Smart Contracts and the Digital Single Market Through the Lens of a "Law + Technology" Approach

DR. THIBAULT SCHREPEL, LL.M.

Internal identification

Contract number: LC-01637682
VIGIE number: VIGIE 2021-438

EUROPEAN COMMISSION

Directorate-General for Communications Networks, Content and Technology
Directorate F — Digital Transformation
Unit F3 — Digital Innovation and Blockchain

Contact: CNECT-F3@*ec.europa.eu*

European Commission
B-1049 Brussels

Smart Contracts and the Digital Single Market Through the Lens of a "Law + Technology" Approach

Directorate-General for Communications Networks, Content and Technology
2021 EN

LEGAL NOTICE

PDF	ISBN: 978-92-76-41173-4	doi:10.2759/562748	Catalogue number: KK-09-21-375-EN-N

Manuscript completed in September 2021

1st edition
The European Commission is not liable for any consequence stemming from the reuse of this publication.
Luxembourg: Publications Office of the European Union, 2021

TABLE OF CONTENTS

EXECUTIVE SUMMARY

The deployment of smart contracts within the European zone could fluidify economic transactions. It also risks fragmenting the Digital Single Market ("DSM"). This conundrum calls for a constructive response to preserve both the benefits brought by smart contracts and a strong DSM. Against this background, this report adopts a "law + technology" approach. It suggests combining law and technology to develop solutions that encourage the evolution of smart contracts (rather than hindering it) in a direction that preserves and reinforces the DSM.

The **first chapter** explains the ins and outs of the "law + technology" approach based on the teachings of evolutionary biology. I explain that it requires considering smart contracts as a technological species composed of different varieties. Doing so allows stakeholders to better anticipate the opportunities, problems, and possibilities created by smart contracts. The "law + technology" approach also calls for an analysis of the characteristics of smart contracts' environment to influence their evolution.

With that in mind, the **second chapter** is dedicated to smart contracts themselves. After studying their origin, I investigate their recent evolution from which it becomes apparent that smart contracts have common characteristics that have persisted over time because they are necessary for their survival. First, they use advanced encryption techniques, and second, they are decentralized and distributed, leading to immutability. But smart contracts also vary in the different forms they take, depending on their nature, their mode of activation, their use, and storage. In the end, these common characteristics and particularities define smart contracts' survival capacity. Their interactions with each other and with the outside world (through oracles) indeed depend on their ability to maintain and use their unique properties.

The **third chapter** is devoted to smart contracts' environment, i.e., its characteristics, how it influences smart contracts, and how it can be shaped to preserve and strengthen the DSM. The analysis first focuses on existing legal rules. I study the legal uncertainty and regulatory barriers that surround smart contracts' legality. I show they are mostly related to (i) the intention to create a contractual obligation and (ii) the form requirements that are not standardized across the European space. Next, I explain that the interpretation of smart contracts also poses new difficulties, as it requires technical expertise to inform legal analysis. Also, the performance of smart contracts raises questions regarding the ability to stop or force their execution. On top of that, I show that the cross-border nature of smart contracts reinforces all these different challenges. Since it seems possible to qualify all smart contracts as cross-border, these issues appear to be pressing.

Besides the rules of contract law, the report analyzes the potential legal barriers created by additional legal rules, such as data protection, data localization, and the mechanisms currently discussed with the "AI Act." I contend that, despite all existing regulations, new solutions are needed to strengthen the DSM. On that basis, the report proposes new mechanisms allowing for an ex-post "law is code" approach. The aim is to enable legal enforcement by modifying the technical environment of smart contracts while preserving their fundamental characteristics (i.e., their means of survival). Doing so may require the creation of legal templates and comfort zones.

The last part of the report questions the extent to which technology could strengthen the DSM. It suggests implementing a "practical immutability" to enforce legal rules and standards Further, it shows that technology can complement these rules. On the one hand, the report details how artificial intelligence could help translate smart contracts' code into natural language and how oracles could help with the compliance of smart contracts. On the other hand, blockchain and smart contracts could reinforce the databases' integrity to train AI systems. In the end, implementing these solutions will only be possible should the unique characteristics of smart contracts be maintained. It requires that regulators pursue this objective and that technologists open the doors to legal enforcement.

RÉSUMÉ EXÉCUTIF

Le déploiement des contrats intelligents (« smart contracts ») au sein de la zone européenne peut fluidifier les transactions économiques. Il peut également fragmenter le marché unique numérique (« MUN »). Ce dilemme appelle à des solutions qui visent à préserver les bénéfices apportés par les contrats intelligents tout en assurant un MUN robuste. Dans ce contexte, le présent rapport adopte une approche « droit + technologie » (« law + technology »). Il suggère de combiner le droit et la technologie afin d'élaborer un plan d'action qui encourage l'évolution des contrats intelligents (plutôt que de l'entraver) dans une direction qui préserve et renforce le MUN.

Le **premier chapitre** explique les tenants et aboutissants de l'approche « droit + technologie » sur la base des enseignements de la biologie évolutive. Il explore la nécessité de considérer les contrats intelligents comme une espèce technologique composée de différentes variétés. Cette approche permet aux parties prenantes de mieux anticiper les opportunités et les problématiques créées par les contrats intelligents. Elle appelle également une analyse des caractéristiques de l'environnement des contrats intelligents pour influencer leur évolution.

Dans cette optique, le **deuxième chapitre** procède à une étude des différentes formes de contrats intelligents. Il analyse leurs origines et évolutions récentes dans le but de faire émerger des caractéristiques communes qui ont persisté dans le temps. Le rapport met ainsi l'accent sur les techniques de cryptage utilisées au sein des contrats intelligents, ainsi que sur leurs méthodes de décentralisation et de distribution qui conduisent à leur immutabilité. Dans le même temps, les contrats intelligents présentent de nombreuses variations liées à leur forme, leur nature, leur mode d'activation, et leur stockage. Ces caractéristiques communes et particularités définissent leur capacité de survie.

Le **troisième chapitre** est consacré à l'environnement des contrats intelligents, à ses caractéristiques, à la manière dont il les influence et à la façon dont il peut être façonné afin de préserver et renforcer le MUN. L'analyse porte d'abord sur les règles juridiques existantes et conduit le lecteur à explorer l'incertitude juridique ainsi que les obstacles réglementaires qui conditionnent la légalité des contrats intelligents. En l'absence d'une normalisation juridique au sein de l'espace européen, (i) l'intention de créer une obligation contractuelle et (ii) les exigences de forme sont au centre de nombreuses problématiques. Par ailleurs, l'interprétation des contrats intelligents pose des difficultés nouvelles dans la mesure où elle nécessite une expertise technique afin d'éclairer l'analyse juridique. De plus, l'exécution des contrats intelligents soulève des questions quant à la capacité d'arrêter ou de forcer leur exécution. Enfin, la nature transfrontalière des contrats intelligents s'accompagne d'obstacles juridiques supplémentaires qui appellent à une réponse coordonnée au sein de l'espace européen.

Outre les règles de droit des contrats, le rapport analyse les difficultés d'adoption créées par d'autres règles juridiques, dont la protection et localisation des données, ainsi que les mécanismes actuellement inclus dans le « AI Act » de la Commission européenne. Il en ressort qu'en dépit des règles existantes, de nouvelles solutions juridiques sont nécessaires pour renforcer le MUN. Sur cette base, le rapport propose des mécanismes nouveaux permettant une approche ex-post « law is code » (« droit est code »). L'objectif est de permettre l'application du droit en modifiant l'environnement technique des contrats intelligents tout en

préservant leurs caractéristiques fondamentales (c'est-à-dire, leurs moyens de survie). Le rapport explore la création de modèles juridiques et de zones de confort afin de tester différentes solutions.

La dernière partie du rapport s'interroge sur l'utilisation de la technologie afin de renforcer le MUN. Il suggère la mise en place d'une « immuabilité pratique » permettant de faire respecter les règles et normes juridiques. Par ailleurs, il procède à l'analyse de diverses solutions techniques permettant de compléter ces règles. D'une part, le rapport détaille comment l'intelligence artificielle pourrait aider à traduire le code des contrats intelligents en langage naturel, et comment les oracles pourraient aider à la conformité de ces derniers. D'autre part, la blockchain et les contrats intelligents pourraient renforcer l'intégrité des bases de données servant à l'entrainement des systèmes d'intelligence artificielle. Le rapport conclut sur la nécessité de préserver les caractéristiques uniques des contrats intelligents afin que ces solutions puissent être mises en œuvre. Cela nécessite que les régulateurs poursuivent cet objectif et que les communautés décentralisées ouvrent les portes de leur blockchain à la règle de droit.

1. Introduction

Blockchain is a transactional infrastructure whose potential is well documented. Against this background, the European Commission (hereinafter "Commission") is asking me to write this report on the assumption of a fully automated environment where "machines will transact with machines through smart contracts."

I have been assigned two objectives. First, I shall study "whether smart contracts could lead to a fragmentation of the Digital Single Market" (hereinafter "DSM") "when used in commercial relationships instead of contracts written in prose." Second, I shall "differentiate between the legal rules that could help develop smart contracts within the DSM versus those that create artificial barriers." In the end, I am tasked to study what environment would be the most conducive to strengthening the DSM with the use of smart contracts.

The scope of the endeavor leads me to take a legal and technological look at the subject. As I shall explain, tensions appear between the law and the technology in the field; however, they are also complementary. I have, therefore, chosen an approach that I would like to describe as "law + technology" (**1**). Rather than studying law and (then) technology from a purely confrontational angle, I explore how the Commission could combine them to increase the common good. As I should like to explain, this implies looking at the different varieties of smart contracts (**2**) before focusing on how to shape their environment (**3**).

2. Approach: why "law + technology"

Looking at smart contracts through the lens of an evolutionary perspective helps understand and capture the dynamism and complexity of the ecosystem (**1.1**).[1] When adopting such a perspective, it appears that two key elements of smart contracts' environment, the law and the technology, call for cooperation to maximize the chances of emergence and fruition of smart contracts (**1.2**).

2.1. An evolutionary perspective

Arthur's and Darwin's bodies of work are complementary: one accounts for the emergence of new technologies, the other helps understand their evolution (**1.1.1**). Complexity theory completes the picture by explaining how technologies affect their environment and vice versa (**1.1.2**).

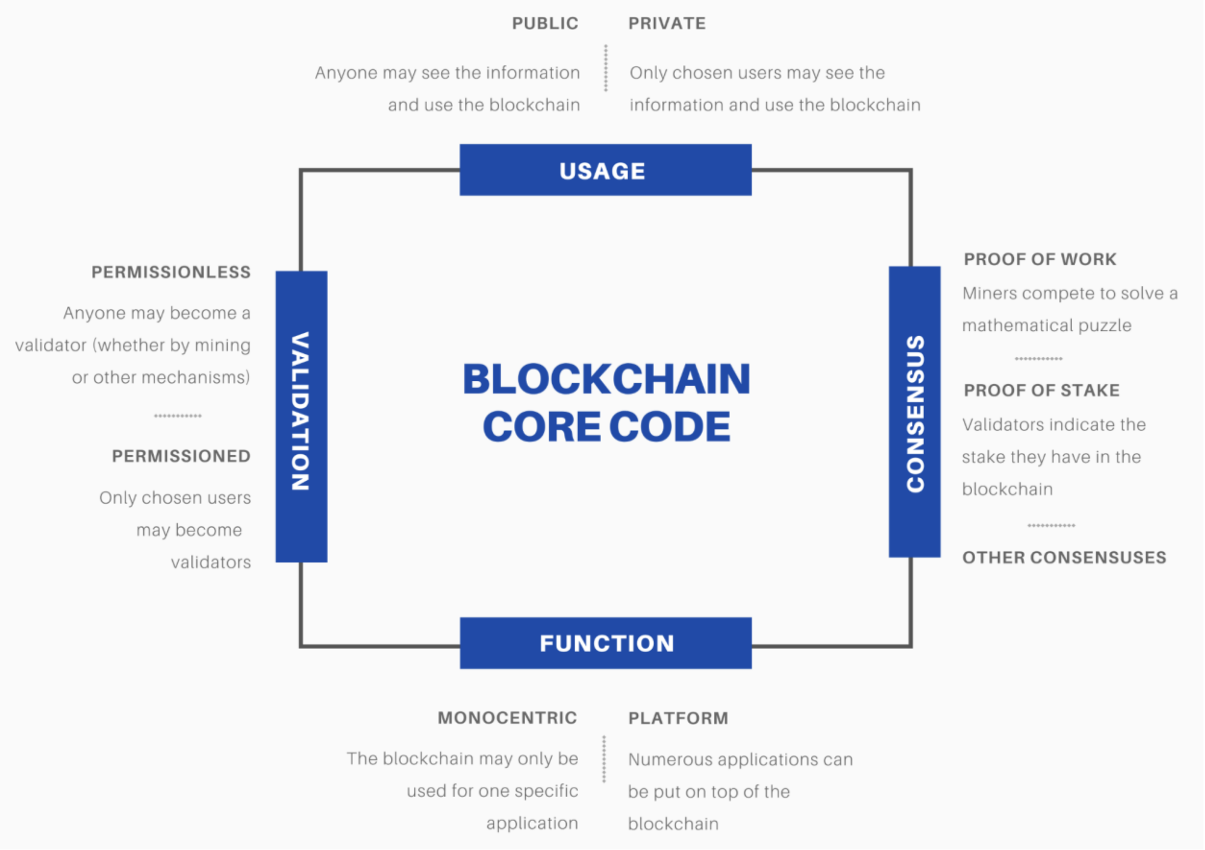
2.1.1. Arthur and Darwin

Blockchain results from a combination of existing technologies.[2] Blockchain is, in this sense, comparable to other technologies. W. Brian Arthur explains this evolutionary — but not Darwinian — process: "Novel technologies must somehow arise by combination of existing technologies. (...) Some technologies-the laser, the jet engine, radar, the Quicksort computer algorithm, the railroad locomotive itself-just appear, or at least they seem to just appear, and unlike novel biological species, they are not versions of earlier objects. (...) What we should really be looking for is not how Darwin's mechanism should work to produce radical novelty in technology, but how 'heredity' might work in technology."[3] The Bitcoin White Paper, for example, refers to and makes use of past research, concepts, and techniques.[4] The combination of these pre-existing elements gave rise to blockchain.

Once a new class of technology has emerged, a Darwinian process of natural selection follows. The technology — a species — moves in several directions simultaneously, leading to the emergence of different varieties. These varieties compete through the elements that differentiate them. Only a few make it as "the struggle [for survival] almost invariably [is] most severe between the individuals of the same species, for they frequent the same districts, require the same food, and are exposed to the same dangers."[5]

[1] "Complex" should not be confused with "complicated." In a complicated system, one cannot remove a single element without leading to its collapse. In a complex system, each element affects the entire ecosystem; removing one element will lead to the emergence of a different system, if any, *see* William Brian Arthur et al., *Complexity Economics: Dialogues of The Applied Complexity Network I* 41 (SFI Press, 2020) (stressing that the environment is not an externality).
[2] Thibault Schrepel, *Blockchain + Antitrust: The Decentralization Formula* 12 (Edward Elgar Publishing, 2021); Also, Arvind Narayanan & Jeremy Clark, *Bitcoin's Academic Pedigree: The Concept of Cryptocurrencies is Built from Forgotten Ideas in Research Literature*, 1 ACM QUEUE 1 (2017).
[3] William Brian Arthur, *The Nature of Technology: What It Is and How It Evolves* 17, 19 (Free Press, 2009).
[4] *See* Satoshi Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System* (2008) (quoting, among others, W. Dai's "b-money," and Ralph C. Merkle's "Protocols for public key cryptosystems").
[5] Charles Darwin, *On the Origin of Species* 75 (1859, Oxford University Press, 2008).

This analogy applies to blockchain. Different varieties of blockchains are being deployed. Some are public (any individual can use them); others are private. Some are permissionless (any individual can help validate transactions); others are permissioned. Different variants regularly appear while others are abandoned. "Natural selection is daily and hourly scrutinising, throughout the world, every variation, even the slightest; rejecting that which is bad, preserving and adding up all that is good."[6]



A look at blockchain varieties
© Thibault Schrepel[7]

The variants that survive will multiply and seek to expand their territory. When doing so, they will make contact with other species and start competing with them. Their ability to survive this new form of competition will depend on their ability to utilize the characteristics that differentiate them from other species. Depending on their environment, some of these characteristics will prove more or less valuable. The process is very much "utilitarian."[8] Accordingly, it requires monitoring how the environment evolves and identifying factors of change. Some are natural, while others are manufactured (i.e., caused by human intervention). Some result from gradual events (e.g., the erosion of rocks), while others are more sudden (e.g., the random appearance of another species on the territory).

[6] *Id.* at 84.
[7] Schrepel, *supra* note 2, at 37.
[8] Darwin, *supra* note 5, at 199.

Blockchain is just beginning to compete with centralized transactional means; see, for instance, what is happening with cryptocurrencies versus fiat money.[9] The competition that is initially strong between the varieties of blockchain is verging on a competition between species (blockchain vs. centralized ecosystems). Blockchain will only survive if it retains strong elements of differentiation to gain a competitive advantage over other species in a given environment. Absent these elements, no one will want to use blockchain instead of centralized transactional means. Against this backdrop, the present study aims at exploring how a technical and legal environment could reinforce the DSM, while not eliminating the means of survival for blockchain.

2.1.2. The complexity of blockchain evolution

Species and varieties influence their environment. The living organisms that survive the natural selection process consume the resources available within the area. Their activities contribute to soil erosion and the displacement of other species, to name a few. "Complexity theory" captures this logic and explains how systems respond to the context they create.[10] Transposed to smart contracts, complexity theory leads us:

- First, to consider the attributes of smart contracts from a technical angle; (Chapter 2).
- Second, to analyze how smart contracts compete and cooperate (between themselves but also with more centralized transactional methods), and how these interactions impact them in the long run (i.e., survival of the fittest); (Chapter 2).
- Third, to study the characteristics of their environment(s). This involves identifying what the law allows,[11] what blockchain allows, what the economy allows, and what the norms allow;[12] (Chapter 3).
- Fourth, to identify how the different dimensions of smart contracts environment collide and thus create dynamism; (Chapter 3).
- Fifth, to explore how the dynamism between organisms affects their environment and how the dynamism of their environment affects these organisms. (Chapter 3).

In the end, complexity theory helps us understand the emerging order of an ecosystem and how to intervene without hindering its chances of survival.

2.2. A harmonized "law + technology" approach

The environment of smart contracts has legal (i.e., soft law, regulations, case-law, etc.) and technical dimensions (i.e., the blockchain).[13] They must be combined. In the absence of cooperation between law and technology, these two aspects would battle to take the upper

[9] Jose Antonio Lanz, *El Salvador Passes Law to Make Bitcoin Legal Tender*, DECRYPT (Jun. 9, 2021), https://decrypt.co/73125/el-salvador-approves-bitcoin-law-btc-is-legal-tender [https://perma.cc/74ZJ-ZSU3].
[10] Arthur et al., *supra* note 1 ("[i]n the complex-systems view, the environment is not an externality"). One should underline that William Brian Arthur is actually one of complexity theory's first proponents.
[11] Here too, we must analyze the interactions between the different components of the law, for example, between the regulations, recommendations, jurisprudence, etc., which form a network, *see* Jamie Murray et al., *Complexity Theory and Law Mapping an Emergent Jurisprudence* 23 (Routledge 2018). This leads us to fractals.
[12] This quadriptych has been developed by Lawrence Lessig, *see* Lawrence Lessig, *Code: Version 2.0* 123 (Basic Books, 2006).
[13] This environment has other dimensions too, such as economic, social, philosophical, etc.

hand.[14] One would have to succeed before the other takes over. This would push smart contracts to develop under the primary influence of either law or technology, depending on which one dominates during a given period. These smart contracts would acquire unbalanced characteristics, for example, by ignoring legal constraints altogether. Their chances of survival would diminish as soon as the law or technology dominates again. A more cooperative (**1.2.1**) and harmonized (**1.2.2**) approach to law and technology is thus preferable so that smart contracts can grow in a cohesive and long-lasting environment.

