4, section 2.1 (Input and output legitimacy).

⁷ *Ibid.*

⁸ I described the PoP in greater detail in Chapter 2, section 3.6 (Technological constitutionalism).

CPU manufacturers) as a ‘legislator’, by dint of the power they wield over the design of the internal rules, or ‘instruction sets’, of the processor.⁹ This is the apotheosis of the PoP metaphor, where the ultimate technical constitution is found in the set of low-level instructions that is defined by the physical hardware of the chip. Recalling Vismann and Krajewski’s definition, the PoP

maintains the ultimate power because he or she, as the constructor of the programming language itself, defines what the ‘normal’ programmer, as a user, will be able to do. Both types of programmers [sic] establish the conditions of using the computer, and, as such, they behave like lawmakers or, rather, code-makers.¹⁰

Although Vismann and Krajewski are concerned primarily with the homologies between legal structure and chip architecture, one can appreciate the broader application of their insight about ‘meta-architecture’, and how it mediates the work of the designer in the production of end-user-facing code. The idea of establishing the conditions under which the ‘normal’ programmer (i.e. the product designer) can go about her work brings to mind Hart’s distinction between primary and secondary rules, the subject of the next section.

2.1 From primary and secondary rules to primary and secondary (dis)affordances and inscriptions

Thinking normatively about the role of the PoP, we can consider how to leverage it to impose elements of the ‘constitutional’ framework of digisprudence on product designers operating later in the production process. The idea, then, is to push for

⁹ Vismann and Krajewski, *supra* n. 1, pp. 96–97. See also F.A. Kittler, ‘Protected Mode’ in J. Johnston (ed.), S. Harris (tr.), *Literature, Media, Information Systems: Essays* (Psychology Press, 1997), arguing that conceptions of power should come not from analysing society but rather from examining chip architectures (*ibid.*, p. 162).

¹⁰ Vismann and Krajewski, *supra* n. 1, p. 100 (emphasis supplied).

‘legitimacy by design, by design’, through the structuring, guiding, and restraining of product design practices according to the requirements and aims of digisprudence. This can (and should) be aimed for whatever the substantive purpose of the code being produced or the underlying business model being pursued.

One can think of this in terms of Hart’s primary and secondary rules. As we saw previously, primary rules are those that require a substantive behaviour (or forbearance) on the part of the addressee.¹¹ Secondary rules are those that define the conditions under which the primary rules can be created and changed (and adjudicated).¹² Secondary rules are thus *ex ante* and ‘constitutional’, defining the process for creating primary rules and the proper form that they should take.

The primary rules find their analogue in the (dis)affordances and inscriptions that make their way into the design of the artefact and directly constrain and enable the behaviour of the end-user, as discussed in detail in Chapter 2. In the digisprudential context, we can envisage including in the design process secondary rules that constrain what primary (dis)affordances and inscriptions the designer may build into her product’s code. This analysis suggests certain homologies between Hart’s thesis, the legisprudential hierarchy, and digisprudence:

Table 2. Hartian-Legisprudential-Digisprudential homologies

<i>Hartian norm</i>	<i>Legisprudential locus</i>	<i>Digisprudential actor (creates)</i>
-	Citizen	End-user
Primary rule	Legislature	Product designer (Primary (dis)affordance/inscription)
Secondary rule	Binding constitution	PoP (Secondary (dis)affordance/inscription)

¹¹ H.L.A. Hart, *The Concept of Law* (2nd edn., Oxford: Clarendon Press, 1994) pp. 91–93.

¹² *Ibid.*, pp. 95–96.

The concept in the central column of a hierarchy of regulative force building up from a base ‘constitutive’ foundation maps onto the legisprudential concept of the ‘proxy model of legitimation’, discussed in Chapter 3.¹³

In terms of operationalisation, where the legislature is constrained by secondary rules and the legisprudential principles in traditional law-making, we can imagine in the design sphere the ‘legislature’ of the design process (including on a concrete technical level the integrated development environment, discussed below) being similarly constrained by secondary rules which guide what primary (dis)affordances and inscriptions can legitimately be created there.

Assessing the embodiment of some of the secondary legisprudential affordances will involve qualitative judgements – for example whether a given delay is sufficient to enable comprehension, or the extent to which oversight is afforded. This may be the point at which structured impact assessment might play a role.¹⁴ However, in addition to those qualitative approaches we can also envisage *secondary* (dis)affordances/inscriptions that are built into the very design environment itself, which in turn guide the work of the ‘designer-legislator’ in her creation of primary (dis)affordances/inscriptions, in a similar fashion to how the end-user is herself regulated by the end-product. The product designer becomes a regulatee, like the end-user. The substantive *primary* (dis)affordances and inscriptions thus remain end-user-facing, while the constitutional *secondary* (dis)affordances and inscriptions are designer-facing. The latter operate, according to the legisprudential perspective, to help produce legitimate instances of the former.

The next section discusses elements of the software development process which may be appropriate targets for this kind of ‘meta-normativity’. The first is the ‘agile’

¹³ Chapter 3, section 2.2 (Legalism according to legisprudence).

¹⁴ I discuss the possibilities for further research of this topic in Chapter 7, section 4.3 (‘Legitimacy impact assessment’).

development process, one of the most prominent contemporary code development methodologies, and the other is the class of software systems that designers use to write code, known as integrated development environments.

2.2 Agile development processes

Gürses and van Hoboken describe the recent shift towards ‘agile’ development processes as a ‘paradigmatic transformation in the production of digital functionality’.¹⁵ As we saw in earlier chapters,¹⁶ there has been a lack of focus by legal scholars on the production of code. Like Goldoni, Asscher, and Hartzog,¹⁷ Gürses and van Hoboken argue for greater focus to be placed on the production, rather than just the observed results, of code. The latter two authors’ analysis pushes the envelope further, however, by looking directly at, and gathering empirical data on, code design practices. Although their primary focus is privacy and the production of platform-based systems rather than individual artefacts, they acknowledge the ‘wider societal implications of the agile turn’,¹⁸ and as we saw in the second part of Chapter 4¹⁹ their concern about production is equally applicable to the more fundamental question of legitimacy. Framed through the perspective of production, Gürses and van Hoboken consider the role of agile development processes in the creation of the conditions that we now observe.

According to the Agile Manifesto, agile development processes are characterised by a focus on end-users, continuous development and testing,

¹⁵ S. Gürses and J. van Hoboken, ‘Privacy after the Agile Turn’ in E. Selinger, J. Polonetsky and O. Tene (eds.), *The Cambridge Handbook of Consumer Privacy* (1st edn., Cambridge University Press, 2018) p. 579.

¹⁶ Chapters 1 and 4, *passim*.

¹⁷ Chapter 4, sections 3.1 (Input and output legitimacy in code) and 3.3.2.1 (Asscher), respectively. See also references in Chapter 1, n. 34.

¹⁸ Gürses and van Hoboken, *supra* n. 15, p. 580.

¹⁹ Chapter 4, section 3 (Normative criteria for code-making).

collaboration, and response to change.²⁰ This approach contrasts with the ‘waterfall’ paradigm, the code development model that was dominant between the 1970s and 1990s,²¹ in which stages of the design process were discrete and sequential, with limited recursion and feedback between them.²² They were thus to an extent somewhat brittle: whereas the focus of agile processes is on producing modularised working code as early as possible and integrating feedback as it is gathered,²³ waterfall processes rely on ‘rigorously regimented practices, extensive documentation and detailed planning and management.’²⁴ Agile processes are thus cyclical and responsive, while waterfall processes move between pre-determined phases that are less flexible vis-à-vis contingencies and feedback. By nature, agile processes also accelerate the code development process because kinks and problems tend to be identified and fixed ‘on-the-fly’, rather than waiting until later testing phases that might uncover fundamental problems that require significant code redesign that is expensive and time-consuming.²⁵

This idea of incremental cycles that are responsive to changing requirements fits in well with the digisprudential framework outlined above.²⁶ In recent work the

²⁰ ‘Manifesto for Agile Software Development’ (2001) <<https://agilemanifesto.org/>>.

²¹ W.W. Royce, ‘Managing the Development of Large Software Systems’, *proceedings of IEEE WESCON* (Los Angeles, 1970). See also Gürses and van Hoboken, *supra* n. 15, p. 582.

²² D.A. Norman, *The Design of Everyday Things* (Cambridge, Mass; London, UK: MIT Press, 2013) p. 234f.

²³ T. Hoeren and S. Pinelli, ‘Agile Programming – Introduction and Current Legal Challenges’ (2018) 34 *Computer Law & Security Review* 1131, p. 1132.

²⁴ Gürses and van Hoboken, *supra* n. 15, p. 582.

²⁵ See for example ‘The Lean Startup | Methodology’ <<http://theleanstartup.com/principles>>, promoting a cyclical ‘build-measure-learn’ approach to code development. See also E. Luger and M. Golembewski, ‘Towards Fostering Compliance by Design; Drawing Designers into the Regulatory Frame’ in M. Taddeo and L. Floridi (eds.), *The Responsibilities of Online Service Providers* (Springer International Publishing, 2017) p. 296.

²⁶ For a practical discussion making this point in relation to Privacy by Design, see Alberto Crespo García et al., ‘PRIPARE Privacy- and Security-by-Design Methodology Handbook’ (EU FP7, 2015) p. 103 *et seq.*

technology ethics thinktank *doteveryone* has suggested augmenting agile cycles with anticipatory assessments of the potential consequences of design choices, thus enabling the mitigation of problems during the design process.²⁷ This chimes with the idea of continually assessing modules of code functionality according to whether (and the extent to which) they reflect the digisprudential affordances – as with Wintgens’ plotting of various proposed legislative norms in Chapter 5,²⁸ a given element of code functionality can be assessed according its embodiment of the affordances and whether and how it balances them. Rather than assessing or testing the design only once it is at or close to the stage of release, when problems are costliest to fix, a digisprudential perspective can operate continually to adjust production throughout agile cycles, pushing the design closer towards digisprudential legitimacy. Given the length and complexity of design processes, the enjoinder merely that ‘by design’ of whatever form (legitimacy, privacy, or legal compliance more generally) take place at the ‘early stages’ of the process is insufficient;²⁹ the proper embodiment of the value-based affordances I am describing requires continual assessment and re-assessment throughout the process, which cyclical agile methodologies might help to facilitate.

2.3 Integrated development environments

Returning to the discussion above of primary and secondary (dis)affordances and inscriptions, one place in which this concept might be implemented in a technically

²⁷ S. Brown, ‘An Agile Approach to Designing for the Consequences of Technology’ (*doteveryone*, 13 February 2019) <<https://medium.com/doteveryone/an-agile-approach-to-designing-for-the-consequences-of-technology-18a229de763b>>.

²⁸ Chapter 5, section 2 (Mapping the criteria), Figure 1 (Differing norm justifications (reproduced from Wintgens)).

²⁹ L. Diver and B. Schafer, ‘Opening the Black Box: Petri Nets and Privacy by Design’ (2017) 31 *International Review of Law, Computers & Technology* 68, p. 76; Gürses and van Hoboken, *supra* n. 15, p. 592. On the inadequacy of a ‘checklist’ approach to privacy by design, see S. Gürses, C. Troncoso and C. Diaz, ‘Engineering Privacy by Design’ (2011) 14 *Computers, Privacy & Data Protection*.

robust way is in the integrated development environment ('IDE') used by the product designer. In practice the developer does not write out the programming code by hand and enter it verbatim into the computer to be executed; the modern approach is to enter the source code into an IDE, which compiles it into code that is executable by the machine. In addition to this fundamental function, IDEs are also 'intended to assist the software lifecycle process'.³⁰ They can do this in a variety of ways; software applications for writing code vary in complexity and sophistication, from applications that are essentially simple text editors which require external tools (e.g. a compiler) to produce working code, to more powerful IDEs with in-built compilers, build-automation tools, version control, debuggers, tools for highlighting syntax and suggested code, etcetera. Most IDEs are able to detect problems in the source code that is entered, for example syntax errors (identified according to the requirements of the programming language being used), naming mistakes (e.g. incorrect variable or method names), and fatal errors (e.g. logically impossible statements and other incorrect programming 'grammar'). More sophisticated IDEs can auto-complete formulaic expressions in the relevant programming language and can keep track of a code project's structure to suggest relevant connections between code modules, on-the-fly as the designer is working (this is sometimes termed 'intelligent code completion'). Some can also suggest more computationally efficient means of achieving a particular outcome, for example by detecting that a particular instruction entered by the designer might be more efficiently achieved using an alternative mechanism provided by the programming language. It is also possible for the IDE to suggest points at which the code might be documented, or explanatory comments added,³¹ to

³⁰ A. Abran et al., *Guide to the Software Engineering Body of Knowledge (SWEBOK)* (Los Alamitos, California: IEEE Computer Society and Angela Burgess, 2004) p. 10-1.

³¹ In modern sophisticated IDEs this kind of functionality is included for documenting traditional code. See for example 'XML Documentation (Visual C++)' (Microsoft, 2016) <<https://docs.microsoft.com/en-us/cpp/ide/xml-documentation-visual-cpp>>.

enable other developers to understand what the code is designed to do (below I discuss two methods of providing such documentation which also assist in facilitating digisprudential contestability). This might be one means by which the transparency of blockchain applications especially – as mentioned in the previous chapter’s discussion³² – could be implemented within the IDE, whereby the software identifies where descriptive tags are missing and requires the designer to include them.

The design of these features – these affordances – can therefore have a significant impact on how the product designer goes about her job.³³ As we have already seen, that design is controlled by the notional PoP, who dictates to a great extent the scope of action of the product designer. The (secondary) affordances of the IDE might therefore feasibly be designed to discourage not only logical or aesthetic infelicities in the product designer’s code (as happens at present), but also the inclusion of code that is digisprudentially illegitimate.

Some initiatives exist already that aim to augment the purely logical aspects of the coding assistance that IDEs give the designer. For example, recent advances use machine learning directly within the IDE to facilitate best practice in the code suggestions that are provided to designers. Leveraging the ‘wisdom of the crowd’,³⁴ sophisticated code predictions can be provided that go beyond merely the static logical suggestions based on the syntax of the programming language that are provided by standard intelligent code completion. Microsoft, for example, has developed ‘IntelliCode’, a system that uses the open source code from popular projects on its GitHub service as training data to refine the code recommendations provided in its

³² Chapter 5, section 3.3.2.1 (Discussion: blockchain applications).

³³ R.B. Kline and A. Seffah, ‘Evaluation of Integrated Software Development Environments: Challenges and Results from Three Empirical Studies’ (2005) 63 *International Journal of Human-Computer Studies* 607.

³⁴ M. Bruch et al., ‘IDE 2.0: Collective Intelligence in Software Development’, *Proceedings of the FSE/SDP workshop on Future of software engineering research – FoSER ’10* (Santa Fe, New Mexico, USA: ACM Press, 2010).

