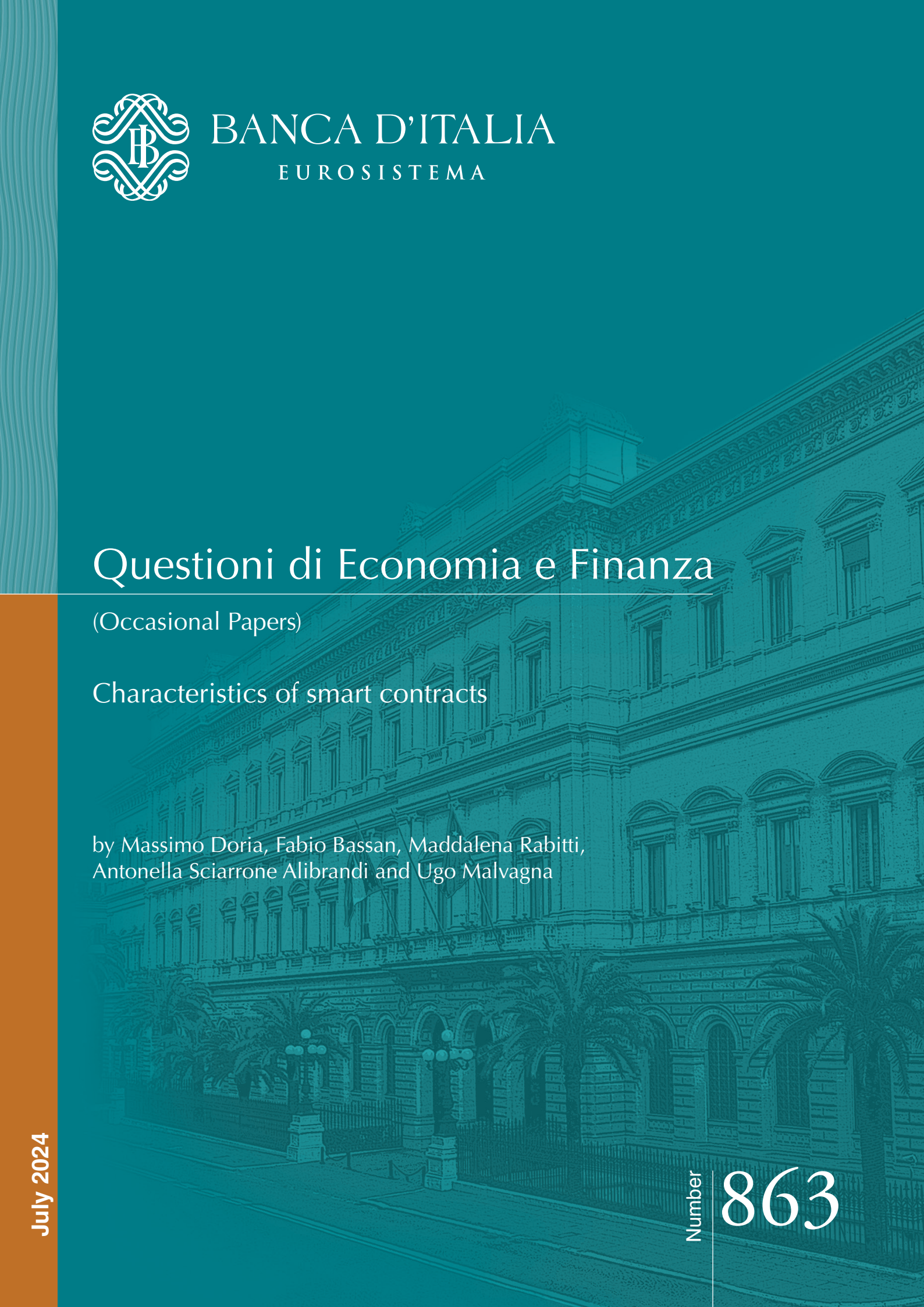
BANCA D'ITALIA
EUROSISTEMA

Questioni di Economia e Finanza

(Occasional Papers)

Characteristics of smart contracts

by Massimo Doria, Fabio Bassan, Maddalena Rabitti,
Antonella Sciarrone Alibrandi and Ugo Malvagna