2.2.1. Two approaches

Two approaches can inform the design of smart contracts' environment. The first is absolutist. It consists of creating legal rules without looking for ways to lean on the technology. It also involves enforcing legal rules and standards without seeking to preserve the elements of differentiation necessary for the survival of the technology.[15] Apropos, it is not uncommon for the law to eliminate (sometimes unintentionally) the support that a technology could provide when tackling its negative aspects.[16] This type of intervention creates a strong disturbance in the environment.

The Commission is asking me to study "which types of legal barriers of substance and form could lead to a fragmentation of the Digital Single Market if smart contracts were used across borders in commercial relationships instead of contracts in prose." The question shows the Commission's awareness regarding that risk. I will then explore different solutions to the challenges created by smart contracts, while striving to endorse the ones that do not put them at risk.

Technical fundamentalism consists of designing technology without relying on legal rules, and sometimes is a way of avoiding it. This approach leads to creating "temporary autonomous zones" (also called TAZs, i.e., zones in which the state cannot enforce all its power).[17] It also leads to compromising the technology's chances of survival. As soon as the technology extends its territory and leaves the TAZ, the application of the law can lead to the technology extinction.[18] The same fundamentalism comprises the rejection of any technical modification under the pretext that it contradicts some founding principles, such as those extolled by Satoshi Nakamoto.

The second approach is cooperative; I would like to call it "law + technology". Here, law and technology complement one another by seeking to preserve their sphere of influence and building on the other's strengths. This approach requires maintaining blockchain's distinctive features (i.e., its chances of survival) while permitting legal enforcement. As Vitalik Buterin puts

[14] The word "law" is used as synonymous with "legal rules and standards".
[15] *See* Thibault Schrepel, *Law and Technology Realism*, MIT COMPUTATIONAL LAW REPORT (Aug. 17, 2020) https://law.mit.edu/pub/lawandtechnologyrealism/release/3 [https://perma.cc/S4J3-ZFRX].
[16] *See id.*; for an example, *see* Michal S Gal & Oshrit Aviv, *The Competitive Effects of the GDPR*, 16 J. COMP. L. & ECON. (2020) (explaining that the GDPR has caused anticompetitive effects such as limiting competition in data markets, therefore preventing a new form of competition offered by technicalities).
[17] Peter Ludlow, *Crypto Anarchy, Cyberstates, and Pirate Utopias* 401 (MIT Press, 2001).
[18] Peer-to-peer file sharing makes a good example of a technology which grew within a TAZ before being tamed by regulation for the greatest part.

it, "[p]rinciples have to serve a social purpose"[19] while "fundamental values, pushed to such extremes, are silly."[20]

Where possible, the "law + technology" approach also leads to mutual support. For example, one can use smart contracts where contract law is difficult to enforce (e.g., because jurisdictions are unfriendly).[21] Smart contracts can also be used, as I shall explain later, where the law cannot achieve an aim on its own,[22] such as preventing corruption.[23] Smart contracts can help, but only if their core characteristics (such as immutability) are preserved. As a result, the "law + technology" approach leads to creating a more harmonious environment where the different varieties experience fewer shocks threatening them with sudden extinction because of a legal or technological takeover.

2.2.2. The need for uniformity

The cooperation of law and technology requires a uniform approach across the European space. Should one member state choose a confrontational path and impact the layer 1 of a blockchain,[24] that state would impose its choice on all blockchain users, regardless of their location. Unlike centralized products and services, no participant has the power of command and control in public permissionless blockchains. There is not just one pilot in the cockpit.[25] This means that such blockchains cannot easily decide to comply with a regulation in only one member state, as a company can do. The "Brussels effect" (i.e., the "unilateral regulatory globalization, where regulations originating from a single jurisdiction penetrate many aspects of economic life across the global marketplace")[26] is then expected to be strong.

With that in mind, EU member states may want coordinated regulatory actions, especially when they concern blockchain layer 1. The present report will introduce avenues for reflection that converge in this direction while also considering solutions at layer 2 and application layer. The implementation of these various solutions will require a consensus on the "law + technology" approach, that is, on the necessity to keep blockchain differentiating elements and build on them.

---

[19] Alyssa Hertig, *The Plot Thickens As DAO Attacker Trades Stolen Funds for Bitcoin*, CoinDesk (Oct. 27, 2016) https://www.coindesk.com/dao-attack-hacker-trades-funds-bitcoin [https://perma.cc/3ZJ2-YWJK].
[20] Matthew Leising, *The Ether Thief*, Bloomberg (Jun. 13, 2017), https://www.bloomberg.com/features/2017-the-ether-thief [https://perma.cc/SH7B-TF8B].
[21] Thibault Schrepel & Vitalik Buterin, *Blockchain Code as Antitrust*, Berkeley Tech. L.J. 2 (2020).
[22] As Justice Holmes once said, "I trust that no one will understand me to be speaking with disrespect of the law, because I criticise it so freely. I venerate the law, and especially our system of law, as one of the vastest products of the human mind," *see* Oliver Wendell Holmes Jr., *Path of the Law*, 10 Harv. L. Rev. 457 (1897). The same observation applies here.
[23] Transparency International, *Corruption Perception Index 2020*, Transparency International (2020) https://www.transparency.org/en/cpi/2020/index/nzl [https://perma.cc/N5RD-6T33].
[24] The "layer 1" refers to the blockchain architecture, including the core software, etc. The "layer 2" comprises the technology and applications that function on top of an underlying blockchain protocol. The "application layer" is the one of smart contracts. For a more complete description of blockchain layers, *see* Schrepel, *supra* note 2, at 59–62.
[25] *Id.* at 211.
[26] Anu Bradford, *The Brussels Effect: How the European Union Rules the World*, Preface (Oxford University Press, 2020).

3. Varieties of smart contracts

In this chapter, I explore the dynamism of smart contracts with the objective of capturing their evolutionary process. I first consider the origins of smart contracts and their current population (**2.1**). I then analyze their common characteristics (**2.2**), i.e., what is making them a species, before establishing a typology of the different varieties (**2.3**). On that basis, I study smart contracts interactions and address their evolution (**2.4**).

3.1. Origins and current state of development

The origins of smart contracts are central to understanding their evolution (**2.1.1**). The current population results from different derivatives from the original idea presented in the 1990s (**2.1.2**).

3.1.1. Yesterday

Smart contracts result from a combination of technologies. Nick Szabo defined them in 1994 as "a computerized protocol that executes the terms of a contract."[27] He added that "[t]he protocols are usually implemented with programs on a computer network, or in other forms of digital electronics, thus these contracts are 'smarter' than their paper-based ancestors."[28] Contrary to common wisdom, that definition of "smart" seems about right. The word "smart" comes from the Latin "intelligere," which means "to choose between."[29] Because smart contracts automate the choice according to pre-defined conditions, they are "smart" in the in the term's original meaning.

To illustrate the concept, Szabo takes the example of a vending machine. The manufacturer predetermines the conditions (insert X coin), so the machine can perform the task (deliver the product) when they are met. One cannot use the machine for other functions, but it has the advantage of certainty: the machine provides a product each time a user inserts the right amount. And in fact, this delivery system has been used since (at least) 215 BC: Egyptian temples used it to dispense holy water.[30]

That said, certainty is limited to the trust one has in the manufacturer (and the machine's proper functioning). Should the manufacturer change the terms of the vending machine, he could deceive the potential buyer. The ability to do so from a distance makes the change even more opaque and easy.

Blockchain provides a solution as it allows users to verify the bytecode of smart contracts. Furthermore, when they are public and permissionless, blockchains prevent a single user from changing the terms. Blockchain is indeed *de facto* immutable; one cannot unilaterally delete

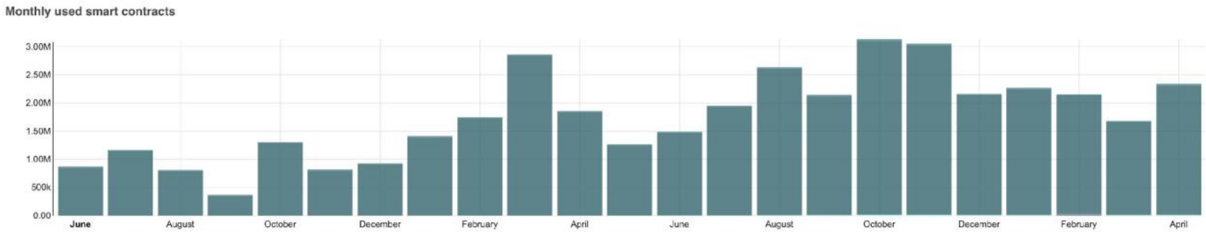
[27] Nick Szabo, *Smart Contracts*, University of Amsterdam (1994). It goes without saying that Szabo's smart contracts were non-blockchain based.
[28] Nick Szabo, *Smart Contract Glossary*, University of Amsterdam (1995).
[29] Alain Rey, *Dictionnaire Historique De La Langue Française* 1037 (Le Robert, 1992).
[30] Max Raskin, *The Law and Legality of Smart Contracts*, 1 GEO. L. TECH. REV. 315 (2017).

(past) transactions from its ledger. Similarly, future transactions (i.e., smart contracts) cannot, in principle, be unilaterally changed or removed. This limit to unilateral actions reinforces trust.[31] In concrete terms, the operation of blockchains' vending machines (the price of the transaction, the conditions necessary for its completion, etc.) is immutable, which adds to the certainty.

For this reason, some have called smart contracts "more contract than contract."[32] Indeed, contracts have the function of deterring opportunistic behaviors,[33] which smart contracts do by preventing technical deviations from the initial agreement.[34] That explains why, since their democratization with the Ethereum blockchain in 2013, smart contracts are gaining momentum.[35]

3.1.2. Today

The number of smart contracts in circulation is increasing. They are used to implement escrow systems, automate royalty payments,[36] manage digital collectibles,[37] organize decentralized marketplaces,[38] enable decentralized finance ("DeFi"),[39] help with prediction markets integrity,[40] etc. The Ethereum blockchain is the most popular amongst developers and smart contracts' users.

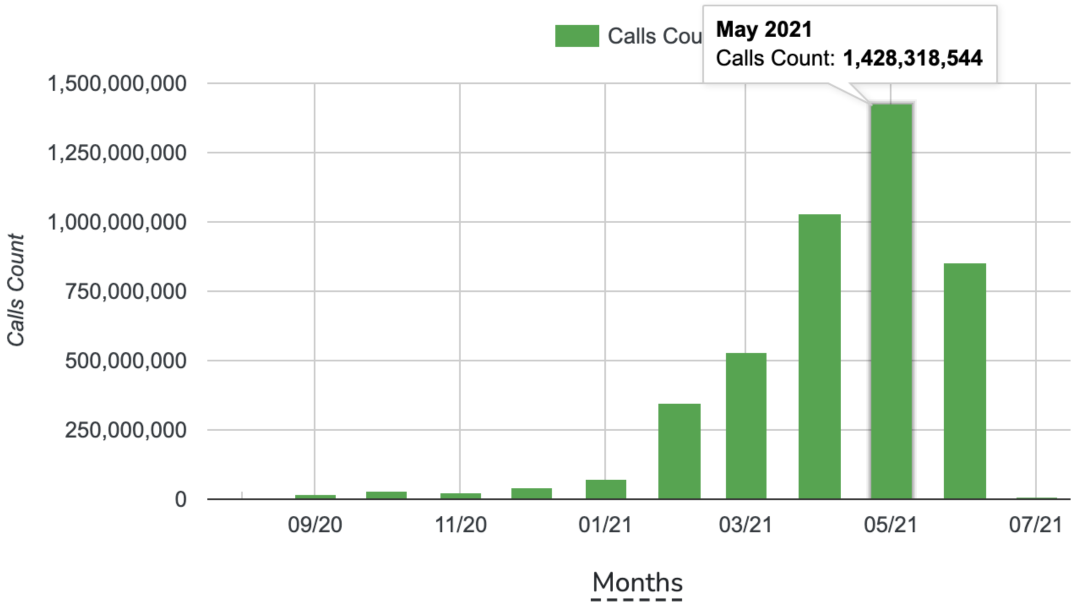


Title: Monthly used smart contracts on the Ethereum
Source: Bloxy
(May 3, 2021)

One can use other blockchains to implement smart contracts, such as Tezos, Stellar, EOS, Celo, TRON, Hyperledger, and Ethereum Classic. Although there is no aggregate data of all

[31] Sarah Templin, *Blocked-Chain: The Application of the Unauthorized Practice of Law to Smart Contracts*, 32 GEO. J. LEGAL ETHICS 957, 960 (2019).
[32] Bill Marino & Ari Juels, *Setting Standards for Altering and Undoing Smart Contracts*, 1 INT. SYMP. ON RULES AND R. M. LANG. FOR THE SEM. W. 151 (2016).
[33] Richard Posner, *Economic Analysis of Law* 81 (Wolters Kluwer, 1986).
[34] Thibault Schrepel, *Collusion by Blockchain and Smart Contracts*, 33 HARV. J.L. & TECH. 117, 163 (2019).
[35] From this point on, I will use the term "smart contract" to refer to "blockchain smart contract."
[36] Sam Daley, *17 Companies Utilizing Blockchain in Music to Reshape a Changing Industry*, BUILTIN BETA (Mar. 16, 2019) https://builtin.com/blockchain/blockchain-music-innovation-examples [https://perma.cc/V7MF-LTF8].
[37] *See* Melanie Kramer & Daniel Phillips, *Non-Fungible Tokens (NFT): Beginner's Guide*, DECRYPT (Feb. 4, 2021) https://decrypt.co/resources/non-fungible-tokens-nfts-explained-guide-learn-blockchain [https://perma.cc/C2CS-8LSL].
[38] *See* Matt Hussey, *What are decentralized marketplaces?*, DECRYPT (Mar. 5, 2020) https://decrypt.co/resources/what-are-decentralized-marketplaces [https://perma.cc/R3H9-EL62].
[39] *See*, for example, MakerDAO, *A Better, Smarter Currency*, MAKERDAO (Jul. 27, 2021) https://makerdao.com/en/ [https://perma.cc/3LLL-BKBL].
[40] *See*, for example, Augur, *Augur: Your Global, No-Limit Betting Platform*, AUGUR (Jul. 27, 2021) https://augur.net/ [https://perma.cc/XDV5-46FS].

monthly used smart contracts, different online "explorers" give an overview of their popularity.[41] As we speak, most of these blockchains are concentrated around a core business, such as Binance Smart Chain which allows token swaps in decentralized money markets.[42] The smart contracts implemented on this blockchain have been called over 1.4 billion times during May 2021 alone.[43]



Title: Smart contracts calls on the Binance Smart Chain
Source: Bitquery, Binance Smart Chain mainnet blockchain explorer: smart contract calls (July 1, 2021)

Looking at one blockchain in-depth (here the Ethereum blockchain), one will notice that only a couple thousand smart contracts (about 20,000) are created daily. Some are called more than a million times a day.[44] They are often "auxiliary" smart contracts that operate basic functions and do not feature a great deal of complexity (similarly to the smart contracts on the Binance Smart Chain).

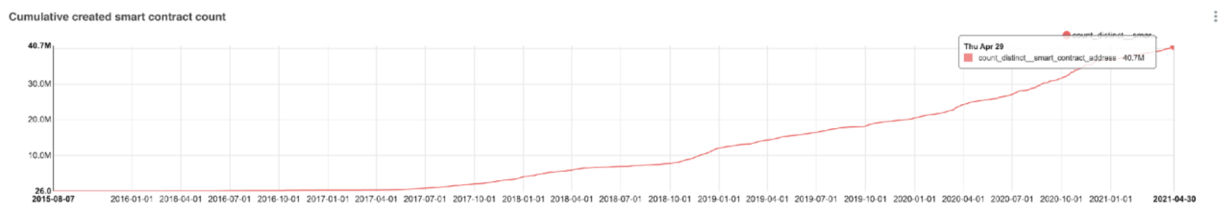
---

[41] Bitquery, *Smart Contracts Blockchains*, BITQUERY (2021) https://explorer.bitquery.io/platform/smart_contract [https://perma.cc/4VAJ-J4YH]. For example, *see* Statistics, *Mainnet*, Statistics (Jul. 22, 2021) https://better-call.dev/stats/mainnet/general [https://perma.cc/XK4C-HXBQ] (exploring the Tezos blockchain). A couple thousands of smart contracts are deployed each month on the mainnet (where tokens have fiat value), while tens of thousands are deployed on testnet such as Florencenet and Edo2net, *see* Statistics, *Florencenet*, Statistics (Jul. 22, 2021) https://better-call.dev/stats/florencenet/general [https://perma.cc/9EX8-Z2JB].

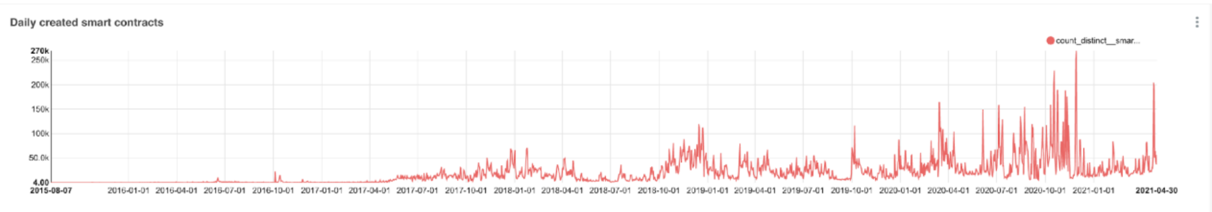
[42] Binance Academy, *How to Get Started with Binance Smart Chain (BSC)*, BINANCE ACADEMY (Feb. 22, 2021) https://academy.binance.com/en/articles/how-to-get-started-with-binance-smart-chain-bsc [https://perma.cc/7Y3D-9XY5].

[43] "Calling" a smart contract means invocating one of its functions. Each time users are calling a smart contract, they are paying fees to do so (e.g., mining fees, gas fees).

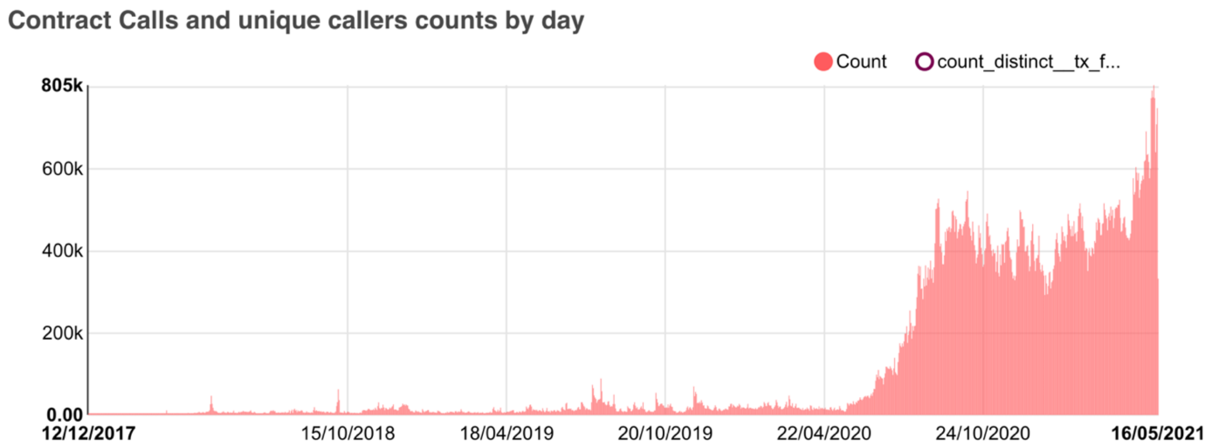
[44] For example, Bloxy, *0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2*, BLOXY (Jul. 26, 2021) http://bloxy.info/address/0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2 [https://perma.cc/H8KP-9M29].



