Technology and Regulation

General Principles of Code

Nadezhda Purtova*, Diletta Huyskes**

Regulation, algorithms, general principles, amplification, automation, technology-specific regulation

n.n.purtova@uu.nl
diletta.huyskes@unimi.it

The General Data Protection Regulation has been seen as an omnibus law which potentially addresses all Internet-related problems. Yet, concerns have been raised about the broad scope of the GDPR that might be overreaching its capacity. Calls have been made to create a scalable and more targeted system of legal protection against digital wrongs. General principles governing the design and use of software should be at the foundation of such a scalable system. While automation via code is often an aspect of existing societal problems, software affordances amplify non-technology-specific problems in a unique way.

1. Introduction

One important piece is missing from the EU grand scheme of regulatory efforts to address the challenges of the digital society. We submit that – in addition to the existing and proposed regulation for the digital world and in order to create a scalable and more targeted system of legal protection against risks of an information society – the design and use of computer software, including but not limited to what is known as Artificial Intelligence systems, should be subject to framework regulation in the form of general principles. We argue that applying one possible set of such principles, namely, Fair Information Principles, will positively affect legal protection against harms associated with information technologies.

In the past decade, the EU has been leading a large-scale effort to reform the legal landscape, tackling various problems of the information society. The effort was kicked off with the 2012 data protection reform that resulted in the adoption of the General Data Protection Regulation (the GDPR)[1] and culminated in the avalanche of other regulatory instruments proposed and adopted in the past five years.[2] The new law has primarily been sectoral, that is focusing on one or several aspects of the information society or specific sectors, such as data governance and especially data sharing,[3] data sharing in the health sector,[4] unfair consumer practices and content moderation in the digital context,[5] algorithmic management of platform workers,[6] or design and use of certain kinds of AI and transparency of the AI

1 Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 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 95/46/EC (General Data Protection Regulation) OJ L 119.

* Nadezhda Purtova is a Professor of Law, Innovation, and Technology at Molengraaff Institute for Private Law, Faculty of Law, Economics, and Governance, Utrecht University, the Netherlands;
** Diletta Huyskes is a PhD candidate at the Department of Philosophy, University of Milan, Italy.

Received 13 Nov 2023 Accpted 9 Jan 2024 Published 26 Apr 2024

2 This is an incomplete list of the relevant proposed and adopted EU legislation: Directive (EU) 2019/944 of the European Parliament and of the Council of 5 June 2019 on common rules for the internal market for electricity and amending Directive 2012/27/EU OJ L 158; Directive (EU) 2019/1024 of the European Parliament and of the Council of 20 June 2019 on open data and the re-use of public sector information OJ L 172; Regulation (EU) 2022/868 of the European Parliament and of the Council of 30 May 2022 on European data governance and amending Regulation (EU) 2018/1724 (Data Governance Act) OJ L 152; Regulation (EU) 2022/2065 of the European Parliament and of the Council of 19 October 2022 on a Single Market For Digital Services and amending Directive 2000/31/EC (Digital Services Act) OJ L 277; Regulation (EU) 2022/1925 of the European Parliament and of the Council of 14 September 2022 on contestable and fair markets in the digital sector and amending Directives (EU) 2019/1937 and (EU) 2020/1828 (Digital Markets Act) OJ L 265; at the time of writing, Council has adopted its general approach on the Artificial Intelligence Act Data Act (Proposal for a Regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union legislative acts - General approach, adopted on 25 November 2022); Regulation (EU) 2023/2854 of the European Parliament and of the Council of 13 December 2023 on harmonised rules on fair access to and use of data and amending Regulation (EU) 2017/2394 and Directive (EU) 2020/1828 (Data Act) OJ L, 2023/2854; Proposal for a Regulation of the European Parliament and of the Council on the European Health Data Space (COM/2022/197 final); Proposal for a Directive of the European Parliament and of the Council on improving working conditions in platform work (COM(2021) 762 final) Brussels, 9.12.2021 (The Platform Work Directive).
3 E.g. Data Act and Data Governance Act (n 2).
4 European Health Data Space (n 2).
5 Digital Services Act (n 2).
6 Platform Work Directive (n 2).

use.[7] In contrast, the GDPR has been seen as an omnibus law which potentially addresses all Internet-related problems,[8] from protecting private sphere to algorithmic firing[9] to discrimination[10] and consumer empowerment and trust in the digital economy.[11] This "catch-all" nature of the GDPR's legal protection results from the broad interpretation of its material scope anchored in the broad meaning of "personal data"[12] and its broad general principles of lawful data processing, including fairness, lawfulness, transparency and proportionality of processing manifested in the form of minimization and storage limitation principles, which make the GDPR suitable to address a broad range of digital problems. To illustrate, the Chair of the Dutch Data Protection Authority (DPA) has recently spoken about his vision of the role of data protection authorities as watchdogs of fundamental rights in the digital society.[13] In Spring 2023, the Italian Data Protection Authority banned the use of ChatGPT, a generative AI chatbot that mimics human responses to prompts, on the ground that the GDPR was violated during its training and use.[14]

Having such a catch-all legal instrument seems like a good thing, considering that it provides a legal remedy for cases where no other legal regimes apply. This was clearly the case with the Italian ban on ChatGPT. While the DPA cited GDPR violations as reasons for the ban, one of the concerns was that the AI "exposes minors to absolutely unsuitable answers compared to their degree of development and awareness".[15] Such effects of AI have not yet been addressed by any existing piece of legislation, and it has been disputed if the transparency requirements in the proposed AI Act would suffice.[16] However, there are a few significant problems with this approach.

7 AI Act (n 2).
8 Bert-Jaap Koops, 'The Trouble with European Data Protection Law' (2014) 4 International Data Privacy Law 250; Bert-Jaap Koops, 'On legal boundaries, technologies, and collapsing dimensions of privacy' (2014) Politica & Societa 247, 258.
9 'Uber Sued by Drivers over 'automated Robo-Firing'' BBC News (26 October 2020) https://www.bbc.com/news/business-54698858 accessed 3 November 2023. ("alleging well over 1,000 individual cases where drivers have allegedly been wrongly accused of fraudulent activity and immediately had their accounts terminated without a right of appeal."). See also Sarah Butler, 'Court Tells Uber to Reinstate Five UK Drivers Sacked by Automated Process' The Guardian (14 April 2021) https://www.theguardian.com/technology/2021/apr/14/court-tells-uber-to-reinstate-five-uk-drivers-sacked-by-automated-process accessed 1 November 2023.
10 E.g. Gianclaudio Malgieri, 'The Concept of Fairness in the GDPR: A Linguistic and Contextual Interpretation', Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (ACM 2020) 158, https://dl.acm.org/doi/10.1145/3351095.3372868 accessed 1 November 2023.
11 The potential for consumer empowerment and trust in digital economy lies primarily with the right to data portability as discussed in Inge Graef, Martin Husovec and Nadezhda Purtova, 'Data Portability and Data Control: Lessons for an Emerging Concept in EU Law' (2018) 19 German Law Journal 1359.
12 Nadezhda Purtova, 'The Law of Everything. Broad Concept of Personal Data and Future of EU Data Protection Law' (2018) 10 Law, Innovation and Technology 40.
13 Aleid Wolfsen, Hans Frankenlezing "Hoedster van de grondrechten in de digitale rechtsstaat?" 20 May 2022, Leiden.
14 The violations include a lack of legal basis for processing training data and a data breach involving conversations and payment data (Shiona McCallum, 'ChatGPT Banned in Italy over Privacy Concerns' BBC News (31 March 2023) https://www.bbc.com/news/technology-65139406 accessed 1 November 2023.). At the time of writing, OpenAI, a company behind ChatGPT, has fixed the GDPR issues and ChatGPT is back on the Italian market, while the problems of exposing children to generative AI that the DPA cited are not resolved.
15 McCallum (n 14).
16 E.g. Nathalie Smuha, Mieke De Ketelaere, Mark Coeckelbergh, Pierre Dewitte and Yves Poullet, 'Onze samenleving is niet klaar voor manipulatieve AI" (Knack, 29 March 2023) https://www.knack.be/nieuws/technologie/onze-samenleving-is-niet-klaar-voor-manipulatieve-ai/ accessed 3 November 2023.