July 2024

Number 863

BANCA D'ITALIA
EUROSISTEMA

Questioni di Economia e Finanza

(Occasional Papers)

Characteristics of smart contracts

by Massimo Doria, Fabio Bassan, Maddalena Rabitti, Antonella Sciarrone Alibrandi and Ugo Malvagna

Number 863 – July 2024

The series Occasional Papers *presents studies and documents on issues pertaining to the institutional tasks of the Bank of Italy and the Eurosystem. The* Occasional Papers *appear alongside the* Working Papers *series which are specifically aimed at providing original contributions to economic research.*

The Occasional Papers *include studies conducted within the Bank of Italy, sometimes in cooperation with the Eurosystem or other institutions. The views expressed in the studies are those of the authors and do not involve the responsibility of the institutions to which they belong.*

The series is available online at www.bancaditalia.it .

ISSN 1972-6643 (online)

Designed by the Printing and Publishing Division of the Bank of Italy

CHARACTERISTICS OF SMART CONTRACTS

by Massimo Doria (coordinator)*, Fabio Bassan**, Maddalena Rabitti**,
Antonella Sciarrone Alibrandi*** and Ugo Malvagna***

Abstract

The Bank of Italy, Università Cattolica del Sacro Cuore of Milan and Roma Tre University have launched a research project on the characteristics of smart contracts for the provision of banking, financial and insurance services. This paper aims to identify and analyse the legal and technical issues related to smart contracts, to then develop guidelines derived from best practices. It examines the concept of smart contract defined as: a) smart contract code, i.e. a software program stored, verified, and executed on a blockchain; b) smart legal contract, i.e. a means of articulating, verifying, and enforcing an agreement between parties that relies on blockchain technology. Additionally it offers an in-depth analysis of the characteristics of blockchain technology.

JEL Classification: O33.

Keywords: smart contracts, DLT/blockchain, governance, tokenomics.

DOI: 10.32057/0.QEF.2024.863EN

* Banca d'Italia.

** Università degli Studi Roma Tre.

*** Università Cattolica del Sacro Cuore di Milano.

Characteristics of Smart Contracts

EXECUTIVE SUMMARY[1]

As part of a Memorandum of Understanding, Banca d'Italia, the Università Cattolica del Sacro Cuore and Università Roma Tre have started a research project on the characteristics of smart contracts for the provision of banking, financial and insurance services.[2]

This topic is of great importance and relevance, in the light of both the diffusion of DLT technologies in this context and the impetus that European legislation is giving to the development of experimental activities (as in the case of the DLT Pilot Regime) or to the profiling of new products (such as crypto-assets). To date, smart contracts represent the most well known application of distributed ledger technology, together with crypto-assets and the tokenisation of assets (i.e. representation of real assets in the form of digital tokens issued on the blockchain, which represent their intrinsic economic value and ownership rights).

A working group was set up, composed of representatives of the signatories, which identified two phases for the development of the project:

a) a survey of the main characteristics of blockchains and smart contracts;

b) the definition, in a best practice view and addressed to market participants, of the characteristics that smart contracts should possess in order to be used in the provision of banking, financial or insurance services.

This document illustrates the results of the first phase of this project by exploring the characteristics of smart contracts, subject to the two definitions of smart contract code, a software program that is stored and executed on a blockchain[3] or, more generally, on a distributed ledger technology (DLT),[4]

---

[1] *For Banca d'Italia, the contributors to the working group set up for this project and coordinated by M. Doria are: C. Iacomini, S. Guida, G. Goretti, A. Lentini, M. Suardi for the Currency Circulation and Retail Payments Directorate General; G. Marcelli and F. Rossi for the Information Technology Directorate General; L. Anchora, R. Mancini and G. Falcone for the Markets and Payment Systems Directorate General; M. Bevilacqua, V. Cappa and C. Lanfranchi for the Financial Supervision and Regulation Directorate General; N. Branzoli for the Economics, Statistics and Research Directorate General; M. L. Cartechini, M. Argirò, R. Loffredo, F. Squartini, C. Tabarrini and G. Tucci for the Consumer Protection and Financial Education Directorate General; V. Profeta and G. Pala for the Legal Services Directorate; L. La Rocca and M. Militello for the Financial Intelligence Unit for Italy; C. Galasso for the Anti-Money Laundering Supervision and Regulation Unit; R. Gabbiadini and E. Rubera for the Milan office and B. De Luca and S. Castrovinci Zenna for the Institute for the Supervision of Insurance (IVASS).*

For Università Roma Tre, Prof. Fabio Bassan, Prof. Maddalena Rabitti, Dr Michela Mastrantonio, Dr Michele Mastrantonio and Dr Stefano De Angelis.