Title: Cumulative created smart count on the Ethereum
Source: Bloxy
(May 3, 2021)



Title: Daily created smart contracts on the Ethereum
Source: Bloxy
(May 3, 2021)

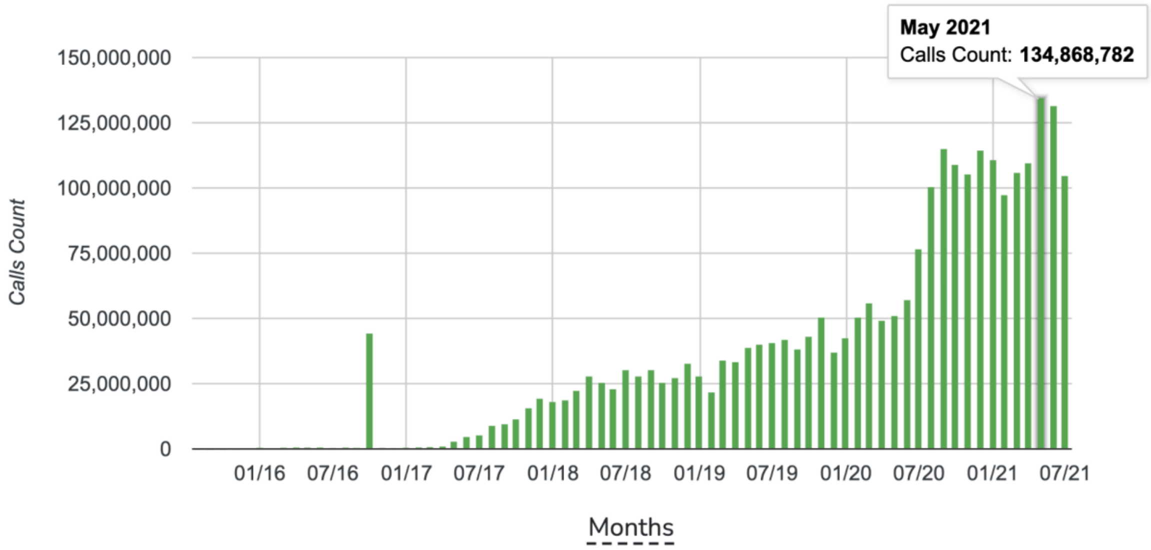


Title: Bloxy, smart contract "0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2"
Source: Bloxy[45]

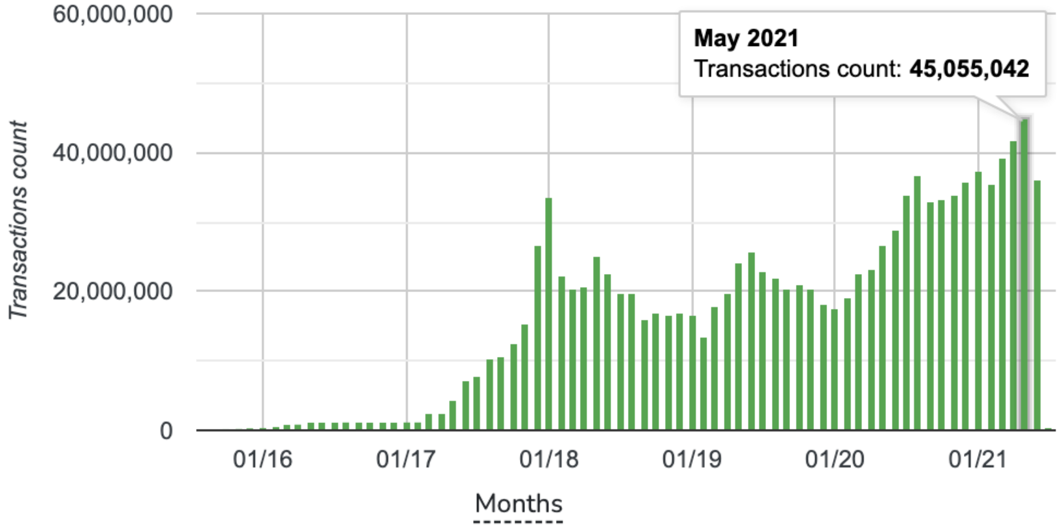
This disparity between the number of smart contracts and calls is explained by the fact that once registered on a blockchain, one can, in theory, use a smart contract an unlimited number of times.[46] That amounts to over 130 million smart contracts' calls on the Ethereum blockchain for the sole month of May 2021, resulting in over 45 million transactions.

[45] Bloxy, *0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2: Smart Contract Dashboard*, BLOXY (2021) https://bloxy.info/smart_contract_stat/0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2 [https://perma.cc/S9WG-63K3].

[46] This opens reentrancy vulnerabilities, *see* Noama Fatima Samreen & Manar H. Alalfi, *Reentrancy Vulnerability Identification in Ethereum Smart Contracts,* 1 IEEE INT. WORK. ON BLOCK. OR. SOFT. EN. 22, 23 (2021); Ayman



Title: The number of smart contracts' calls on the Ethereum
Source: Bitquery, Ethereum smart contract calls
(July 25, 2021)



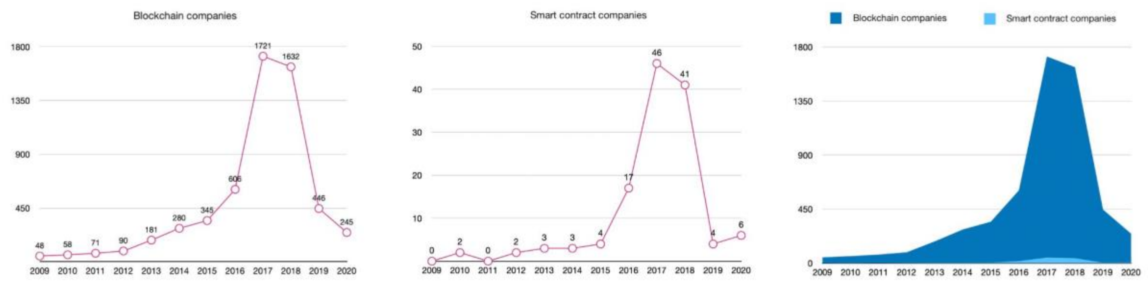
Title: The number of smart contract transactions on the Ethereum
Source: Bitquery, Ethereum transactions count
(July 1, 2021)

Finally, it should be emphasized that smart contracts can be directly coded by individuals or built using templates and no-code solutions provided by specialized companies.[47] The number of startups operating blockchain systems, products, or services has increased significantly in

Alkhalifah et al, A *Mechanism to Detect and Prevent Ethereum Blockchain Smart Contract Reentrancy Attacks*, 3 FRONT. COMPUT. SCI. 1 (2021).

[47] For example, Open Law, *Real World Contracts for Ethereum*, OPEN LAW (Jul. 26, 2021) https://www.openlaw.io/ [https://perma.cc/R2EF-GGC7].

recent years, but the number of such companies specializing in smart contracts remains relatively small.



Title: A look at blockchain companies vs. smart contract companies (2010 – 2020)
Source: Crunchbase[48]
© Thibault Schrepel (2021)

3.2. One species: key characteristics

Smart contracts have common characteristics. The combination of encryption techniques with decentralized and distributed mechanisms (**2.2.1**) makes them immutable (**2.2.2**) and differentiates them from other transactional methods.

3.2.1. Smart contract functioning

Understanding how smart contracts function requires an appreciation of private and public keys. As a principle, each user keeps a private key, while public keys are publicly available.[49] They are reciprocal; what one key does, only the other one can undo.

When users want to send certified information, they need to run the data (such as a picture) through a hash function (e.g., MD5). This process results in a hash value (a numeric value) identifying the data. Users encrypt that hash value using their private key. The encryption results in a new encrypted hash, called the signature. Ultimately, private key holders send the original data (here, the picture) along with the signature.

The receiver decrypts the signature using the sender's public key—the decryption results in a hash value. In parallel, the receiver also runs the original data into the sender's hash function and gets another hash value. Should the two hashes be identical, this proves the validity of the signature as only the sender's public key can undo the encryption.

---

[48] On the left, one will find the number of companies listed in the "blockchain" industry on Crunchbase and founded between 2010 and 2020. In the middle, one will find the number of companies listed in the "blockchain" industry, founded between 2010 and 2020, and using the term "smart contract" in their description. On the right, one will find a comparative overview of these companies.

[49] A private key on the Ethereum blockchain comprises 64-bit hexadecimal characters, such as fffffffffffffffffffffffffffffffebaaedce6af48a03bbfd25e8cd036415f (*see* Ethereum, *Ethereum Accounts*, ETHEREUM (Jul. 26, 2021) https://ethereum.org/vi/developers/docs/accounts [https://perma.cc/JVP6-3R5K].

Blockchain wallets facilitate that verification process by automatically generating and storing each users' private key. For example, the wallet's software signs a transaction with the sender's private key when transferring data or tokens. This signature indicates to the entire blockchain community that the sender has the authority to send that data or token.

Smart contracts follow that logic. They are scripts activated when a valid call is sent to them, i.e., a call that can be authenticated and linked to a single user. Taking the Ethereum blockchain as an example, two types of blockchain accounts play a role in that process of activating smart contracts. There are Externally Owned Accounts (EOAs) and contract accounts.

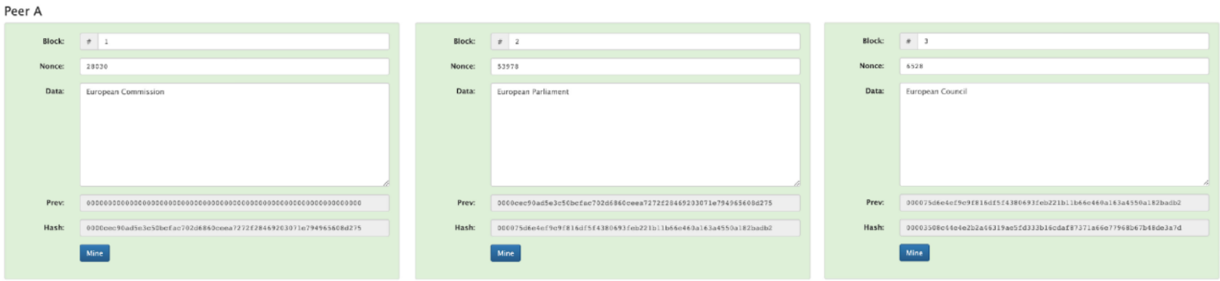
Each EOA is controlled by a single user. It comprises a combination of public and private keys. This type of account can receive and send data or tokens. In parallel, a new contract account is generated each time one puts a smart contract on the Ethereum blockchain.[50] These accounts can receive and send data or tokens, similarly to EOAs, but differ in two respects. First, they do not have a private key.[51] Second, they can execute the smart contract bytecode using the Ethereum Virtual Machine ("EVM"), a simulated computer. They do so every time an EOA calls one of the functions of a smart contract,[52] but if and only if the signature (created by the private key) is valid.

The validity of the signature proves that the person making the call is the custodian of the assets. Once the verification is completed, the smart contract code is executed, not on the Ethereum blockchain itself (i.e., the ledger) but outside of it. Only the inputs and outputs of this transaction are recorded on the blockchain.[53]

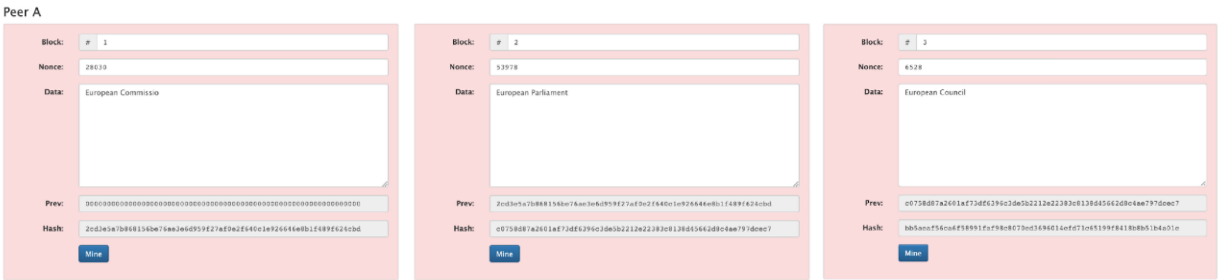
3.2.2. Smart contract immutability

Smart contracts deployed on a blockchain are said to be immutable by default.[54] The bytecode of a smart contract is indeed recorded in a transaction that is mined into a block along with other transactions. An address is assigned to the contract,[55] and each block is assigned a hash value that represents all the transactions (each with an address) it contains. That hash value of block #1 is systematically recorded into block #2, and so on. Changing even the slightest information in a block generates a new hash value and, therefore, invalidates the entire chain. In the following example, three blocks have been mined. Should the data in block #1 be changed (e.g., from "European Commission" to "European Commissio", from 10 transactions to 9 transactions, etc.), the blockchain software would generate a new hash value, therefore changing the hash value of block #2.

[50] They contain the smart contract's bytecode.
[51] A contract account is run by the logic of its smart contract's code.
[52] A smart contract does not self-execute.
[53] Eliza Mik, *Smart Contracts: Terminology, Technical Limitations and Real World Complexity*, 2 L., INNOV. AND TECH. 269, 277 (2017).
[54] Making a distinction between weak immutability (when it must be monitored) and strong immutability (when ensured by way of computation), *see* Esteban Landerreche & Marc Stevens, *On Immutability of Blockchains*, 2 REP. OF THE EUR. S. FOR SOC. EMB. TECH. 1, 2 (2018).
[55] This address indicates where the smart contract resides on the blockchain.



Title: A look at blockchain immutability (Part I)
© Thibault Schrepel (2021)



Title: A look at blockchain immutability (Part II)
© Thibault Schrepel (2021)

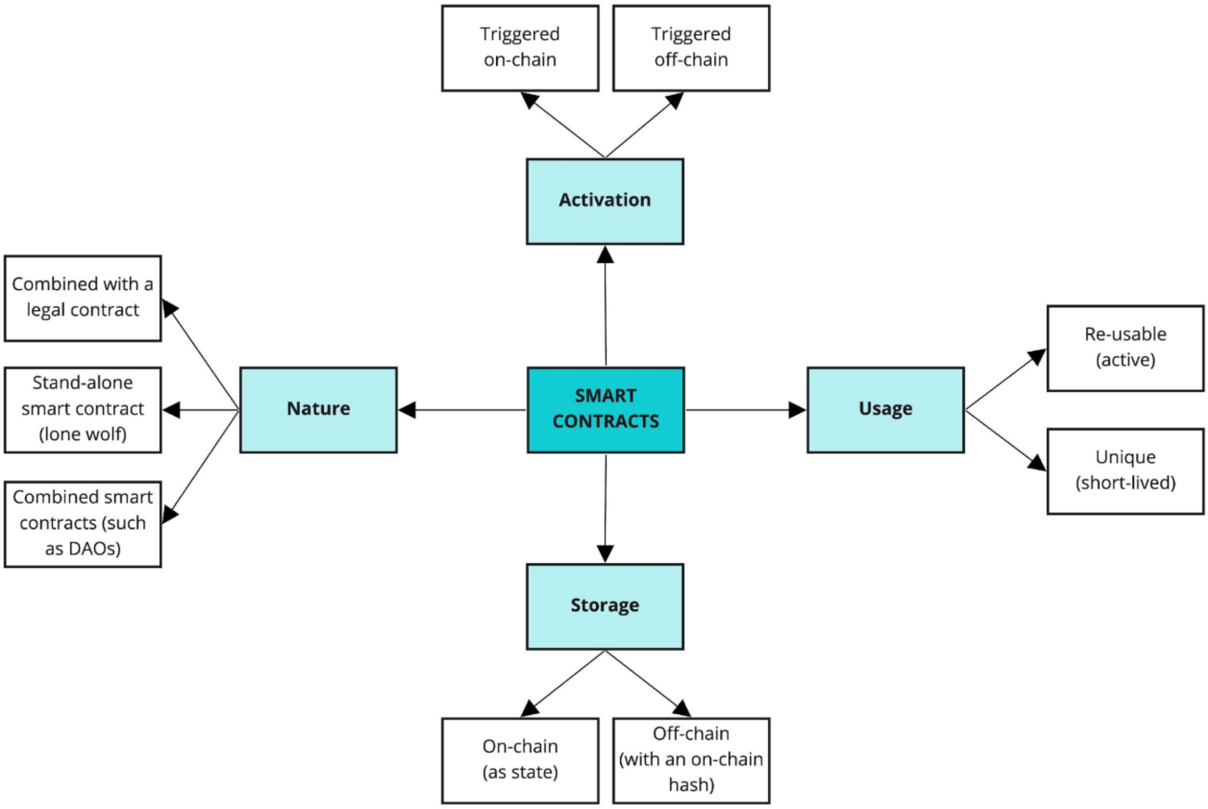
One would then need to recalculate all the subsequent blocks' hash values to re-validate the entire chain. And one would need to do so before the addition of a new block. Indeed, the protocol of most proof-of-work blockchains validates the chain with the most chain work put into it.[56] Should anyone want to change blockchain history, they would need to out-compete the computational power put in by all the miners and have the majority of them agree to the changed history. Doing so — i.e., a 51% attack — would be costly (for example, in electricity, should the blockchain operate under Proof of Work) and could be impossible in practice (should the blockchain have enough miners) with public permissionless blockchains.

Therefore, the collective work of miners ensures practical immutability,[57] which is a strong, but emergent blockchain feature.[58] The more users a blockchain has and is used, the more it becomes *de facto* immutable. Smart contracts stay visible on the blockchain even after they have been used. The gradual emergence of immutability means, on the contrary, that one can more easily alter a blockchain ledger with very few miners.[59]

[56] This is true for Bitcoin and other proof of work blockchain. By "chain work," I mean computational work.
[57] Satoshi Nakamoto described miners' activity "as a cooperative effort to make a chain," *see* Satoshi Nakamoto, *BitcoinTalk Repost: Bitcoin Maturation*, SATOSHI NAKAMOTO INSTITUTE (Nov. 22, 2009) https://satoshi.nakamotoinstitute.org/posts/bitcointalk/8/ [https://perma.cc/VA9P-TS36].
[58] Eugenia Politou et al., *Blockchain Mutability: Challenges and Proposed Solutions* 6 (2019).
[59] *See* Zibin Zheng, *An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends*, 1 IEEE INT. CONG. ON B. DAT. 557, 562 (2017).

3.3. Several varieties: a typology

Now that we have explored the typical characteristics of smart contracts, I want to analyze their different varieties, and at the same time, begin to address legal opportunities and challenges.



Title: An overview of blockchain smart contracts
© Thibault Schrepel (2021)

One can make a first distinction between the different **natures** of smart contracts. First, one can combine a smart contract with a "legal contract," i.e., a contract recognized by the law.[60] This legal contract can itself be materialized in paper or digital format. For example, a rental contract could be written in prose between the owner of an apartment and a tenant, while the smart contract could automate payment. The smart contract's code could also complement the contract in prose instead of matching existing rights and duties.[61] In a nutshell, these two transactional forms compete and complement each other simultaneously.[62]

[60] Daniel Kraus et al., *Blockchains, Smart Contracts, Decentralised Autonomous Organisations and the Law : Perspectives of a Distributed Future, Cheltenham* 111 (Edward Elgar Publishing 2019). Here, I do not wish to assume that a smart contract cannot be a "legal contract". For a discussion, *see* 3.1.
[61] Smart Contracts Alliance, *Smart Contracts: Is the Law Ready?*, CHAMBER OF DIGITAL COMMERCE 25 (2018).
[62] Should the contract in prose be void, the performances automated by the smart contract will be the subject to restitution.

Second, one can implement a smart contract without the support of a legal contract. This scenario captures the majority of the smart contracts in circulation today. I call them "lone wolves" because they intend to be self-sufficient. They can be recognized as legal contracts, but this is not necessarily the case — as I will later explain.

Third, smart contracts can be combined with other smart contracts to create the conditions for the decentralized governance of (autonomous) ecosystems. Decentralized Autonomous Organizations ("DAOs") are good examples of a nexus of smart contracts creating a coherent whole.[63] In more concrete terms, smart contracts are conditioned to other smart contracts. The occurrence of an event may then trigger a chain of such smart contracts.
One can make a second distinction depending on the **use** of smart contracts. The conditions for calling a smart contract are rarely specific to a single party.[64] This general-purpose nature means that any user can call them.[65] That said, a smart contract can be conditioned to real-life events that only one party can trigger. It then becomes *intuitu personæ*. And should the smart contract be called regularly, whether by one user or a multitude of them, it is called active (as opposed to dormant).

This discussion leads us to explore a third distinction between varieties of smart contracts depending on how they are **activated**. Some are triggered on-chain; the call is made following an event on the chain. For example, a smart contract can be designed to be called only when the value of a specific token exceeds a certain level. This information comes from the chain itself, which eases the transmission. Others are triggered off-chain; the call is made following an event outside the chain. For instance, a smart contract providing a refund when a train is delayed requires that the information (i.e., the delay) be correctly registered on the chain (*see* 2.4.2. for a discussion of "oracles").