Serious concerns have been raised about the broad scope of the GDPR that might be overreaching its capacity. The underfunded and understaffed DPAs are not in a position to meaningfully and equitably enforce the GDPR in all cases where it applies, and even less so after another leap to the new context of effectively moderating developments in generative AI or generally supervising algorithms,[17] a mandate that the Dutch DPA has recently assumed.[18] Such expansions have contributed to the GDPR losing its identity, where controllers and data subjects alike do not understand what this legal instrument is for,[19] an important factor if we want this law to be used. Finally, as broad as the scope of the GDPR is, it is still defined by the concept of personal data, and therefore, the availability of the GDPR remedies is conditional, among others, on whether or not a natural person affected by a certain digital practice is identified or identifiable. Many so-called "privacy-enhancing technologies" (PETs), such as multi-party computation, synthetic data and federated learning have been developed to minimize the processing of identified or identifiable data, though how successful they have been is questionable.[20] While a very broad interpretation of identification and identifiability is possible to bring data processing with the use of PETs within the ambit of the data protection law,[21] the debates on whether or not a data subject is identified or identifiable and if personal data is processed in many cases can obscure what really is at stake regarding a certain practice, for instance, in the example above, if the content generated by ChatGPT is appropriate for vulnerable content recipients, such as children. For these reasons, it has been argued elsewhere that the system of legal protection against harms of the digital society needs to be revised, made more scalable and targeted at the phenomena that are in a more direct causal connection with those harms.[22]

In this paper, we argue that one way to achieve such a scalable and more targeted system of legal protection and an alternative to the GDPR as one law that "rules them all" is regulation of the design and use of computer code with impact on people by means of the framework of general principles as opposed to highly specific rules. While many such general principles can be devised, we consider the general data protection principles, the latest reincarnation of the Fair Information Principles, as a blueprint for the general principles of code and examine what effects they will have on legal protection when applied to code.

Before we proceed with the analysis, it is important to define what we mean by code. In very simple terms, by code, we understand computer software, i.e. instructions that "tell" the computer hardware what to do. Depending on its role in the human programmer-computer interaction, all computer software can be generally divided into

17 Purtova (n 12); Nadezhda Purtova and Ronald Leenes, 'Code as Personal Data: Implications for Data Protection Law and Regulation of Algorithms' (2023) 13(4) International Data Privacy Law 245, https://academic.oup.com/idpl/advance-article/doi/10.1093/idpl/ipad019/7308779 accessed 1 November 2023.
18 'Dutch DPA to Enhance Algorithm Supervision' https://iapp.org/news/a/dutch-dpa-to-enhance-algorithm-supervision/ accessed 3 November 2023.
19 Koops (n 8).
20 E.g. Nadezhda Purtova, 'From Knowing by Name to Targeting: The Meaning of Identification under the GDPR' (2022) 12(3) International Data Privacy Law 163; Michael Veale, 'Rights for Those Who Unwillingly, Unknowingly and Unidentifiably Compute!' (SocArXiv, 31 July 2023) https://osf.io/preprints/socarxiv/4ugxd/ accessed 3 November 2023.
21 Purtova (n 20).
22 Purtova (n 12); Nadezhda Purtova and Bryce Clayton Newell, 'Against Data Fixation: Why "Data" Fails as a Regulatory Target for Data Protection Law and What to Do About It' (2023) (unpublished manuscript).

three types: *source code* written by a human programmer in one of the human-readable programming languages such as Python-, C++ or Javascript, which usually relies on the natural-language words, abbreviations and punctuation;[23] *object code*, i.e. the machine-readable detailed instructions on which operations the hardware should execute; and software that "translates" source code to object code, called a *compiler* (for ad hoc translations) or *interpreter* (for every time an application runs). We will use "code", "computer software", and "algorithm" interchangeably in this paper, which is in line with how these terms are currently used in the legal and ethical discourse.[24]

Our analysis will proceed as follows. In Section 2, we review the literature that came before our argument where computer code is problematized, and calls are made for its regulation. We explain how our analysis differs from and contributes to this literature in at least three ways. (1) We focus on the broad category of computer software rather than its narrow subtype, such as AI, and (2) on its entire lifecycle. (3) Finally, we share the view held by many that code is a manifestation of power and a mode of regulation. Yet, instead of presuming that for this reason, a code-specific regulatory intervention is warranted, we make the issue of the need for code-specific regulation vs technology-neutral regulation a central point of our analysis. In Section 3, we explain why computer code warrants code-specific regulation. While computer code is often an aspect of existing societal problems, affordances of computer code such as its scalability and stickiness of the code-generated outcomes amplify those non-technology specific problems in a unique way. In Section 4, we argue that, based on the insights of the legal and regulatory theory, general principles, as opposed to specific rules, will deliver more legal certainty and better legal protection in the complex and highly dynamic context of a digital society with high economic stakes. Finally, in Section 5, we illustrate how one set of such general principles – built around the general data protection principles – would work.

2. Situating the argument in relation to state-of-the-art

We are certainly not the first to problematize code or to suggest that computer code needs to be designed and used with certain values or principles in mind or that it should be regulated. In light of the increasing availability and sophistication of software and the empirical consequences it has for our public and private lives, many proposals have been made in recent years from perspectives of various disciplines outlining how software should be constructed or regulated in order to minimize its negative effects. Without aiming at producing a complete catalogue of these analyses, in this section, we cluster and review these proposals and situate ours in relation to them.

The risks associated with applying computer code to society are very diverse in nature. These risks have been approached by many disciplines from different perspectives which resulted in multiple accounts of the relationship between values and code design, which rarely communicate and often fail to successfully inform each other.[25] We identify several clusters of such literature below.

The idea of values designed in code has its intellectual roots in the 1970s the science and technology studies (STS) literature. One of the central ideas of STS is that technological design is not neutral but inherently shaped by cultural and moral values. Social constructivist theories and value-sensitive design (VSD) scholarship[26] stressed how the social context, specific and diverse interests of people and organizations shape technological artifacts. The sociology of technology, more broadly, looked at the actors and social groups that were constructing the technologies and their fundamental role in influencing them with subjective values that would then be transferred back to the technology users.[27] The paradigm shift inaugurated by these studies lies in understanding the ways in which a technology is used only in relation to the social context in which it is embedded, where the boundaries between the social and the technical are increasingly blurred. Langdon Winner, among others, famously demonstrated how political ideas influence the design and goals of a technology.[28] According to philosopher Andrew Feenberg, no technology is built in neutral spaces, demonstrating how it is always a political phenomenon, an "extension of the existing".[29] Relatedly, feminist scholars critiqued technology for not including women and other (non-gender) marginalized social groups in the design process and noted that, as a result, their relationship with technology was different from that of dominant social groups.[30]

While these are the points raised in relation to technology generally, more recently, STS scholarship subjected computer code to similar critique, pointing out that it is intrinsically value-laden.[31] Yet, this academic field primarily provides critical reflections on code and does not offer directions on how to assess computer code against certain values or incorporate those values into code, through regulation or otherwise. Many VSD theories, given their link to human-computer interaction (HCI), critique computer software from the perspectives of usability and accessibility and do not look at other societal values, with notable but rare exceptions such as analyses proposing traditional VSD principles as design principles of AI.[32]

Another way in which computer software was problematized is through bias. The first analysis on bias in computer software was done within VDS theory in 1996 by Friedman and Nissenbaum.

23 Rob Kitchin and Martin Dodge, Code/Space: Software and Everyday Life (The MIT Press 2011), 25

24 For a more at-length explanation of terminology see, e.g. Purtova and Leenes (n 17).

25 Andreas Tsamados and others, 'The Ethics of Algorithms: Key Problems and Solutions' (2022) 37 AI & Society 215.

26 Batya Friedman and David G Hendry, Value Sensitive Design: Shaping Technology with Moral Imagination (The MIT Press 2019) https://direct.mit.edu/books/book/4328/Value-Sensitive-DesignShaping-Technology-with accessed 3 November 2023.

27 Wiebe E Bijker, Thomas Hughes and Trevor Pinch (eds), The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology; [Papers of a Workshop Held at the University of Twente, The Netherlands, in July 1984] (MIT Press 1987); Bruno Latour, Science in Action: How to Follow Scientists and Engineers through Society (Harvard University press 1987).

28 Langdon Winner, 'Do Artifacts Have Politics?' (1980) 109 Daedalus 121.

29 Andrew Feenberg, 'Critical Theory of Technology: An Overview', Between Reason and Experience: Essays in Technology and Modernity (MIT Press 2010).