For the Università Cattolica del Sacro Cuore, Prof. Antonella Sciarrone Alibrandi, Prof. Ugo Malvagna, Dr Giulia Schneider,and Dr Federico Panisi.

The document provides an account of the work carried out from December 2022 to December 2023.

[2] The initiative was launched in line with Banca d'Italia's Communication of 15 June 2022 on decentralised technologies in finance and crypto-assets, which drew attention to the opportunities and risks associated with their use [1].

[3] Blockchain is a distributed ledger technology (DLT). Transactions performed on the blockchain are stored in the ledger grouped in a 'block' data structure. These blocks are connected to each other by cryptography, through hash functions. In this way, a record is generated in a chronological and non-editable order of all transactions made up to that time [1]. This data structure ensures the immutability and traceability of transactions, as modifying a single element of the register would require the cascading modification of subsequent ones. Please refer to Part II of this document for an in-depth analysis.

[4] Banca d'Italia [1, 2, 3] addressed the issue on the basis of the consideration that 'The development of decentralised technologies in the field of financial services rests on the central role of cryptography and distributed ledger technology (DLT/blockchain). The two technological paradigms are highly complementary. The first allows you to protect transaction information and its non-repudiation; it ensures the integrity and, where applicable, confidentiality of the same information and underpins the transaction authorisation mechanism. The second (DLT/blockchain) consists of a shared electronic register in which data is protected both by cryptographic techniques and by "redundancy" (copies of the same information can be validated and stored by all active participants in the register)' [1].

and smart legal contract, a smart contract code when used to articulate, verify and apply an agreement between the parties. The work highlights the issues that smart contracts pose on both legal and technical levels, with the goal of producing, in a second phase, a set of guidelines that can be derived from best practices.

In summary, it is argued that in the development of applications based on decentralized technologies, the specificities of the adopted blockchain must be taken into account, not limited to technological characteristics but also considering governance profiles, tokenomics and environmental profiles. Finally, it should be noted that smart contracts introduce specific risk profiles in terms of cybersecurity.

The document suggests a methodological approach for the acquisition of the information necessary to evaluate the most appropriate platforms to be used in developing a specific application. In introducing the technical peculiarities of smart contracts, the document adheres to a high-level treatment in order to avoid any undue influence on the second phase of the project, which is intended to provide the definition of certain guidelines.

In detail, the document is structured in two parts, each divided into Sections.

Firstly (PART I), we illustrate the main legal aspects addressed by Italian and foreign literature in relation to the use of smart contracts, qualified as *smart contract codes* and *smart legal contracts*. In this work, the distinction is relevant in order to identify the functions characterizing smart contracts. With reference to the smart contract code, the work focuses on the technical characteristics of both the code-language used and the blockchain on which it operates. With regard to the smart legal contract, on the other hand, the document focuses on the suitability of the smart contract to perform its function in such a way as to be solid and to protect the parties involved from a legal point of view.

On the basis of existing regulatory sources, within the European Union and at national level (Section I), we set out the analyses carried out in the legal doctrine (Section II) and the solutions offered. Some extra-European references are then proposed when they are deemed useful to understand certain open issues (Box 1).

A further analysis of smart contracts in terms of the technology (PART II) completes the representation of the issues raised by the doctrine to identify possible solutions through a specific configuration of smart contracts or smart legal contracts, taking into account, where appropriate, the relevant characteristics of the blockchain technology on which they insist.

The technical analysis is therefore divided into two sections. The first deals with blockchains, while the second with smart contracts.