Finally, one can make a distinction between smart contracts depending on where they are **stored**: on-chain or off-chain. As previously explained, the bytecode of a smart contract is stored on a transaction put on-chain. Doing so ensures immutability but leads to a lack of secrecy — one can use a decompiler to revert the bytecode into the original programming language.[66] That said, data (including smart contract) can also be stored off-chain, with only the hash being recorded on-chain. When doing so, the smart contract is encrypted locally using a public key; only the users with the corresponding private key can decrypt it. The hash value is then recorded onto the blockchain instead of recording the raw data (i.e., the smart contract bytecode). The immutability of the smart contract remains *de facto* guaranteed because changing it would automatically generate a new hash value that would not correspond to the original one recorded onto the blockchain.[67]

[63] DAOs can also consist of a single extensive smart contract. For examples, *see* Open Bazaar, *Buy and Sell Freely*, Open Bazaar (Jul. 5, 2021) https://openbazaar.org/ [https://perma.cc/MZZ7-WATL], and Arcade City, *Connect Freely*, Arcade City (Jul. 5, 2021) https://arcade.city/ [https://perma.cc/3CKP-C398?type=image].
[64] A few blockchains are focusing on preventing Reentrancy, *see* for instance Agoric, Build Fast, Earn Fast (Jul. 15, 2021) https://agoric.com/ [https://perma.cc/2434-YFDP]. Shuai Wang et al, *An Overview of Smart Contract: Architecture, Applications, and Future Trends*, 4 IEEE INT. VEH. SYMP. 108, 111 (2018).
[65] James Grimmelmann, *All Smart Contracts Are Ambiguous* 2 J. OF L. & INN. 2, 9 (2019).
[66] Information may be lost in the process, similarly to converting a French text into English and back to French.
[67] In such a scenario, the main drawback would be to recover the original smart contract should it be modified, *see* Xiwei Xu et al., *A Pattern Collection for Blockchain-based Applications*, 18 PROC. OF THE 23RD EUR. CONF. ON PAT. LANG. OF PROG. 10 (2018).

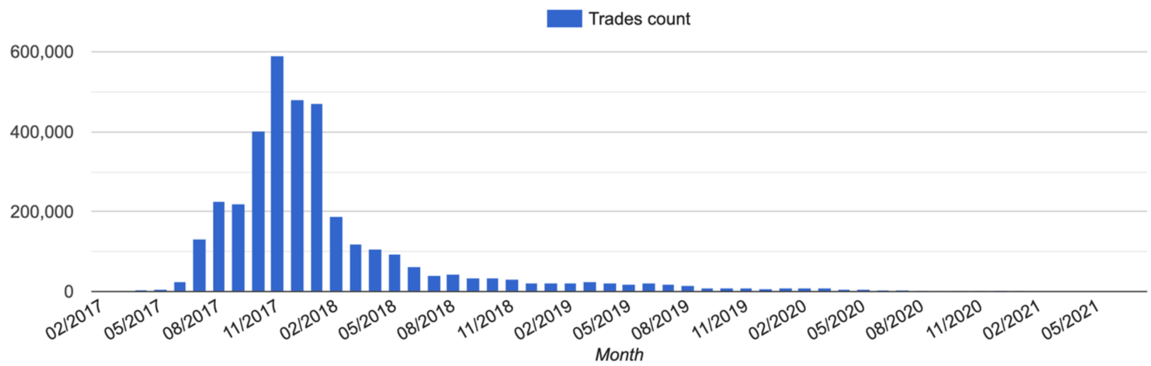
3.4. Smart contracts interactions

The interaction of smart contracts between each other (**2.4.1**) and outside the blockchain (**2.4.2**) leads to multiple transformations. These mutations end up impacting smart contract's evolutionary process.

3.4.1. Interactions between varieties

Smart contracts interact with each other, either to compete or to cooperate. First, several blockchains are competing for smart contracts. Depending on the chain on which they are built, smart contracts have unique characteristics. For example, Polkadot, Cardano, and EOS smart contracts are, on average, validated more rapidly than Ethereum smart contracts. Tezos allows more secrecy. Polkadot uses bridges to enable the transfer of tokens or data from one blockchain to another. Cardano focuses on on-chain governance mechanisms and uses Ouroboros as a consensus mechanism.[68] NEM uses code located off the blockchain, while Ethereum enjoys the strongest reputation. Innovations — mutations — are introduced every day.

Second, there is also competition between smart contracts built on the same blockchain. Some become more attractive than others because they are better designed, introduce new functions, or enjoy a stronger network effect.[69] Others disappear. For example, the smart contract shown below was used over 589,000 times in November 2017 in Uniswap transactions.[70] A few months later, blockchain users did not use it anymore.



Title: Smart contract "0x8d12a197cb00d4747a1fe03395095ce2a5cc6819"
Source: Bloxy[71]

Smart contracts cooperate when they are technically linked to each other.[72] They also cooperate through the network effects they generate. The more a blockchain is used for the creation of smart contracts, the higher is the utility users can derive from implementing new

[68] Aggelos Kiayias et al., *Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol*, 1 ANN. INT. CRYPT. CONF. 357 (2019).
[69] The template of successful smart contracts is enjoying network effects.
[70] Uniswap is a "fully decentralized protocol for automated liquidity provision on Ethereum," *see* Uniswap, *Decentralized Trading Protocol*, UNISWAP (Jul. 5, 2021) https://uniswap.org/ [https://perma.cc/X92T-XVTH].
[71] Bloxy, *0x8d12a197cb00d4747a1fe03395095ce2a5cc6819: Activities by Time*, BLOXY (2021) https://bloxy.info/dexes/0x8d12a197cb00d4747a1fe03395095ce2a5cc6819 [https://perma.cc/CJL9-GW7E].
[72] *See* our explanations regarding DAOs in 2.3.

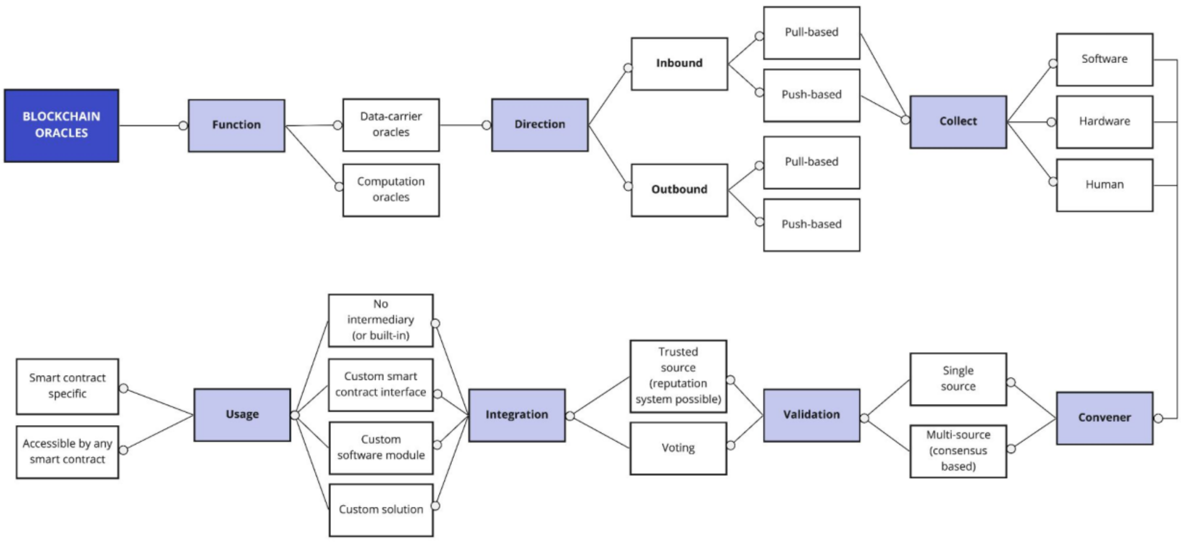
ones. In effect, this leads more users to own native tokens and be willing to transact using that blockchain.

More generally, the entire blockchain ecosystem benefits from network effects. Indeed, the more blockchains are used, the more familiar decentralized transactions become. In addition, these effects are reinforced by the existence of multichains and parachains that allow the interoperability of various legacy blockchains (e.g., Ethereum and Bitcoin), such as Polkadot and Cosmos.[73]

3.4.2. Interactions with the outside world

Smart contracts can interact with the "outside" world (i.e., outside the blockchain). They do so by organizing it, that is, by enforcing obligations between individuals. That said, smart contracts cannot make a lasting and sustainable impact if they remain blind to what is happening outside the blockchain. This is where oracles come in.

The concept of oracle goes back to ancient times. Originally, an oracle was a person in charge of reporting the prophecy whispered by divine sources. The meaning of the word has changed with information sciences. One can find the roots of the concept in the work of Turing, who, in 1939, devoted a few lines to oracle-machine ("o-machines") connecting with external databases or other devices.[74] Bellare and Rogaway (1993)[75] then used it to designate a process by which the cryptographic hash function is chosen randomly.[76]



Title: An overview of blockchain oracles
© Thibault Schrepel (2021)

[73] Polkadot, *Polkadot is Live*, Polkadot (Jul. 5, 2021) https://polkadot.network/ [https://perma.cc/R9D7-5ZKS?type=image] and Cosmos, *The Internet of Blockchains*, COSMOS (Jul. 27, 2021) https://cosmos.network/ [https://perma.cc/2VSR-GPNC?type=image].
[74] Marta Poblet et al., *From Athens to the Blockchain: Oracles for Digital Democracy*, 3 FRONT. BLOCKCHAIN 1, 3 (2020).
[75] Mihir Bellare & Phillip Rogaway, *Random Oracles Are Practical: Random Oracles Are Practical: a Paradigm For Designing Efficient Protocols*, 93 PROC. OF THE 1ST ACM CONF. ON COMP. AND COM. SEC. (1993).
[76] Poblet et al., *supra* note 74, at 3.

The word "oracle" takes on a different meaning with blockchain. Generally, it designates the intermediary who reports information from the real world to the blockchain[77] or vice versa. It is then a data carrier. Alternatively, the oracle can have a computational **function** when it performs calculations.[78]

When being a data carrier, the information can take two **directions**. First, it can be outbound; information about the blockchain (or already recorded on chain) is then carried to the outside world. Here, one can pull information by request made outside the chain[79] or push information following an on-chain request. Second, the oracle can be inbound; it then brings off-chain information to the chain. Here, it can be pull-based when the request comes from the blockchain or push-based when the transmission is started outside the blockchain.[80]

When the oracle is inbound, one must distinguish between different **ways of collecting** information.[81] The oracle can be a computer program interacting with (existing) online information and transmitting it to the blockchain.[82] The oracle can retrieve information from a trusted website, databases, or servers that provide, for example, weather data and send it to a smart contract implemented by an insurance company.

The oracle can be a piece of hardware that transforms real-world information into digital information.[83] A humidity sensor can transmit information to the blockchain. Connected objects (Internet of Things) fall into this category, along with robots, electronic chips, etc. Finally, the oracle can be a human being. One can appoint a trusted third party (e.g., an individual, an institution, etc.) to provide real-world information to the blockchain, such as a notary who certifies that two persons got married.

Regardless of where the information comes from, the oracle can use a single **source** or several of them.[84] When a single source is used, the oracle recentralizes the blockchain by introducing a single point of failure and requiring trust in just one entry point. This is often referred to as the "oracle problem."[85] For that reason, the combination of several sources is preferred whenever possible. For example, a software oracle can fetch the weather from several websites and calculate the average temperature. Similarly, multiple physical devices or human beings may transmit information. Doing so requires properly designed governance rules.

---

[77] *See* Kamran Mammadzada et al., *Blockchain Oracles: A Framework for Blockchain-Based Applications*, INT. CONF. ON BUS. PROC. MANAG. 19, 20 (2020) (defining orales as "trusted entities that enable the collection, validation, and transmission of data from external source").
[78] Abdeljalil Beniiche, *A Study of Blockchain Oracles* 2 (2020).
[79] When the oracle is outbound and pull-based, one needs permission to sign transactions using a private key.
[80] Roman Mühlberger et al., *Foundational Oracle Patterns: Connecting Blockchain to the Off-Chain World*, 1 INTER. CONF. ON BUSIN. PROC. MANAG. 35, 36 (2020).
[81] Beniiche, *supra* note 78, at 1.
[82] *See* Chainlink, *Chainlink 2.0 and the Future of Decentralized Oracle Networks*, CHAINLINK (2021) https://chain.link/whitepaper [https://perma.cc/N2FF-V6P4].
[83] These oracles raise the question of where to store the information once extracted. Discussing the issue, *see* Hamda Al Albreiki, *Decentralized Access Control for IoT Data Using Blockchain and Trusted Oracles*, 1 IEEE INTER. CONF. ON IND. INT. 248 (2019).
[84] Aleksandre Asatiani et al., *Business Process Management Blockchain and Robotic Process Automation Forum* 26 (Springer 2020).
[85] For a discussion of the "oracle problem", *see* ChainLink, *What is Blockchian Oracle Problem?*, CHAINLINK (Aug. 27, 2020) https://blog.chain.link/what-is-the-blockchain-oracle-problem/ [https://perma.cc/F978-E5HB].

One must then **validate** the information once transmitted. The validation can be automatic if users decide to trust the oracle. In such a scenario, a reputation system that establishes a quality score for each oracle based on the accuracy of the information transmitted can validate only the information of the best-ranked oracles. One can also set up a system allowing each user to challenge the oracle's information. The claim is then evaluated more or less easily depending on whether the oracle documents binary facts (true or false)[86] or more nuanced ones.[87] Alternatively, the validation of the information can be the subject of a vote submitted to blockchain users.[88] In such a hypothesis, each user is incentivized to vote on the accuracy of the information. Here again, a reputation system can be used so that the vote of the most reliable users is given more importance than one of the users without a positive record.

After that, the information needs to be **integrated**. The first way of doing this consists of developing a solution within the blockchain network. Here, smart contracts automatically self-execute when the conditions are met;[89] there is no intermediary. A second solution, more classical, is to use a custom interface to integrate the information provided by the oracle.[90] Decentralized web applications ("dApps") are helpful for this purpose. A third solution comprises using custom software modules that process the information before transmitting it.[91] Finally, other customized solutions can be developed, for instance, by requiring the fingerprint of a specific individual to prevent falsification.[92]

Once the information is integrated, one can **use** it for two purposes. First, it can be used in a single smart contract; it is then "contract-specific." For example, an oracle can track a product within a specific production chain. Second, one can use this information in multiple smart contracts. This is done, for instance, when an oracle transmits financial values and thus can be used to activate different transactions. It is also the case when oracles are used to pull together a database ("pool data") accessible by several companies operating in a specific industry.

Overall, oracles are the centerpiece of the puzzle for connecting the blockchain space with the outside world. As we shall see, they could play the role of an entry door for legal enforcement by providing solutions to the difficulties arising from the automation of smart contracts. Despite their importance, only a handful of companies are currently devoted to providing new solutions in the blockchain space.

---

[86] Such as Boolean values.

[87] William George & Clément Lesaege, *A Smart Contract Oracle for Approximating Real-World, Real Number Values*, 1 INT. CONF. ON BLOCK. ECON., SEC. AND PROT. 1 (2019).

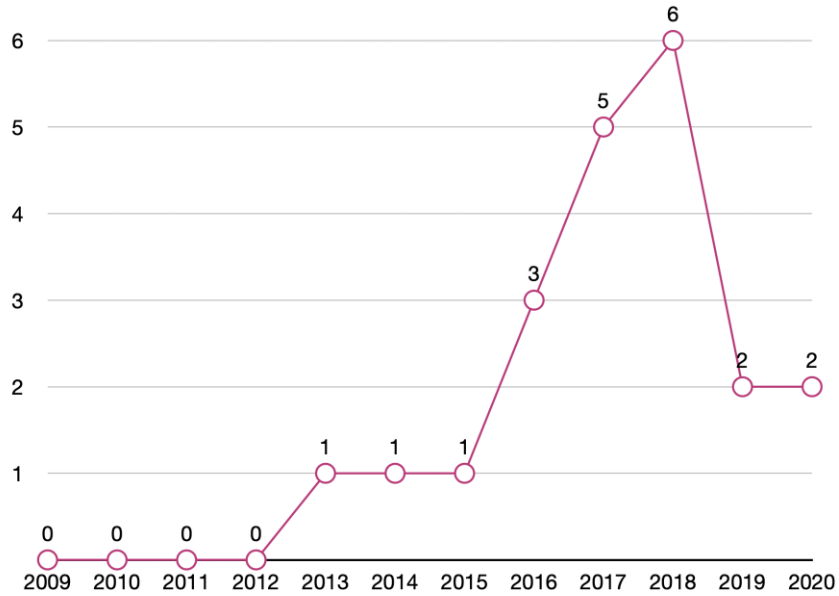
[88] *See* Naman Goel et al., *Infochain: A Decentralized, Trustless and Transparent Oracle on Blockchain,* 1 PROC. OF THE 29TH INT. J. CONF. ON ART. INT. 4604 (2020); Also, John Adler et al., *ASTRAEA: A Decentralized Blockchain Oracle* (2018).

[89] Paul Sztorc, *Truthcoin: Peer-to-Peer Oracle System and Prediction Marketplace* (2015); Paul Sztorc, *Truthcoin: Decentralized Bitcoin Prediction Markets*, GITHUB (Jul. 26, 2021) https://github.com/psztorc/Truthcoin [https://perma.cc/J7N7-Z7TF].

[90] James Ahn, *EdenChain: The Programmable Economy Platform Version 1.2* (2018).

[91] Bing Moet al., *A Solution for Internet of Things Based on Blockchain Technology*, 1 IEEE INT. CON. ON SER. OP. AND LOG., AND INF. 112 (2018).

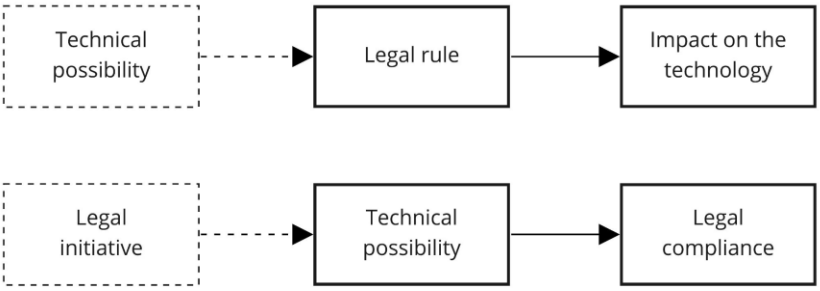
[92] Zhimin Gao et al., *Blockchain-Based Identity Management with Mobile Device*, CRYBLOCK'18: PROC. OF THE 1ST WORK. ON CRYPT. AND BLOCK. FOR DISTR. S. 66–70 (2018).



<u>Title</u>: Number of founded companies listed in the industry "blockchain" and using the word "oracle" in their purpose description
<u>Source</u>: Crunchbase © Thibault Schrepel (2021)

4. Toward a more harmonious ecology

In this chapter, I focus on the legal and technical dimensions of smart contracts' environment.[93] I explore how the environment influences smart contracts, and how smart contracts influence it in return. In line with a "law + technology" approach, I wish to lay down some first principles for creating a more harmonious framework that allows smart contracts to maintain their differentiating elements while fostering the DSM. It requires implementing solutions that are grounded in law (**3.1**) and technology (**3.2**), and that enable each other.



© Thibault Schrepel (2021)

4.1. Solutions enabled by the law

As a reminder, the first objective of the present study is "to conduct research on regulatory barriers when smart contracts are used instead of conventional contracts under the current European legal framework. In particular, the potential problems arising out of 'automation' should be considered." Doing so requires discussing if the existing legal framework constitutes a barrier to the deployment of smart contracts (**3.1.1**) and which legal rules could better support them (**3.1.2**).

4.1.1. The current legality of smart contracts

4.1.1.1. A closer look at (smart) contracts lifecycle

To begin with, I shall compare the main requirements of contract law with the creation and performance of a smart contract. But first, I want to recall that smart contracts can complement a contract in prose by automating the execution of certain obligations (such as payment). The contract is then hybrid.[94] That said, this chapter addresses standalone smart contracts. Two questions need to be discussed: can smart contracts be recognized as legal contracts, and if so, under what conditions?