30 Judy Wajcman, Feminism confronts Technology (Polity Press 1991); Donna Haraway, 'Situated Knowledges. The Science Question in Feminism and the Privilege of Partial Perspective' (1988) Feminist Studies 14 (3), 575–599; Steven Yearley and Sandra Harding, 'Whose Science? Whose Knowledge?: Thinking from Women's Lives' (1993) British Journal of Sociology 44; Keith Grint and Rosalind Gill, The Gender-Technology Relation: Contemporary Theory and Research, vol 38 (Taylor & Francis 1997).

31 Felicitas Kraemer, Kees van Overveld and Martin Peterson, 'Is There An Ethics of Algorithms?' (2011) 13 Ethics and Information Technology 3; Brey, P. and Søraker, J. (2009). 'Philosophy of Computing and Information Technology' Philosophy of Technology and Engineering Sciences. Vol. 14 of the Handbook for Philosophy of Science. (ed. A. Meijers) (gen. ed. D. Gabbay, P. Thagard and J. Woods), Elsevier.

32 See e.g. Steven Umbrello and Ibo van de Poel, 'Mapping value sensitive design onto AI for social good principles' (2021), AI Ethics 1, 283–296.

They distinguished three types of bias: pre-existing (social) bias that precedes and impacts the design of code, technical bias embedded in code, and emergent bias (from the use context). They also suggested that the quality of computer systems should be assessed, considering criteria such as freedom from bias, reliability, accuracy and efficiency.[33]

Against this background, yet another cluster of literature emerged which draws attention to what we call "procedural design justice". Some scholars in this broad field advocate for participatory data collection and design.[34] Others in the "design justice" literature critique technology design on the grounds of structural inequalities and discrimination of marginalized groups. They object against universalistic principles and the practice of standardization underlying design that disadvantage certain groups.[35] Drawing on the literature on gender and technology, more recent approaches such as D'Ignazio and Klein's Data Feminism have emphasized how data science has settled as a practice excluding some already marginalized groups.[36]

Others focused on real-life cases of harm related to code and the uses of algorithms in sensitive contexts to document those and raise awareness about risks associated with algorithms and data analytics among data- and computer scientists. For example, much attention is devoted to the amplification of gender- and race-based discrimination through computer code. In particular, the quality of data or the choice of specific statistical variables, indicators and weights may exacerbate existing biases.[37] In response, more inclusive and socially representative design spaces and teams are proposed as strategies to ensure non-discrimination.[38]

(STS-informed) law and technology- and privacy scholarship have also produced relevant problematizations of code. For example, Philip Agre in his work on the capture model of privacy argued that design of any information system includes modelling human behavior by breaking it into smaller tasks (what he calls 'grammar of behaviour') and then imposes this 'grammar of behaviour' on users, whose behavior is shaped by this model.[39] Code only allows for certain behavioral possibilities (what philosophy of technology calls "affordances")[40] and disallows others ("disaffordances").[41] Thus, any computer code, even when not explicitly intended by its makers, will impact behaviour and can impact users' rights and interests. Relatedly, Lessig famously coined "Code is Law" and articulated it as a mode of regulation through architecture.[42]

Most recently, Diver argued that, to be acceptable in a democracy, code as a form of regulation should have certain formal characteristics, deriving from the ideas of legitimacy in legal philosophy.[43] As many social constructivists argued, designers play a key role in designing norms into code, which determine user behaviour.[44] As a society, we expect lawmakers to adhere to the values of legitimacy when introducing legal norms. In a similar fashion, when designers "legislate" the "norms" in the form of code, code design should meet standards of legitimacy, or create certain affordances such as (i) transparency of provenance, purpose, and operation, (ii) delay, (iii) choice, (iv) oversight, and (v) contestability, what Diver calls "digisprudence".[45]

With the rise of Artificial Intelligence (AI) enabled by Machine Learning (ML) as a subset of computer software, a rapidly growing body of literature has emerged that raises concerns and suggests solutions specific to the AI/ML context. Some focus on the problem of bias in ML, proposing solutions in the form of curation of data and the procedures around it (collection, cleaning, analysis, labeling). Several authors from law and ethics have addressed the issue of algorithmic bias by looking at the quality of data sets and their distribution.[46] Yet others critique the data-centered approaches to AI problems as limited and distracting from a broader picture of the AI-induced harms, proposing instead to refocus attention on a broader societal impact of ML and human rights.[47] There is yet another branch of the

33 Friedman, B. and Nissenbaum, H. (1996). Bias in computer systems. ACM Transactions on Information Systems (TOIS) 14(3): 330–347.

34 Catherine D'Ignazio and Lauren Klein, 'Data Feminism' (MIT Press 2020).

35 Design justice, as defined by Costanza-Chock in 2020, has roots in social movements and local practices, rather than universalistic principles which, the author argues, disadvantage certain groups of discriminated people in turn. According to several recent scholars in science and technology studies, discriminatory design is related to standardization. It starts with statistical "norms" that favor certain categories (less "marginal" because they represent the status quo and not outliers) (see e.g. Caroline Criado Perez, 'Invisible Women: Data Bias in a World Designed for Men' (Random House 2019); Sara Wachter-Boettcher, 'Technically Wrong: Sexist Apps, Biased Algorithms, and Other Threats of Toxic Tech' (W.W. Norton & Company 2017).

36 D'Ignazio and Klein (n 34) and Criado Perez (n 35).

37 E.g. Joy Buolamwini, Timnit Gebru, 'Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification' (Conference on Fairness, Accountability, and Transparency 2018) Proceedings of Machine Learning Research 81, 1–15; Cathy O'Neil 'Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy' (Crown 2016); Michael Rovatsos, Brent Mittelstadt, Ansgar Koene, 'Landscape Summary: Bias in Algorithmic Decision-Making' (2019) In What is bias in algorithmic decision-making, how can we identify it, and how can we mitigate it? UK Government http://Inhttps://www.gov.uk/government/publications/landscape-summaries-commissioned-by-the-centre-for-data-ethics-and-innovation Accessed 10 november 2023; Teresa Scantamburlo, Andrew Charlesworth, Nello Cristianini 'Machine Decisions and Human Consequences' In Karen Yeung, Martin Lodge (eds), Algorithmic Regulation (Oxford University Press 2019); Criado Perez (n35); Susan Leavy, 'Gender bias in artificial intelligence: the need for diversity and gender theory in machine learning' In Proceedings of the 1st International Workshop on Gender Equality in Software Engineering (GE 2018), Association for Computing Machinery, 14–16.

38 See also the European Fundamental Rights Agency has published guidelines on discrimination in data-supported decision making https://fra.europa.eu/en/publication/2018/bigdata-discrimination-data-supported-decision-making

39 Philip E Agre, 'Surveillance and Capture: Two Models of Privacy' (1994) 10 The Information Society 101., 110.

40 Donald A. Norman, The Design of Everyday Things (MIT Press 2013) 11.

41 Dan Lockton, 'Architectures of control in product design' (2006) Engineering Designer: The Journal of the Institution of Engineering Designers 28.

42 Lawrence Lessig, Code 2.0 (Basic Books 2006), 128.

43 Laurence Diver, 'Digisprudence: The Design of Legitimate Code' (2021) 13(2) Law, Innovation and Technology 325.

44 Wiebe E Bijker, Thomas Hughes and Trevor Pinch (eds), The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology; [Papers of a Workshop Held at the University of Twente, The Netherlands, in July 1984] (MIT Press 1987).; Diver (n 43).

45 Diver (n 43).

46 Seeta Pena Gangadharan, Virginia Eubanks, Solon Barocas 'Data and discrimination: Collected essays' (2014) Open Technology; Solon Barocas, Andrew D. Selbst, 'Big Data's Disparate Impact' (2016) California Law Review, 104(3), 671–732; Catherine D'Ignazio and Lauren Klein (n34); danah boyd, Kate Crawford, 'Critical Questions for Big Data' (2012) Information, Communication & Society, 15:5, 662-679; Aline Shakti Franzke, Iris Muis, Mirko T. Schäfer, 'Data Ethics Decision Aid (DEDA): A Dialogical Framework for Ethical Inquiry of AI and Data Projects in the Netherlands' (2021) Ethics and Information Technology 23, 551–567.