Visual Studio IDE,³⁵ thus providing suggestions based on usage in real-world projects that inevitably includes aesthetic and, potentially, value-based code design choices.³⁶ If the code which embodies the sorts of legitimacy-creating practices I have discussed begins to promulgate widely, such systems might ultimately contribute to a greater standardisation of these concepts at the level of source code, perhaps contributing to the shift in the ‘institutional fabric’ described by Luger et al.³⁷

2.3.1 *Facilitating contestability in the IDE*

As I discussed in the previous chapter, one of the elements of contestability is affording the evidence of the code’s operation that is necessary for its legal effects to be assessed.³⁸ Not only are legal institutions empowered to make sense of the code (even though they have access to expert witnesses), but depending on the approach used it might be possible to demonstrate the effects of the code had a different design choice been made (this is the case with the Petri net approach, discussed below). I have elsewhere explored, with Burkhard Schafer, the possibility of using visual representation to map code states onto legal requirements.³⁹ Although that analysis concerned substantive doctrinal law, specifically data protection by design, the approach considered is agnostic with respect to the (legal) norms that are intended to be implemented in

³⁵ ‘Visual Studio IntelliCode’ (Microsoft, May 2018) <<https://visualstudio.microsoft.com/services/intellicode/>>.

³⁶ On values in source code, see for example A. Schlesinger, ‘Feminism and Programming Languages’ <<https://www.hastac.org/blogs/ari-schlesinger/2013/11/26/feminism-and-programming-languages>>; P. Graham, ‘Beating the Averages’ (2003) <<http://www.paulgraham.com/avg.html>> accessed 20 October 2017; J. Weizenbaum, *Computer Power and Human Reason: From Judgment to Calculation* (San Francisco: Freeman, 1976) pp. 102–104.

³⁷ E. Luger et al., ‘Playing the Legal Card: Using Ideation Cards to Raise Data Protection Issues Within the Design Process’, *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (New York, NY, USA: ACM, 2015) p. 464.

³⁸ Chapter 5, section 2.1 (Contestability as an overarching affordance).

³⁹ Diver and Schafer, *supra* n. 29.

code.⁴⁰ The approach entwines the gap between formal proof of technical requirements and legal proof in the courtroom setting.

As an ancillary benefit, the approaches below can also help to facilitate democratic participation in the design process. This is, as I have mentioned, explicitly not the focus of the thesis, given that the digisprudential framework is concerned more with form than with substance. Nevertheless, as previously discussed, such participatory design methods are complementary to digisprudence. The two approaches below can support them through the provision of a high-level language that assists comprehension of complex code by stakeholders in such participatory processes. Furthermore, because legitimacy does not *end* with digisprudence (rather, it begins with it), the PoP ought not to preclude or impede other elements of the design process that are useful or beneficial for other reasons. As we shall see, then, while the approaches below are aimed at attaining legitimacy in accordance with digisprudence, they are at the same time supportive of other normatively desirable practices involved in the production of code.

2.3.1.1 Formal verification of code versus legal proof

The importance of transparency and contestability in code governance has led to increased interest in the computer science community in formal methods that not only guarantee a certain outcome, but do so verifiably, and ideally in a way that allows relevant parties to see the ‘why’ of the code’s behaviour.

From a computer science perspective, the goal is to verify in advance that a system will operate according to a predefined set of characteristics (this is, of course, the core of code’s ruleishness). Although this is an important and helpful development from the perspective of legal compliance – recall the distinction, made in Chapter 1,

⁴⁰ *Ibid.*, p. 82f.

between compliance by design and legitimacy⁴¹ – it is not necessarily sufficient to afford true contestability. Legal proof and formal verification of code share similarities, but there are also crucial differences. A legal proof contains not just the substantive evidence to support a given conclusion, but also evidence of procedural propriety as distinct from that conclusion. In other words, the code may be compliant, but in order for legitimate contestability properly to be afforded it must provide evidence of ‘due process’ in order for the matter to be proven according to the necessary legal standard. Doing justice (i.e. complying) is insufficient; due process requires evidence of the procedure that was followed. In other words, justice must be seen to be done. Apprehending the right person in a criminal case is only one part of the equation; if their confession is obtained without legal representation then it is *de facto* illegitimate. Thus, due process under the law takes what *might* have happened to be as important as what *actually* happened. Evidence of this proof must be communicable in a specific way; legitimacy of the legal process requires a form of evidential transparency that goes beyond merely telling the ‘whole truth’; it requires that this truth is demonstrated to external observers (including the courts) in a form that is intelligible to them. In the context of institutional legal processes this idea finds its expression in the principle of the public trial.

Affording true contestability, then, requires enabling the end-user to detect code conditions that are susceptible to contest (transparency of operation, discussed in the previous chapter⁴²). Crucially, however, it can also be interpreted as requiring the demonstration of how the code was developed in the first place.⁴³ In a sense, the documenting of the digisprudential approach (particularly the outcomes of agile cycles discussed above) will in itself provide this, but it is also feasible to integrate a measure

⁴¹ Chapter 1, section 1.4 (Legitimacy vs. ‘compliance by design’).

⁴² Chapter 5, section 3.2.2 (Affordance: transparency of operation).

⁴³ T.J. Bench-Capon and F.P. Coenen, ‘Isomorphism and Legal Knowledge Based Systems’ (1992) 1 *Artificial Intelligence and Law* 65, p. 70f.

of this into the IDE through the use of approaches such as Petri net modelling and Behaviour-Driven Design.

2.3.1.2 Petri net modelling

The Petri net, originally conceived by Carl Petri in 1962,⁴⁴ is a standardised formal modelling approach designed to represent processes in terms of ‘states’ and ‘transitions’. Petri nets have been applied in many domains, not least in the modelling of legal provisions and processes.⁴⁵ More commonly, the nets are used in the early stages of the design of code to map graphically the changing states of the system over time. Despite their graphical appearance and apparent simplicity, the flow of a Petri net model can be both easily simulated and formally (mathematically) verified. They allow for a kind of ‘live documentation’ of the system, describing the functionality of the code in a way that is both intelligible to non-technologists but that is also isomorphic to the concrete behaviour of the system.⁴⁶ Prior research has demonstrated the automated generation of Petri nets from object-oriented source code⁴⁷ as well as (contrariwise) the automated generation of *code* from Petri models of intended functionality.⁴⁸ The validation qualities of Petri nets (through formal proofs and

⁴⁴ C.A. Petri, ‘Kommunikation Mit Automaten’ (PhD, University of Bonn 1962) <<http://epub.sub.uni-hamburg.de/informatik/volltexte/2011/160/>>.

⁴⁵ J. Freiheit et al., ‘Lexecute: Visualisation and Representation of Legal Procedures’ (2006) 3 *Digital Evidence & Elec. Signature L. Rev.* 19; J.A. Meldman, ‘A Petri-Net Representation of Civil Procedure’ (1977) 19 *Idea* 123; J.A. Meldman and A.W. Holt, ‘Petri Nets and Legal Systems’ (1971) 12 *Jurimetrics Journal* 65.

⁴⁶ B. Lin, ‘Software Synthesis of Process-Based Concurrent Programs’, *Proceedings of the 35th annual Design Automation Conference* (ACM, 1998); S.M. Shatz and W.K. Cheng, ‘A Petri Net Framework for Automated Static Analysis of Ada Tasking Behavior’ (1988) 8 *Journal of Systems and Software* 343.

⁴⁷ Lin, *supra* n. 46; Shatz and Cheng, *supra* n. 46.

⁴⁸ K.H. Mortensen, ‘Automatic Code Generation Method Based on Coloured Petri Net Models Applied on an Access Control System’ in M. Nielsen and D. Simpson (eds.), *Application and Theory of Petri Nets* (Berlin, Heidelberg: Springer Berlin Heidelberg, 2000).

reachability analysis⁴⁹) mean we can be sure of isomorphism between the code and the net, thus making the graphical representation a valuable evidential tool for making intelligible the concrete behaviour of the code. Petri nets can thus balance intuitive comprehension and analytical certainty,⁵⁰ both of which are important for evidential purposes. What follows is an extremely brief summary of how Petri nets operate.

The states and transitions in the model are represented by circles and rectangles, respectively. These are connected with arcs that represent the flow of the process, which at any given moment is represented by the distribution of ‘tokens’ across the model’s states. These four basic elements (states, transitions, arcs, and tokens) are the essence of all Petri nets:⁵¹

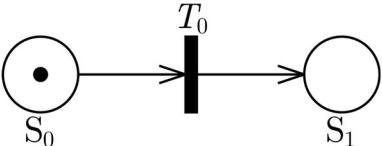


Figure 5. A basic Petri net

⁴⁹ Diver and Schafer, *supra* n. 29, p. 82f.
⁵⁰ K. Salimifard and M. Wright, ‘Petri Net-Based Modelling of Workflow Systems: An Overview’ (2001) 134 *European Journal of Operational Research* 664, p. 667.
⁵¹ This and the following Petri net models are reproduced from Diver and Schafer, *supra* n. 29.

A state containing a token (a dot) currently *holds*. Multiple states can lead to, or from, a given transition, and they can hold simultaneously, as shown below:

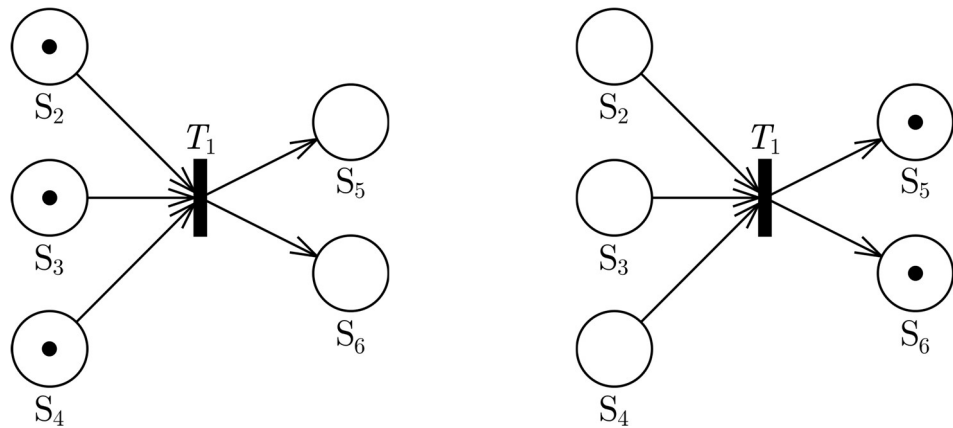


Figure 6. Multiple token transitions

When a transition fires, all the states leading to it will lose x tokens, and all the states leading from it will gain y tokens, where x and y correspond to the numerical weightings alongside each of the relevant arcs (no weighting implies a default of 1). A transition can only fire – and will *always* fire – where the number of tokens in its preceding state(s) is greater than or equal to the weighting of the relevant arc. This is demonstrated in Figure 7, where the transitions T_1 and T_2 are competing, with T_2 ‘winning’ because the two states leading to it have the requisite number of tokens to trigger that transition:

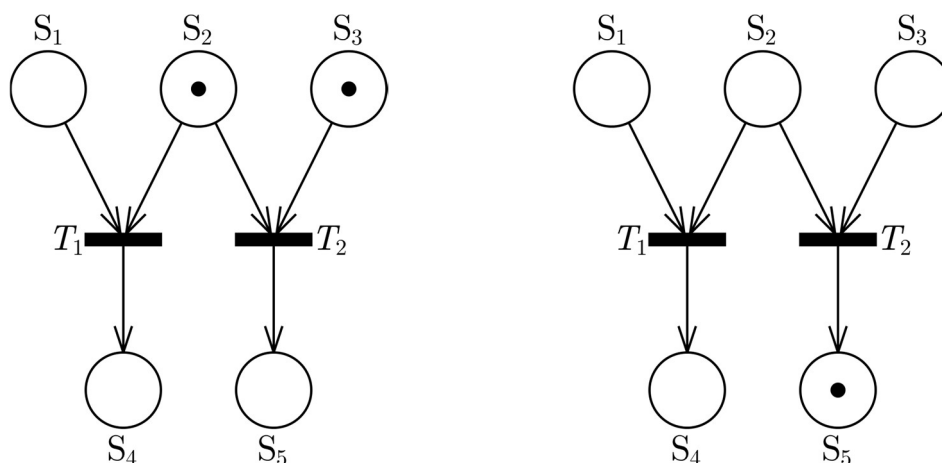


Figure 7. Competing transitions

This allows for control over the flow of the net, as tokens are distributed across the net according to the outcomes of prior transitions. This simple semantics enables complex real-world processes to be simplified into these graphical representations without losing formal validity.⁵²

Mechanistic elements of complex processes can be abstracted into ‘sub-nets’ and then subsequently into transitions, thus mirroring the basic concept of abstraction in object-oriented programming.⁵³ Recursive abstraction of this kind allows for the modelling of even very complex systems, whilst simultaneously enabling the (legal) observer to drill down into the particulars of the code’s logic as required. It can be appreciated how these representations might be useful from an evidential perspective, should the code ultimately be contested in court.

⁵² This is necessarily a very basic overview of Petri nets. For a more detailed treatment of their application in the legal domain see *ibid.*, and for a theoretical background see either Petri’s thesis (Petri, *supra* n. 44) or T. Murata, ‘Petri Nets: Properties, Analysis and Applications’ (1989) 77 *Proceedings of the IEEE* 541.

⁵³ Bench-Capon and Coenen, *supra* n. 43, p. 72.

In prior work with Burkhard Schafer, I demonstrate the normative complexity that can be represented by Petri nets.⁵⁴ For example, the following net shows a model of Article 8 of the Data Protection Directive.⁵⁵

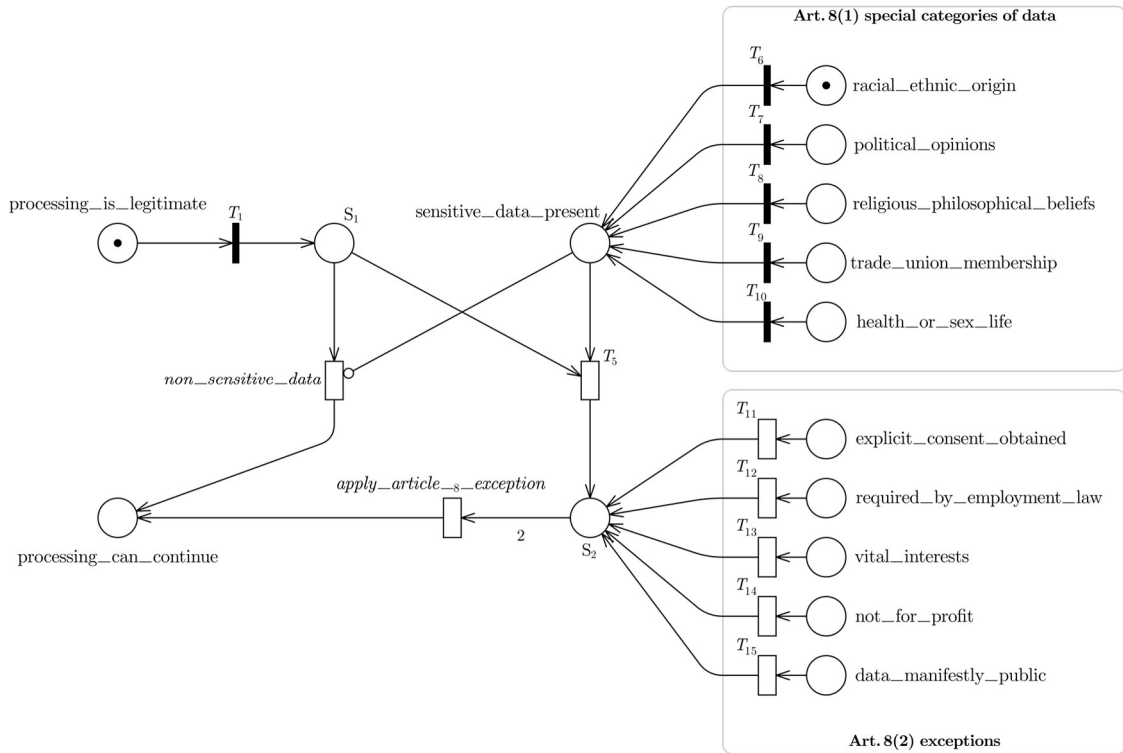


Figure 8. Petri net model of Art. 8 of the DPD

⁵⁴ Diver and Schafer, *supra* n. 29, p. 77 *et seq.*

⁵⁵ Although superseded by the GDPR (Art. 9) the principles demonstrated are still applicable.