[93] Social, political, normative, philosophical, and economic constraints (among others) shall be added in subsequent studies.
[94] Law Commission, *Smart Contracts: Call for Evidence* 15 (2020).

A complete answer would require an analysis of the contract law of all 27 Member States independently — because of the absence of harmonized European contract law. By default, I will limit myself to analyzing contract law as it seems to be shared by most Member States and pointing out the most contentious rules and principles for our subject.[95]

→ Creation

Validity. From a technical point of view, the creation of a smart contract is unilateral.[96] A single user puts it on the blockchain.[97] The smart contract then becomes "valid" as soon as a miner integrates it into a block.[98] The miner or validator does not check the content of the smart contract, but its technical validity by running all its functions.

Despite being technically unilateral, the smart contract can be legally binding. Member States acknowledge that a contract, although materialized by one party only, can indeed be valid if several conditions are met.[99] The same logic applies to smart contracts.[100] The following discussion points out the two main contentious elements that will need to be studied for each member state: (i) the exchange of consents, and (ii) forms requirements.

(i) To begin with, the *intention* to create a contract (e.g., the validity of the exchange of consents) can be more easily challenged with smart contracts than contracts in prose. Both parties must indeed understand the obligations to which they consent. The intention is deemed absent when one party could not have known the ins and outs.[101] The fact that smart contracts have a computer language format (e.g., Solidity, C++) may raise doubts as to the intention of a party that does not understand these computer languages to commit to the smart contract's obligations.

Against this background, and because *ex-post* enforcement is more difficult in the case of smart contracts than contracts in prose, disclosure could have a role to play. That said, it could also prove ineffective or endanger smart contracts, depending on the degree of the obligation being imposed. Regulators will thus be required to find the right balance, somewhere between the Belarus (which presumes that contracting parties have been duly informed of their rights when they use a smart contract, unless one party provides proof to the contrary)[102] and heavy

---

[95] "Certainty" is a recurring element across all Member States — it implies that one must clearly define contractual obligations. But this element is not particularly problematic in the context of smart contracts because they are expert systems ("if/then") requiring binary rules. I have therefore decided not to explore it. Similarly, I have deliberately chosen not to discuss "consideration" or "cause." They both require that the parties agree to a change in their present situation. This is also not especially problematic in the context of smart contracts.

[96] Kelvin F. K. Low & Eliza Mik, *Pause the Blockchain Legal Revolution*, 69 INT. COMPARATIVE L. QUARTERLY 7 (2020).

[97] Christian Sillaber & Bernhard Waltl, *Life Cycle of Smart Contracts in Blockchain Ecosystems*, 41 DATENSC. UND DATENSI. 498 (2017).

[98] Miners are generally called "validators" when the blockchain uses Proof of Stake instead of Proof of Work.

[99] These conditions are: offer, acceptance, and intention to be legally bound. *See* Jan M. Smits, *Contract Law a Comparative Introduction* (Edward Elgar Publishing, 2017).

[100] This calls for studying validity rules within each Member State. Difficulties will arise when a smart contract is recognized as being valid in one country but is denied legality in another.

[101] In case of doubt, the intention is analyzed retroactively and objectively, i.e., reading the contract and pre-contractual discussions.

[102] Agata Ferreira, *Regulating Smart Contracts: Legal Revolution Or Simply Evolution?*, 45 TELECOM. POL. 1, 12 (2021). To be sure, the Belarus is not a Member State, but it benefits from the DSM, *see* European Commission, *Eastern Partnership: Promoting Stronger Digital Economies for the EU's Six Eastern Partners*, EUROPEAN COMMISSION (Mar. 8, 2021) https://digital-strategy.ec.europa.eu/en/news/eastern-partnership-promoting-stronger-digital-economies-eus-six-eastern-partners [https://perma.cc/7FRF-6GMJ].

paper-based disclosures. A start could be disclosing smart contracts' immutable nature to consumers or considering technical solutions, as explained in B.2.

Proposal #1: Harmonize disclosure rules in a way that smart contracts are not disadvantaged compared to other transactional means.

Further, a question arises as to *when* the consents are exchanged, in other words, when does the contract come into being.

In general contract law, there must be an offer and its acceptance. In the context of a smart contract, one user deploys it on the blockchain, and another one calls its functions. Therefore, a first question arises whether deploying the smart contract counts as an offer. The answer will be negative should one consider the deployment of the smart contract to be an invitation to negotiate. However, it will be positive should one consider it an offer. Member States should lean towards the positive by regarding it as an offer, as smart contracts embed fixed conditions. It will nonetheless prove necessary to ensure harmonization in the matter.

Proposal #2: Recognize the deployment of smart contracts on a blockchain as a legally binding offer.

Next, one must question the legal qualification to give the calling of a smart contract. In practice, calling a smart contract function amounts to agreeing to it as it is. This acceptance derives from the fact that smart contracts are immutable, i.e., their terms cannot be negotiated after they are recorded on the blockchain. Consents are therefore exchanged every time a call is made.[103]

Proposal #3: Recognize the calling of smart contracts as proof that consents have been exchanged.

Last, a legal question concerns whether the machine's *automated* activation of a smart contract (e.g., when an oracle transmits the information triggering the transaction) could constitute an exchange of consents. Should that not be the case, only the manual activation would be sufficient. In all likelihood, imposing activation by hand would put a technical burden on a users' shoulders and deter the use of smart contracts. It should be rejected for this reason.

Proposal #4: Recognize the automated exchange of consents as contractually valid.

(ii) In addition to substantial rules, different European countries impose form requirements. Even if the general principle is the absence of such requirements, there are many exceptions. First, specific *mediums* may be required for sending *pre-contractual information*. For example, the Directive 2011/83/EU holds that "[w]ith respect to off-premises contracts, the trader shall give the information provided for in Article 6(1) to the consumer on paper or, if the consumer

[103] Kristian Lauslahti et al., *Smart Contracts - How Will Blockchain Technology Affect Contractual Practices?*, 68 ETLA REP. 1, 16 (2017). Declaration of intent does not, therefore, appear to occur separately from the conclusion or execution of a smart contract. It is rather an immovable part of the contract itself.

agrees, on another durable medium."[104] Blockchain should be recognized as such a durable medium; after all, immutability makes it durable by definition.[105]

Second, the *medium* used for the *transaction itself* may be subject to formal requirements. A proposal for amending the EU Regulation 910/2014 suggests that "[a]n electronic attestation of attributes shall not be denied legal effect and admissibility as evidence in legal proceedings solely on the grounds that it is in electronic form."[106] This proposal would cover smart contracts and appears essential to them. Denying smart contracts' legal effect because they are recorded on a blockchain with no equivalent in the physical space discriminates against their very essence. Doing so further deeply affects their chances of survival against other transactional modes.

For this reason, the recent German initiative allowing electronic securities to be recorded on blockchain,[107] therefore creating alternatives to the requirement for a paper-based certificate, should be commended.[108] Conversely, obligations to register contracts in a specific State-owned registry prove to be problematic. For example, article 28(1) of the Law no.51/1995 for the organisation and practice of the legal profession in Romania specifically provides that "[a] lawyer appearing on the bar's table shall be entitled to assist and represent any natural or legal entity, based on a contract concluded in a written form, which acquires a certified date after being recorded in the official register of evidence."[109] A similar rule exists for the transfer of *in-*

[104] Directive 2011/83/EU of the European Parliament and of the Council of 25 October 2011 on consumer rights *OJ L 304*, Article 7.
[105] *See* Catalina Goanta, *Information Duties in the Internet Era: Case Note on Content Services Ltd v. Bundesarbeitskammer*, 3 EUR. REV. OF PRIV. L. 643 (2013).
[106] Proposal for a Regulation of the European Parliament and of the Council amending Regulation (EU) No 910/2014 as regards establishing a framework for a European Digital Identity COM/2021/281 final. For a similar idea implemented outside Europe, *see* the State of Illinois, Illinois General Assembly, Public Act 101-0514 LRB101 11071 RJF 56276 b, Section 10(a) ("A smart contract, record or signature may not be denied legal effect or enforceability solely because a blockchain was used to create, store or verify the smart contract, record or signature").
[107] Currently, the scope of the German law addressing electronic securities only addresses bearer bonds, *see* Bundesministerium der Justiz und für Verbraucherschutz, *Gesetz zur Einführung von elektronischen Wertpapieren*, Bundesministerium der Justiz und für Verbraucherschutz (Jun. 10, 2021) https://www.bmjv.de/SharedDocs/Gesetzgebungsverfahren/DE/Einfuehrung_elektr_Wertpapiere.html [https://perma.cc/Z2FL-UWCD]. When registering the electronic securities, the bearer bond issuers have the choice of using the existing paper form or one of the two electronic forms of a central register: central securities depository (CSD see section 12 clause 2 eWpG) or the crypto securities register, *see id*. at section 4(3). Also, the Principality of Liechtenstein introduced the Token and Trusted Technology Service Provider Act (TVTG) more than a year ago (1st of January 2020), *see* Gesetz Vom 3. Oktober 2019 über Token und VT-Dienstleister, Nr. 301, Ausgegeben Am 2. Dezember 2019, https://www.gesetze.li/konso/pdf/2019301000?version=3 [https://perma.cc/B3RD-TMV4]. Lastly, outside of the EU, the Parliament in Switzerland has unanimously adopted the bill to adapt federal law to developments in distributed ledger technologies. It amended the country's securities law, therefore providing a secure legal basis for the trading of rights through electronic registers, *see* State Secretary For International Finance SIF, *Blockchain*, State Secretary For International Finance SIF (Jul. 27, 2021) https://www.sif.admin.ch/sif/en/home/finanzmarktpolitik/digit_finanzsektor/blockchain.html [https://perma.cc/733D-UEMU].
[108] *See* Bundesministerium der Justiz und für Verbraucherschutz, *Gesetz zur Einführung von elektronischen Wertpapieren*, Bundesministerium der Justiz und für Verbraucherschutz (Jun. 10, 2021) https://www.bmjv.de/SharedDocs/Gesetzgebungsverfahren/DE/Einfuehrung_elektr_Wertpapiere.html [https://perma.cc/Z2FL-UWCD]. Vermont also recognizes that "[a] fact or record verified through a valid application of blockchain technology is authentic," *see* General Assembly of The State of Vermont, House Bill 868: An Act Relating To Miscellaneous Economic Development Provisions No. 157, VT H0868, 118, https://legislature.vermont.gov/bill/status/2016/H.868 [https://perma.cc/5EW3-P926]. A similar legislation exists in Arizona, *see* Legislature of The State of Arizona, House Bill 2417: Signatures; Electronic Transactions; Blockchain Technology AZ HB2417 https://www.azleg.gov/legtext/53leg/1r/bills/hb2417p.pdf [https://perma.cc/E82N-GGR6].
[109] *See* Law no. 51/1995 for the Organisation and Practice of the Legal Profession (Lege nr. 51 din 7 iunie 1995: Pentru Organizarea şi Exercitarea Profesiei De Avocat, https://www.unbr.ro/legea-nr-100-din-01-iulie-2020-privind-completarea-legii-nr-51-1995-pentru-organizarea-si-exercitarea-profesiei-de-avocat/ [https://perma.cc/3AAR-

rem rights.[110] This obligation amounts to imposing the creation of smart contracts' paper or digital equivalents, which harms the opportunities to use smart contracts in such cases.

Proposal #5: Harmonize the recognition of blockchain as a durable medium for sending pre-contractual information, storing contracts and other legal documents.

Apropos of creating *paper versions*, several Member States impose that specific contracts have a "written form." In the absence of dedicated legislation, it appears that smart contract "code" will only meet this criterion depending on the courts' interpretation.[111] This could be harmonized so that it always does.

Proposal #6: Harmonize the interpretation of the "written form" requirement to include code.

Third, specific contracts (e.g., those with consumers) can require a *signature*.[112] Thanks to the EU regulation on electronic identification and trust services for electronic transactions, electronic signatures produce the same effect as handwritten signatures.[113] That said, Member States can require handwritten signatures under specific circumstances. And when they do not, they can impose specific guarantees for the validity of electronic signatures, such as requiring a digital certificate or the like.

EU regulation may want to harmonize the rules regarding the validity of smart contracts' signatures. Simply put, smart contracts should not be put at a disadvantage relative to other transactional means because existing constraints turn out to be particularly troublesome for smart contracts. This implies that the type and method of signature that exists already should be recognized as valid.[114]

Proposal #7: Harmonize the rules regarding the type of electronic signatures and related constraints for signing a smart contract.

WYNW]. Also, the decision no. 64/2011 on the Legal Profession, Adopted by the National Association of the Romanian Bars details the above provision, *see* Statut Din 3 decembrie 2011 al Profesiei De Avocat (la data 19-dec-2011 Actul A Fost Adoptat De Hotărârea 64/2011, http://legislatie.just.ro/Public/DetaliiDocumentAfis/133881 [https://perma.cc/895G-YMRD].

[110] Article 1244 of the New Civil Code in Romania (Law no. 287/2009 Codul civil al României) provides the registration form required for a contract that establishes or transfers real rights (*in-rem*). Unless the contract is authenticated/certified and registered in the land registry, the respective contract becomes null and void. *Id*, article 885 provides that *in rem* rights over registered property are acquired only by registering them in the land registry. *Id.* Article 885(2) further emphases that *in-rem* rights can be forfeited or extinguished only by deleting them from the land registry with the consent of the right holder, drafted and acknowledged by a notary.

[111] In Italy, the law no. 12/2019 expressly acknowledges that smart contracts satisfy the written form requirement. *See* Consiglio Nazionale del Notariato, L. 12/2019 - Smart Contract E Tecnologie Basate Su Registri Distribuiti - Prime Note, https://www.notariato.it/sites/default/files/S-1-2019-DI.pdf [https://perma.cc/Z2FL-UWCD]. The implementing measures have not yet been published, despite the injection to do so within 90 days.

[112] For example, Article L. 221-9 of the French Consumer Code (Code de la consommation, Edition 2021) holds that "[t]he trader shall provide the consumer with a dated copy of the off-premises contract, on paper, signed by the parties or, with the consumer's agreement, on another durable medium, confirming the parties' express commitment." [free translation].

[113] Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market *OJ L 257* ("An electronic signature shall not be denied legal effect and admissibility as evidence in legal proceedings solely on the grounds that it is in an electronic form or that it does not meet the requirements for qualified electronic signatures.").

[114] *See* our developments on "smart contract functioning", 2.2.1.

In the end, one may want to harmonize smart contracts' form requirements so that the same smart contract is legally valid all across Europe, and the legal environment is neutral as to contractual means (e.g., in prose, on a blockchain, etc.).

Admittedly, harmonization issues are nothing new. But these issues are becoming more pressing with smart contracts because the law cannot apply *ex-post* as it does with contracts in prose. That is explained by the fact that in the absence of centralization, smart contracts cannot easily adapt to different legal regimes as contracts in prose can. To be sure, one could draft different versions of the same smart contract for each Member State, but the core characteristics (e.g., immutability, visibility) will remain without action at the layer 1. Also, because they are immutable, new regulations taken by one Member State will prove ineffective against existing smart contracts.

The early convergence of European (smart) contract law across Europe will thus benefit all users of blockchain smart contracts. Moreover, it will increase smart contracts' chances of survival, effectiveness, and reliability.

Interpretation. New questions arise regarding the interpretation of smart contracts' content (i.e., the translation of code into a legal duty and vice-versa). With contracts in prose, one must determine the applicable law and explore the legal obligations they create. Here, courts and agencies are facing an additional problem even before interpreting the content of smart contracts: understanding it.[115]

In all likelihood, they will call upon experts (e.g., computer scientists) for translating smart contracts into natural language. The potential lack of legal training of these experts will be problematic. For this reason, one may recommend that courts and agencies dealing with blockchain and smart contracts hire computer scientists and involve them in their day-to-day activities.[116] Doing so will help the convergence of technical and legal expertise.[117]

Proposal #8: Set up data units within European and national courts or regulatory agencies with technical expertise in blockchain and smart contracts.

Courts and agencies will also use technologies to translate computer language into natural language, as I show in the second section of this chapter. Here again, the lack of these systems' legal expertise will create issues in the translation process.

[115] Reggie O'Shields, *Smart Contracts: Legal Agreements for the Blockchain*, 21 N.C. BANKING INST. 177, 190 (2017) (explaining that smart contracts may pose evidentiary issues in the absence of a proper translation).

[116] Several regulatory agencies have already started at the national level, *see* for example the Authority for Consumers & Markets in the Netherlands, *Toezichthouden ondergaat een digitale transformatie*, ACM (Jul. 27, 2021) https://www.acm.nl/nl/organisatie/werken-bij-de-acm/over-onze-mensen/kijkje-de-keuken/toezichthouden-ondergaat-een-digitale-transformatie [https://perma.cc/9Y2K-577L], and the CMA's DaTA unit in the United Kingdom, Stefan Hunt, *The CMA DaTA unit – We're Growing!*, Gov.UK (May 28, 2019) https://competitionandmarkets.blog.gov.uk/2019/05/28/the-cma-data-unit-were-growing/ [https://perma.cc/5VR2-NHUQ]. A network of competition agencies interested in cooperating with computer scientists has been formed at Stanford University, *see* CodeX, *Computational Antitrust*, Stanford Law School (July 27, 2021) https://law.stanford.edu/computationalantitrust [https://perma.cc/TD62-2CUR].

[117] Alternatively, the European Commission could provide support to national courts and agencies seeking to appoint third-party experts by creating a database of experts in blockchain and smart contracts.

In either case, courts and agencies will be dependent on the translation offered by third parties. They will not be able to review it and engage in a constructive dialog. To remedy this critical issue, one could start offering training programs to judges willing to learn programming and computational thinking basics.[118] One could then create specialized technology chambers, similar to chambers specialized in patent law for which technical expertise is often required. To be clear, I do not say that courts and agencies should know how to code a smart contract. But I do say that they should have the level necessary to evaluate the results submitted to them critically. Our legal system must have the technical means and knowledge to maintain its independence.

Basic technical knowledge is all the more important as smart contracts are incomplete and therefore require courts to attach legal obligations to them. This is so, first, because smart contracts cannot automate all contractual obligations.[119] Like a vending machine that does not automate the guarantee clause (it is simply displayed on the machine), smart contracts will not be able to compute the entire legal system.[120] Second, unexpected events occur, requiring the courts to interpret how to adapt smart contracts' obligations.[121] Here again, learning about code and its limitations proves necessary for the proper administration of legal rules.

Proposal #9: Support or develop education programs dedicated to technology literacy (i.e., to teach the basics of programming languages and computational methods) for judges and regulatory agencies.

→ **Performance**

Apprehending the performance of smart contracts requires an exploration of their technical functioning. It also entails studying the ability of the parties, or third parties, to pre-program different methods in order to stop them or force their execution, in other words, to interfere with their natural dynamism.

Execution. As I explained in Chapter 2, smart contracts' bytecode is put on the chain — within a block — once its technical integrity is validated. The code is then executed when the conditions are met.[122] These conditions can be simplistic (e.g., a user is willing to call the smart contract) or more complex (e.g., an oracle provides specific information). Some smart contracts are never called. When they are, the transaction is executed and then recorded on the blockchain. The ledger may or may not keep track of the fact that a smart contract actually triggered the transaction.[123]

[118] For starters, *see* Thibault Schrepel, *A list of open-access resources to learn computer science*, CONCURRENTIALISTE (May 13, 2021) https://leconcurrentialiste.com/computer-science-resources/ [https://perma.cc/B3U9-9LHT]. Custom-made courses should further be designed for the purpose.
[119] This is all the more true that the law could be amended during the smart contract's lifecycle.
[120] The incomplete nature of smart contracts poses legal difficulties, especially when conditions not anticipated in the contract arise. That being said, their incompleteness is also a feature. Indeed, the parties may want to use a smart contract precisely to automate performance despite unexpected real-world events. They may want certainty in its execution, however problematic it may be. On the completeness of contracts, *see* Oliver Hart and John Moore, *Incomplete Contracts and Renegotiation*, 56 ECON. 755 (1988).
[121] *See* Schrepel, *supra* note 2, at 241. Also, Kraus et al., *supra* note 60, at 101 et seq. ("If a smart contract contains a gap, the immutable character of the blockchain prevents it from filling that gap"). Lastly, *see* Nassim Nicholas Taleb, *The Black Swan: The Impact of the Highly Improbable* 1–10 (Random House, 2010).
[122] Sillaber & Waltl, *supra* note 97.
[123] This depends on each blockchain.