Section I analyses the state of art and goes on to list and describe the main characteristics of blockchain technology divided into classes: technical characteristics,[5] economic model,[6] ecosystem[7]

[5] Among the technical characteristics are considered: architectural aspects of blockchain networks; security parameters, such as confidentiality, integrity, availability, consistency (in the two meanings referring to the finality and fork blockchain) and finally quantum resistance; efficiency parameters, such as scalability and decentralisation (number of nodes, distribution of validation power and fairness); parameters from the point of view of application and usability: flexibility as programmability; system configurability and interoperability as available techniques to facilitate interaction between different systems; energy consumption and environmental impacts.

[6] Among the parameters of the economic model taken into account were: transaction costs; native token distribution techniques (to ensure the proper functioning of permissionless blockchains, it is necessary to use a native token, which allows you to define financial incentives for validators to act honestly and keep the network functioning); capitalisation (meaning the total market value of all native tokens that have been put into circulation, valued at the market price at a given time. This parameter can play a role in the analysis of the robustness and security of a blockchain, as it can have an impact on the economic effort required to control the network).

[7] The governance of a blockchain platform is another useful parameter for assessing its sustainability and durability. In fact, blockchain systems, being decentralized, cannot have central authorities making governance decisions regarding, for

and on-chain data.[8] In the document the classes are the analysed more deeply, motivating the choices made for the selection of quality parameters. In conclusion, a methodological approach is proposed that could be used for the acquisition of the information needed to describe and analyse the different blockchains in relation to the identified characteristics. The proposed approach aims to identify a taxonomy of parameters to answer the following question: 'What are the main parameters or requirements that should be taken into account when analysing the characteristics of a blockchain platform?' Identifying a set of main parameters makes it possible to analyse blockchain platforms and could help develop the regulation of this technology.

The first part of Section II analyses the main components of smart contracts and presents account-based and token-based status models, discussing their peculiarities and differences. In the context of account-based models, the life cycle of smart contracts and known methodologies for model updating and governance are also described. Subsequently, a taxonomy of the fundamental characteristics of smart contracts is proposed, highlighting the tradeoffs between stateful and stateless execution environments and between complete and non-complete Turing programming languages. The section concludes with the high-level features, focusing the study on the costs of production and execution of smart contracts. In the second part of the Section, we propose an in-depth analysis on security, identifying the challenges that developers face for the creation of secure and reliable decentralized applications and a classification of the possible vulnerabilities that can affect smart contracts.

The aim for the second stage of the project is to develop useful guidelines for blockchains (and the technology used by them) and for smart contracts. The topic is of great importance: so much so that both European authorities and international institutes have recently expressed their interest by publishing works[9] that will be examined and taken into consideration in the second stage of our analysis.

PART I - SMART CONTRACTS — TECHNOLOGY AND LEGISLATION

Introduction

To date, smart contracts represent the most well-known application of a distributed ledger technology, together with crypto-assets and the tokenisation of assets, defined as the representation of real assets in the form of digital tokens issued on the blockchain, which represent their intrinsic economic value and ownership rights [4, 5].

The first to theorize the design of smart contracts, even before the advent of blockchain technology, was the American computer scientist and cryptographer Nick Szabo[10] in the early 1990s, with the aim

example, protocol updates or the management of undistributed native tokens; it is therefore necessary to identify which governance mechanisms can be adopted depending on the context, to allow the actors involved to be able to participate in the decision-making process.

[8] Another useful parameter for the characterization of a blockchain could be the actual use of the platform; to evaluate the use of a platform, the volume of transactions carried out on the chain could be considered.

[9] Reference is made in particular to the principles developed by the International Institute for the Unification of Private Law (Unidroit) 'PRINCIPLES ON DIGITAL ASSETS AND PRIVATE LAW' and to an article by the European Securities and Markets Authority (ESMA) entitled 'Decentralised Finance: A categorisation of smart contracts'.

[10] The definition provided by Nick Szabo is the following: 'A smart contract is a computerised transaction protocol that executes the terms of a contract. The general objectives of smart contract design are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries. Related economic goals include lowering fraud loss, arbitration and enforcement costs, and other transaction costs'. Szabo had also described a decentralized system of generation and

to: meet common contractual conditions (e.g.: terms of payment, rights, confidentiality) and minimise the risk of default and limit the use of trusted intermediaries or traditional enforcement mechanisms [6, 7, 8, 9, 10]. According to this original scheme, smart contracts make possible to reduce, or even eliminate, any associated costs [114].