This model of a legal provision can be ‘plugged in’ to an abstracted model of a code system, providing a way of communicating between the states generated, and required, by each:

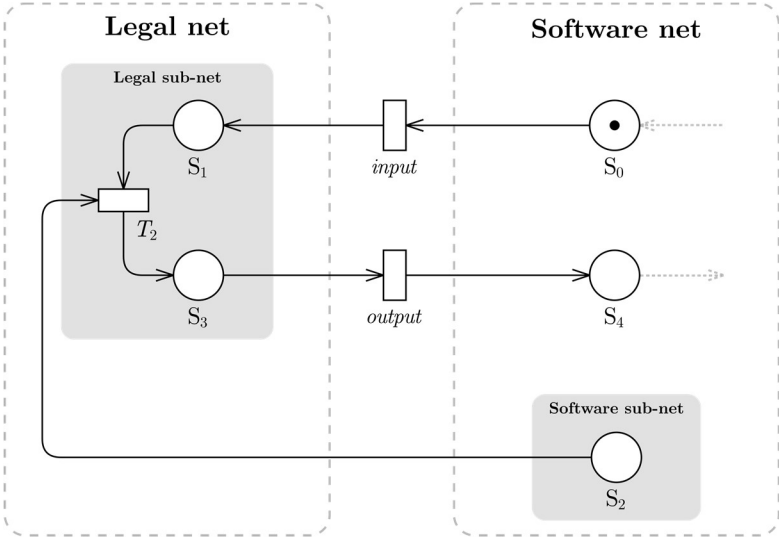


Figure 9. Communication between legal and software (code) models

As one can appreciate from this example, in order for the software net (right) to traverse between states S_0 and S_4 , it must first pass the test in the legal net (left), which is itself contingent on an input from a sub-element of the software net. The idea is that the legal net will only permit the code to ‘continue’ (i.e. reach state S_4) if there is some other condition in place that demonstrates the existence of a legally-required state. In the article from which the example is taken, the discussion of the model above envisaged a registration form that collects sensitive personal data (in that case, the end-user’s ethnic origin). This fact was represented by S_2 , which thus communicated to the legal net that a special category of data was being processed, which in turn would require one of the Article 8(2) exceptions to apply.⁵⁶

⁵⁶ For a more in-depth discussion, see Diver and Schafer, *supra* n. 29, p. 79, *et seq.*

Of course, these latter examples concern compliance with the substantive law. This is ‘compliance by design’, which was the subject of that article. Despite this difference in focus, the very existence of the model as a form of documentation demonstrates the second aspect of contestability that I am concerned with. The formal verification of the model begets compliance by design, but simultaneously the model itself *affords* contest by providing evidence of operation. One can appreciate the flexibility and abstraction of this approach, and its ability to model in a single representation more than one regulatory modality.

2.3.1.3 Behaviour-Driven Development

Another approach that can help to facilitate the aspect of contestability I am here concerned with is known as Behaviour-Driven Development (‘BDD’). BDD combines aspects of the agile methodology with concrete tools that are used within the IDE. Like the Petri net, it facilitates isomorphism between code and a representation that is intelligible to non-technologists. In this case, that representation is a natural language textual description of the behaviour of the system (hence the title). Dan North, the originator of BDD, describes it as an ‘outside-in’ methodology that starts from a set of desired outcomes and evolves towards the code features that can achieve them.⁵⁷ Although usually viewed from the perspective of bridging the domains of business requirements and code development, the isomorphism of the approach means it can operate as a post hoc evidentiary mechanism as much as an ex ante design specification mechanism.

⁵⁷ Dan North, ‘What’s in a Story?’ (*Dan North & Associates*, 11 February 2007) <<https://dannorth.net/whats-in-a-story/>>.

BDD uses natural language⁵⁸ templates for defining code features. These are combined with IDE tools that generate both the framework of code functions from those specifications and the ‘unit tests’ (small, modular tests for individual elements of code) that can verify that they behave as expected. Code features are defined using natural language, making them de facto intelligible for evidentiary purposes (and indeed, feasibly, for end-users). As mentioned, features in BDD are defined using a natural language template. For example, this feature specification describes an online shopping basket:

```
Feature: Online shop basket
  In order to buy products
  As a customer
  I need to be able to put interesting products into a basket
Rules:
  Delivery for basket under £10 is £3
Scenario: Buying a single product under £10
  Given there is a "Product X", which costs £5
  When I add the "Product X" to the basket
  Then I should have 1 product in the basket
  And the overall basket price should be £859
```

The keywords specified in the template (‘feature’, ‘scenario’, ‘given’, ‘when’, and ‘then’) are parsed by the IDE to generate the necessary code functions at a level of granularity that encourages the designer to write separate chunks of code that implement discrete functions. This in turn facilitates more modular testing and verification. For example:

⁵⁸ Known as a ‘ubiquitous language’, or a ‘business readable domain specific language’. See Martin Fowler, ‘Business Readable Domain Specific Language’ (*martinfowler.com*, 15 December 2008) <<https://martinfowler.com/bliki/BusinessReadableDSL.html>>.

⁵⁹ This and the next example are adapted from ‘Behat Documentation’ (*Behat*) <http://docs.behat.org/en/latest/quick_start.html>. Behat is a set of tools for implementing BDD in the PHP programming language.

```

/**
 * @Given there is a(n) :arg1, which costs £:arg2
 */
public function thereIsAWhichCostsPounds($arg1, $arg2) { }

```

Without going into the detail of how exactly this works, one can see how the function name ('thereIsAWhichCostsPounds') has been automatically generated from the natural language 'Given' line in the feature description. Within the curly parentheses – { and } – the designer writes the code that corresponds to that precise element of functionality (and nothing else). The small number of arguments (\$arg1 and \$arg2, respectively the product and its cost) means she is constrained in the amount of functionality the code she writes within this function can reasonably be written to perform. Furthermore, titling the functions in this way tends towards thinking in terms of discrete functionalities, which is ideal for modularity (recall the discussion on 'tussle' in the previous chapter), and therefore for testing. The ultimate goal is to achieve 'living documentation' that is intelligible by non-developers but is simultaneously isomorphic with the underlying instrumental code.

2.3.1.4 Conclusion

The above examples demonstrate two ways in which the evidentiary aspects of contestability can be facilitated partly in the IDE, maintaining an isomorphic connection between the underlying code and its documentation. Although the designer is ultimately still free to write the code as she sees fit (i.e. she can ignore these approaches, or override them), they nevertheless suggest one way in which the PoP, as represented in the IDE, can encourage the evidential aspect of the affordance of contestability. Furthermore, although optional at present, there is no principled reason why such approaches cannot be integrated more deeply into the process of IDE-based development.

3. Conclusion

This chapter discussed ‘agile’, one of the dominant code design methodologies in use today, setting out how the iterative process of digisprudential legitimation might conceivably fit within it. Drawing on Hart’s theory of primary and secondary rules mentioned earlier in the thesis, the discussion above has strengthened the parallel between constitutional and parliamentary law-making on the one hand, and the programmer of the programmer (represented *inter alia* in the affordances of the integrated development environment) and the product designer on the other. By considering what the (secondary) affordances of the design environment are and ought to be, it then becomes possible to conceive of the constitutional binding of the (primary) technological normativity that ultimately makes it into the finished product. To that end, I discussed the affordances of IDEs, and how these might themselves be tailored to fit this schema. With respect to the digisprudential affordance of contestability, I discussed two candidate mechanisms that can help to satisfy legal evidentiary requirements from the perspective of the courts – the latter being as important an aspect of contestability as the affordance, to the end-user, of the opportunity to contest the code.

Chapter 7

Conclusion

When technologies are always influencing human actions, we had better try to give this influence a desirable form.¹

1. Introduction

This multidisciplinary thesis has sought to bring together the practical question of how code regulates with a legal-theoretical view of what constitutes legitimate regulation. It has contributed both a descriptive legal-theoretical analysis, culminating in the concept of *computational legalism*, and a normative framework – digisprudence – that synthesises the latter with design theory in order to suggest a set of affordances that, when present, legitimise the code at a ‘constitutional’ level that lies below considerations of its commercial purposes or whether or not it complies with the requirements of substantive law. The digisprudential affordances therefore serve as a guide for the production of legitimate code. The thesis has also considered practical elements of the software development process, and how elements of the digisprudential framework can be employed there.

This final chapter concludes by discussing the relevance of the research carried out, before considering each of the research questions that were set out in Chapter 1 and how the ensuing analysis has answered them. It then goes on to suggest some avenues for potential further research that have emerged in the course of carrying out this work.

¹ P.-P. Verbeek, ‘Materializing Morality: Design Ethics and Technological Mediation’ (2006) 31 *Science, Technology, & Human Values* 361, p. 370.

2. The relevance of the thesis

I highlighted in Chapter 1 the tension between law as the paradigmatic normative order on the one hand, and code as an alternative order on the other. This speaks to fundamental questions of law and of normativity – what it is for an a-legal order to arise in parallel with (or even to supplant) democratically-legitimated law, particularly when that alternative order is commercially-motivated and benefits from the ‘legalistic’ characteristics of ruleishness, opacity, immediacy, immutability, and pervasiveness. When we fail to enquire as to the processes by which private code-based normativity is created and imposed, the result is a situation that is deeply problematic on two fronts, each of which compounds the other: we have technical rules which are by their very nature opaque and instrumental, created through commercial processes that lack democratic incentives, ratification, and oversight. The implications are profound, particularly given the ever-increasing role of code in ordering our social, political, and economic lives.

It is clear from the literature that this is a fundamental but under-studied problem. While the regulation of code by law is a significant field, it fails adequately to account for the myriad ways in which designers impose normativity in practice, perhaps without awareness, and likely without full cognisance of the substantive law which notionally they should be applying to and through their practices. The assumption is also that more law will result in better code. I have argued that the translation of legal text into code – assuming the designer is aware of the text in the first place – is problematic, and so there are inevitably gaps between what the law expects and what the code actually does. Appealing for *more* law may help close some of these gaps, but it is unlikely to solve the problem at a foundational level, not least because ever-more complex and precise sets of textual rules militate against

compliance.² In the absence of a more computationally-friendly form of legislation,³ then, designers need – in light of the characteristics of computational legalism – to be guided in their creation of code that is not necessarily ‘legal’ *per se* (although of course they should in any case be aiming for compliance by design) but whose design embodies constitutional protections that minimise the possibility of substantive illegality and facilitate judicial action should such illegality be found.

As we saw in Chapter 4, the legal literature on normative criteria for code is very small indeed, and while some legal scholars have either argued for greater engagement with other disciplines (particularly design studies, and science and technology studies), or have indeed done so themselves,⁴ my impression is that in the main there is, ironically, an element of the legalistic perspective operating, whereby lawyers are perhaps less willing to look outside the boundaries and the conceptual lenses of their discipline in order to engage with what lies beyond.⁵ This is unfortunate because it is not possible fully to understand the alternative normative order of code by observing it only through a legal lens,⁶ far less is it possible to think productively about how to solve the problems, discussed in this thesis, that it raises.

3. Research questions and contributions

Chapter 1 gave a broad overview of the thesis’ narrative, its primary theoretical sources, and the research questions that the analysis has aimed to answer. This section surveys

² C. Reed, ‘How to Make Bad Law: Lessons from Cyberspace’ (2010) 73 *The Modern Law Review* 903, p. 904 *et seq.*; J.N. Shklar, *Legalism* (Harvard University Press, 1964) p. 2.

³ I consider this possibility in the section on further research, *infra*.

⁴ We saw the primary examples of this in Chapter 4, section 3 (Normative criteria for code-making).

⁵ Cf. S. Gutwirth, P. De Hert and L. De Sutter, ‘The Trouble with Technology Regulation: Why Lessig’s “Optimal Mix” Will Not Work’ in R. Brownsword and K. Yeung (eds.), *Regulating technologies: legal futures, regulatory frames and technological fixes* (Oxford ; Portland, Or: Hart, 2008).

⁶ J.E. Cohen, *Configuring the Networked Self: Law, Code, and the Play of Everyday Practice* (Connecticut: Yale University Press, 2012) p. 27.

the questions and considers how the findings of Chapters 2-6 have contributed to answering them.

3.1 Research question 1: How does code in fact regulate end-user behaviour?

This question was dealt with in Chapter 2, which alongside Chapter 3 constitutes the descriptive element of the thesis. As alluded to above, this is a question that cannot be answered from within the legal discipline alone – the phrase ‘in fact’ in the question alludes to the need to consider what code *actually does*, a question that cannot be answered by law alone. Following the work of scholars such as van den Berg, Leenes, and Hildebrandt,⁷ the chapter engages directly with design theory in order to set out the ways in which code architectures impose technological normativity that enables and constrains what it is possible for end-users to do. The theories of affordance⁸ and inscription⁹ are particularly relevant, speaking to the conscious ways in which designers shape the interfaces and ‘stories’ of their artefacts in order to affect what behavioural possibilities end-users are given. The analysis demonstrates the sheer power of designers to enable and constrain behaviour, and even to define reality for the end-user.¹⁰ That power is direct and purposive in a way that the text-bound law can never be, rendering the latter but ‘a paper dragon in the age of the “digital tsunami”’.¹¹ In this way, the chapter showed the extent to which, as a regulator of behaviour, code is indeed ‘more’ than law.

⁷ B. van den Berg and R.E. Leenes, ‘Abort, Retry, Fail: Scoping Techno-Regulation and Other Techno-Effects’ in M. Hildebrandt and J. Gaakeer (eds.), *Human Law and Computer Law: Comparative Perspectives* (Dordrecht: Springer Netherlands, 2013); M. Hildebrandt, ‘Legal and Technological Normativity: More (and Less) than Twin Sisters’ (2008) 12 *Techné: Research in Philosophy and Technology* 169.

⁸ Chapter 2, section 2 (Affordance).

⁹ *Ibid.*, section 3.4 (Code mediating action).

¹⁰ *Ibid.*, section 3 (Infusing code with normativity).

¹¹ M. Hildebrandt and B.-J. Koops, ‘The Challenges of Ambient Law and Legal Protection in the Profiling Era’ (2010) 73 *The Modern Law Review* 428, p. 440.

3.2 Research question 2: To what extent are the characteristics of legalistic illegitimacy reflected in the normativity of code?