In any case, the new state of the blockchain (i.e., with the latest transaction) reflects the transaction in the registry while the smart contract does not disappear. It remains in the original block, and it can be "called" by other users if conditions allow. Even after it is used, this persistence of the smart contract led Vitalik Buterin, co-creator of Ethereum, to express his regrets for having used the term "smart contract." He would have found it more appropriate to call them "persistent scripts."[124]

Stop. The question that logically arises is how to stop a smart contract without dedicated mechanisms (such as described in 3.2). A solution comprises relying on the contracting parties to integrate a kill switch in their design. The parties can modulate the conditions for killing them, for example, by submitting the switch to the agreement of all the co-contractors or certain trusted entities (e.g., external users operating in decentralized court systems or institutions).
In the absence of empirical data documenting the percentage of smart contracts featuring a kill switch, I should like to underline two opposite tendencies. On the one hand, some users are likely tempted to include a kill switch if they think they will change their minds and, therefore, fear smart contracts' immutability. On the other hand, some users will not anticipate a potential problem in the contract's execution (because of overconfidence) or will prefer to reduce monitoring costs by maintaining total immutability.[125] A technical question will then arise as no user — not even the parties — will be able to stop the smart contract.

This inability will create issues in ensuring the rights of these parties. It will also prove problematic on a broader scale, such as maintaining the coherence of the DSM in a context where legal enforcement cannot rely on the usual means. In short, the smart contracts that do not respect the principles of European law must be stopped, whether the parties have pre-agreed to it or not.

Enforcement. Other difficulties arise with enforcing the obligations that smart contracts cannot automate. For example, a smart contract may automate the payment of a product once it has been shipped. Should it be lost or damaged, restitution could need to be ordered. Simply put, smart contracts automate certain obligations (here, the debtor's), but they do not replace legal enforcement with real-world obligations.[126] The legal system has thus an important role to play, even if decentralized dispute resolution systems are gradually emerging.[127]

Now, it turns out that blockchain communities have a role to play in opening the door for legal enforcement. This technical necessity is a significant difference from other technologies where one can simply impose the law from above. Absent such gate opening, an infertile confrontation between law and technology will deprive all users of the ability to enforce dispute resolution decisions without endangering the blockchain environment they use. For example, enforcers will be tempted to force the termination of smart contracts by imposing drastic measures (e.g., shutting down all the computers running the blockchain) should they not be

[124] Vitalik Buterin, *Replying to @CleanApp @cryptoecongames and 4 others*, TWITTER (Oct. 13, 2018) https://twitter.com/VitalikButerin/status/1051160932699770882 [https://perma.cc/YYM3-KH6L].
[125] Discussing the overconfidence bias, *see* Daniel Kahneman & Amos Tversky, *Intuitive Prediction: Biases and Corrective Procedures* (Cambridge University Press, 1997).
[126] Sai Agnikhotram & Antonios Kouroutakis, *Doctrinal Challenges for the Legality of Smart Contracts: Lex Cryptographia or a New, Smart Way to Contract*, 19 J. HIGH. TECH. L. 300, 318 (2019).
[127] *See* Kleros, The Justice Protocol, Kleros (Jul. 6, 2021) https://kleros.io/ [https://perma.cc/46JZ-F79P], *also*, Aragon, Govern Better, Together (Jul. 12, 2021) https://aragon.org/ [https://perma.cc/2QML-9SMD].

given technical access.[128] As I shall discuss, that access could be granted at the level of each smart contract or at the blockchain layer 1.[129]

4.1.1.2 Cross-border environment

The cross-border nature of contractual relationships can be assessed in two different ways. If established, it produces a double effect that existing laws applicable to international smart contracts do not entirely capture.

→ **When is a transaction by smart contract cross-border?**

Cross-border transactions are the subject of extensive case law. The first way to characterize them is to use a legal criterion.[130] A contract is cross-border when all of its elements are located in more than one legal system. The second way uses an economic angle. The contract is cross-border when assets flow across the borders of at least two countries.[131]

In this context, a contract is cross-border when it involves an economic transaction between parties in two different countries, as it fulfills the legal and economic criteria. This is so, for example, when an Italian citizen buys a product from a French distributor. Conversely, the cross-border nature of the transaction is harder to establish when it involves two parties in the same country but who transact using a tool located and operated in another country. Such a transaction does not fulfill the economic criterion but could satisfy the legal one depending on the court's interpretation.

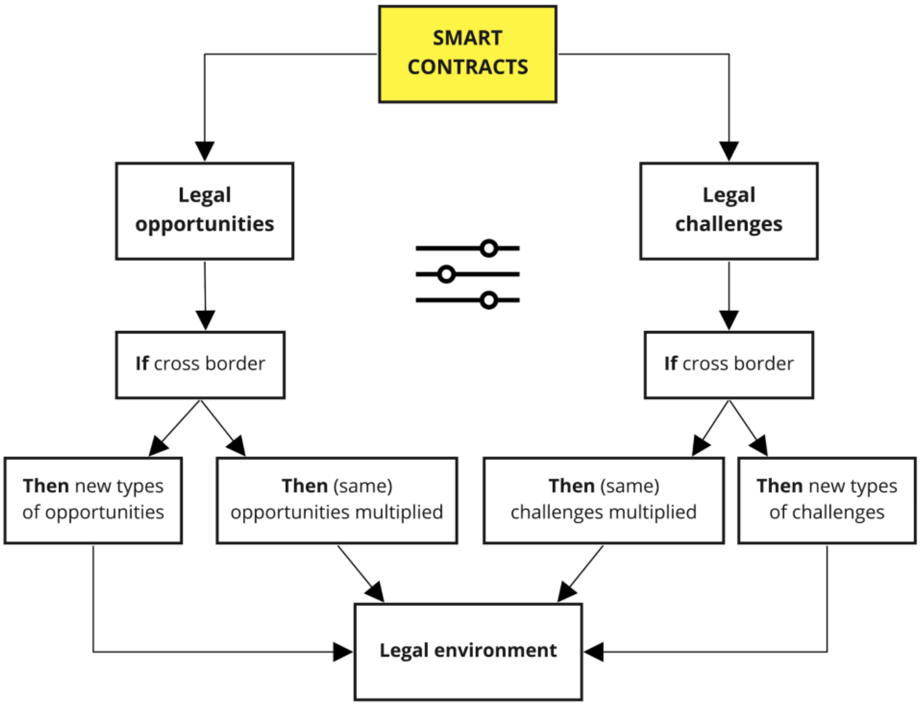
In that regard, a transaction involving two Italian citizens who enter a sales contract and use a payment system located in the United States (e.g., Stripe) is already difficult to qualify. But blockchain goes a step further: it not only *facilitates* the transaction, but it also uses nodes to authorize and *execute* the script (at least the payment part).[132] These nodes may be located in different countries when it comes to large public blockchains, meaning that all smart contracts operated on such blockchains could be considered cross-border.

Proposal #10: Use soft law to clarify the criteria for a smart contract to be cross-border.

→ **Why does it matter?**

The cross-border nature of a smart contract has a double effect: it multiplies existing opportunities and challenges while creating new ones.

[128] This measure would be effective with respect to private blockchains.
[129] Discussing these drastic measures, *see* Schrepel, *supra* note 2, at 236–238.
[130] *See*, for instance Case C-55/98 *Vestergaard* ECLI:EU:C:1999:533 [1999] para 19; Case C-381/93 *Commission v France* ECLI:EU:C:1994:370 [1994] para 14; Case C-154/89 *Commission v France* ECLI:EU:C:1991:76 [1991] para 10.
[131] *See*, for instance Case C-386/05 *Color Drack GmbH* ECLI:EU:C:2007:262 [2007] para 40; Case C19/09 *Wood Floor Solutions* ECLI:EU:C:2010:137 [2010] paras 40–41.
[132] Weiqin Zou et al., *Smart Contract Development: Challenges and Opportunities*, 1 IEEE TRANS. SOFTW. ENG. 1, 3 (2019).



© Thibault Schrepel (2021)

On the one hand, blockchain users may find a particular interest in using smart contracts in cross-border transactions. First, this is so because the international nature of the relationship multiplies the existing benefits of smart contracts. The latter reduces friction points by automating the execution of contractual obligations. This decrease is even more helpful in international transactions, where legal recourse might be more complicated and expensive than it is with domestic transactions. Second, the cross-border aspect of the transaction creates new *opportunities* for smart contracts. Their immutable nature prevents hostile courts of justice from canceling them, even when they are legally formed.[133] This could prove handy when involving specific non-Member States.

On the other hand, the cross-border nature of the relationship multiplies existing *challenges*. For example, I have explained that immutability can create enforcement difficulties, like in case a transaction needs to be stopped or canceled. These issues will be reinforced when a foreign court seeks to enforce national law in another Member State. Further, the cross-border aspect creates new challenges. The legal divergence regarding the recognition of smart contracts may put the parties in an unsustainable situation. In addition, one can use the same smart contract across Europe for several years. When its validity is not admitted in one country, the parties will have to deal with the disconnect between the technical obligations and legal ones each time they use it.

That said, these opportunities and challenges are intrinsically linked to each other. The advantages and drawbacks of immutability are particularly telling in this regard. As a response, legal rules must help create a reliable ecosystem in which smart contracts can flourish while preserving and enhancing the DSM. Doing so requires a uniform legal environment across

[133] Discussing the issue, *see* Schrepel & Buterin, *supra* note 21.

Europe.[134] In the absence of such uniformity, smart contracts will adapt to one legal environment while possibly causing greater legal challenges in other countries.

→ The law applicable to international smart contracts

The law applicable to cross-border relationships provides answers to some challenges created by smart contracts.[135] I would like to explain that it remains incomplete.

Several international treaties govern contractual relationships. For example, the United Nations Convention on Contracts for the International Sale of Goods of April 11, 1980, covers international sales contracts. This body of law which supersedes national laws thus covers smart contracts of this type. But many cross-border contractual relationships are not covered by a *sui generis* law.[136] The parties can then choose the law applicable to their (smart) contract, although various conventions limit this freedom. This is the case of Rome 1, which applies to cross-border contractual obligations,[137] and the 1955 Hague Convention, which applies to the international sale of goods.[138] These conventions attach (smart) contracts to national legal systems according to specific rules.

Article 3 of Rome 1 repeats the principle according to which the parties may choose the applicable law to their contract[139] while holding that it needs not to be explicit.[140] Thus, a smart contract concluded between two Italian citizens can be expected to be governed by Italian law. That said, if the contract falls into one category set out in Article 4(1) (e.g., contracts for the sale of goods, contracts for the provision of services, franchise contracts, etc.), special rules are substituted for the general rules. For example, "a contract for the sale of goods shall be governed by the law of the country where the seller has his habitual residence." Where the contract falls outside these categories and does not contain a clause specifying the applicable law elected by the parties, the applicable law is attached to their domicile.[141]

Finally, articles 3 and 4 of Rome 1 leave room for articles 5 to 8 if one party is more vulnerable than the other. This is the case for sales contracts between consumers and professionals. There, the parties cannot elect a law that "result[s] in depriving the consumer of the protection (…) which, in the absence of choice, would have been applicable."[142] This provision entails that professionals cannot offer the same smart contract to consumers across EU Member States.[143] To begin with, this requires that smart contracts' creators distinguish between professional and non-professionals.

[134] On the legal environment for smart contracts, *see* Tonya M. Evans, *The Role of International Rules in Blockchain-Based Cross-Border Commercial Disputes*, 65 WAYNE L. REV. 1, 7–8 (2019).
[135] On the subject, Benedetta Cappiello & Gherardo Carullo, *Blockchain, Law and Governance* 159–180 (2020).
[136] The parties may also decide to opt-out of the convention, *see* Convention on the Law Applicable to International Sale of Goods (The Hague Convention) - The Hague (1955), Article 6.
[137] Rome 1 Regulation only applies to smart contracts when they create legal obligations between the parties, see the criteria listed above. When this is not the case, smart contracts are never more than a piece of software that escapes Rome 1.
[138] The Hague Convention, *supra* note 136.
[139] Regulation (EC) No 593/2008 of the European Parliament and of the Council of 17 June 2008 on the law applicable to contractual obligations (Rome I) OJ L 177, Article 3(1)).
[140] *Id.* at Article 3 ("The choice shall be made expressly or clearly demonstrated by the terms of the contract or the circumstances of the case").
[141] *Id*. at Article 4(2).
[142] *Id*. at Article 6.
[143] *See* Cappiello & Carullo, *supra* note 135, at 159–180.

4.1.1.3. Other legal considerations

Many rules that fall outside contract law have a substantial effect on smart contracts. I discuss two of them as they touch upon the very nature of smart contracts and may therefore affect their evolution: data protection and data localization, as well as the AI Act.

→ Data protection & data localization

Smart contracts can make use of personal data. This is so, for example, when they use location data to document the movement of goods and persons. Moreover, smart contracts can create new personal data, as when they are used to transfer and record the ownership of certain assets. In a nutshell, the GDPR can apply to them depending on which data they use and generate (i.e., whether or not these data qualify as personal data).[144]

This leads to considering smart contracts in the face of the "right to be forgotten." According to article 17 of the GDPR, "[t]he data subject shall have the right to obtain from the controller the erasure of personal data concerning him or her without undue delay." Despite a vast literature on the subject, the *a priori* incompatibility of blockchain and GDPR remains an issue.[145] Technical proposals are put on the table to solve it, such as blocking access to data, not storing personal data in clear-text, or using zero-knowledge proofs.[146]

I shall raise one important point in that regard. Regulators and policy-makers should want to preserve the characteristics differentiating smart contracts from other contractual means. This implies that they compare different solutions to data protection-related problems and evaluate which alternatives maintain the best distinction between smart contracts and other ways of contracting. To be clear, a balance may have to be struck between compliance with the GDPR and maintaining smart contracts' chances of survival.[147] For example, one could argue that an exception to the right to be forgotten should be created for smart contracts. Alternatively, one could narrow down the interpretation of the GDPR in a way that blockchain public addresses are not considered personal data.

Proposal #11: Compare technical solutions allowing smart contacts' compliance with GDPR and, if necessary, introduce exceptions to it.

Data localization requirements also come to mind as a barrier to smart contracts.[148] Public permissionless blockchains cannot comply with requirements to store data within the European

[144] World Economic Forum, *Bridging the Governance Gap: Interoperability for Blockchain and Legacy Systems* 28 (2020).
[145] Michèle Finck, *Blockchains and Data Protection in the European Union*, 4 EUROP. DATA PROTECT. L. REV. 17 (2018), also, Emmanuelle Ganne, *Can Blockchain Revolutionize International Trade?*, 102 (2018).
[146] Using zero-knowledge proofs, one could verify the accuracy of a statement without access to the data underlying it. For a discussion, *see* Teresa Alameda, *Zero Knowledge Proof: How To Maintain Privacy In A Data-Based World*, BBVA (Jun. 23, 2020) https://www.bbva.com/en/zero-knowledge-proof-how-to-maintain-privacy-in-a-data-based-world [https://perma.cc/3SN2-XKQ5].
[147] I say "may" not to exclude the possibility that a new technical solution will maintain that differentiation while enabling full compliance with the GDPR.
[148] These requirements are also referred to as data sovereignty. Discussing their impact on blockchain, *see* Ganne, *supra* note 145, at 101. More generally, *see* Nigel Cory & Luke Dascoli, *How Barriers to Cross-Border Data Flows Are Spreading Globally, What They Cost, and How to Address Them*, INFORMATION TECHNOLOGY & INNOVATION FOUNDATION (Jul. 19, 2021) https://itif.org/publications/2021/07/19/how-barriers-cross-border-data-flows-are-spreading-globally-what-they-cost [https://perma.cc/D4LN-H5T8] ("The number of data-localization measures in

Union or an EU Member State because, by definition, they are stored all around the globe. Here again, one could find workarounds, for example, by recording only the hash value of data on the blockchain (should the hash not be considered personal data). Still, data localization requirements, even coordinated at the EU level, will hamper the use of smart contracts by forcing information out of the blockchain, and therefore endangering its integrity. Should the European Union enact such data localization requirements, one will want to create exceptions for the data used and generated by smart contracts.

Proposal #12: Allow exceptions to data sovereignty regulations for the data used and generated by smart contracts.

→ AI Act

The current draft of the AI Act covers smart contracts. According to Annex I, this Act applies to "[l]ogic- and knowledge-based approaches, including knowledge representation, inductive (logic) programming, knowledge bases, inference and deductive engines, (symbolic) reasoning and expert systems." As previously described, smart contracts function with "if/then" rules, making them an expert system.

For the AI Act to apply, smart contracts will also have to "bring (…) new risks or negative consequences for individuals or the society."[149] Some will. Depending on why smart contracts are used, they will create "unacceptable" risk, high-risk, limited risk, or minimal risk. For example, the "[m]anagement and operation of critical infrastructure" is considered high-risk by the AI Act, meaning that smart contracts of this nature will have to undergo a conformity assessment procedure operated by the European Commission[150] if they are not already on the market.[151] Going through such a procedure will create a powerful incentive not to use smart contracts but to use contracts in prose instead.

Further, the technical environment of smart contracts (i.e., the blockchain) is not compatible with the AI Act. This Act has been designed for centralized expert systems where a single agent controls the data that feeds into it and the system's evolution. One can easily conceive of submitting that agent to the obligations of the AI Act. Doing so will create an incentive for the agent to comply with the law. But when, on the contrary, an expert system is decentralized, no single agent controls it.[152] This is so with public permissionless blockchains where the power is shared amongst participants. Submitting decentralized expert systems to the AI Act will therefore be ineffective. In the absence of control, one can hold no agent liable for what is happening at the blockchain level, meaning that they have no incentive to comply with the law.[153] It will also prove detrimental to the development of smart contracts by creating legal

force around the world has more than doubled in four years. In 2017, 35 countries had implemented 67 such barriers. Now, 62 countries have imposed 144 restrictions.").

[149] Proposal for a Regulation of the European Parliament and of the Council Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act) COM(2021) 206 final 1.

[150] *See id.* at Title III, Chapter 2, Article 43.

[151] *Id.* at Article 83.

[152] That said, should a kill-switch or a possibility to amend the smart contract be pre-programmed, a user is controlling it.

[153] Holmes, *supra* note 22, at 3 ("If you want to know the law and nothing else, you must look at it as a bad man, who cares only for the material consequences which such knowledge enables him to predict, not as a good one, who finds his reasons for conduct, whether inside the law or outside of it, in the vaguer sanctions of conscience").

uncertainty for all users who would be liable (despite having no power of command and control) should the regulator refuse *de facto* non-liability. For that reason, one may want to exclude smart contracts from the AI Act.

Proposal #13: Redraft Annex I of the AI Act by adding the qualifier "centralized" to expert systems to exclude smart contracts: "Logic- and knowledge-based approaches, including knowledge representation, inductive (logic) programming, knowledge bases, inference and deductive engines, (symbolic) reasoning; and centralized expert systems;"

4.1.2. New rules and legal mechanisms

4.1.2.1. Adopting a "law is code" approach, but ex-post

Smart contracts were imagined and designed to automate transactions between parties. They were meant to escape the need for relying on legal institutions. Yet, smart contracts achieve the exact opposite of that. Because they are preventing (or at least complicating) ex-post intervention, smart contracts and blockchain are pushing for ex-ante intervention.[154] This creates a true "smart contract paradox."

One will thus not be surprised to observe that scholars have explored several avenues for enabling these ex-ante solutions. Two of them are amongst the most common. The first is an ex-ante "law is code" approach — a derivative of the maxim "code is law" according to which the code of digital things constrains online behaviors.[155] This approach is rooted in the idea that the regulator may want to convert legal obligations into code so that users will comply by design.[156] But adopting such a "law is code" approach may have a disruptive impact on the subject being regulated. Indeed, it not only changes the legal environment but also seeks to alter smart contracts themselves. It, therefore, comes at great risk.