47 Alessandro Mantelero, 'Beyond Data: Human Rights, Ethical and Social Impact Assessment in AI', Information Technology and Law Series (Asser Press 2022); Elizabeth M. Renieris, 'Beyond Data: Reclaiming Human Rights at the Dawn of the Metaverse' (MIT Press 2022).

ML-focused legal and ethics literature critiquing this technology from the perspectives of transparency, non-discrimination, explainability and accountability, and suggesting respective solutions.[48] But these analyses – and related policy initiatives[49] - are highly specific to the ML-context and do not extrapolate to a broader discussion of regulating code as such.

We build on and contribute to the ongoing debates in at least three ways. (1) Our focus is on the broad category of computer software generally, in contrast to the sizeable and growing literature on bias, discrimination, accountability, etc. in AI. (2) Unlike a lot of the AI scholarship and the literature on design justice that problematize and propose solutions with regard to certain stages of code lifecycle such as design or data analysis and training, we focus on the entire lifecycle of code from inception to design, use and destruction. (3) Finally, like philosophers of technology and privacy and law and technology scholars, we embrace code as a manifestation of power and a mode of regulation that imposes certain (sometimes problematic) values on its users. Yet, instead of presuming that for this reason a code-specific regulatory intervention is warranted, we make the issue of the need for code-specific regulation vs technology-neutral regulation a central point of our analysis.

3. Why design and use of code needs to be regulated

Computer code does not get created or used in legal vacuum. Some general laws applicable to human behavior also apply to computer code. If someone will intentionally manipulate software running a medical device with intent to cause death of a patient, this will qualify as murder in many jurisdictions. Product safety requirements apply to the design of some code, e.g. when the code is a (part of a) medical device.[50] The question this section examines is not if code should be subjected to regulation as such, or be immune to it, but rather if that regulation should be code-specific.

3.1 Theoretical framework: regulatory theory and technology neutrality of law

Two bodies of scholarship inform our analysis. The first is regulatory theory, specifically, work of Schauer and Black on construction of (legal) rules. Legal rules - as any form of intentional regulation - are constructed to achieve certain outcomes, "the evil sought to be eradicated or the goal sought to be served".[51] These outcomes form a part of a legal rule called the justification[52] and determine further substance of rules.[53] More precisely, the substance of the rule is informed by the ideas the regulator has about how the world works, what causes the desired outcome and how the world needs to be manipulated to achieve the desired objective. Thus, a legal rule is based on "the idea of a causal process" leading to the desired objective.[54] A rule targets a part of that process that, according to a regulator, is the factor of causal relevance for the regulatory objective. Schauer and Black call this factor a factual predicate[55] or operative fact[56] of a rule. This means that for legal regulation to target the design and deployment of computer software, there should be something specific about software that is of causal relevance for the evils to be avoided or goals sought.

The second relevant theoretical framework is formed by the law and technology scholarship on technology neutrality of law. Relatedly to the regulatory theory-based insight that regulation should target phenomena in causal connection to what regulation should achieve, this second body of scholarship suggests that in order to be transparent and proportionate, accommodating and sustainable for technological change, regulation should generally be technology-neutral, i.e. focus "on the effects of actions"[57] rather than target a specific technology.[58] This is except for the cases where a specific technology raises specific moral objections,[59] or presents specific risks, and to provide equal legal protection in cases where the technology in question is and is not involved.[60] The next section argues that the design and use of computer software cause specific risks, and hence need to be regulated through technology-specific legislation.

3.2 Software-specific problems

Does computer software present specific problems that necessitate specific regulation? Yes and no. As Bennet Moses aptly notes, "[w]e tend to be more concerned about technological dimensions of what are in fact broader [societal] problems",[61] so problems discussed in the context of automation and computing are oftentimes a manifestation of broader social processes such as discrimination or social injustice. Similarly, several accounts of technology we discussed earlier in this paper consider technology, and therefore software, as a socio-technical practice where technology cannot be understood or tackled in isolation from society. Software, like any other practice, is social because it is produced by people who act based on their experiences and worldviews and are situated in social, political and economic contexts.[62]

48 E.g. Sandra Wachter, Brent Mittelstadt and Chris Russell, 'Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR' (2017) 31 Harvard Journal of Law & Technology, 841.

49 To name one such initiative, the EU High Level Expert Group (HLEG) on Artificial Intelligence proposed four ethical principles based on fundamental rights to achieve Trustworthy AI, namely (i) respect for human autonomy; (ii) prevention of harm; (iii) fairness; (iv) explainability. To ensure the practical implementation of Trustworthy AI, seven key requirements were identified: (i) human agency and oversight, (ii) technical robustness and safety, (iii) privacy and data governance, (iv) transparency, (v) diversity, non-discrimination and fairness, (vi) societal and environmental wellbeing, and (vii) accountability. This effort eventually led to the proposal of the EU AI Act. (High-Level Expert Group on Artificial Intelligence, 'Ethics Guidelines for Trustworthy AI' https://ec.europa.eu/futurium/en/ai-alliance-consultation.1.html.)

50 Regulation (EU) 2017/745 of the European Parliament and of the Council of 5 April 2017 on medical devices, amending Directive 2001/83/EC, Regulation (EC) No 178/2002 and Regulation (EC) No 1223/2009 and repealing Council Directives 90/385/EEC and 93/42/EEC, *OJ L 117, 1–175. Recital 19 clarifies that* "software in its own right, when specifically intended by the manufacturer to be used for one or more of the medical purposes set out in the definition of a medical device, qualifies as a medical device".

51 Frederick F Schauer, Playing by the Rules: A Philosophical Examination of Rule-Based Decision-Making in Law and in Life (Repr 2002, Clarendon Press 2002), 26.

52 Schauer (n 51).

53 E.g. Julia Black, 'Using Rules', Rules and regulators (Clarendon Press ; Oxford University Press 2012). E-book available online at oxfordscholarship.com.

54 E.g. Antony Honoré, 'Causation in the Law' in Edward N Zalta (ed), The Stanford Encyclopedia of Philosophy (Winter 2010, Metaphysics Research Lab, Stanford University 2010) https://plato.stanford.edu/archives/win2010/entries/causation-law/ accessed 3 November 2023.

55 Schauer (n 51), 27.

56 Black (n 53).

57 Bert-Jaap Koops, 'Should ICT Regulation Be Technology-Neutral' in Bert-Jaap Koops and others (eds), Starting Points for ICT Regulation, vol 9 (TMC Asser Press 2006). 77-108.

58 E.g. Lyria Bennet Moses, 'How to Think about Law, Regulation and Technology: Problems with "Technology" as a Regulatory Target' (2013) 5(1) Law, Innovation and Technology 1; Koops (n 57).

59 Bennet Moses (n 58).

60 Koops (n 57).

61 Bennet Moses (n 58). 17.

62 E.g. Kitchin and Dodge (n 23). 25 et seq.

For instance, optimization and optimization-related concerns might seem to be computing-specific. Optimization is said to be the underlying goal of computer science, as computing is engaged in order to solve problems more efficiently or using less resources,[63] e.g. find a solution to a mathematical problem faster or calculate a shorter route between two locations. In the domain of A, optimization is defined as "the best state according to an objective function"[64] that mathematically evaluates the desirability of an outcome.[65] Optimization is also a feature that underlies most of the algorithm-related concerns discussed in the context of algorithmic decision-making, e.g. optimization comes at the price of complexity and opacity of algorithms, exclusion, limited contestability, reducing humans to numbers, or limiting human agency.[66] At the same time, optimization in the sense of finding better solutions to problems faster has been a goal of mathematics pursued hundreds of years before computers,[67] and therefore is not a computing-specific phenomenon. Yeung similarly describes optimization as part of a broader societal context in which computing is used. The use of automation and algorithmic data analytics in public administration (what she calls New Public Analytics) is a manifestation of the New Public Management ('NPM') combined with techno-solutionism, both aimed at "improving" public services, a.o. through subjecting them to market rules, which is a manifestations of neo-liberalism.[68]