This idea was put into practice for the first time in the distributed ledger system [11]. This has allowed developers to interact with smart contracts in order to build applications that run in a decentralized way directly on blockchain technology.

In this work, we focus on smart contracts as programs on blockchain platforms. In addition, the analysis focuses on the forms of smart contracts used in the insurance, banking and financial sectors, which have specific needs and particular characteristics, imposed (also) by sector-specific legislation. This document aims to illustrate the state of the art, in doctrine and in practice. It will be followed by another research aimed at proposing guidelines for the use of smart contracts in the financial, banking and insurance markets.

1. 'Smart contract code' and 'Smart legal contract'

The term 'smart contract', now predominantly associated with blockchain platforms,[11] does not have a clear definition [12, 13, 14]. Some refer to the concept of 'autonomous machines', others to 'contracts between parties stored on a blockchain', others still associate the smart contract more generally with 'any calculation that takes place on a blockchain'. The defining attempts can be traced mainly to two macro-categories [15, 16, 17, 18, 19] that we use here for the sake of brevity and clarity, aware of the fact that it is not the only possible taxonomy and that it is not shared by all: a) *smart contract code,* to identify a specific technology or code that is stored, verified and executed on a blockchain; and (b) *smart legal contract*, as a specific application of this technology as a complement to, or substitute for, traditional contracts.

In this work, the distinction is relevant in order to identify the functions of a smart contract [20, 21]. **With reference to the smart contract code, it focuses on the technical characteristics of both the code-language used and the blockchain on which it operates.**

With regard to the smart legal contract, on the other hand, the work focuses on the suitability of the smart contract to perform its function in a solid manner and to protect the parties from a legal point of view.

1.1 Smart contracts as 'Smart contract code'

The concept of 'smart contract code'[12] is commonly used by developers working on blockchain technology.[13] Smart contracts have unique features compared to other types of software because: (i)

exchange of digital currency called 'Bit gold', a precursor to today's and more famous Bitcoin. He advocated the use of the smart contract in other areas, such as the purchase of an asset in instalments, for example a car, assuming a system whereby the buyer's default results in an automatic locking of the vehicle through the interaction of software and hardware capable of recognising the fulfilment of a pre-established condition (e.g. the non-payment or late payment of the period installment), without further human intervention being necessary or possible in order to achieve the relevant consequences.

[11] Almost 20 years later, Vitalik Buterin in the Ethereum White Paper (Published online on 6 April 2014: V. Buterin, A Next-Generation Smart Contract and Decentralized Application Platform) describes smart contracts from a technical-computer point of view.

[12] The reference to 'Smart contract code' was initially used in Ethereum documentation, on stackexchange and in technical articles. Today, however, the term is used generically to refer to any complex program that is stored and executed on a blockchain.

[13] While the first blockchains were designed to perform a small set of simple operations – mainly, transactions of a token similar to a currency – techniques were then developed to perform more complex operations, defined in real programming languages.

the program code is registered on the blockchain and thus acquires the characteristics of immutability, security and transparency[14] guaranteed by the shared register; (ii) the execution of the programme is deterministic and the result of the execution is stored on the blockchain and (iii) the program automates the execution of certain transactions on the blockchain; for example it can act as a repository of digital assets (including crypto-assets), approve the transfer of assets, or save certain information in the register.

The smart contract code does not have the typical characteristics of a contract. Moreover, in many cases, smart contracts do not have an autonomous function, but are instrumental to the success of a broader application[15], executed on the blockchain − and therefore decentralized[16] [13].

1.2 Smart contracts as 'Smart legal contracts'

Among lawyers, the term 'smart contract' is often understood as a tool that insists on blockchain technology to articulate, verify and enforce an agreement between the parties, with the aim of complementing, or in some cases replacing, traditional contracts. This is known as a smart legal contract. It is ultimately a combination of programming code and legal language[17] [15].