A second solution consists not in impacting the code of smart contracts but having the regulator pre-approve the code of the smart contracts featuring the higher risks.[157] For example, one could subject smart contracts intended for consumers to pre-approval. Doing so would require the creation of an agency in charge of ensuring this sole mission. It would entail significant expenses. Implementing such a process of unnatural selection would also create strong discrimination between smart contracts and contracts in prose. One would systematically require a pre-approval for smart contracts while only a few contracts in prose are reassessed ex-post (e.g., in case of litigation). It would most likely end up having harmful consequences.

Against this backdrop, I would like to propose another approach. It consists of implementing a "law is code" approach, but ex-post. This variant of the classical "law is code" approach aims to be less intrusive, and therefore more respectful of smart contracts' evolutionary perspectives. It allows legal enforcement only once an illegal practice has been

[154] Kevin Werbach & Nicolas Cornell, *Contracts Ex Machina*, 67 DUKE L.J. 313, 377 (2017). Also, Jerry I-H Hsiao, *Smart Contract on the Blockchain-Paradigm Shift for Contract Law*, 14 US-CHINA L. REV. 685 (2017).
[155] Lessig, *supra* note 12, at 124 (explaining that "an analog for architecture regulates behavior in cyberspace—code. The software and hardware that make cyberspace what it is constitute a set of constraints on how you can behave.").
[156] Discussing how to enforce this, *see* 3.1.2.2, also, Schrepel, *supra* note 2, at 244–246.
[157] Daniel Drummer & Dirk Neumann, *Is Code Law? Current Legal and Technical Adoption Issues and Remedies for Blockchain-Enabled Smart Contracts*, 35 J. OF INFORMATION TECH. 11 (2020).

implemented.[158] It seeks to preserve the technology while ensuring the cohesion of the DSM by offering a means of intervention. All blockchain users would benefit from it. For example, this approach would allow regulators to stop smart contracts that are deemed illegal and thus avoid perpetuity traps.[159] And it would create an incentive not to lock consumers into abusive smart contracts. I will therefore explore the technical aspects of this approach in 3.2.

4.1.2.2. Implementing new rules

In this section, I shall discuss how to enable an *ex-post* "law is code" approach. Having said that, the mechanisms I discuss also prove relevant for implementing other regulatory methods.

→ Smart contract factory

Creating a "smart contract factory" containing templates of code is a recurring theme in the academic literature. Whether public authorities[160] or private actors[161] create these templates, one could make them mandatory or simply offer them for use on a voluntary basis.

Should they concern the first layer of blockchain, they would create a *de facto* compliance of all the higher levels. For example, impacting the immutability of a blockchain would affect the smart contracts built on it. But this effectiveness would come at the risk of disrupting the entire ecosystem as it could reduce blockchain differentiation from centralized ecosystems. Also, the implementation of these templates would create a strong "Brussels effect."[162] As previously explained, the power relationships within public and permissionless blockchains like Bitcoin and Ethereum are horizontal rather than vertical.[163] It entails the absence of a power of command and control, which applies to core developers, users, and miners. Imposing templates at the layer 1 will thus affect the entire ecosystem, which cannot be as easily forked as centralized systems do for each country.

Alternatively, these templates could impact the code of smart contracts or specific clauses.[164] These templates are less intrusive than the first ones, but the characteristics of the first layer constrain them. For example, limiting smart contracts' immutability by changing their code (e.g., forcing kill switch functions) requires designing workarounds to the layer 1. This is not an architectural solution, meaning that non-compliance is easier and monitoring costs are higher. Finally, these templates could concern the code and operation of oracles. As I will explain, oracles can ensure that smart contracts are called upon only when they comply with specific

[158] The topic of detecting illegal practices and smart contracts will also need to be discussed. Proposing the introduction of regulatory nodes, *see* Federal Reserve: Bank of Boston, *Beyond Theory: Getting Practical with Blockchain Boston Fed Learns by Doing With Blockchain Technology* 19 (2019). Suggesting the use computational tools for the purposes of detecting illegal practices, *see* Thibault Schrepel, *Computational Antitrust: An Introduction and Research Agenda*, 1 STAN. COMP. ANTITRUST 1 (2021).
[159] Werbach & Cornell, *supra* note 154, at 357 (discussing the desirability of requiring all smart contracts to introduce kill switch functions).
[160] Lyudmila Efimova et al., *Smart contracts between freedom and strict legal regulation*, 1 INF. & COM. TECH. 16 (2021) (discussing the potential role of the International Chamber of Commerce)
[161] Pete Rizzo, *How Barclays Used R3's Tech to Build a Smart Contracts Prototype*, COINDESK (Apr. 26, 2016), https://www.coindesk.com/barclays-smart-contracts-templates-demo-r3-corda [https://perma.cc/3DP2-ZKAG].
[162] Bradford, *supra* note 26.
[163] Schrepel, *supra* note 2, at 102. Also, Philipp Hacker et al., *Regulating Blockchain: Techno-Social and Legal Challenges* 16 (Oxford University Press, 2019).
[164] Xiwei Xu et al., *supra* note 67 (exploring how one could use templates).

legal rules.[165] One could imagine that public institutions will want to design such oracles and offer templates. They could also promote the templates designed by private actors. In any case, the monitoring costs will remain high as it will require to screen constantly which smart contracts are using them.

Proposal #14: Should the European Commission undertake the design of templates, it is recommended to make them voluntary while limiting them to smart contracts and oracles.

→ Comfort zones and interoperability

The mechanisms enabling the "law is code" *ex-post* approach (or any regulatory approach) could be more or less decentralized. Depending on whether they require a vote of a majority of blockchain users, selected users, or the sole decision of external individuals and institutions, these approaches will strike a unique balance between efficiency and the ability to preserve smart contract unique characteristics.[166]

The centralized mechanisms tend to be efficient, but they bring blockchain closer to existing transactional means. From this point of view, the legislator must be careful not to create a hostile environment for blockchain by preferring a technical solution that endangers the survival of the technology while another solution would have been (almost) equally effective but not as dangerous.

The more collective mechanisms provide a safeguard against the endangerment of blockchain, but they are not as effective. One must therefore seek to achieve the right balance through testing.[167]

One way of testing regulatory mechanisms is to use computer or lab simulations. Agent-based modeling can be helpful in the matter.[168] Experimental economics has also produced evidence of lab testing's usefulness.[169] They could allow measuring the impact of specific technical solutions on the use of blockchain.

Proposal #15: Set up experiments to preserve the integrity of smart contracts while enabling legal enforcement through an *ex-post* "law is code" approach.

Another way, perhaps complementary, is to implement legal comfort zones, i.e., innovation hubs, sandboxes, and safe harbors.[170] Faced with difficulties regulating blockchains and their participants, comfort zones allow policy-makers and regulators to turn the table around. They

[165] *See* 3.2.2.1.
[166] *See* Werbach & Cornell, *supra* note 154, at 375 (2017). Also, Samuel Bourque & Fung Ling Tsui, *A Lawyer's Introduction to Smart Contracts* (2014) (discussing the idea of having a central authority to stop smart contracts).
[167] Generally, *see* Politou et al., *supra* note 58.
[168] *See* Vernon L. Smith, *Microeconomic Systems as An Experimental Science*, 72 AM. ECON. REV. 923 (1982); *See* Michael Laver, *Agent-Based Models of Social Life: Fundamentals* (2020).
[169] *See* Daniel Kahneman & Vernon L. Smith, *Prize Lecture: Constructivist and Ecological Rationality in Economics*, THE NOBEL PRIZE (Dec. 8, 2002) https://www.nobelprize.org/prizes/economic-sciences/2002/smith/lecture/ [https://perma.cc/LN64-XU9Y]. Also, Peter Cane & Herbert M. Kritzer, *The Oxford Handbook of Empirical Legal Research* (Oxford University Press, 2010).
[170] Schrepel, *supra* note 2, at 244–247. Also, Dirk A. Zetzscheet al., *Regulating a Revolution: From Regulatory Sandboxes to Smart Regulation*, 23 FORDHAM J. CORP. & FIN. L. 31, 99 (2017) (discussing the path between these different tools).

make these participants want to be compliant to benefit from various privileges such as getting no-action letters. In addition, regulators could adopt a "comply or explain" approach, pushing companies that do not want to cooperate (i.e., work on implementing new solutions) to explain their approach.[171] These explanations could be made mandatory in some instances, and regulators should be able to impose sanctions when these explanations are not satisfactory.

Proposal #16: Create comfort zones to test the effectiveness of different regulatory mechanisms addressing smart contracts' most pressing issues.

4.2. Solutions enabled by the technology

In this section, I explore how blockchain could be designed for greater harmonization with legal constraints (and, generally, to foster the DSM). It leads me to consider how to implement the "law is code" *ex-post* approach to creating a more consistent environment for smart contracts. I explain that it entails enabling and increasing the influence of the law through technical means (**3.2.1**) and offering new solutions that the law alone cannot develop (**3.2.2**).

4.2.1. Enabling the law within blockchain

4.2.1.1. Rationale

Blockchain creates a fortress, not an impregnable fortress, but a fortress nonetheless.[172] This leads to enforcement concerns, but it also builds trust for users who can often rely on technology instead of relying on legal rules. That said, adding legal enforcement to the mix could create more trust (that is if the one created by blockchain is preserved). Applied to smart contracts, it amounts to maintaining their immutability as a principle while allowing exceptions to stop them when necessary.

There are several reasons for wanting to stop a smart contract.[173] First, the reason may be specific to the code of a smart contract. There are between 1 and 25 errors every 1000 lines of code.[174] This means that a vast majority of smart contracts contain coding errors. Further, smart contract code may end up not reflecting the intention of one (or more) of the parties.[175] This type of issue often appears only after the smart contract is executed.

Second, the reason may be related to the substance of the smart contract. The parties may want to terminate it because the obligations have been fulfilled, because they are exhausted or because they were not valid in the first place.[176] Here, evidence could be provided by only

[171] Hacker, *supra* note 163, at 33–34 (discussing the idea of introducing a mechanism that detects non-compliant smart contracts and raises a flag).
[172] Schrepel, *supra* note 2, at 230.
[173] Marino & Juels, *supra* note 32.
[174] Steve McConnell, *Code Complete: Practical Handbook of Software Construction* 521 (Microsoft Press, 2004).
[175] Law Commission, *supra* note 94, at 66–68 (making a distinction between situations where "the code fails to reflect the terms of a prior natural language contract (...) the parties' common intention (...) [or] one of the parties' intentions").
[176] For example, a smart contract automating payment after an organ donation is not valid.

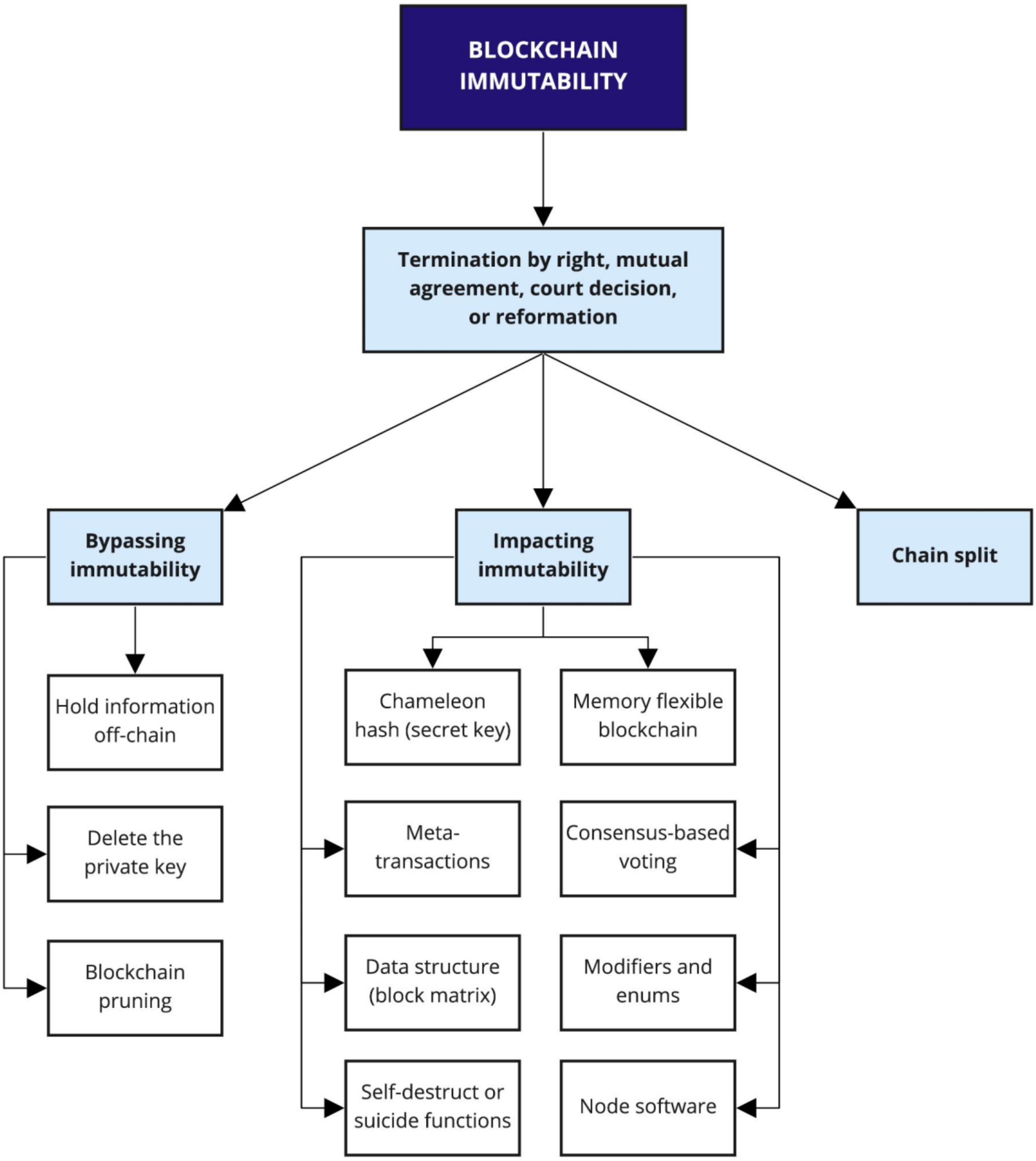
one party. They may also wish to terminate it by mutual agreement. In this situation, the proof that there is such a mutual agreement will be required.

Finally, the reason may be external to the substance of the smart contract itself but related to another statutory right. For example, a consumer may wish to make use of her or his right to withdraw from a smart contract in the context of e-commerce transactions.[177] Similarly, a court may want to cancel a smart contract if one party has not freely consented to it.[178] Overall, the ability to impact blockchain immutability is central in fostering the DSM and ensuring that all users enjoy the same legal protection, regardless of the Member State in which they are located.

4.2.1.2 How to proceed

Blockchain immutability results from its fundamental characteristics. The decentralization (i.e., control) and distribution (i.e., location) of blockchain prevent its participants from imposing vertical and unilateral decisions. Altering immutability thus comes at the risk of recentralizing blockchain. The importance of that risk explains why one may want to explore and compare different methods to select the one that allows for creating a "pragmatic immutability"[179] while maintaining most of its original identity.

[177] Directive 2011/83/EU, *supra* note 104, at Article 9.
[178] This court can also be digital, *see* Hitoshi Matsushima & Shunya Noda, *Mechanism Design with Blockchain Enforcement*, 1027 KIER WORK. PAP. (2020).
[179] Finck, *supra* note 145, at 17 (coining the term "pragmatic immutability").



© Thibault Schrepel (2021)

A first set of solutions comprises bypassing immutability, therefore creating a technical workaround without impacting it. To start with, users may store the hash value of the information on the blockchain, but not the information itself. Doing so allows them to delete the information if necessary and offers a practical solution to enforcement problems.[180] Alternatively, users may delete the private key used for the encryption. As previously explained, this will prevent the retrieval of the information stored on the blockchain. One may qualify this solution as a deliberate omission. Lastly, users may "prune" the blockchain, i.e.,

[180] Jacob Eberhardt & Stefan Tai, *On or Off the Blockchain? Insights On Off-Chaining Computation and Data*, 1 EUR. CONF. ON SER. OR. AND CL. COMP. 3-15 (2017).

reduce the size of the ledger by eliminating information from it and mine new blocks without it.[181]

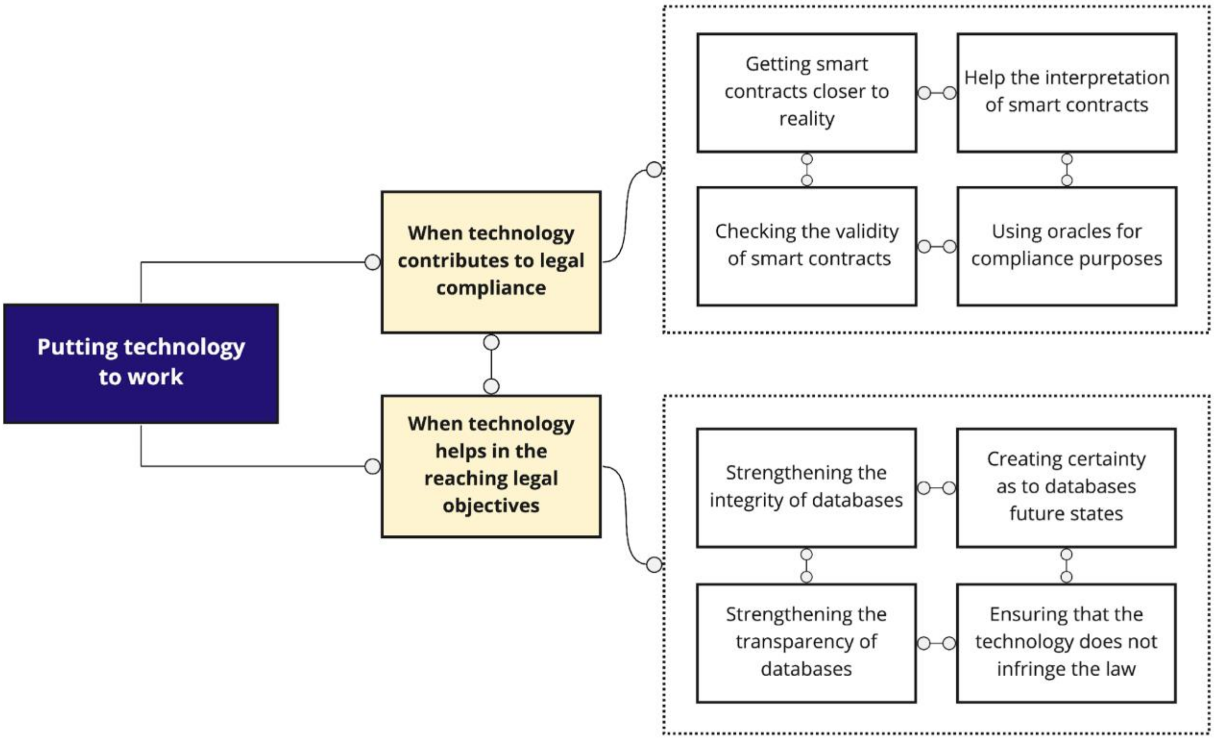
A second set of solutions consists in impacting immutability.[182] One can do so using chameleon hash functions.[183] Here, trusted users (e.g., blockchain users or institutional actors) have access to a trapdoor key that allows them to change the information while maintaining chain integrity.[184] This solution reintroduces centralization. Alternatively, mutable blockchains may record various versions of a transaction. Should the parties want to adopt another version, they will use meta-transactions to do so.[185] Third, another solution consists of using a data structure that allows blockchain users to retract information without changing the hash value of recorded blocks.[186] Fourth, smart contracts may feature suicide functions enabling the parties to kill them. As I discussed already, they are also called self-destruct or kill switch functions that can be designed in a way to allow specific third parties (e.g., decentralized courts, public agencies, etc.) to activate them.[187] They render smart contracts unresponsive, although the latter remains on the chain.[188] Fifth, a memory flexible blockchain may allow changes while preserving the integrity of hash values.[189] Sixth, one may use consensus-based voting mechanisms to change blockchain history depending on specific voting rules and procedures.[190] Users may also want to use modifiers, thanks to which they can change the behavior of a function.[191] Finally, nodes can flag data for erasure using specifically designed node software.[192]

A third set of solutions consists of creating a chain split in which users create a new version of the ledger history without pre-targeted information. Such split often results from hard forks, i.e., when new rules for validating blocks are implemented while rejecting the blocks complying with the original rules. Without a fork, chain splits are unlikely to be implemented. They indeed require the agreement of a majority of blockchain users that would mine blocks on a new