Similarly, Diver discusses immediacy of code as one of its key affordances and as something that is in its "very nature".[69] In contrast with the so-called text-based law (or any text-based rule) which allows a delay between representation of the rule, understanding of the rule, and a rule-based action, also referred to as a 'hermeneutic gap', when rules are translated into computational systems, the delay is eliminated, together with a possibility for a human to engage with and reflect on the rule.[70] At the same time, lack of reflection and consideration have also been said to characterise the "analogue" bureaucracy. While human bureaucrats in theory can make use of the inbuilt delay of the text-based policies to reflect on their meaning, they often do not do so for a number of reasons, including but not limited to no discretion left in the language of the rules or due to hierarchical pressure[71] and limited resources leading to the lack of time to engage in any meaningful reflections. According to Piliavsky, bureaucracy "violates our capacity to imagine, create, play, or even think clearly; and in so doing it infringes upon the very essence of what it means to be human".[72] Graeber writes: "Bureaucratic procedures... have an uncanny ability to make even the smartest people act like idiots. [...] [Bureaucracy] radically strips down, simplifies, and ultimately prevents communication ... it is [...] a form of anti-action".[73] Interestingly, as a metaphor to describe the core of the "database problem" intimately connected to computing, Daniel Solove uses Kafka's depiction of bureaucracy in *The Trial*, referring in particular to "a more thoughtless process of bureaucratic indifference, arbitrary errors, and dehumanization, a world where people feel powerless and vulnerable, without any meaningful form of participation in the collection and use of their information."[74] While the immediacy of code-based decisions leaving no space for reflection can be compared to a domino run where an individual falling domino has no say in the process but does enable the domino effect, individual bureaucrats often operate in a similar fashion, if not in terms of speed, certainly in terms of the automatism of a cog in the machinery of public and private bureaucracies.

Similarly, concerns related to code as a mechanism of power and control discussed earlier[75] are not unique to software and the digital context. Architecture as a mode of regulation and instrument of control is not limited to code but includes any designed objects or space.[76]

In other words, some characteristics of computer code that have been problematized in the literature are not technology-specific and are often an aspect of broader social phenomena such as prioritization of efficiency of (governmental) processes at the expense of the ensuing costs to citizens. What seems specific to code though is the degree of amplification that it provides to those broader societal problems. We explain this high degree of amplification by what we call the scalability of code and stickiness of outcomes produced or presented through code. Below we briefly explain both.

In information technology literature, scalability is understood as the ability of software to maintain performance even when the field of its application is expanded.[77] Scalability is one of the major computing tasks and software's ability to achieve an output that is applicable to a large number of cases at once is often the reason for software to be used. Software scalability has both a quantitative and a temporal dimension. The quantitative dimension refers to the fact that a rule or instruction, once translated into computer code, gains the ability to apply to and affect a large group of people. For example, a certain affordance of computer code structures the behaviour of its users

63 E.g. John S Conery, Explorations in Computing: An Introduction to Computer Science and Python Programming (CRC Press, Taylor & Francis 2015). 2 et seq. and multiple references to solutions to problems achieved by computation which are more or less optimal (e.g. 325, 334, etc.).

64 Stuart Russell, Peter Norvig 'Artificial intelligence: a modern approach' (Third edition, Upper Saddle River 2010)

65 Jonathan Stray, 'Aligning AI Optimization to Community Well-Being' (2020) International Journal of Communication 3, 443–463.

66 For a broad overview of concerns related to algorithmic decision-making see Margot E. Kaminski, Binagy Governance, 92 S. CAL. L. Rev. (identifying three categories of concerns: dignitary, justificatory, and instrumental).

67 Conery (n 63) 3 et seq.

68 Karen Yeung, 'The New Public Analytics as an Emerging Paradigm in Public Sector Administration' (2023) 27(2) Tilburg Law Review 1.

69 Laurence Diver, 'Computational Legalism and the Affordance of Delay in Law' (2021) 1 Journal of Cross-disciplinary Research in Computational Law 1.

70 Diver (n 69).

71 For instance, one of the key factors in the infamous child benefit scandal in the Netherlands was that the rules applicable in case of incorrect benefit applications did not allow the bureaucrats apply any discretion or indeed the principle of proportionality when minor mistakes or omissions in the application led to the withdrawal of the paid benefits in full. This point is discussed in Leo Damen, 'Ik was het niet, ik was het niet, het was de wetgever! | Navigator' (2021) 2021 Nederlands Juristenblad 354. 371 et seq.

72 Anastasia Piliavsky, 'The Wrong Kind of Freedom? A Review of David Graeber's The Utopia of Rules: On Technology, Stupidity and the Secret Joys of Bureaucracy (Brooklyn/London: Melville House, 2015, 261 Pages)' (2017) 30(1) International Journal of Politics, Culture, and Society 107.

73 David Graeber, The Utopia of Rules: On Technology, Stupidity, and the Secret Joys of Bureaucracy (Melville House 2015). The authors are also aware of the alternative views on bureaucracy, notably, Michael Lipsky, Street-Level Bureaucracy: Dilemmas of the Individual in Public Services (expanded ed, Russell Sage Foundation 2010).

74 Daniel J Solove, 'Privacy and Power: Computer Databases and Metaphors for Information Privacy' (2001) 53(6) Stanford Law Review 1393, 1398.

75 Supra note 29 and related text.

76 E.g. see examples of architecture-based regulation in Karen Yeung, 'Can We Employ Design-Based Regulation While Avoiding Brave New World ?' (2011) 3(1) Law, Innovation and Technology 1. Also think of Panoptic surveillance as a tool of social control. As the name referring to a certain design of prisons suggests, Panoptic surveillance does not have to be effectuated via code.

77 E.g. Liu explains scalability of software as the ability to maintain performance with increased load Henry Liu, Software Performance and Scalability: A Quantitative Approach (Wiley 2009), 1.

equally regardless of the users' numbers, be it one or a million. A traffic rule "translated" into a rules-based algorithm of a smart traffic camera (e.g. "drivers driving at a speed above X are in violation and ought to be fined"), when that algorithm is in action, equally affects all the drivers that speed on the relevant piece of the road. Be it one driver or a thousand, they all receive an automatically generated speeding ticket.

The temporal dimension refers to the fact that a rule translated into computer code, can be applied to multiple people *at the same time*, further contributing to the amplification. A good illustration of the qualitative and temporal dimensions of software scalability leading to the amplification of broader societal problems is the automation of administrative processes. Many governments employ automation to be able to enforce welfare policies at scale. For instance, Article 40 of the Dutch GDPR implementation act creates a broad exception from the GDPR prohibition of automated decision-making "on the basis other than profiling, (...) necessary to fulfil (...) a service in the common interest" since there is "no added value" of human involvement in automated decision-making on the basis of individual traits where the legal basis of such decisions leaves no discretionary space as in case of the allocation of certain social benefits.[78] In this context, hypothetically, if the government were to decide to question legitimacy of working mothers receiving a public benefit and withdraw the benefit, it could do so in a matter of seconds by using software and the indicators such as gender (female), number of children (>0), and employment status (employed). In this scenario, when an employed woman – regardless of a total number of affected individuals – becomes a mother, this policy is automatically and immediately effectuated against her and thousands like her. This has a scalable and immediate impact on a large portion of the population sharing the relevant characteristics, regardless of where they are and their specific history. This amplification, in the absence of software, would not have been possible.

Finally, another characteristic that is specific to the contexts where software is used and that amplifies broader societal problems, is that the outcomes produced or presented by software tend to stick, i.e. theyare not easily altered. When we outsource a task to software, we tend to blindly trust its outcome without questioning it. This comes as a result of a much-studied empirical phenomenon referred to as "automation bias".[79] Sociologist Thomas Veblen proposed the idea of technology as the initiator of all social transformations and therefore objectively unquestionable.[80] Automation bias occurs when a technological result is interpreted as pronounced by an "oracle" and an indisputable truth. Historically, the fact that software is based on mathematical logic and statistical models has given it a flair of rigour and objectivity, especially contrasted with human judgment that is often biased by emotions, culture, or values. Therefore, automation should in theory reduce human error and is often considered (while might not be) more reliable as the result of objective science expressed in computer code. Moreover, code-generated outcomes are often used to obfuscate human choices by the "computer said so"-type of arguments. While this over-reliance on automation risks eroding human control when it matters, especially in highly sensitive decision-making contexts, and leads to oversimplification and reduction of complexity and nuance, primarily it cements the outcomes of decisions encoded in software, and amplifies any societal problems that those decisions cause.

When imposed on broader societal problems such as bias and discrimination, these affordances of computer code create a "perfect storm" and amplify their non-technology specific effects. This amplification is code-specific and justifies technology-specific regulation of code.