Taken together, Chapters 2 and 3 can be viewed as the first significant multidisciplinary contribution of the thesis. Where Chapter 2 uses design theory to analyse the characteristics of code, Chapter 3 considers it through the legal-theoretical lens of *legalism*. This analysis proceeds from the observation that the characteristics of code are strikingly similar to those of legalism, and particularly its stronger variant, discussed in the legal-theoretical literature¹² and in particular depth in the work of Wintgens.¹³ The analysis considers the characteristics of legalism and how they are exhibited analogously in code, the overarching observation of the thesis being that normative approaches that exist to mitigate legalism in the legal-institutional world (particularly Wintgens' own theory of *legisprudence*) might be susceptible to application in an a-legal context that exhibits similar characteristics. As Chapter 4 details, legisprudence is concerned with the creation of legitimate legal normativity, based on rules being constituted according to certain formal requirements rather than simply as arbitrary exercises of sovereign power. The focus of the latter on the application of rules without enquiry as to how they were made is analogous to the way in which technological normativity is in many cases simply a given of the code's design: end-users can simply 'take or leave' the code as they find it, in much the same way as legalism views laws as 'just there'.

The computational form of legalism, however, goes further than the legal-theoretical concept. Not only are computational rules 'just there', they are also in most cases hidden by the *opacity* of code, removing the possibility – still present under orthodox legalism – of choosing to reinterpret or disobey them. This absence of interpretation is further embodied in code's *ruleishness*: first, it only ever operates under

¹² See the discussion in Chapter 3, section 2 (What is legalism?).

¹³ Particularly in ch. 5 of his *Legisprudence: Practical Reason in Legislation* (Surrey: Routledge, 2012).

precisely the conditions that have been laid down in advance, no matter how closely extant conditions might be to the code's requirements; second, it *always* operates when those conditions exist; and third, its consequences can only ever be those that are predetermined in the code. These characteristics are in turn made more problematic by code's *immediacy* (the effects are executed in many cases without the end-user having the opportunity to comprehend what is happening or to arrest that execution) and by code's *immutability* (once released, code is not readily susceptible to its normativity being altered, unless this possibility is consciously anticipated in its design). The *pervasiveness* of code as a regulator amplifies these characteristics, when their aggregation and proliferation are taken into account. The analysis connects these characteristics with Wintgens' discussion of the 'veil of sovereignty', which views the exercise of legislative power as a mysterious thing, the province of politics rather than law.¹⁴ This central element of legalism is in evidence in the computational context where the 'sovereignty' of the designer is veiled by market orthodoxy, the inherent opacity of code, and the unwillingness of lawyers to look at the processes of production (cf. legislation¹⁵).

The result of all these characteristics is a *sui generis* form of legalism – what I call computational legalism – that is stronger even than that envisaged by Wintgens (and others).¹⁶ In this way, those elements that are present when law-making is done *well* are conspicuously absent in the context of code as a normative enterprise. Expressed another way, code is 'less' than (good) law.

Chapter 3 demonstrates one way in which rule-making in the code sphere can be analysed from the perspective of rule-making in the legal sphere. There are various ways this might be done, but in considering the issue from the perspective of legalism,

¹⁴ See Chapter 3, section 3.5 (The veiling of code's production).

¹⁵ L. Wintgens, 'Legisprudence as a New Theory of Legislation' (2006) 19 *Ratio Juris* 1, p. 1 ('The way law is created through the process of legislation does not appear on the screen of the legal theorist.').

¹⁶ See *ibid.*, section 3 (Computational legalism).

I set the stage for importing from the legal sphere to the computational sphere the means of combating legalism.

3.3 Research question 3: Can mechanisms for designing legitimate *legal* normativity be adopted to ensure the design of legitimate *technological* normativity?

Chapter 4 considers in greater detail two influential normative theories of law-making, namely Fuller's principles of legality and, as previously discussed, Wintgens' legisprudence.¹⁷ It discusses the concept of legality (in contrast to legalism, its perversion), and the important distinction between input and output legitimacy,¹⁸ before summarising Fuller's and Wintgens' theories.¹⁹ The latter part of the chapter then shifts to consider the existing literature on normative criteria for the use of code as a regulator, observing the important differences between the application of procedural (input) and substantive (output) criteria.²⁰ The literature considers only to a limited extent the important temporal question of the point in the design process at which the desirability or legitimacy of a code's normativity should be considered. The argument that production is the appropriate point to focus on (cf. operation) is made by Goldoni²¹ and Koops,²² but the literature does not develop this argument significantly, nor does it consider what this might mean practically from a design perspective. Questions of input legitimacy are important because computational

¹⁷ Chapter 4, section 2 (Normative criteria for law-making: the aspirations of legality).

¹⁸ *Ibid.*, section 2.1 (Input and output legitimacy in law).

¹⁹ *Ibid.*, sections 2.2 (Fuller's principles of legality) and 2.3 (Wintgens' legisprudence), respectively.

²⁰ *Ibid.*, section 3 (Normative criteria for code-making).

²¹ M. Goldoni, 'The Politics of Code as Law: Toward Input Reasons' in J. Reichel and A.S. Lind (eds.), *Freedom of Expression, the Internet and Democracy* (Leiden: Brill, 2015).

²² B.-J. Koops, 'Criteria for Normative Technology: The Acceptability of "Code as Law" in Light of Democratic and Constitutional Values' in R. Brownsword and K. Yeung (eds.), *Regulating Technologies: Legal Futures, Regulatory Frames and Technological Fixes* (Oxford; Portland, Or: Hart, 2008).

legalism tends towards fixed, opaque, and brittle constellations of normativity; the appropriate locus of attention is therefore the stage of production, in order to ensure that those constellations are legitimate *ab initio*. This brings us back to Wintgens' and (elements of) Fuller's theories, concerned as they are with the crafting – the design – of legitimate laws, as distinct from what happens in the ex post adjudicative processes that apply those laws. The analysis also considers the distinction between substantive and formal legitimacy – akin to 'thick' and 'thin' conceptions of the rule of law – and what these mean from the perspective of providing constitutional principles for the guidance of design.²³

Having set out the requirements of the normative frameworks for law-making, and reviewed the criteria specified by legal scholars as to the legitimate use of code, the stage is then set for a synthesis in Chapter 5.

3.4 Research question 4: What might such mechanisms look like in the design context?

Chapter 5 synthesises the analyses in Chapters 2-4 into a framework of affordances that can translate the requirements for the legitimisation of legal normativity into a conceptual apparatus appropriate for their application through the design of code. This critical and normative theory of *digisprudence* (legitimised code) is the most significant contribution of the thesis. The analysis maps the Fullerian and legisprudential principles onto the characteristics of computational legalism that they are conceptually linked with, and proposes a set of affordances that can embody the goals of those principles and thereby ameliorate the characteristics of computational legalism. These can be used both as a mechanism for critiquing an existing design and as a normative guide for what features ought to be built into code during the production process.

²³ Chapter 4, sections 2.1 (Input and output legitimacy in law) and 3.1 (Input and output legitimacy in code), respectively.

This analysis is augmented in Chapter 6 by a discussion of the code production process and a practical consideration of how the ‘constitutional’ role of the *programmer of the programmer* (‘PoP’) might assist in achieving the aims of digisprudence. Adapting Hart’s theory of primary and secondary rules,²⁴ we can conceptualise the PoP as a constitutional actor who, via technological normativity, enables and constrains the forms of normativity the product designer can in turn create.²⁵ This was especially relevant to the affordance of contestability, the legal-institutional element of which requires evidential standards that (legitimate) code ought to meet. The chapter considers some mechanisms by which the provision of such evidence can be anticipated *ex ante* during the design process.

4. Further research

In the course of researching this thesis I have identified various potential avenues for further work. The thesis covers a wide range of issues and there are therefore many possibilities; this section summarises three of the most interesting and salient potentials.

4.1 Drawing law-making and code-making closer together – the future of compliance by design

I have mentioned on a few occasions that this thesis is not about compliance with substantive law, but that such compliance is nevertheless a separate and important goal. As I set out in Chapter 1, the thesis consciously takes the position that code is a *parallel* normative order, operating separately from the law. Because of this, the creation of normativity in that context requires to be legitimate(d), hence the digisprudential

²⁴ H.L.A. Hart, *The Concept of Law* (2nd edn., Oxford: Clarendon Press, 1994) pp. 91–96.

²⁵ Chapter 6, section 2.1 (From primary and secondary rules to primary and secondary (dis)affordances and inscriptions).

framework. This raises the question however of how the parallel orders might in future be brought more closely together. As code is increasingly the medium upon which other parts of social, political, and commercial life are built, it seems reasonable to assume that it will become the target of more and more law. However, laws that fail properly to be embodied in the code that they target tacitly undermine law-making as an expression of democratic will. This is the problem I referred to early in Chapter 5, where the belief in the validity of the rule that animates the use of code transfers into a belief that the resulting code is itself valid. The limitations of this belief and the resulting negative effects might come to have significant negative effects as code proliferates every further into society.

Continuing in the legisprudential spirit, then, one avenue for further research is the consideration of how legislators might better couch the terms of the norms they promulgate in order that they are more susceptible to application in the design environment, specifically using the design theories I discussed in Chapter 2. Much of the work that aims to bridge the gap between code and law views legislative norms as a passive source of rules to be grappled with by various computational processes.²⁶ It is therefore in a sense legalistic in outlook, because it does not directly question the practice of norm creation itself. While the general question of how to develop legal norms that are appropriate for application in the technological context is not new,²⁷ I do not believe there has been any systematic analysis of how the substantive content of those laws (i.e. the primary rules, in Hartian terms) might be expressed in the language of affordances, which might render them capable of more direct

²⁶ See, for example, P. Lippe, D.M. Katz and D. Jackson, 'Legal by Design: A New Paradigm for Handling Complexity in Banking Regulation and Elsewhere in Law' (2014) 93 *Oregon Law Review* 833; D. Oberle et al., 'Engineering Compliant Software: Advising Developers by Automating Legal Reasoning' (2012) 9 *SCRIPTed* 280; D.M. Katz, 'Quantitative Legal Prediction – or – How I Learned to Stop Worrying and Start Preparing for the Data Driven Future of the Legal Services Industry' (2012) 62 *Emory Law Journal* 909.

²⁷ Reed, *supra* n. 2.

implementation by designers and manufacturers. Hildebrandt, for example, mentions in passing ‘detecting, configuring or designing affordances that are compatible with specific legal norms’,²⁸ but does not discuss the opposite concept of *laws* that are compatible with *affordances* (i.e. the language and theory of the latter). I have briefly considered this idea in other work, derived from this thesis.²⁹

A nascent literature is developing that views the goal of compliance from the perspective of informatics and how designers can be involved in the implementation of regulatory goals.³⁰ This may be a useful multidisciplinary starting point for future research in this area, identifying in greater detail the kinds of practices that legislators ought to have in mind when they design legal norms, and providing qualitative empirical evidence for what practices are effective.³¹ In any event, we have seen how the concepts of (dis)affordance and inscription are simultaneously both concrete and technology-agnostic; the digisprudential affordances are specific enough to identify the presence or absence of defined capabilities, but abstract enough to apply across a wide spectrum of technologies. This might suggest affordance theory as a good candidate for expressing a wide range of substantive legal requirements in terms that lie closer to the actual practices of those expected to comply with them. The expressive texture of such an approach might be enhanced by the work of theorists like Davis and

²⁸ M. Hildebrandt, *Smart Technologies and the End(s) of Law: Novel Entanglements of Law and Technology* (London: Edward Elgar Publishing, 2015) p. 218.

²⁹ L. Diver, ‘Law as a User: Design, Affordance, and the Technological Mediation of Norms’ (2018) 15 *SCRIPTed* 4, p. 39f.

³⁰ E.g. E. Luger and M. Golembewski, ‘Towards Fostering Compliance by Design; Drawing Designers into the Regulatory Frame’ in M. Taddeo and L. Floridi (eds.), *The Responsibilities of Online Service Providers* (Springer International Publishing, 2017); L. Urquhart and T. Rodden, ‘New Directions in Information Technology Law: Learning from Human–Computer Interaction’ (2017) 31 *International Review of Law, Computers & Technology* 150.

³¹ See for example E. Luger et al., ‘Playing the Legal Card: Using Ideation Cards to Raise Data Protection Issues Within the Design Process’, *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (New York, NY, USA: ACM, 2015).

Chouinard, couching affordances in deontological terms apt for legal application,³² Hartson, who classifies affordances according to their cognitive, physical, sensory, and functional characteristics,³³ or even Ihde's various relationships of technological mediation.³⁴

Building on the 'thin' concept of digisprudential legitimacy, then, we might envisage legislators conceptualising the *substance* of traditional legislative instruments in terms of (dis)affordance and inscription. Indeed, were legislative practice to develop in this direction it would be only a small step to 'convert' digisprudence into this 'design-sensitive' law, thus narrowing the gap between law and code even further. One might imagine a 'Code Legitimacy Act', perhaps part of consumer protection law, requiring all technologies to embody such standards by default, with the technology-facing aspects of those bodies of substantive law that require 'by design' approaches (e.g. copyright or data protection) operating as *leges speciales* of that Act's techno-constitutional regime. Indeed, as mentioned above, it is the current absence of this kind of design-sensitive law-making that necessitates the analysis and the practical measures put forward in this thesis.

4.2 Design and private law

Another avenue for research that I have been unable to explore in the thesis is the question of private law-making, i.e. contract, and its relationship with design (recall the discussion in Chapter 1 of Relationship (c) and the commixture of law and code³⁵). The observation that online contracting is a form of non-state legal ordering is not a

³² J.L. Davis and J.B. Chouinard, 'Theorizing Affordances: From Request to Refuse' [2017] *Bulletin of Science, Technology & Society*.

³³ R. Hartson, 'Cognitive, Physical, Sensory, and Functional Affordances in Interaction Design' (2003) 22 *Behaviour & Information Technology* 315.

³⁴ D. Ihde, *Technology and the Lifeworld: From Garden to Earth* (Indiana University Press, 1990) ch. 5.

³⁵ Chapter 1, sections 1.4.1 (Normative relationships online) and 1.4.1.1 (The commixture of law and code), respectively.

new one,³⁶ but some of the observations about design, and about code as a regulator, have implications within that context that might be worth exploring. Building on the discourse around ‘clickwrap’ licensing from the mid to late 2000s, Hartzog has mooted the idea of design elements *per se* being viewed as terms of privately-ordered contracts.³⁷ This very clearly intertwines design practice with legal practice, and is something that design theory might helpfully inform, in a similar fashion to the role for legislators described above. Although Hartzog does not employ such theories, their role is implicit in his analysis, where he argues that specific features of design (such as Facebook’s privacy settings, as defined for the purposes of the end-user by the affordances of its interface) ought to be deemed to be part of the contract between the end-user and a website’s operator.³⁸

As with contract terms that are deemed unfair or unconscionable, we can envisage affordances that ought to be deemed illegitimate terms of such a design-based contract. In his recent book Hartzog discusses ‘promissory design’, or ‘the implicit (and sometimes even explicit) promises embedded in and expressed through design.’³⁹ He also discusses affordance theory, demonstrating the emergence of this new literature.⁴⁰ He questions the apparent disparity between the imposition of liability flowing from textual contract terms and the comparative lack of accountability for promises expressed via design. Given the interface of the website is frequently the only medium by which end-users communicate with online providers, their expression of preferences

³⁶ See for example L. Belli and J. Venturini, ‘Private Ordering and the Rise of Terms of Service as Cyber-Regulation’ [2016] *Internet Policy Review*; W. Schulz and K. Dankert, “Governance by Things” as a Challenge to Regulation by Law’ (2016) 5 *Internet Policy Review*; M.J. Radin, ‘Regulation by Contract, Regulation by Machine’ (2004) 160 *Journal of Institutional and Theoretical Economics (JITE)* 142

³⁷ W. Hartzog, ‘Website Design as Contract’ (2010) 60 *American University Law Review* 1635.