181 Nakamoto, *supra* note 4; GitHub, *Bitcoin Core version 0.11.0 Release*, GITHUB (July 10, 2015) https://github.com/bitcoin/bitcoin/blob/v0.11.0/doc/release-notes.md/#block-file-pruning [https://perma.cc/8BJM-VZHN].
182 Generally, *see* Giuseppe Ateniese et al., *Redactable Blockchain-Or-Rewriting History In Bitcoin And Friends In Security And Privacy*, 1 IEEE EUR. SYMP. ON SEC. AND PR. 111 (2017).
183 Politou et al., *supra* note 58 ("A chameleon hash is a cryptographic hash function that contains a trapdoor, and the knowledge of this trapdoor allows collisions to be generated efficiently"). Kondapally Ashritha, *Redactable Blockchain using Enhanced Chameleon Hash Function*, 5 INT. CONF. ON ADV. COMP. & COM. S. 323 (2019)*;* Ke Huang, *Achieving Intelligent Trust-Layer for IoT via Self-Redactable Blockchain*, 16 IEEE TRAN. ON IND. INF. 2677 (2019); Ateniese et al., *supra* note 185, at 111.
184 Jan Camenisch et al., *Chameleon-Hashes with Ephemeral Trapdoors*, 20 IACR INT. CONF. ON PR. AND T. IN P.-KEY CRYPT. 152-182 (2017).
185 Politou, *supra* note 58, at 10.
186 Richard D. Kuhn, *A Data Structure for Integrity Protection with Erasure Capability (Draft*) (2018)
187 Kill switch functions can be designed so that specific third parties can activate them, such as decentralized courts or public agencies.
188 Massimo Bartoletti and Livio Pompianu, *An Empirical Analysis of Smart Contracts: Platforms, Applications, And Design Patterns*, 1 INT. CONF. ON FIN. CRYPT. AND D. SEC*.* 494-509 (2017). Also, Marino & Juels, *supra* note 32, at 158.
189 Ali Dorri et al., *MOF-BC: A Memory Optimized and Flexible Blockchain for Large Scale Networks*, 92 F. GEN. COMP. S. 357 (2019).
190 Ivan Puddu & Alexandra Dmitrienko, *µchain: How To Forget Without Hard Forks*,1 IACR CRYPT. 106 (2017); Dominic Deuber et al., *Redactable Blockchain in the Permissionless Setting,* IEEE EUR. SYMP. ON SEC. AND PRI. 124 (2019).
191 Marino & Juels, *supra* note 32, at 163.
192 Martin Florian et al., *Erasing Data from Blockchain Nodes*, 1 IEEE EUR. SYMP. ON SEC. AND PRI. 368 (2019).

version of the ledger. Notably, creating a split to remove one piece of data is unlikely to get the majority approval as it endangers the trust users have in that blockchain.[193]

4.2.2. Putting technology to work



© Thibault Schrepel (2021)

4.2.2.1. When technology contributes to legal compliance

Technology can serve the law by helping legal enforcement against seemingly hermetic ecosystems such as blockchain. AI and oracles are useful in that regard.[194]

→ Putting AI to work

When Nick Szabo introduced the concept of smart contract, he stressed that "[n]o use of artificial intelligence is implied."[195] This does not have to be true anymore. Smart contracts follow the logic of expert systems made of "if/then" rules, which the European Commission

[193] Such approval was obtained in the case of TheDAO hack, *see* Matt Hussey & Adriana Hamacher, *What Is a Decentralized Autonomous Organization (DAO)?*, DECRYPT (May 21, 2021) https://decrypt.co/resources/decentralized-autonomous-organization-dao [https://perma.cc/9MH7-KPPE].
[194] On top of helping with smart contracts, AI systems will also contribute to the entire blockchain ecosystem. Some will ensure blockchain security by detecting intrusions in the ledger, on the subject, *see* Phillip Sandner et al., *Convergence of Blockchain, IoT, and AI*, 3 FRONT. BLOCKCHAIN 1, 2 (2020). Others will improve the performance of hash functions, help with hardware design, and *see* Bo Xing & Tshilidzi Marwala, *The Synergy of Blockchain and Artificial Intelligence* 6 (2018).
[195] Szabo, *supra* note 27.

puts under the heading of artificial intelligence.[196] But going one step further, smart contracts could embed machine learning instead of simple "if/then" rules.[197] They will undoubtedly do so in the future. This could get them closer to reality, for example, by allowing smart contracts to be called only when complex real-world events (e.g., events that humans we cannot identify on their own) have been verified using unsupervised machine learning.

In addition, AI systems will help verify the validity of smart contracts, whether simply technical or even legal. Some will test smart contracts, assisting with formal verifications and search-based software engineering.[198] Others will help to analyze the content of smart contracts and ensure their compliance with legal rules.[199] Natural language processing systems will help in the matter, while machine learning systems will compare their content with valid contracts.

AI systems will also help with interpreting smart contracts. They will supplement the experts capable of translating the code of smart contracts into natural language. These experts will be in high demand and often out of reach for certain parties (e.g., consumers). This will prove not only problematic in private enforcement (should one party be unable to translate the smart contract) but also in the absence of litigation. Article 6 of Directive 2011/83/EU on consumer rights subjects traders (i.e., professionals) to various information requirements for distance and off-premises contracts.[200] Should they not provide this information "in a clear and comprehensible manner," they face heavy penalties. This threat creates a strong incentive to provide consumers with a natural translation of smart contracts.

AI systems will assist traders and consumers in the matter. Several APIs, such as GPT-3 or NaturalyCode, translate natural language into code and vice versa.[201] Here is an example of a basic smart contract automating an auction:[202]

```
pragma solidity ^0.4.17;
contract Auction {
      address public highestBidder;
      uint highestBid;
      mapping(address => uint) refunds;

      function bid() public payable {
            require(msg.value >= highestBid);
            if (highestBidder != 0) {
                  // record the underlying bid to be refund
                  refunds[highestBidder] += highestBid;
            }
```

[196] Artificial Intelligence Act Proposal, *supra* note 149.
[197] Riccardo de Caria, *The Legal Meaning of Smart Contracts*, 1 EUR. REV OF PRIVATE L. 731, 737 (2018).
[198] Xing & Marwala, *supra* note 194, at 4.
[199] Joanna Hardy, *Continued EY Investments in Blockchain Market to Support Increased Demand*, EY: Building a Better Working World (May 17, 2021) https://www.ey.com/en_gl/news/2021/05/continued-ey-investments-in-blockchain-market-to-support-increased-demand [https://perma.cc/E5VV-LWH4] (discussing the smart contract "Testing Studio").
[200] Directive 2011/83/EU, *supra* note 104.
[201] *See* AIM, *OpenAI's GPT-3 Can Now Generate The Code For You*, AIM (Jul. 20, 2020) https://analyticsindiamag.com/open-ai-gpt-3-code-generator-app-building/ [https://perma.cc/2AGU-ZMWY].
[202] Maximilian Wöhrer & Uwe Zdun, *Design Patterns for Smart Contracts in the Ethereum Ecosystem*, 1 IEEE CONF. ON BLOCK. 1515 (2018).

```
            highestBidder = msg.sender;
            highestBid = msg.value;
      }

      function withdrawRefund() public {
            uint refund = refunds[msg.sender];
            refunds[msg.sender] = 0;
            msg.sender.transfer(refund);
      }
}
```

Now, here is the translation provided by GPT-3:[203]

```
"The contract is simple. It has a single function, bid(), which accepts
a bid in ether and returns the amount of ether that the caller is
willing to pay for the item. The function also has a withdrawRefund()
function that allows the caller to withdraw the amount of ether that
was paid for the item.

The contract has a single address, highestBidder, which is the address
of the highest bidder. The highestBid variable is a uint variable that
is initialized to 0. The highestBidder variable is a mapping from
address to uint. The mapping is initialized to an empty mapping. The
mapping is used to store the highest bid that has been made.

The function bid() is payable. This means that the caller can call the
function and pay ether to the contract. The function also has a
require() function that checks if the caller has enough ether to pay
for the item. If the caller does not have enough ether, the function
returns a failure.

The function bid() also has a function that allows the caller to
withdraw the amount of ether that was paid for the item. The function
withdrawRefund() is also payable. This means that the caller can call
the function and pay ether to the contract. The function also has a
require() function that checks if the caller has enough ether to pay
for the item. If the caller does not have enough ether, the function
returns a failure."
```

This translation is not very precise.[204] Therefore, one may want to change GPT-3 “temperature” (e.g., how random is the output) to obtain a different result when running it again. Doing so

[203] Perhaps the lack of proper understanding is best illustrated by the first sentence of the translation: “the contract is simple”.

[204] NaturalyCode provides the following translation: “This Python code interacts with an Ethereum contract called Auction. Auction is a contract that has a mapping of addresses to the amount of money that is owed in refunds. The function of Auction is to allow bids. The bidder can bid as many times as they want, but the bids need to be greater than the previous bid. The refund is recorded to the address of the person who made the bid.” NaturalyCode does not translate Solidity. Therefore, I used the Python translator (as the two languages have common roots). Similarly, when giving the following Serpent code:

could help increase the comprehension of the smart contract. Here is another GPT-3 translation of the same smart contract:

```
"The contract is very simple, it has two functions: bid and withdrawRefund.

The bid function is called by a single party that wants to place a bid. This function verifies that the minimum deposit is met and that the bid is greater than the highest bid already in the contract. If both conditions are met, the function adds the bid as the highest bid.

The withdrawRefund function is called by the owner of the contract to refund a previously made bid. This function first verifies that the deposit is still available to be refunded. If not, the refund is denied. The function then verifies that the bid does not exceed the highest bid ever made in the contract. If that is true, the bid is removed from the highest bid and the refund is made to the caller."
```

GPT-3 (and other APIs) needs to improve. One should not expect these systems to become highly accurate any time soon. That said, despite not being standalone solutions, they can help. Future versions of these models will continue to progress and will play a role in interpreting smart contracts. Today, state-of-the-art deep-learning models reach around 90% common-sense reasoning accuracy, while humans reach 94%.[205] These models are quickly closing the gap.

Proposal #17: Support the technical solutions translating computer code into natural languages and vice versa.

→ Putting oracles to work

Oracles can play a role in helping legal enforcement. One could indeed use oracles to send the clauses of smart contracts to legal experts. One could also push these clauses into legal analysis systems stored outside the blockchain and using natural language processing (e.g., they could flag consumer contracts that seem to violate consumer protection legislation). Outbound oracles could then transmit the result coming out of this expertise or AI systems back to the blockchain to block illegal smart contracts.

"from = msg.sender to = msg.data[0] value = msg.data[1]
if contract.storage[from] >= value:
contract.storage[from] = contract.storage[from] value contract.storage[to] = contract.storage[to] + value", NaturalyCode returns "This Python code checks to see if the sender of the message has enough tokens to pay for the transaction in front. If they do, it updates the sender's balance and the recipient's balance.".

205 Commonsense Reasoning, *Winograd Schema Challenge*, COMMONSENSE REASONING (Jul. 26, 2021) http://commonsensereasoning.org/winograd.html [https://perma.cc/NDM9-YPZH]. *See* Jimmy Chain, *AI Still Doesn't Have the Common Sense to Understand Human Language*, MIT TECH. REV. (Jan. 31, 2020) https://www.technologyreview.com/2020/01/31/304844/ai-common-sense-reads-human-language-ai2/ [https://perma.cc/4R2U-Y5LW].

Of course, building such systems will require the creation of large databases. The task will not be easy, but it will contribute to the integrity of the DSM by preventing the circulation of smart contracts whose object or conditions infringe the law.

Proposal #18: Undertake the creation of AI systems functioning under human supervision to verify the legal validity of smart contracts and ensure technical compatibility with outbound oracles.

Finally, smart contracts could call legal provisions stored outside the blockchain (e.g., a database run by the European Union) using APIs as oracles. Trusted parties could update this database. It would align smart contracts with current legislation by pulling information from the database and making it a clause of the smart contract.[206] Doing so would require converting legal obligations into code.[207] Using these clauses would create a risk of centralizing smart contracts, but it would also help ensure the integrity of the DSM.

Proposal #19: Undertake the codification of contract law in computer language, starting with the main principles recognized throughout Europe.

4.2.2.2. When technology helps legal objectives

Technology can help achieve some goals set by regulators and policy-makers.[208] This observation applies to blockchain and smart contracts. I do not wish to say they are standalone solutions, but I do say that regulators should use them. I want to drive the point home by discussing the "requirements of high quality data, documentation and traceability, transparency, human oversight, accuracy and robustness" stressed by the European Commission for high-risk AI systems.[209]

Numerous studies document the importance of using error-free databases for training AI systems. When they are not, AI reproduces these errors and amplifies them. Unfortunately, errors are commonplace. The ten most used databases in computer vision, natural language, and audio datasets include about 3% of erroneous data (i.e., information that has been mislabeled).[210] Further, databases should be complete and diverse, or else machine learning will malfunction for populations outside the training set of data.[211]

[206] Raskin, *supra* note 30, at 327.
[207] This process would reveal inconsistencies in contract law and will prove difficult where the terms and concepts are open. That said, although *all* the rules of contract law could not be made computable, I am of the opinion that several of the most basic and important ones could and should be translated into computer code. In fact, there is a rich history behind making programming languages for law. For recent examples, *see* Denis Merigoux et al., *Catala: A Programming Language for The Law*, 1 INT. CONF. ON FUNC. PROG. 1 (2021) and Accord Project, *Ergo*, ACCORD PROJECT (Jul. 27, 2021) https://accordproject.org/projects/ergo/. [https://perma.cc/NV5C-UURH]. A history of why earlier languages did not take off would be interesting to document.
[208] Schrepel, *supra* note 15.
[209] Artificial Intelligence Act Proposal, *supra* note 149, at 7.
[210] Curtis G. Northcutt et al., *Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks*, 1 ICLR ROBUSTML AND W. SUP. LEAR. WORK. 1 (2021) ("Errors in test sets are numerous and widespread: we estimate an average of 3.4% errors across the 10 datasets"').
[211] Joy Buolamwini & Timnit Gebru, *Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification*, 81 PROC. OF MACH. LEAR. RES. 77 (2018).

Blockchain and smart contracts can provide technical support, not in providing errorless databases, but in increasing the ability to prevent hacks and audit them.

To begin with, public permissionless blockchains are immutable infrastructures. This immutability proves central in terms of cybersecurity. It means that blockchains have no single point of failure, i.e., hacking a single copy of the ledger has no impact on the other copies. In addition, blockchain visibility ensures the transparency of the data used in AI systems.[212] Such accessibility will prove helpful for auditing the databases used to train and operate AI. Moreover, one cannot modify blockchain ledgers without leaving a trace. No single user will thus be able to delete data, e.g., as one may be tempted to do in case of AI malfunction to hide the roots of the problem.

Smart contracts have a complementary utility. Their immutability, as challenging as it may be for legal enforcement, also creates certainty. This certainty is a significant asset. In effect, it ensures the flow of information such as it has been established, which serves three purposes. First, it creates trust in automating the transfer of data when specific conditions are met. Second, it ensures the traceability of the transfer. Third, it prevents the hack of that transfer since smart contracts are equally censorship-resistant as the blockchain that supports them.
In a nutshell, blockchain and smart contracts will reinforce the integrity of the databases used to train AI systems. Despite the difficulties associated with the immutability of blockchains and smart contracts, this feature is a natural ally for institutions seeking to create or encourage transparent and auditable data infrastructures that support AI systems.

Proposal #20: Promote the use of blockchains and smart contracts in the European Data Strategy to create and maintain public databases supporting AI systems.

[212] Schrepel, *supra* note 34, at 117 (2020) (speaking of "visibility effect").

5. Conclusion

In this conclusion, I first summarize the proposals I have made throughout the report (**4.1**) before offering some final thoughts (**4.2**).

5.1. Summary of the proposals

Summary of all the proposals	
1. <u>Substantial proposals</u>	
Proposal # (in order of appearance)	**Description**
#1	Harmonize disclosure rules in a way that smart contracts are not disadvantaged compared to other transactional means.
#2	Recognize the deployment of smart contracts on a blockchain as a legally binding offer.
#3	Recognize the calling of smart contracts as proof that consents have been exchanged.
#4	Recognize the automated exchange of consents as contractually valid.
#5	Harmonize the recognition of blockchain as a durable medium for storing contracts and other legal documents.
#6	Harmonize the interpretation of the "written form" requirement to include code.
#7	Harmonize the rules regarding the type of electronic signatures and related constraints for signing a smart contract.
#10	Use soft law to clarify the criteria for a smart contract to be cross-border.
#12	Allow exceptions to data sovereignty regulations for the data used and generated by smart contracts.
2. <u>Actions to be undertaken</u>	
Proposal # (in order of appearance)	**Description**
#8	Set up data units within European and national courts or regulatory agencies with technical expertise in blockchain and smart contracts.
#9	Support or develop education programs dedicated to technology literacy (i.e., to teach the basics of programming languages and computational methods) for judges and regulatory agencies.
#11	Compare technical solutions allowing smart contacts' compliance with GDPR and, if necessary, introduce exceptions to it.

#13	Redraft Annex I of the AI Act by adding the qualifier "centralized" to expert systems to exclude smart contracts: "Logic- and knowledge-based approaches, including knowledge representation, inductive (logic) programming, knowledge bases, inference and deductive engines, (symbolic) reasoning; and centralized expert systems;"
#14	Should the European Commission undertake the design of templates, it is recommended to make them voluntary while limiting them to smart contracts and oracles.
#15	Set up experiments to preserve the integrity of smart contracts while enabling legal enforcement through an ex-post "law is code" approach.
#16	Create comfort zones to test the effectiveness of different regulatory mechanisms addressing smart contracts' most pressing issues.
#17	Support the technical solutions translating computer code into natural languages and vice versa.
#18	Undertake the creation of AI systems functioning under human supervision to verify the legal validity of smart contracts and ensure technical compatibility with outbound oracles.
#19	Undertake the codification of contract law in computer language, starting with the main principles recognized throughout Europe.
#20	Promote the use of blockchains and smart contracts in the European Data Strategy to create and maintain public databases supporting AI systems.

5.2. Final thoughts

The interaction between law and smart contracts is relatively new. Many questions arise, leading to issues that might threaten the cohesion of the DSM and therefore deserve a firm response. That being said, smart contracts open up new perspectives that benefit the European citizen and economy. In this context, legal rules and standards must not compromise smart contracts' chances of survival. Moreover, the EU legal framework has a role to play in stimulating their dynamism and helping smart contracts evolve in a way that strengthens the DSM.

The "law + technology" approach is fit for purpose. It seeks to combine law and technology to reinforce each other rather than create a confrontation that harms the common good. In the end, there is much to be done, and perhaps we should begin by anchoring the mutual benefits that law and technology can yield when they cooperate. Smart contracts illustrate well the merits of such cooperation.

GETTING IN TOUCH WITH THE EU

In person

All over the European Union there are hundreds of Europe Direct information centres. You can find the address of the centre nearest you at: https://europa.eu/european-union/contact_en

On the phone or by email

Europe Direct is a service that answers your questions about the European Union. You can contact this service:

– by freephone: 00 800 6 7 8 9 10 11 (certain operators may charge for these calls),

– at the following standard number: +32 22999696 or

– by email via: https://europa.eu/european-union/contact_en

FINDING INFORMATION ABOUT THE EU

Online

Information about the European Union in all the official languages of the EU is available on the Europa website at: https://europa.eu/european-union/index_en

EU publications

You can download or order free and priced EU publications at: https://publications.europa.eu/en/publications. Multiple copies of free publications may be obtained by contacting Europe Direct or your local information centre (see https://europa.eu/european-union/contact_en).

EU law and related documents

For access to legal information from the EU, including all EU law since 1952 in all the official language versions, go to EUR-Lex at: http://eur-lex.europa.eu

Open data from the EU

The EU Open Data Portal (http://data.europa.eu/euodp/en) provides access to datasets from the EU. Data can be downloaded and reused for free, for both commercial and non-commercial purposes.

KK-09-21-375-EN-N

Publications Office
of the European Union

doi: 10.2759/562748
ISBN: 978-92-76-41173-4