4. Why principles and not rules?[81]

Considering that computer code is widely used and affects many aspects of our lives, introducing an elaborate regulation of its design and use akin to what the GDPR is for personal data will simply amount to yet another "law of everything" and therefore be counterproductive in relation to the reasons why we propose regulation of code, i.e. to build a scalable system of legal protection. While some software, e.g. as part of a medical device or a child's toy, should certainly be subject to comprehensive regulation to ensure that it is safe, we do not believe that this should be the case for all code. Instead, to balance the broad scope of the regulation (all code) with its low intensity, we propose framework legislation that sets out general principles for how computer code should be designed and used to address the problems described in Section 3.2.[82] These principles should be complemented by sectoral and more specific legislation of potentially problematic practices amplified by code, such as unfair consumer practices regulated via consumer protection law, public decision-making regulated in administrative and procedural law, etc.[83] The resulting system would be scalable since the more general and less compliance-intensive principles will apply to all code, and more intensive obligations would target specific problematic areas.

What do we mean by general principles? In legal and regulatory theory, there are many ways to define and distinguish principles from rules and the authors disagree on whether or not there is a fundamental difference between rules and principles. Some scholars such as Dworkin[84] and Alexy,[85] maintain that there is a principal difference between rules and principles. Both rules and principles equally point to particular decisions about legal obligations in particular circumstances, but they differ in the character of the direction they give. Rules offer a conclusive resolution of a situation ("[a] will is invalid unless signed by three witnesses") and principles are inconclusive as to the outcome ("[n]o man may profit from his own wrong").[86] Therefore, two rules cannot conflict with each other (in case of conflict, one of them must be invalid) and when principles conflict, the reasons behind those principles must be weighed against each other, one principle will prevail or be given priority and the principle that did not prevail will remain a valid principle of law.[87] Yet, for others

78 Kamerstukken II, vergaderjaar 2017–2018, 34 851, nr. 3, 120.

79 Linda J. Skitka, Kathleen L. Mosier, Mark Burdick, 'Does automation bias decision-making?' (1999) International Journal of Human-Computer Studies 51(5), 991–1006; Karen Yeung, 'Can We Employ Design-Based Regulation While Avoiding Brave New World ?' (2011) 3 Law, Innovation and Technology 1.; Daan Kolkman, 'The (in)credibility of algorithmic models to non-experts' (2022) Information, Communication & Society 25:1, 93-109.

80 Thomas Hauer, 'Education, Technological Determinism And New Media' (2017) International Journal of English and Literature 2: 239174.

81 We thank Douwe de Lange for his research support in writing this section.

82 A similar suggestion was made by Koops, who proposed to redesign data protection law into framework legislation in Koops (n 8), 259.

83 See generally Purtova and Newell (n 22).

84 Ronald Dworkin, Taking Rights Seriously (Harvard Univ Press 2001).

85 Robert Alexy, 'On the Structure of Legal Principles' (2000) 13 Ratio Juris 294.

86 Dworkin (n 84), 25.

87 Dworkin (n 84), 27, Alexy (n 85).

like Raz[88] and Sartor,[89] the difference between rules and principles is not as clear-cut. Both rules and principles are defeasible[90] and can conflict without losing validity.[91] They can be weighed against each other albeit the weighing works differently for these two types of norms.[92] What distinguishes legal principles from rules for Raz is their more general character: their subjects are not specified, their area of application is not narrowed by possible conflict with more specific rules, and principles prescribe highly unspecific actions.[93] Yet, another school of thought[94] does not see any principal difference between rules and principles at all and argues that what distinguishes rules and principles is a degree of specificity. A rule to a principle is what a blueprint is to a plan, the former "being a more detailed form" of the latter.[95] Where all three bodies of scholarship converge is that, where rules are specific, principles are general norms, and this is how we understand principles in this paper.

Two objections were usually made when we presented the idea to regulate computer code with general principles. The *first* objection is that general principles are non-binding and therefore regulation by such general principles amounts to self-regulation, which, when it comes to regulation of technology and other spheres, has a bad track record.[96] The *second* objection is that regulation by general principles which are not prescribing specific action in specific circumstances will lead to ununiform and unpredictable application and to legal uncertainty, which will leave too much room for maneuver to the industry and other actors and negatively impact the quality of legal protection. To the first objection, we answer that the general nature of principles does not imply that they are non-binding. The second objection requires a more elaborate response.

We have already argued that introducing a comprehensive regime of specific rules akin to the GDPR would defeat the purpose of the proposal, which was, among others, to resolve the problem of over-inclusive and unscalable regulation. However, in addition, we submit that considering the high complexity of the field of design and use of computer code and high (economic) stakes, regulation of computer code by highly specific legal rules will lead to increased uncertainty, while general principles combined with rules will likely deliver more legal certainty.

Our argument is based on legal theory and empirical regulatory research, in particular the works of Baldwin[97] and Braithwaite.[98] Baldwin objects to the positivist account of law where precision is key to the effectiveness of rules, captured best by Raz:

> *Since the law should strive to balance certainty and reliability against flexibility, it is ... wise legal policy to use rules as much as possible for regulating human behavior because they are more certain than principles and lend themselves more easily to uniform and predictable application.*[99]

Yet, according to Baldwin, the world is full of specific legal rules that do not work, and this is in part because those rules were drafted without account of how compliance with them is secured in regulatory practice.[100] According to Braithwaite, highly specific rules may work and deliver legal certainty in relatively simple and static contexts without high economic stakes, such as traffic rules.[101] However, contrary to the intuitive appeal of Raz' maxim, regulation by highly specific rules often has opposite effects and erodes legal certainty:

> *As the complexity, flux and the size of regulated economic interests increase, certainty progressively moves from being positively associated with the specificity of the acts mandated by rules to being negatively associated with rule specificity.* [102]

In other words, in situations which are complex, dynamic and involve considerable economic interests, the factors which arguably often go together, "[general] principles are more likely to enable legal certainty".[103] This is explained by the mechanics of compliance with and enforcement of the rules. According to Braithwaite's account which is more nuanced than what we can recapitulate here, every rule has a core where its meaning is clear and a penumbra where its meaning is more uncertain. The more complex and dynamic the context of the application of the rules is, a.o. as a result of technological developments, the larger the penumbra grows.[104] The economic stakes play an especially significant role in the growth of uncertainty, as these are exactly the economic stakes that motivate industry and other subjects of regulation with resources to become or engage legal entrepreneurs such as expensive and skilled consultancies and "play the penumbra game" to their advantage, to litigate "to expand the penumbra of one rule to slightly overlap the penumbra of another, creating compliant non-compliance."[105] This dynamic is illustrated especially well by the case of taxation and tax law.[106] As the regulator creates more rules to plug the resulting loopholes, they become "a set of sign-posts that show the legal entrepreneur precisely what they have to steer around to defeat the purposes of the law."[107] This creates another form of inequality where people without sufficient resources to afford sophisticated legal advice do not understand the rules and cannot structure their actions accordingly, while the rich take advantage of legal entrepreneurs to create and navigate the legal uncertainty. The rich also actively invest their resources in order not just to profit from

88 Joseph Raz, 'Legal Principles and the Limits of Law' (1972) 81 The Yale Law Journal 823.
89 Giovanni Sartor, 'A Formal Model of Legal Argumentation' (1994) 7 Ratio Juris 177.
90 Sartor (n 89).
91 Raz (n 88), 830.
92 Raz (n 88), 833.
93 Raz (n 88), 836-836.
94 Robert E Goodin, Political Theory and Public Policy (9. Dr, Univ of Chicago Press 1992); John Braithwaite, 'Rules and Principles: A Theory of Legal Certainty' (2002) 27 Journal of Legal Philosophy 47; Arend Soeteman, 'Rechtsbeginselen en positivisme!?' (2009) 38 Rechtsfilosofie en Rechtstheorie 5.
95 Goodin (n 94), 63.
96 E.g. Derek Wilding, 'Regulating News and Disinformation on Digital Platforms: Self-Regulation or Prevarication?' (2021) 9 Journal of Telecommunications and the Digital Economy 11; Dennis D Hirsch, 'The Law and Policy of Online Privacy: Regulation, Self-Regulation, or Co-Regulation' (2010) 34 Seattle University Law Review 439.
97 Robert Baldwin, 'Why Rules Don't Work' (1990) 53 The Modern Law Review 321.
98 Braithwaite (n 94).
99 Raz (n 88), 841.
100 Baldwin (n 97), 321.
101 Braithwaite (n 94), 53.
102 Braithwaite (n 94), 53.
103 Braithwaite (n 94), 53 and fn 32.
104 Braithwaite (n 94), 54.
105 Braithwaite (n 94), 54 et seq.
106 "Uncommon transactions that are taxed inappropriately become common as taxpayers discover how to take advantage of them." Braithwaite (n 94), 55, citing David A Weisbach, 'Formalism in Tax Law' (1999) 66 University of Chicago Law Review, 869.
107 Braithwaite (n 94), 56.