³⁸ *Ibid.*, p. 1650 *et seq.*

³⁹ W. Hartzog, *Privacy’s Blueprint: The Battle to Control the Design of New Technologies* (Cambridge, Mass: Harvard University Press, 2018) p. 169 *et seq.*

⁴⁰ *Ibid.*, p. 31 *et seq.*

through the configuration of website settings (i.e. by configuring its affordances) perhaps ought to constitute a form of agreement. From the perspective of the *production* of those interfaces, the provision of a setting by a provider perhaps ought in turn to imply a legal duty on her to ensure the background code operates in accordance with (a reasonable interpretation of) the technical state the setting appears to the end-user to create.⁴¹ The role that design plays in end-users' understanding of the products and services they use suggests the potential to explore further the role that design plays in foundational legal concepts of negotiation, consensus, and performance. This is perhaps especially important in areas where the end-user's understanding of the affordances of the interface has a bearing on their fundamental rights, for example the provision of consent to the processing of personal data. It is also an avenue of research that could benefit from analysis of how contractual and promissory requirements differ between jurisdictions.⁴²

4.3 'Legitimacy Impact Assessment'

A growing area of research activity, particularly in the fields of privacy and data protection,⁴³ is impact assessment. These assessments aim to provide 'a systematic process for evaluating the potential effects of privacy of a project, initiative, or proposed system or scheme' and to assist in 'finding ways to mitigate or avoid any adverse effects'.⁴⁴ The UK Information Commissioner describes them as a way both to meet

⁴¹ Recall the mismatch between cookie setting interfaces and what appears to happen on the technical level. See the example in Chapter 5 at n. 118.

⁴² The Scots law of unilateral promise, for example, is unusual in its lack of the requirement of consideration or even action (cf. the English gratuitous contract). See Robert Black, '612. Gratuitous Contracts', *Stair Memorial Encyclopaedia* (Reissue, Edinburgh: Law Society of Scotland: Butterworths, 1999).

⁴³ The latter is required by Art. 35 of the GDPR under certain circumstances, including where new technologies are being deployed.

⁴⁴ D. Wright, 'Should Privacy Impact Assessments Be Mandatory?' (2011) 54 *Communications of the ACM* 121, p. 123.

data protection obligations and to ‘meet individuals’ expectations of privacy.’⁴⁵ The European Commission has published guidance on their use for the Internet of Things,⁴⁶ and they are a common feature of government procurement processes.⁴⁷ As Clarke observes, one interpretation of why impact assessments have emerged in recent years is as a reaction to the ‘increasingly privacy-invasive actions of governments and corporations’ in the late 20th century.⁴⁸ Because of these actions, ‘people want to know about organisations’ activities, and want to exercise control over their excesses’, with the privacy impact assessment demonstrating a ‘ceding by large organisations of some of the substantial power that they exercise over citizens’.⁴⁹ One can appreciate the overlap with the ethos of the current thesis. Key to privacy impact assessment processes are their focus on a single project or initiative, their anticipatory (ex ante) nature, their wide scope in considering forms of privacy and the actors whose interests might be affected, their desire to identify both problems *and* solutions, and their focus on organisational engagement.⁵⁰

From a broader perspective, in previous work with Lilian Edwards and Derek McAuley I have considered the idea of a ‘social impact assessment’,⁵¹ which aims to take wider considerations into account than just data protection, for example security,

⁴⁵ ‘Data Protection Impact Assessments’ (*Information Commissioner’s Office*, 20 December 2018) <<https://icoumbraco.azurewebsites.net/for-organisations/guide-to-data-protection/guide-to-law-enforcement-processing/accountability-and-governance/data-protection-impact-assessments/>>.

⁴⁶ European Commission, ‘Privacy and Data Protection Impact Assessment Framework for RFID Applications’ (European Commission, 2011) <<http://cordis.europa.eu/fp7/ict/enet/documents/rfid-pia-framework-final.pdf>>.

⁴⁷ L. Edwards, D. McAuley and L. Diver, ‘From Privacy Impact Assessment to Social Impact Assessment’, *Security and Privacy Workshops (SPW), 2016 IEEE* (IEEE, 2016) p. 54.

⁴⁸ R. Clarke, ‘Privacy Impact Assessment: Its Origins and Development’ (2009) 25 *Computer law & security review* 123, p. 124.

⁴⁹ *Ibid.*

⁵⁰ *Ibid.*, p. 124f.

⁵¹ Edwards, McAuley and Diver, *supra* n. 47.

transparency, sustainability, resilience, and interoperability.⁵² One can of course appreciate the overlap between transparency and the equivalent affordance under digisprudence.⁵³ There is scope for research into how the outcomes of the digisprudential framework might be integrated into a kind of ‘legitimacy impact assessment’, investigating how existing impact assessment methodologies might be adapted to formalise the process further than has been possible within the scope of the thesis. This research might also consider how such an enterprise would run alongside other substantive initiatives aimed at guiding the production of code, for example data protection by design,⁵⁴ value-sensitive design,⁵⁵ and participatory models for design.⁵⁶ As I have previously mentioned, these are not incompatible with digisprudence, it being a foundational ‘constitutional’ framework that runs beneath other considerations of the characteristics an artefact’s design should include. A consideration of the (appropriate) interplay between the baseline constitutional requirements of digisprudence and the higher-level substantive outputs of such approaches could be a useful starting point for further work.

4.4 Application to data-driven applications

An avenue for further research that would be of great contemporary relevance is consideration of how the digisprudential affordances might be applied to data-driven (as opposed to code-drive) applications. The same goals might be sought, but the

⁵² *Ibid.*, p. 56f.

⁵³ See Chapter 5, section 3.2 (Opacity).

⁵⁴ Mandated by Art. 25 of the GDPR.

⁵⁵ M. Flanagan, D.C. Howe and H. Nissenbaum, ‘Embodying Values in Technology: Theory and Practice’ in J. Van Den Hoven and J. Weckert (eds.), *Information Technology and Moral Philosophy* (New York: Cambridge University Press, 2008); B. Friedman, ‘Value-Sensitive Design’ (1996) 3 *interactions* 16.

⁵⁶ Verbeek, *supra* n. 1; J. Schot and A. Rip, ‘The Past and Future of Constructive Technology Assessment’ (1997) 54 *Technological Forecasting and Social Change* 251.

means of doing so in such applications would likely differ. The precise contours of the affordances would likely change, in particular in relation to the affordances of transparency, a topic that at the moment is attracting much attention from scholars working in the legal (especially data protection),⁵⁷ computer science,⁵⁸ and STS/critical studies fields.⁵⁹ The question of production is of great importance in that debate, given that the design of machine learning algorithms can set in motion (negative) path dependencies that are exacerbated when those algorithms are trained using data that reflect social biases.⁶⁰ Moving the present work in this direction might also involve further consideration of the parties to whom the affordances are directed, particularly the affordance of contestability. It may not be necessary for the end-user, at the level of the interface, to understand the intricacies of the algorithm with which she is

⁵⁷ See for example L. Edwards and M. Veale, 'Slave to the Algorithm: Why a Right to an Explanation Is Probably Not the Remedy You Are Looking For' (2017) 16 *Duke Law & Technology Review* 18; S. Wachter, B. Mittelstadt and L. Floridi, 'Why a Right to Explanation of Automated Decision-Making Does Not Exist in the General Data Protection Regulation' (2017) 7 *International Data Privacy Law* 76.

⁵⁸ See for example S. Wachter, B. Mittelstadt and C. Russell, 'Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR.(2017)' (2017) 31 *Harvard Journal of Law and Technology* 841; J.A. Kroll et al., 'Accountable Algorithms' (2016) 165 *University of Pennsylvania Law Review* 633; 'Principles for Accountable Algorithms and a Social Impact Statement for Algorithms :: FAT ML' <<https://www.fatml.org/resources/principles-for-accountable-algorithms>>.

⁵⁹ See for example B. Wagner, 'Ethics as an Escape from Regulation: From Ethics-Washing to Ethics-Shopping?' in I. Baraliuc et al. (eds.), *Being Profiled: Cogitas Ergo Sum - 10 Years of Profiling the European Citizen* (Amsterdam University Press, 2018); J. Danaher et al., 'Algorithmic Governance: Developing a Research Agenda through the Power of Collective Intelligence' (2017) 4 *Big Data & Society* <<https://doi.org/10.1177/2053951717726554>>; B.D. Mittelstadt et al., 'The Ethics of Algorithms: Mapping the Debate' (2016) 3 *Big Data & Society* 1; J. Danaher, 'The Threat of Algocracy: Reality, Resistance and Accommodation' (2016) 29 *Philosophy & Technology* 245; N. Diakopoulos, 'Algorithmic Accountability' (2015) 3 *Digital Journalism* 398.

⁶⁰ See for example J. Angwin et al., 'Machine Bias' (*ProPublica*, 23 May 2016) <<https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>>.

interacting,⁶¹ but such information ought to be afforded to e.g. the courts or other overseeing institutions. Structuring research on algorithms according to the legal-theoretical frame I have developed might bring some useful foundational insights to bear.

5. Conclusion

This chapter has summarised the contribution of the thesis and has considered avenues for potential future research. In answering the research questions the thesis has contributed to knowledge in its use of multidisciplinary literature to describe both the reality and the legal-theoretical implications of code as a regulator. It has also contributed a critical and practical theory of digisprudence, developing what is currently a very minimal literature bridging the gap between legal theory and concrete design practice.

The approach of the thesis has been to identify fundamental design characteristics that ought always to be reflected in code, in order not only that end-users are provided with mechanisms that enable them to resist its heteronomy, but also so that the courts, as the arbiter of last resort, are at all times able to exercise their function as guardians of the rule of law.

⁶¹ A.D. Selbst and J. Powles, 'Meaningful Information and the Right to Explanation' (2017) 7 *International Data Privacy Law* 233.

Bibliography

All URLs were accessed on 21 April 2019.

1. Primary sources

Black R., '612. Gratuitous Contracts', *Stair Memorial Encyclopaedia* (Reissue, Edinburgh: Law Society of Scotland: Butterworths, 1999)

Microsoft Corp. v Commission of the European Communities (Case T-201/04)

Regulation on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 85/46/EC (General Data Protection Regulation) 2016

'Request for Comments (RFC) 2109 HTTP State Management Mechanism' (Internet Engineering Task Force, 1997)

'Request for Comments (RFC) 2965 HTTP State Management Mechanism (obsoleting RFC2109)' (Internet Engineering Task Force, 2000)

'Request for Comments (RFC) 6265 HTTP State Management Mechanism (obsoleting RFC 2965)' (Internet Engineering Task Force, 2011)

2. Books

Bratton B.H., *The Stack: On Software and Sovereignty* (Cambridge, Massachusetts: MIT Press, 2016)

Cohen J.E., *Configuring the Networked Self: Law, Code, and the Play of Everyday Practice* (New Haven, Connecticut: Yale University Press, 2012)

Daly A., *Private Power, Online Information Flows, and EU Law: Mind the Gap* (Oxford, United Kingdom; Portland, Oregon: Hart, 2016)

De Filippi P. and A. Wright, *Blockchain and the Law: The Rule of Code* (Cambridge, Massachusetts: Harvard University Press, 2018)

Dennett D.C., *From Bacteria to Bach and Back: The Evolution of Minds* (London: Penguin Books, 2018)

DiSalvo C., *Adversarial Design* (Cambridge, Massachusetts: MIT Press, 2012)

Dworkin R., *Law's Empire* (Cambridge, Massachusetts: Belknap Press, 1986)

Floridi L., *The Ethics of Information* (Oxford University Press, 2013)

- , *The Fourth Revolution: How the Infosphere Is Reshaping Human Reality* (Oxford University Press, 2014)
- Fogg B.J., *Persuasive Technology: Using Computers to Change What We Think and Do* (Amsterdam; Boston: Morgan Kaufmann Publishers, 2003)
- Foucault M., *The Birth of Biopolitics: Lectures at the Collège de France, 1978–79* (Basingstoke; New York: Palgrave Macmillan, 2008)
- Fuller L.L., *The Morality of Law* (New Haven, Connecticut: Yale University Press, 1977)
- Galloway A.R., *Protocol: How Control Exists after Decentralization* (Cambridge, Massachusetts: MIT Press, 2004)
- Gibson J.J., *The Ecological Approach to Visual Perception* (Classic Edition, New York: Psychology Press, 2015)
- , *The Senses Considered as Perceptual Systems* (Boston, Massachusetts: Houghton Mifflin, 1966)
- Goldsmith J.L. and T. Wu, *Who Controls the Internet? Illusions of a Borderless World* (Oxford University Press, 2006)
- Golumbia D., *The Cultural Logic of Computation* (Cambridge, Massachusetts: Harvard University Press, 2009)
- Hart H.L.A., *The Concept of Law* (2nd edn., Oxford: Clarendon Press, 1994)
- Hartzog W., *Privacy's Blueprint: The Battle to Control the Design of New Technologies* (Cambridge, Massachusetts: Harvard University Press, 2018)
- Harvey D., *A Brief History of Neoliberalism* (Oxford University Press, 2005)
- Hildebrandt M., *Smart Technologies and the End(s) of Law: Novel Entanglements of Law and Technology* (London: Edward Elgar Publishing, 2015)
- Ihde D., *Technology and the Lifeworld: From Garden to Earth* (Indiana University Press, 1990)
- Latour B., *The Making of Law: An Ethnography of the Conseil d'Etat* (Revised ed., Cambridge, UK; Malden, Massachusetts: Polity, 2009)
- Lessig L., *Code: Version 2.0* (New York: Basic Books, 2006)
- Luhmann N., *Law as A Social System* (Fatima Kastner et al. eds., Klaus Ziegert tr.: Oxford University Press, 2004)
- MacCormick N., *Legal Reasoning and Legal Theory* (Oxford University Press, 1994)
- Moody G., *Rebel Code* (New York: Basic Books, 2002)
- Norman D.A., *The Design of Everyday Things* (Cambridge, Massachusetts; London, UK: MIT Press, 2013)
- Pasquale F., *The Black Box Society: The Secret Algorithms That Control Money and Information* (Cambridge, Massachusetts: Harvard University Press, 2015)

- Petri C.A., 'Kommunikation Mit Automaten' (Ph.D. thesis, University of Bonn 1962)
- Reed C. and A. Murray, *Rethinking the Jurisprudence of Cyberspace* (Cheltenham, UK: Edward Elgar Publishing, 2018)
- Scharpf F., *Governing in Europe: Effective and Democratic?* (Oxford University Press, 1999)
- Searle J.R., *The Construction of Social Reality* (New York: Free Press, 1995)
- Shelley M., *Frankenstein: A New Edition for Scientists and Engineers* (Ed Finn, David Guston and Jason Scott Robert eds, Cambridge, Massachusetts: MIT Press, 2017)
- Shklar J.N., *Legalism* (Cambridge, Massachusetts: Harvard University Press, 1964)
- Smith A., *The Theory of Moral Sentiments* (2nd edn., London: Millar, 1761)
- Vallor S., *Technology and the Virtues: A Philosophical Guide to a Future Worth Wanting* (Oxford University Press, 2016)
- Verbeek P.-P., *What Things Do: Philosophical Reflections on Technology, Agency, and Design* (Penn State Press, 2005)
- Weizenbaum J., *Computer Power and Human Reason: From Judgment to Calculation* (San Francisco: Freeman, 1976)
- Williams J., *Stand out of Our Light: Freedom and Resistance in the Attention Economy* (Cambridge University Press, 2018)
- Winner L., *Autonomous Technology: Technics-out-of-Control as a Theme in Political Thought* (2nd edn., Cambridge, Massachusetts: MIT Press, 1977)
- Wintgens L., *Legisprudence: A New Theoretical Approach to Legislation* (Oxford: Hart, 2002)
- , *Legisprudence: Practical Reason in Legislation* (Surrey: Routledge, 2012)
- Zittrain J., *The Future of the Internet and How To Stop It* (New Haven, Connecticut: Yale University Press, 2008)

3. Journal articles and book chapters

- Akrich M., 'The De-Description of Technical Objects' in W.E. Bijker and J. Law (eds.), *Shaping Technology/Building Society: Studies in Sociotechnical Change* (MIT Press, 1992)
- Asscher L., "Code" as Law: Using Fuller to Assess Code Rules' in E. Dommering and L. Asscher (eds.), *Coding Regulation: Essays on the Normative Role of Information Technology* (The Hague: TMC Asser Press, 2006)
- Bamberger K.A., 'Technologies of Compliance: Risk and Regulation in a Digital Age' (2010) 88 *Texas Law Review* 669

- Bańkowski Z., 'Bringing the Outside In: The Ethical Life of Legal Institutions' in T. Gizbert-Studnicki and J. Stelmach (eds.), *Law and Legal Cultures in the 21st Century* (Warsaw: Wolters Kluwer, 2007)
- , 'Don't Think About It: Legalism and Legality' in M.M. Karlsson, Ó. Páll Jónsson and EM Brynjarsdóttir (eds.), *Rechtstheorie: Zeitschrift für Logik, Methodenlehre, Kybernetik und Soziologie des Rechts* (Berlin: Duncker & Humblot, 1993)
- Bańkowski Z. and N. MacCormick, 'Legality without Legalism' in W. Krawietz et al. (eds.), *The Reasonable as Rational? On Legal Argumentation and Justification; Festschrift for Aulis Aarnio* (Berlin: Duncker & Humblot, 2000)
- Bańkowski Z. and B. Schafer, 'Double-Click Justice: Legalism in the Computer Age' (2007) 1 *Legisprudence* 31
- Bartoletti M. and L. Pompianu, 'An Empirical Analysis of Smart Contracts: Platforms, Applications, and Design Patterns' [2017] *arXiv preprint arXiv:1703.06322*
- Bayamloğlu E., 'On the Possibility of Normative Contestation of Automated Data-Driven Decisions' in I. Baraliuc et al. (eds.), *Being Profiled: Cogitas Ergo Sum - 10 Years of Profiling the European Citizen* (Amsterdam University Press, 2018)
- Bayamloğlu E. and R. Leenes, 'The "Rule of Law" Implications of Data-Driven Decision-Making: A Techno-Regulatory Perspective' [2018] *Law, Innovation and Technology* 1
- Belli L. and J. Venturini, 'Private Ordering and the Rise of Terms of Service as Cyber-Regulation' [2016] *Internet Policy Review*
- Bench-Capon T.J. and F.P. Coenen, 'Isomorphism and Legal Knowledge Based Systems' (1992) 1 *Artificial Intelligence and Law* 65
- Benoliel D., 'Technological Standards, Inc.: Rethinking Cyberspace Regulatory Epistemology' (2004) 92 *California Law Review* 1069
- Ben-Shahar O. and C.E. Schneider, 'The Failure of Mandated Disclosure' [2011] *University of Pennsylvania Law Review* 647
- Birnhack M. and N. Elkin-Koren, 'The Invisible Handshake: The Reemergence of the State in the Digital Environment' [2003] *Virginia Journal of Law & Technology*
- Black J., 'Critical Reflections on Regulation' (2002) 27 *Australian Journal of Legal Philosophy* 1
- Blichner L.C. and A. Molander, 'What Is Juridification?' (Centre for European Studies, University of Oslo, 2005)
- Boyle J., 'Foucault in Cyberspace: Surveillance, Sovereignty, and Hardwired Censors' (1997) 66 *University of Cincinnati Law Review* 177

- Brownsword R., 'What the World Needs Now: Techno-Regulation, Human Rights and Human Dignity' in R. Brownsword (ed.), *Global Governance and the Quest for Justice*, vol. 4 (Oxford: Hart Publishing, 2004)
- , 'Code, Control, and Choice: Why East Is East and West Is West' (2005) 25 *Legal Studies* 1
- , 'Lost in Translation: Legality, Regulatory Margins, and Technological Management' (2011) 26 *Berkeley Technology Law Journal* 1321
- , 'In the Year 2061: From Law to Technological Management' (2015) 7 *Law, Innovation and Technology* 1
- , 'Technological Management and the Rule of Law' (2016) 8 *Law, Innovation and Technology* 100
- Bruch M. et al., 'IDE 2.0: Collective Intelligence in Software Development', *Proceedings of the FSE/SDP workshop on Future of software engineering research - FoSER '10* (Santa Fe, New Mexico: ACM Press, 2010)
- Bucher T., 'A Technicity of Attention: How Software "Makes Sense"' (2012) 13 *Culture Machine*
- Burrell J., 'How the Machine "Thinks": Understanding Opacity in Machine Learning Algorithms' (2016) 3 *Big Data & Society*
- Caine K.E., 'Supporting Privacy by Preventing Misclosure', *CHI09 Extended Abstracts on Human Factors in Computing Systems* (ACM, 2009)
- Calo R., 'Code, Nudge, or Notice' (2013) 99 *Iowa Law Review* 773
- Campbell C.M., 'Legal Thought and Juristic Values' (1974) 1 *British Journal of Law and Society* 13
- Citron D.K., 'Open Code Governance' [2008] *University of Chicago Legal Forum* 355
- , 'Technological Due Process' (2008) 85 *Washington University Law Review* 1249
- Clack C.D., V.A. Bakshi, and L. Braine, 'Smart Contract Templates: Foundations, Design Landscape and Research Directions' [2017] *arXiv:1608.00771 [cs]*
- Clark D.D. et al., 'Tussle in Cyberspace: Defining Tomorrow's Internet' (2005) 13 *IEEE/ACM Transactions on Networking (ToN)* 462
- Clarke R., 'Privacy Impact Assessment: Its Origins and Development' (2009) 25 *Computer law & security review* 123
- Cohen J.E., 'The Regulatory State in the Information Age' (2016) 17 *Theoretical Inquiries in Law* 369

- Conti G. and E. Sobiesk, 'Malicious Interface Design: Exploiting the User', *Proceedings of the 19th International Conference on the World Wide Web* (Raleigh, North Carolina: ACM Press, 2010)
- Cranor L.F. and R.N. Wright, 'Influencing Software Usage' [1998] *arXiv:cs/9809018*
- Danaher J., 'The Threat of Algocracy: Reality, Resistance and Accommodation' (2016) 29 *Philosophy & Technology* 245
- , 'Algorithmic Governance: Developing a Research Agenda through the Power of Collective Intelligence' (2017) 4 *Big Data & Society*
- Davis J.L. and J.B. Chouinard, 'Theorizing Affordances: From Request to Refuse' [2017] *Bulletin of Science, Technology & Society*
- Diakopoulos N., 'Algorithmic Accountability' (2015) 3 *Digital Journalism* 398
- Diver L., 'Law as a User: Design, Affordance, and the Technological Mediation of Norms' (2018) 15 *SCRIPTed* 4
- Diver L. and B. Schafer, 'Opening the Black Box: Petri Nets and Privacy by Design' (2017) 31 *International Review of Law, Computers & Technology* 68
- Dizon M.A.C., 'From Regulating Technologies to Governing Society: Towards a Plural, Social and Interactive Conception of Law' in H.M. Morgan and R. Morris (eds.), *Moving Forward: Tradition and Transformation* (Cambridge Scholars Publishing, 2011)
- Dworkin R., 'Law as Interpretation' (1982) 9 *Critical Inquiry* 179
- Edwards L., 'Data Protection and E-Privacy: From Spam and Cookies to Big Data, Machine Learning and Profiling' in L. Edwards (ed.), *Law, Policy and the Internet* (Oxford, UK; Portland, OR: Hart, 2018)
- Edwards L., D. McAuley, and L. Diver, 'From Privacy Impact Assessment to Social Impact Assessment', *2016 IEEE Security and Privacy Workshops (SPW)* (San Jose, CA: IEEE, 2016)
- Edwards L. and M. Veale, 'Slave to the Algorithm: Why a Right to an Explanation Is Probably Not the Remedy You Are Looking For' (2017) 16 *Duke Law & Technology Review* 18
- Endicott T., 'Law Is Necessarily Vague' (2001) 7 *Legal Theory* 379
- Faraj S. and B. Azad, 'The Materiality of Technology: An Affordance Perspective' in P.M. Leonardi, B.A. Nardi and J. Kallinikos (eds.), *Materiality and Organizing: Social Interaction in a Technological World* (Oxford University Press, 2012)
- Flanagan M., D.C. Howe and H. Nissenbaum, 'Embodying Values in Technology: Theory and Practice' in J. van den Hoven and J. Weckert (eds.), *Information Technology and Moral Philosophy* (New York: Cambridge University Press, 2008)

- Floridi L., 'Soft Ethics, the Governance of the Digital and the General Data Protection Regulation' (2018) 376 *Phil. Trans. R. Soc. A* 20180081
- Fraser A. and R. Kitchin, 'Slow Computing' (Maynooth University, 2017) The Programmable City Working Paper Series 36
- Freiheit J. et al., 'Lexecute: Visualisation and Representation of Legal Procedures' (2006) 3 *Digital Evidence & Elec. Signature L. Rev.* 19
- Friedman B., 'Value-Sensitive Design' (1996) 3 *interactions* 16
- Friedman B. and D. Hendry, 'The Envisioning Cards: A Toolkit for Catalyzing Humanistic and Technical Imaginations', *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY: ACM, 2012)
- Fuller L.L., 'Positivism and Fidelity to Law: A Reply to Professor Hart' (1958) 71 *Harvard Law Review* 630
- Gaver W.W., J. Beaver and S. Benford, 'Ambiguity As a Resource for Design', *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY: ACM, 2003)
- Goldoni M., 'The Politics of Code as Law: Toward Input Reasons' in J. Reichel and A.S. Lind (eds.), *Freedom of Expression, the Internet and Democracy* (Leiden: Brill, 2015)
- Governatori G. and S. Sadiq, 'The Journey to Business Process Compliance' in J. Cardoso and W. Van der Aalst (eds.), *Handbook of Research on Business Process Modeling* (IGI Global, 2009)
- Grigg I., 'The Ricardian Contract', *Proceedings of the First IEEE International Workshop on Electronic Contracting* (IEEE, 2004)
- Grimmelmann J., 'Regulation by Software' (2005) 114 *The Yale Law Journal* 1719
- Gürses S., C. Troncoso and C. Diaz, 'Engineering Privacy by Design' (2011) 14 *Computers, Privacy & Data Protection*
- Gürses S. and J. van Hoboken, 'Privacy after the Agile Turn' in E. Selinger, J. Polonetsky and O. Tene (eds.), *The Cambridge Handbook of Consumer Privacy* (Cambridge University Press, 2018)
- Gutwirth S., P. De Hert and L. De Sutter, 'The Trouble with Technology Regulation: Why Lessig's "Optimal Mix" Will Not Work' in R. Brownsword and K. Yeung (eds.), *Regulating technologies: legal futures, regulatory frames and technological fixes* (Oxford; Portland, Or: Hart, 2008)
- Halderman J.A. and E.W. Felten, 'Lessons from the Sony CD DRM Episode', *15th USENIX Security Symposium* (USENIX Association, 2006)

- Hart H.L.A., 'Positivism and the Separation of Law and Morals' (1958) 71 *Harvard Law Review* 593
- Hartson R., 'Cognitive, Physical, Sensory, and Functional Affordances in Interaction Design' (2003) 22 *Behaviour & Information Technology* 315
- Hartzog W., 'Website Design as Contract' (2010) 60 *American University Law Review* 1635
- et al., 'Inefficiently Automated Law Enforcement' [2015] *Michigan State Law Review* 1763
- Hartzog W. and E. Selinger, 'The Internet of Heirlooms and Disposable Things' (2016) 17 *North Carolina Journal of Law & Technology* 581
- Hildebrandt M., 'A Vision of Ambient Law' in R. Brownsword and K. Yeung (eds.), *Regulating Technologies: legal futures, regulatory frames and technological fixes* (Oxford; Portland, Or: Hart, 2008)
- , 'Legal and Technological Normativity: More (and Less) than Twin Sisters' (2008) 12 *Techné: Research in Philosophy and Technology* 169
- , 'Legal Protection by Design: Objections and Refutations' (2011) 5 *Legisprudence* 223
- , 'Radbruch's Rechtsstaat and Schmitt's Legal Order: Legalism, Legality, and the Institution of Law' (2015) 2 *Critical Analysis of Law* 42
- , 'Law As an Affordance: The Devil Is in the Vanishing Point(s)' (2017) 4 *Critical Analysis of Law* 116
- , 'Algorithmic Regulation and the Rule of Law' (2018) 376 *Philosophical Transactions of the Royal Society A* 20170355
- Hildebrandt M. and B.-J. Koops, 'The Challenges of Ambient Law and Legal Protection in the Profiling Era' (2010) 73 *The Modern Law Review* 428
- Hoepman J.-H., 'Privacy Design Strategies', *ICT Systems Security and Privacy Protection* (Springer, 2014)
- Hoeren T. and S. Pinelli, 'Agile Programming – Introduction and Current Legal Challenges' (2018) 34 *Computer Law & Security Review* 1131
- Huldtgren A., 'Design for Values in ICT' in J. van den Hoven, P.E. Vermaas and I. van de Poel (eds.), *Handbook of Ethics, Values, and Technological Design* (Dordrecht: Springer Netherlands, 2014)
- Joerges B., 'Do Politics Have Artefacts?' [2016] *Social Studies of Science*
- Johnson D.R. and D.G. Post, 'Law and Borders - The Rise of Law in Cyberspace' (1995) 48 *Stanford Law Review* 1367
- Kant I., 'Metaphysical First Principles of the Doctrine of Right' in M. Gregor (tr.), *The Metaphysics of Morals* (Cambridge University Press, 1996)