but to contrive change and complexity of the regulated context.[108] This hypothesis is confirmed by a study Braithwaite conducted comparing certainty and consistency of enforcement in nursing homes in Australia where care was regulated by general principles of the quality-of-life vs highly specific rules governing the provision of care in the nursing homes in the US. In line with the hypothesis, the assessment of care by inspectors was more consistent and in the spirit of the purpose of regulation to ensure quality of life of the homes' inhabitants than the assessment based on highly detailed rules.[109]

No dedicated empirical research has been done to study if and to what extent these patterns are present when the regulation of digital technologies and associated problems are concerned. Still, in the authors' opinion, the data protection law that has so far dominated this space certainly provides plenty of illustrations fitting these patterns that will likely reemerge if the regulation shifts towards highly detailed regulation of computer code. The context of regulation, i.e. digital transformation, is highly dynamic and involves high economic stakes, both in terms of the data-driven business models, compliance costs and high fines sanctioning non-compliance.[110] Technological developments by themselves are causing a lot of legal uncertainty and extend the penumbra of the rules as to how the GDPR applies, for instance, in case of the meaning of identification and identifiability,[111] or the assignment of controllership.[112] At the same time, the industry actors in the regulated field are highly motivated to contribute to and navigate the uncertainty through complex compliance schemes on the one hand and litigation on the other. One recent example of the former is the infamous "Transparency and Consent Framework" developed by the IAB Europe as a GDPR and ePrivacy compliance tool for behavioural advertising.[113] The so-called "confidentiality computing" and much of what is called privacy enhancing technologies are developed by the industry to "steer clear" from the sign-posts, to technically comply with the GDPR while still affecting people.[114] The latter is evidenced by the avalanche of the GDPR litigation involving Big Tech both on the EU and national level,[115] making data protection a highly dynamic field of law. Even when Big Tech lose in a case, they gain another sign-post to optimize their business strategies for. Consider the ongoing saga with Facebook (now Meta) relying, first, on consent, then on contract and possibly legitimate interest, and then again on consent to legitimize the processing of the personal data of its users for behavioural advertising.[116] Each decision of the data protection authorities striking down Meta's previous choice of a ground of lawful data processing did not stop Meta from processing personal data for behavioural advertising, but simply shifted their compliance strategy to the next yet untested ground. Should no ground of lawful processing under Article 6 GDPR turn out suitable, it is highly likely that the next move would be towards one or another form of confidentiality computing and an argument that behavioural advertising does not involve the processing of personal data at all. In this specific case, a general principle that no one should be targeted with commercial content based on one's behavior or traits would create far more certainty than the current regime of conditions and qualifications of the lawful processing of personal data. In the same vein, general and simple principles regulating design and use of computer code are more likely to deliver more consistency in application and legal certainty.

While general principles sometimes can benefit from specification in (public and contestable) rules, e.g. to provide a context-specific interpretation of the principles or help manage risks, these rules should never provide a "safe harbour" from responsibility for disrespecting the underlying principles. This is what Braithwaite calls "binding principles backing non-binding rules".[117] In case of regulation of code, these non-binding rules could be sectoral legislation,[118] public and publicly scrutinized codes of conduct or standards developed by the industry. Yet, a complementary and better way to achieve certainty and predictability in applying the general principles would be through institutionalized "regulatory conversation",[119] i.e. embedding in the regulatory process the creation of epistemic communities of the regulatees and enforcers that share an understanding of what those principles mean. In the case of computer code, if one of the principles is that computer code must be designed and used fairly and proportionately, it should become part of the computer scientists' curriculum to study the legal notions of procedural and substantive fairness and proportionality, while studying design, affordances and limitations of computer code should become part of the legal education

108 Braithwaite (n 94), 57.
109 Braithwaite (n 94), 60 et seq.
110 At the same time, doubts have been raised as to the persuasive effect of those fines. Consider a EUR 345 mln fine imposed on TikTok by the Irish Data Protection Commission 'Data Protection Commission' (Data Protection Commission) https://www.dataprotection.ie/news-media/press-releases/DPC-announces-345-million-euro-fine-of-TikTok accessed 6 November 2023. Compared to the reported USD 9,4 billion revenue for 2023 ('TikTok Revenue and Usage Statistics (2023)' (Business of Apps) https://www.businessofapps.com/data/tik-tok-statistics/ accessed 6 November 2023.).
111 See generally Purtova (n 20).
112 See generally Lilian Edwards and others, 'Data Subjects as Data Controllers: A Fashion(Able) Concept?' [2019] Internet Policy Review <https://policyreview.info/articles/news/data-subjects-data-controllers-fashionable-concept/1400> accessed 6 November 2023; Rene Mahieu, Joris van Hoboken and Hadi Asghari, 'Responsibility for Data Protection in a Networked World: On the Question of the Controller, Effective and Complete Protection and Its Application to Data Access Rights in Europe' (2019) 10 Journal of Intellectual Property, Information Technology and Electronic Commerce Law 84, and Benjamin Wong, 'Problems with Controller-Based Responsibility in EU Data Protection Law' (2021) 11(4) International Data Privacy Law 375.
113 Autorité de Protection Des données (Gegevensbeschermingsautoriteit) 'The BE DPA to Restore Order to the Online Advertising Industry: IAB Europe Held Responsible for a Mechanism That Infringes the GDPR' https://www.dataprotectionauthority.be/citizen/iab-europe-held-responsible-for-a-mechanism-that-infringes-the-gdpr accessed 6 November 2023.
114 E.g. Michael Veale, 'Confidentiality Washing in Online Advertising' in Corinne Cath-Speth (ed), Eaten by the Internet (Meatspace Press Forthcoming) https://osf.io/53ays accessed 1 November 2023.
115 As a rough indication, a search on https://curia.europa.eu for judgements and pending references for preliminary ruling in the area of data protection since May 2018 when the GDPR became effective till 01 November 2023 has rendered 98 hits. This does not include decisions of the European Data Protection Board and national data protection authorities or national litigation.
116 This saga is best documented in two binding decisions of the European Data Protection Board: European Data Protection Board, 'Binding Decision 3/2022 on the Dispute Submitted by the Irish SA on Meta Platforms Ireland Limited and Its Facebook Service (Art. 65 GDPR)' https://edpb.europa.eu/our-work-tools/our-documents/binding-decision-board-art-65/binding-decision-32022-dispute-submitted_en accessed 6 November 2023. and European Data Protection Board, 'Urgent Binding Decision on Processing of Personal Data for Behavioural Advertising by Meta' https://edpb.europa.eu/news/news/2023/edpb-urgent-binding-decision-processing-personal-data-behavioural-advertising-meta_en accessed 6 November 2023.
117 Braithwaite (n 94). 70 et al.
118 Purtova and Newell (n 22).
119 Braithwaite (n 94). citing Julia Black, 'Talking about Regulation' (1998) Spring Public Law 77.

of the future judges and legal councils.[120] This is just one example of how the regulatory conversations about general principles of code could look like.

We do not argue that the general principles governing computer code on their own will be sufficient to solve all the problems of the digital society. As proposed elsewhere,[121] these principles should come together with and underly sectoral legislation such as consumer, administrative, non-discrimination and civil- and criminal procedure law, that have been traditionally regulating certain societal contexts, now updated for the digital realities. There is room for sector-specific detailed rules applicable to certain types of code, such as product safety regulations applicable to code in and as consumer products and medical devices, or AI in this scheme. Explaining in further detail the exact interaction between the principles and sectoral legislation is beyond the scope of this paper. It suffices to note that the important element of this relationship should be that the general principles prevail over the rules of the sectoral legislation, which do not create "safe harbors" from responsibility when the general principles are violated.