- Katz D.M., 'Quantitative Legal Prediction – or – How I Learned to Stop Worrying and Start Preparing for the Data Driven Future of the Legal Services Industry' (2012) 62 *Emory Law Journal* 909
- Kesan J. and R. Shah, 'Setting Software Defaults: Perspectives from Law, Computer Science and Behavioral Economics' [2006] *University of Illinois Law and Economics Working Papers*
- Kesan J.P. and R.C. Shah, 'Setting Software Defaults: Perspectives from Law, Computer Science and Behavioral Economics' (2006) 82 *Notre Dame Law Review* 583
- Kiran A.H. and P.-P. Verbeek, 'Trusting Our Selves to Technology' (2010) 23 *Knowledge, Technology & Policy* 409
- Kittler F.A., 'Protected Mode' in J. Johnston (ed.), S. Harris (tr.), *Literature, Media, Information Systems: Essays* (Psychology Press, 1997)
- Kiviat T.I., 'Beyond Bitcoin: Issues in Regulating Blockchain Transactions' (2015) 65 *Duke Law Journal* 569
- Kline R.B. and A. Seffah, 'Evaluation of Integrated Software Development Environments: Challenges and Results from Three Empirical Studies' (2005) 63 *International Journal of Human-Computer Studies* 607
- Knuth D.E., 'Literate Programming' (1984) 27 *The Computer Journal* 97
- Koops B.-J., 'Criteria for Normative Technology: The Acceptability of "Code as Law" in Light of Democratic and Constitutional Values' in R. Brownsword and K. Yeung (eds.), *Regulating Technologies: Legal Futures, Regulatory Frames and Technological Fixes* (Oxford; Portland, Or: Hart, 2008)
- Kroll J.A. et al., 'Accountable Algorithms' (2016) 165 *University of Pennsylvania Law Review* 633
- Latour B., 'Where Are the Missing Masses? The Sociology of a Few Mundane Artifacts' in W.E. Bijker and J. Law (eds.), *Shaping Technology/Building Society: Studies in Sociotechnical Change* (MIT Press, 1992)
- , 'The Berlin Key or How To Do Words with Things' in P. Graves-Brown (ed.), *Matter, Materiality and Modern Culture* (London: Routledge, 2000)
- , 'Biography of an Inquiry: On a Book about Modes of Existence' (2013) 43 *Social Studies of Science* 287
- Leenes R., 'Framing Techno-Regulation: An Exploration of State and Non-State Regulation by Technology' (2011) 5 *Legisprudence* 143
- Leenes R. and B.-J. Koops, "'Code" and Privacy or How Technology Is Slowly Eroding Privacy' in E. Dommering and L. Asscher (eds.), *Coding Regulation: Essays on the Normative Role of Information Technology* (The Hague: TMC Asser Press, 2006)

- Le Sueur A., 'Robot Government: Automated Decision-Making and Its Implications for Parliament' in A. Horne and A. Le Sueur (eds.), *Parliament - Legislation and Accountability* (Oxford; Portland, Oregon: Hart, 2016)
- Lessig L., 'The Zones of Cyberspace' (1995) 48 *Stanford Law Review* 1403
- Levy K.E.C., 'Book-Smart, Not Street-Smart: Blockchain-Based Smart Contracts and The Social Workings of Law' (2017) 3 *Engaging Science, Technology, and Society* 1
- Lin B., 'Software Synthesis of Process-Based Concurrent Programs', *Proceedings of the 35th annual Design Automation Conference* (ACM, 1998)
- Lippe P., D.M. Katz and D. Jackson, 'Legal by Design: A New Paradigm for Handling Complexity in Banking Regulation and Elsewhere in Law' (2014) 93 *Oregon Law Review* 833
- Lockton D., 'Architectures of Control in Product Design' [2006] *Engineering Designer: The Journal of the Institution of Engineering Designers* 28
- Longford G., 'Pedagogies of Digital Citizenship and the Politics of Code' (2005) 9 *Techné: Research in Philosophy and Technology* 68
- Luger E. et al., 'Playing the Legal Card: Using Ideation Cards to Raise Data Protection Issues Within the Design Process', *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (New York City: ACM, 2015)
- Luger E. and M. Golembewski, 'Towards Fostering Compliance by Design; Drawing Designers into the Regulatory Frame' in M. Taddeo and L. Floridi (eds.), *The Responsibilities of Online Service Providers* (Springer International Publishing, 2017)
- Luhmann N., 'Self-Organization and Autopoiesis' in B. Clarke et al. (eds.), *Emergence and Embodiment: New Essays on Second-Order Systems Theory* (Duke University Press, 2009)
- MacCormick N., *Law as Institutional Fact* (Edinburgh: University of Edinburgh, 1973)
- , 'The Ethics of Legalism' (1989) 2 *Ratio Juris* 184
- Mackay W.E., 'Triggers and Barriers to Customizing Software', *Proceedings of the SIGCHI conference on Human factors in computing systems Reaching through technology - CHI '91* (New Orleans, Louisiana: ACM Press, 1991)
- Maier J.R. and G.M. Fadel, 'Affordance-Based Methods for Design', *Proceedings of DETC* (Chicago, Illinois: The American Society of Mechanical Engineers, 2003)
- Manders-Huits N., 'What Values in Design? The Challenge of Incorporating Moral Values into Design' (2011) 17 *Science and Engineering Ethics* 271
- Matassa A. and R. Simeoni, 'Eliciting Affordances for Smart Objects in IoT Era', *Internet of Things. User-Centric IoT* (Springer, Cham, 2015)

- Mateas M. and N. Montfort, 'A Box, Darkly: Obfuscation, Weird Languages, and Code Aesthetics', *Proceedings of the 6th Digital Arts and Culture Conference* (Copenhagen: IT University of Copenhagen, 2005)
- Mayer-Schönberger V., 'Demystifying Lessig' [2008] *Wisconsin Law Review* 713
- McGeeveran W., 'The Law of Friction' (2013) 2013 *University of Chicago Legal Forum* 15
- Meldman J.A., 'A Petri-Net Representation of Civil Procedure' (1977) 19 *Idea* 123
- Meldman J.A. and A.W. Holt, 'Petri Nets and Legal Systems' (1971) 12 *Jurimetrics Journal* 65
- Mezrich J.L., 'Source Code Escrow: An Exercise in Futility' (2001) 5 *Marquette Intellectual Property Law Review* 117
- Mittelstadt B.D. et al., 'The Ethics of Algorithms: Mapping the Debate' (2016) 3 *Big Data & Society* 1
- Mohr R. and F. Contini, 'Reassembling the Legal: "The Wonders of Modern Science" in Court-Related Proceedings' (2011) 20 *Griffith Law Review* 994
- Mortensen K.H., 'Automatic Code Generation Method Based on Coloured Petri Net Models Applied on an Access Control System' in M. Nielsen and D. Simpson (eds.), *Application and Theory of Petri Nets* (Berlin, Heidelberg: Springer, 2000)
- Mulligan D.K. and A. Perzanowski, 'The Magnificence of the Disaster: Reconstructing the Sony BMG Rootkit Incident' (2007) 22 *Berkeley Technology Law Journal* 1157
- Murata T., 'Petri Nets: Properties, Analysis and Applications' (1989) 77 *Proceedings of the IEEE* 541
- Murray A., 'Looking Back at the Law of the Horse: Why Cyberlaw and the Rule of Law Are Important' (2013) 10 *SCRIPTed* 310
- Nagy P. and G. Neff, 'Imagined Affordance: Reconstructing a Keyword for Communication Theory' (2015) 1 *Social Media + Society*
- Nakamoto S., 'Bitcoin: A Peer-to-Peer Electronic Cash System' (2008)
- Nemitz P., 'Constitutional Democracy and Technology in the Age of Artificial Intelligence' (2018) 376 *Philosophical Transactions of the Royal Society A* 20180089
- Norman D.A., 'Affordance, Conventions, and Design' (1999) 6 *Interactions* 38
- Oberle D. et al., 'Engineering Compliant Software: Advising Developers by Automating Legal Reasoning' (2012) 9 *SCRIPTed* 280
- O'Hara K., 'Smart Contracts - Dumb Idea' (2017) 21 *IEEE Internet Computing* 97
- Ohm P. and J. Frankle, 'Desirable Inefficiency' (2019) 70 *Florida Law Review* 1
- Pasquale F., 'A Rule of Persons, Not Machines: The Limits of Legal Automation' [2019] *George Washington Law Review*

- Philipps L. and G. Sartor, 'Introduction: From Legal Theories to Neural Networks and Fuzzy Reasoning' (1999) 7 *Artificial Intelligence and Law* 115
- Piekarski M. and W. Wachowski, 'Artefacts as Social Things: Design-Based Approach to Normativity' (2018) 22 *Techné: Research in Philosophy and Technology* 400
- Pilkington M., 'Blockchain Technology: Principles and Applications' in F.X. Olleros and M. Zhegu (eds.), *Research Handbook on Digital Transformations* (Cheltenham, UK; Northampton, MA: Edward Elgar Publishing, 2016)
- Pols A. and A. Spahn, 'Designing for the Values of Democracy and Justice' in J. van den Hoven, P.E. Vermaas and I. van de Poel (eds.), *Handbook of Ethics, Values, and Technological Design: Sources, Theory, Values and Application Domains* (Dordrecht: Springer, 2015)
- Radin M.J., 'Machine Rule: The Latest Challenge to Law' (2005) (unpublished manuscript, on file)
- , 'Regulation by Contract, Regulation by Machine' (2004) 160 *Journal of Institutional and Theoretical Economics (JITE)* 142
- Raz J., 'The Rule of Law and Its Virtue', *The Authority of Law: Essays on Law and Morality* (Oxford University Press, 1979)
- Reed C., 'How to Make Bad Law: Lessons from Cyberspace' (2010) 73 *The Modern Law Review* 903
- Reidenberg J.R., 'Lex Informatica: The Formulation of Information Policy Rules through Technology' (1997) 76 *Texas Law Review* 553
- Royce W.W., 'Managing the Development of Large Software Systems', *proceedings of IEEE WESCON* (Los Angeles, 1970)
- Salimifard K. and M. Wright, 'Petri Net-Based Modelling of Workflow Systems: An Overview' (2001) 134 *European Journal of Operational Research* 664
- Schmidt A., 'Radbruch in Cyberspace: About Law-System Quality and ICT Innovation' (2009) 3 *Masaryk University Journal of Law and Technology* 195
- Schot J. and A. Rip, 'The Past and Future of Constructive Technology Assessment' (1997) 54 *Technological Forecasting and Social Change* 251
- Schot J., 'Constructive Technology Assessment and Technology Dynamics: The Case of Clean Technologies' (1992) 17 *Science, Technology, & Human Values* 36
- Schulz W. and K. Dankert, "'Governance by Things" as a Challenge to Regulation by Law' (2016) 5 *Internet Policy Review*
- Scott C., 'Regulation in the Age of Governance: The Rise of the Post-Regulatory State' in J. Jordana and D. Levi-Faur (eds.), *The politics of regulation: Institutions and regulatory reforms*

- for the age of governance* (Cheltenham, UK; Northampton, MA: Edward Elgar Publishing, 2004)
- Selbst A.D. et al., 'Fairness and Abstraction in Sociotechnical Systems' (Rochester, NY: Social Science Research Network, 2018) SSRN Scholarly Paper ID 3265913 <<https://papers.ssrn.com/abstract=3265913>>
- Selbst A.D. and J. Powles, 'Meaningful Information and the Right to Explanation' (2017) 7 *International Data Privacy Law* 233
- Shah R.C. and C. Sandvig, 'Software Defaults as De Facto Regulation: The Case of the Wireless Internet' (2008) 11 *Information, Communication & Society* 25
- Sharkey N., 'Grounds for Discrimination: Autonomous Robot Weapons' (2008) 11 *RUSI Defence Systems* 86
- Shatz S.M. and W.K. Cheng, 'A Petri Net Framework for Automated Static Analysis of Ada Tasking Behavior' (1988) 8 *Journal of Systems and Software* 343
- Shay L.A. et al., 'Confronting Automated Law Enforcement' in R. Calo, A. Froomkin and I. Kerr (eds.), *Robot Law* (Edward Elgar Publishing, 2016)
- , 'Do Robots Dream of Electric Laws? An Experiment in the Law as Algorithm' in R. Calo, A. Froomkin and I. Kerr (eds.), *Robot Law* (Edward Elgar Publishing, 2016)
- Smith D.J., 'Changing Situations and Changing People' in A. von Hirsch, D. Garland and A. Wakefield (eds.), *Ethical and Social Perspectives on Situational Crime Prevention* (Bloomsbury Publishing, 2004)
- Stahl B.C. et al., 'From Computer Ethics to Responsible Research and Innovation in ICT' (2014) 51 *Information & Management* 810
- Sunstein C.R. and R.H. Thaler, 'Libertarian Paternalism Is Not an Oxymoron' (2003) 70 *University of Chicago Law Review* 1159
- Surden H., 'Values Embedded in Legal Artificial Intelligence' (Rochester, NY: Social Science Research Network, 2017) SSRN Scholarly Paper ID 2932333 <<https://papers.ssrn.com/abstract=2932333>>
- Swierstra T. and K. Waelbers, 'Designing a Good Life: A Matrix for the Technological Mediation of Morality' (2012) 18 *Science and Engineering Ethics* 157
- Tasioulas J., 'The Rule of Law' in J. Tasioulas (ed.), *The Cambridge Companion to the Philosophy of Law* (Cambridge University Press, 2019)
- Tien L., 'Architectural Regulation and the Evolution of Social Norms' (2004) 7 *Yale Journal of Law & Technology* 1

- Urquhart L. and T. Rodden, 'New Directions in Information Technology Law: Learning from Human-Computer Interaction' (2017) 31 *International Review of Law, Computers & Technology* 150
- van den Berg B. and R.E. Leenes, 'Abort, Retry, Fail: Scoping Techno-Regulation and Other Techno-Effects' in M. Hildebrandt and J. Gaakeer (eds.), *Human Law and Computer Law: Comparative Perspectives* (Dordrecht: Springer Netherlands, 2013)
- Verbeek P.-P., 'Materializing Morality: Design Ethics and Technological Mediation' (2006) 31 *Science, Technology, & Human Values* 361
- Vismann C. and M. Krajewski, 'Computer Juridisms' [2007] *Grey Room* 90
- von Schomberg R., 'A Vision of Responsible Research and Innovation' in R. Owen, J. Bessant and M. Heintz (eds.), *Responsible Innovation* (Chichester, UK: John Wiley & Sons, 2013)
- Wachter S., B. Mittelstadt and L. Floridi, 'Why a Right to Explanation of Automated Decision-Making Does Not Exist in the General Data Protection Regulation' (2017) 7 *International Data Privacy Law* 76
- Wachter S., B. Mittelstadt and C. Russell, 'Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR' (2017) 31 *Harvard Journal of Law and Technology* 841
- Wagner B., 'Ethics as an Escape from Regulation: From Ethics-Washing to Ethics-Shopping?' in I. Baraliuc et al. (eds.), *Being Profiled: Cogitas Ergo Sum - 10 Years of Profiling the European Citizen* (Amsterdam University Press, 2018)
- Wagner R.P., 'On Software Regulation' (2005) 78 *Southern California Law Review* 457
- Waldron J., 'The Core of the Case against Judicial Review' (2006) 115 *Yale Law Journal* 1346
- , 'The Concept and the Rule of Law' (2008) 43 *Georgia Law Review* 1
- , 'Can There Be a Democratic Jurisprudence?' (2009) 58 *Emory Law Journal* 675
- , 'The Rule of Law and the Importance of Procedure' (2011) 50 *Nomos* 3
- Weber R.H., "'Rose Is a Rose Is a Rose Is a Rose" – What about Code and Law?' (2018) 34 *Computer Law & Security Review* 701
- Weiser M., 'The Computer for the 21st Century' [1991] *Scientific American* 94
- Winner L., 'Do Artifacts Have Politics?' [1980] *Daedalus* 121
- Wintgens L., 'Legislation as an Object of Study of Legal Theory: Legisprudence', *Legisprudence: A New Theoretical Approach to Legislation* (Oxford: Hart, 2002)
- , 'Legisprudence as a New Theory of Legislation' (2006) 19 *Ratio Juris* 1
- , 'The Rational Legislator Revisited. Bounded Rationality and Legisprudence', *The Rationality and Justification of Legislation* (Springer, Heidelberg, 2013)