5. Which principles? General data protection principles as a blueprint for general principles of code

The main thrust of this paper is that we need to govern software design and use and that it is best to be done by way of general principles rather than highly specific rules. Which set of principles can best address the problems that code presents needs further examination, which is beyond the scope of this paper.[122] Yet, we argue that it is certainly worth examining the potential of the general data protection principles to serve as a blueprint for the general principles of computer software, considering the long track record of the general data protection principles as a basic set of rules "for decent treatment of people" in an information society,[123] with their roots in the OECD data protection guidelines.[124]

The benefits of applying general data protection principles to software have already been demonstrated elsewhere.[125] Those principles can offer stronger protection compared to the status quo regulation of personal data and have added value compared to the AI Act, as they apply to software generally rather than AI, its narrow sub-type, and would govern the entire lifecycle of software from inception to deletion.[126] Yet, these principles need further development and adjustment to the context of code. Below is a very brief recap of how we envisage how these principles can be reformulated to better fit the software context.[127]

The principle of fair lawful and transparent data processing would translate into the principle of *fair, transparent and lawful design and use of computer software*. Fairness can mean both fairness of process (no deception or deployment without the user's or target's knowledge,[128] inclusion of the affected groups in the code design) and outcome. As is the case with the principle of fairness in data protection, the principle of fair design and use of software would be a flexible and powerful tool to tackle new and unpredicted challenges posed by software in new situations. Granted, fairness is a very broad concept which is difficult to define. There can be no absolutely fair procedures,[129] e.g. procedures that favor no one. Yet, the imperfect choices still can be justified for a certain context. Finally, the principle of fairness would serve to outlaw the use of software for practices that are considered unfair.

Transparency would translate into the requirement of *transparency* of the code's design, as well as the fact and purposes of its use. *Lawfulness* would require both that the design and use of code follow existing legal norms.

The principle of *purpose limitation* would translate into the requirement that computer software is only designed and used for specific and legitimate purposes and not repurposed in ways incompatible with the original purpose, where the affordances of code appropriate for the original context are transposed into the new context without thinking.

The principles of data minimization and storage limitation would translate into the general principle of *proportionality* in code's design and deployment, limiting both to what is necessary and proportionate for the declared legitimate purposes, forbidding excessive and disproportionate code uses (is automation necessary and proportionate for a given purpose?) and functionalities (are the features of code necessary and proportionate for a given purpose?).

The principle of *accuracy* would entail that computer code performs as intended and that its design and use are based on sound science rather than marketing claims. The principle of *integrity and confidentiality* would translate into the requirement that code is designed and used with cybersecurity in mind.

Finally, the principle of *accountability* would mean accountability for the design and use of software and an obligation to document and justify relevant choices.

120 Mireille Hildebrandt, 'Grounding Computational "Law" in Legal Education and Professional Legal Training' in Bartosz Brożek, Olia Kanevskaia and Przemysław Pałka (eds), Research handbook on law and technology (Edward Elgar Publishing 2023).
121 Purtova and Newell (n 22).
122 Consider this approach to designing principles governing AI: Laura Weidinger and others, 'Using the Veil of Ignorance to Align AI Systems with Principles of Justice' (2023) 120 Proceedings of the National Academy of Sciences e2213709120.
123 Koops (n 8), 258.
124 OECD, Recommendation of the Council concerning Guidelines Governing the Protection of Privacy and Transborder Flows of Personal Data, adopted on 23/09/1980, amended on 11/07/2013.
125 Purtova and Leenes (n 17).
126 Purtova and Leenes (n 17), Section 5.
127 The analysis below is a recap of the argument presented in one of the authors' earlier work.
128 Cécile de Terwangne 'Article 5 Principles relating to processing of personal data', in Christopher Kuner and others (eds), The EU General Data Protection Regulation (GDPR): A Commentary (New York, 2020; online edn, Oxford Academic); Purtova and Leenes (n 17).
129 Toon Calders, Sicco Verwer, 'Three naïve Bayes approaches for discrimination-free classification' (2010) Data Mining and Knowledge Discovery 21, 277–292.; Solon Barocas, Andrew D. Selbst, 'Big data's disparate impact' (2016) California Law Review 104:671; Jon Kleinberg, Sendhil Mullainathan, Manish Raghavan, 'Inherent Trade-Offs in the Fair Determination of Risk Scores' In 8th Innovations in Theoretical Computer Science Conference (ITCS 2017). Leibniz International Proceedings in Informatics (LIPIcs), Volume 67, pp. 43:1-43:23, Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

6. Conclusion

While a lot of regulatory attention has been drawn to regulation of data and AI, in this piece, we have argued that regulation of all computer software in the form of general binding principles should form a foundation of the scalable and more targeted system of legal protection of people against harms of the digital society.

We first laid a theoretical foundation for our argument, which is the regulatory- and regulation of technology theory. According to these bodies of scholarship, regulation is based on an approximation of reality and should target phenomena that are causally relevant to the desired regulatory outcomes. Technology, while it often attracts regulatory attention, is often just a dimension of broader societal processes and, therefore, should be subjected to specific regulation only under a limited set of circumstances, such as when it presents a technology-specific problem.

We examined a range of concerns associated with the design and use of computer software. We concluded that, while many of those concerns indeed are a manifestation of broader societal phenomena from discrimination to neoliberalism and are not unique to the context of computation, the scalability of software and stickiness of code-generated outcomes amplify non-technology-specific problems in a unique way. An affordance of software is its scalability, i.e. ability to maintain performance when the field of its application is expanded. This is how a problematic rule or practice - when embedded in code - can reach a large number of people without significant additional loss in performance and do it immediately. The rules embedded in computer code and code-generated outcomes are often perceived as backed up by science and hence objective and thus become harder to change. Scalability and stickiness make computer code a unique amplifier of non-digital societal problems and thus warrant technology-specific regulation.

Yet, such regulation should not take the shape of an elaborate regime of specific rules akin to how the GDPR currently regulates the processing of personal data. Considering how widely computer code is used, such a regime is bound to become another "law of everything" defeating the objective of the proposal to create a scalable alternative to the all-encompassing GDPR. But also, and importantly, general principles are a more suitable form of regulation in the field of digital society, which is highly complex, dynamic and involves high economic stakes. As regulatory studies have shown, regulating such contexts with specific rules will deliver uncertainty and uneven and unpredictable legal protection. The practice of data protection law illustrates how specific rules encourage legal entrepreneurship, "playing the game", legal complexity and uncertainty, favouring actors with resources while not necessarily delivering the needed protections. Should the practice of design and use of computer code be regulated by a similar regime, the pattern would likely repeat. Framework regulation of code through general principles would avoid this problem and, when combined with sectoral regulation of problematic practices, deliver a scalable system of legal protection.

Figuring out which general principles should govern the design and use of software was beyond the ambition of this paper. Yet, we proposed that, considering a long track record of fair information principles, currently reincarnated in the general principles of data protection, in setting a standard of decent treatment of people in a digital society, these are a good place to start.

Finally, we are not oblivious to the fact that regulating code presents its own challenges that will have to be resolved. To name a few, considering modern modular and software-as-a-service approaches to software engineering,[130] should the principles be applied to stand alone software or also its components which can be supplied by different actors? How should responsibilities be distributed among those actors? To what extent some principles such as purpose limitation are applicable to code of general use capable of versatile tasks, a debate resembling the one currently going on in the context of regulation of AI and the so-called foundation models.[131] Considering the low intensity of compliance with the general principles, we are inclined to argue in favour of the answers supporting broad application of the principles, albeit this is also subject to further research. Yet, while these issues are important, in our opinion, they are secondary to the primary conclusion of this paper that design and use of computer code should be regulated through general principles. Taking this next step towards better legal protection in a digital society will undoubtedly be challenging, but necessary.

Aknowledgements

This article reports on the results of the project 'Understanding information for legal protection of people against information-induced harms' ('INFO-LEG'). This project received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 716971). The paper reflects only the authors' views, and the ERC is not responsible for any use that may be made of the information it contains. The funding source was not involved in study design, in the collection, analysis and interpretation of data, in the writing of the report, and in the decision to submit the article for publication.

130 E.g. Nadezhda Purtova, 'eHealth Spare Parts as a Service: Modular eHealth Solutions and Medical Device Reform' (2017) 24(4) European Journal of Health Law 463.

131 E.g. Kai Zenner, 'A Law for Foundation Models: The EU AI Act Can Improve Regulation for Fairer Competition - OECD.AI' https://oecd.ai/en/wonk/foundation-models-eu-ai-act-fairer-competition accessed 8 November 2023.