- Wittkower D.E., 'Principles of Anti-Discriminatory Design', 2016 IEEE International Symposium on Ethics in Engineering, Science and Technology (ETHICS) (IEEE, 2016)
- Wright A. and P. De Filippi, 'Decentralized Blockchain Technology and the Rise of Lex Cryptographia' (Rochester, NY: Social Science Research Network, 2015) SSRN Scholarly Paper ID 2580664 <<https://papers.ssrn.com/abstract=2580664>>
- Wright D., 'Should Privacy Impact Assessments Be Mandatory?' (2011) 54 *Communications of the ACM* 121
- Yee K.-P., 'User Interaction Design for Secure Systems' in R. Deng et al. (eds.), *Information and Communications Security* (Berlin, Heidelberg: Springer Berlin Heidelberg, 2002)
- Yeung K., 'Can We Employ Design-Based Regulation While Avoiding Brave New World?' (2011) 3 *Law, Innovation & Technology* 1
- Young M.T., 'Artifacts as Rules: Wittgenstein and the Sociology of Technology' (2018) 22 *Techné: Research in Philosophy and Technology* 377
- Yu Z. et al., 'Tracking the Trackers', *Proceedings of the 25th International Conference on World Wide Web - WWW '16* (Montreal, Quebec, Canada: ACM Press, 2016)

4. Reports, technical publications, and whitepapers

- Abran A. et al., *Guide to the Software Engineering Body of Knowledge (SWEBOK)* (Los Alamitos, California: IEEE Computer Society and Angela Burgess, 2004)
- 'ACM Code of Ethics and Professional Conduct (2018)' <<https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct>>
- 'Android Developers Guide - Permissions Overview' <<https://developer.android.com/guide/topics/permissions/overview>>
- 'Behat Documentation' (*Behat*) <http://docs.behat.org/en/latest/quick_start.html>
- Cardozo Blockchain Project, "'Smart Contracts" & Legal Enforceability' (New York: Benjamin N Cardozo School of Law, 2018)
- Crespo García A. et al., 'PRIPARE Privacy- and Security-by-Design Methodology Handbook' (EU FP7, 2015)
- 'Data Protection Impact Assessments' (*Information Commissioner's Office*, 20 December 2018) <<https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/accountability-and-governance/data-protection-impact-assessments/>>
- 'Deceived by Design: How Tech Companies Use Dark Patterns to Discourage Us from Exercising Our Rights to Privacy' (Consumer Council of Norway (Forbrukerrådet), 2018)

<<https://fil.forbrukerradet.no/wp-content/uploads/2018/06/2018-06-27-deceived-by-design-final.pdf>>

‘Ethereum Natural Specification Format’, *The Ethereum Wiki* (Ethereum Foundation, 2018) <<https://en.ethereum.wiki/Ethereum-Natural-Specification-Format>>

‘Ethereum White Paper’ (*The Ethereum Wiki*, 22 August 2018) <<https://github.com/ethereum/wiki/wiki/White-Paper>>

European Commission, ‘Privacy and Data Protection Impact Assessment Framework for RFID Applications’ (European Commission, 2011) <http://ec.europa.eu/newsroom/dae/document.cfm?doc_id=8218>

Fowler M., ‘Business Readable Domain Specific Language’ (*martinfowler.com*, 15 December 2008) <<https://martinfowler.com/bliki/BusinessReadableDSL.html>>

‘Google Algorithm Change History’ (*Moz*, 2018) <<https://moz.com/google-algorithm-change>>

‘Immaterial Labour and Data Harvesting’ <<https://labs.rs/en/facebook-algorithmic-factory-immaterial-labour-and-data-harvesting/>>

‘Internet of Things: Privacy and Security in a Connected World’ (Federal Trade Commission, 2015)

‘Investigation into the Use of Data Analytics in Political Campaigns - Investigation Update’ (Information Commissioner’s Office, 2018) <<https://ico.org.uk/media/action-weve-taken/2259371/investigation-into-data-analytics-for-political-purposes-update.pdf>>

‘Mattereum Protocol: Turning Code into Law’ (Mattereum Project, 2018) <https://www.mattereum.com/upload/iblock/784/mattereum-summary_white_paper.pdf>

Murphy S. and C. Cooper, ‘Can Smart Contracts Be Legally Binding Contracts? White Paper Key Findings’ (R3 and Norton Rose Fulbright, 2016) <<https://www.nortonrosefulbright.com/-/media/files/nrf/nrfweb/imported/norton-rose-fulbright--r3-smart-contracts-white-paper-key-findings-nov-2016.pdf>>

‘Open Source Software Strategy’ (European Commission, 2016) <https://ec.europa.eu/info/departments/informatics/open-source-software-strategy_en>

Papadopoulos G. et al., ‘Statistics on Small and Medium-Sized Enterprises’ (European Commission, 2018) <https://ec.europa.eu/eurostat/statistics-explained/index.php/Statistics_on_small_and_medium-sized_enterprises>

‘Principles for Accountable Algorithms and a Social Impact Statement for Algorithms :: FAT ML’ <<https://www.fatml.org/resources/principles-for-accountable-algorithms>>

‘Smart Contracts’ (*Law Commission*) <<https://www.lawcom.gov.uk/project/smart-contracts/>>

'Study on the Economic Benefits of Privacy-enhancing Technologies (PETs)' (London Economics, 2010)

'Visual Studio IntelliCode' (*Microsoft*, May 2018)
<<https://visualstudio.microsoft.com/services/intellicode/>>

'XML Documentation (Visual C++)' (Microsoft, 2016) <<https://docs.microsoft.com/en-us/cpp/ide/xml-documentation-visual-cpp>>

5. Grey literature (news articles, blogs, etcetera)

'14 Predictions For The Future Of Smart Home Technology' (*Forbes*)
<<https://www.forbes.com/sites/forbestechcouncil/2018/01/12/14-predictions-for-the-future-of-smart-home-technology/>>

'Amazon.co.uk Help: Set Up Your Dash Button'
<https://www.amazon.co.uk/gp/help/customer/display.html/ref=amb_link_1?nodeId=201746340>

Anastasio M, 'EU Governments Support First Set of Laws for More Repairable Products' (*EEB - The European Environmental Bureau*) <<https://eeb.org/eu-governments-support-first-set-of-laws-for-more-repairable-products/>>

'Android Flashlight App Developer Settles FTC Charges It Deceived Consumers' (*Federal Trade Commission*, 5 December 2013) <<https://www.ftc.gov/news-events/press-releases/2013/12/android-flashlight-app-developer-settles-ftc-charges-it-deceived>>

Angelova E., 'Microsoft Embraces Open Source' [2018] *Fordham Intellectual Property, Media & Entertainment Law Journal* <<http://www.fordhamiplj.org/2018/11/28/microsoft-embraces-open-source/>>

Angwin J. et al., 'Machine Bias' (*ProPublica*, 23 May 2016)
<<https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>>

Barlow J.P., 'A Declaration of the Independence of Cyberspace' (1996)
<<https://www.eff.org/cyberspace-independence>>

Blackburn S., 'Deontological Ethics', *The Oxford Dictionary of Philosophy* (Oxford University Press, 2008)

Booth R., 'Facebook Reveals News Feed Experiment to Control Emotions' *The Guardian* (29 June 2014) <<https://www.theguardian.com/technology/2014/jun/29/facebook-users-emotions-news-feeds>>

- Brodkin J., 'EU Fines Microsoft €561 Million for Not Giving Users a Browser Choice' (*Ars Technica*, 6 March 2013) <<https://arstechnica.com/tech-policy/2013/03/eu-fines-microsoft-e561-million-for-not-giving-users-a-browser-choice/>>
- Brown S, 'An Agile Approach to Designing for the Consequences of Technology' (*doteveryone*, 13 February 2019) <<https://medium.com/doteveryone/an-agile-approach-to-designing-for-the-consequences-of-technology-18a229de763b>>
- Buterin V., 'DAOs, DACs, DAs and More: An Incomplete Terminology Guide' <<https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide/>>
- , 'On Public and Private Blockchains' (*Ethereum Blog*, 8 June 2015) <<https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/>>
- , (*Twitter*, 13 October 2018) <<https://twitter.com/VitalikButerin/status/1051160932699770882>>
- Cadwalladr C. and E. Graham-Harrison, 'Revealed: 50 Million Facebook Profiles Harvested for Cambridge Analytica in Major Data Breach' *The Guardian* (17 March 2018) <<http://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election>>
- CleanApp, 'Crypto Legal Theory' <<https://medium.com/cryptolawreview/crypto-legal-theory-299fa35be21f>>
- 'Connections Counter: The Internet of Everything in Motion' (*Cisco: The Network*, 29 July 2013) <<https://newsroom.cisco.com/feature-content?type=webcontent&articleId=1208342>>
- Felten E., 'Smart Contracts: Neither Smart nor Contracts?' <<https://freedom-to-tinker.com/2017/02/20/smart-contracts-neither-smart-not-contracts/>>
- Franceschi-Bicchierai L., 'Hacked Toy Company VTech's TOS Now Says It's Not Liable for Hacks' [2016] *Motherboard* <<http://motherboard.vice.com/read/hacked-toy-company-vtech-tos-now-says-its-not-liable-for-hacks>>
- @gael, 'How Did We Get Here?' (*Twitter*, 8 January 2019) <https://twitter.com/_gael_/status/1082702284746883072>
- 'Gartner's 2016 Hype Cycle for Emerging Technologies Identifies Three Key Trends That Organizations Must Track to Gain Competitive Advantage' (*Gartner*, 16 August 2016) <<https://www.gartner.com/en/newsroom/press-releases/2016-08-16-gartners-2016-hype-cycle-for-emerging-technologies-identifies-three-key-trends-that-organizations-must-track-to-gain-competitive-advantage>>
- Graham P., 'Beating the Averages' (2003) <<http://www.paulgraham.com/avg.html>>

'Institution, n.' (Oxford English Dictionary Online)

Kaminska I., 'Growing Scepticism Challenges the Blockchain Hype' (*Financial Times*, 20 June 2017) <<https://www.ft.com/content/b5b1a5f2-5030-11e7-bfb8-997009366969>>

Krebs B., 'IoT Reality: Smart Devices, Dumb Defaults' (*Krebs on Security*, 16 February 2008) <<https://krebsonsecurity.com/2016/02/iot-reality-smart-devices-dumb-defaults/>>

Leiner B.M. et al., 'Brief History of the Internet' (Internet Society, 1997) <<https://www.internetsociety.org/internet/history-internet/brief-history-internet/>>

Leyden J., 'Users Protest over "creepy" Facebook Update' *The Register* (9 July 2006) <https://www.theregister.co.uk/2006/09/07/facebook_update_controversy/>

Lockton D., 'Disaffordances and Engineering Obedience' <<http://architectures.danlockton.co.uk/2006/10/22/disaffordances-and-engineering-obedience/>>

'Manifesto for Agile Software Development' (2001) <<https://agilemanifesto.org/>>

'Meltdown and Spectre' (2018) <<https://meltdownattack.com/>>

North D., 'What's in a Story?' (*Dan North & Associates*, 11 February 2007) <<https://dannorth.net/whats-in-a-story/>>

Noyes K., 'Is Linux Really Harder to Use?' (*PCWorld*, 2 August 2010) <https://www.pcworld.com/article/202364/is_linux_really_harder_to_use.html>

Olejnik L., 'Am I Logged in or Not? GDPR Case Study on the Example of Chrome Browser Change' <<http://blog.lukaszolejnik.com/am-i-logged-in-or-not-gdpr-case-study-on-the-example-of-chrome-browser-change/>>

Omale G., 'Gartner Identifies Top 10 Strategic IoT Technologies and Trends' (*Gartner*, 7 November 2018) <<https://www.gartner.com/en/newsroom/press-releases/2018-11-07-gartner-identifies-top-10-strategic-iot-technologies-and-trends>>

Parkin S., 'Has Dopamine Got Us Hooked on Tech?' *The Observer* (4 March 2018) <<https://www.theguardian.com/technology/2018/mar/04/has-dopamine-got-us-hooked-on-tech-facebook-apps-addiction>>

Perez J., 'XYO Game-Changer: We've Executed a Smart Contract With a Drone!' *Medium* (21 November 2018) <<https://medium.com/xyonetwork/xyo-game-changer-weve-executed-a-smart-contract-with-a-drone-4deb414af67b>>

Porup J.M., 'How to Search the Internet of Things for Photos of Sleeping Babies' (*Ars Technica UK*, 19 January 2016) <<http://arstechnica.co.uk/security/2016/01/how-to-search-the-internet-of-things-for-photos-of-sleeping-babies/>>

Roose K., 'Is Tech Too Easy to Use?' *The New York Times* (13 December 2018) <<https://www.nytimes.com/2018/12/12/technology/tech-friction-frictionless.html>>

Russinovich M., 'More on Sony: Dangerous Decloaking Patch, EULAs and Phoning Home' <<http://blogs.technet.com/b/markrussinovich/archive/2005/11/04/more-on-sony-dangerous-decloaking-patch-eulas-and-phoning-home.aspx>>

Schlesinger A., 'Feminism and Programming Languages' <<https://www.hastac.org/blogs/ari-schlesinger/2013/11/26/feminism-and-programming-languages>>

Schneier B., 'I've Seen the Future, and It Has a Kill Switch' [2008] *Wired* <<https://www.wired.com/2008/06/securitymatters-0626/>>

'Scientific Method, n.' <<http://www.oed.com/view/Entry/383323>>

Sengupta S., 'Private Posts on Facebook Revealed' <<https://bits.blogs.nytimes.com/2013/01/18/private-posts-on-facebook-revealed/>>

'Sony Slated over Anti-Piracy CD' *BBC News* (3 November 2005) <<http://news.bbc.co.uk/1/hi/technology/4400148.stm>>

'Stack Overflow Developer Survey 2019' (Stack Overflow) <<https://insights.stackoverflow.com/survey/2019/>>

Szabo N., 'Smart Contracts' (1994) <<https://web.archive.org/web/19971008001022/http://www.virtualschool.edu/mon/Economics/SmartContracts.html>>

'The Lean Startup | Methodology' <<http://theleanstartup.com/principles>>

'The Problem' (*Center for Humane Technology*) <<http://humanetech.com/problem/>>

Thierer A., 'The Pacing Problem, the Collingridge Dilemma & Technological Determinism' *Technology Liberation Front* (16 August 2018) <<https://techliberation.com/2018/08/16/the-pacing-problem-the-collingridge-dilemma-technological-determinism/>>

Tillman K., 'How Many Internet Connections Are in the World? Right. Now.' <<https://blogs.cisco.com/news/cisco-connections-counter>>

Torchia M. and M. Shirer, 'IDC Forecasts Worldwide Spending on the Internet of Things to Reach \$745 Billion in 2019, Led by the Manufacturing, Consumer, Transportation, and Utilities Sectors' *IDC* (Framingham, Mass, 3 January 2019) <<https://www.idc.com/getdoc.jsp?containerId=prUS44596319>>

Twentyman J., 'IoT Drives Progress towards Low-Power Technology' *Financial Times* (8 January 2018) <<https://www.ft.com/content/f2b4de5a-d8ee-11e7-9504-59efdb70e12f>>

Ulanoff L., 'How to create a Twitter collage' (*Mashable*, 1 April 2014) <<https://mashable.com/2014/04/01/how-to-make-a-twitter-collage/>>