Section I – Regulatory sources

2. European regulation of smart contracts

The EU legislator has intended to regulate smart contracts since the European Parliament asked the Commission 'to assess the development and use of distributed ledger technologies, including blockchains and, in particular, smart contracts' in 2020.[18] On that occasion, the European Parliament acknowledged the widespread use of smart contracts and the absence of an appropriate legal framework; it has, therefore, presented legislative proposals regarding their use, the possibility of intervention in transactions in the event of suspicious financial transactions, and protective measures for small and medium-sized enterprises that decide to use these instruments. The Resolution adopted by the European Parliament on this subject is part of the Blockchain Strategy[19] [22]: the European Union intends not only to regulate automated contracts but, recognising their innovative potential for online transactions, to support European companies and technologies active in this sector.

[14] The smart contract code is saved on the register shared by all participants in the network, so it is easily searchable and verifiable.

[15] The reference made is to DApps (Decentralized Applications), defined as applications that can operate autonomously, typically through the use of smart contracts, which are run on a decentralized computer system, a blockchain or another distributed ledger system. Each DApp or other blockchain-based application is built using smart contract code to perform operations on the chosen blockchain. What differentiates them, therefore, from most common applications is that their back-end code is running on a decentralized peer-to-peer network.

[16] The term 'smart contract' is debated in legal literature because it emphasises a single restricted use case. Smart contract programs can hold crypto-asset balances themselves or even control other smart contract programs. Once created, they can act autonomously when called upon to perform an action. For this reason, many prefer the term 'smart agent', analogous to the more general concept of software agent.

[17] Commercial contracts often contain clauses that protect the parties from various borderline cases and that do not always lend themselves to being represented and executed by code. Let's imagine that a supplier of goods enters into a smart contract with a retailer. Payment terms may be coded and executed automatically upon delivery. But the retailer will likely insist that the contract include an indemnity clause, whereby the supplier agrees to indemnify and hold the retailer harmless from claims for compensation arising from a defective product. It would not make sense to represent this clause in the code, since it is a clause that must not be self-executed, but interpreted and applied by the parties and, in the event of a dispute, by the competent court.

[18] European Parliament resolution of 20 October 2020 with recommendations to the Commission on the Digital Services Act: adapting commercial and civil law rules for commercial entities operating online (2020/2019(INL)).

[19] For more information, see the European Commission's website https://digital-strategy.ec.europa.eu/en/policies/blockchain-strategy

The European Commission has also set up the pilot project 'European Blockchain Observatory and Forum', managed by the Directorate-General for Communications Networks, Content and Technology (DG CONNECT), with the aim of: (i) accelerating the development of blockchain innovation in Europe; (ii) monitor European blockchain initiatives; (iii) propose recommendations on the role that the EU could play in the blockchain sector. Among the Reports produced by the Forum, there is also a specific one on smart contracts [23], which lists the benefits for the large-scale adoption of smart legal contracts compared to real contracts; at the same time, it highlights its limitations and proposes solutions.

In order to include a smart legal contract in the blockchain, the Report stresses the need for the legal language to be fully translated into computer code. This is a major issue, as lawyers usually do not have the technical skills of code developers, and vice versa. However, it is required a certain level of trust and expertise to ensure that all parties can trust that the smart contract code truly reflects the legal content and purpose [of the real agreement]' (p. 29). Among the critical issues, there is also the difficulty of assessing real-world events against which the Report identifies the oracle[20] as a tool to connect reliable data to the blockchain.[21]

In addition to the issues mentioned above, there are those relating to the risk of fraud, the protection and management of personal data, as well as cybersecurity and vulnerability to attacks. Compared to the latter, the Report proposes techniques to prevent operational risks, starting with preventive security auditing tools and penetration tests. With regard to the financial sector, the instruments identified in the DORA Regulation can also be useful, since it standardises and harmonises the rules on the security of network and information systems that underpin the business processes of financial entities. DORA's objectives include the achievement of high standards in the field of cyber security and governance of ICT and cyber risks, mandating financial institutions to define an organisational and procedural governance framework, integrated into the broader operational risk framework.[22]

---

[20] In the ELI Principles (European Law Institute) the oracle is described as a 'Service that updates a distributed ledger (eg a blockchain) using data from outside a distributed ledger system (outside the blockchain context). an oracle transmits off-chain information in a computer-readable form to the network' [5]. In other words, an oracle provides the smart contract with the information from outside the network, which is the basis on which the smart contract performs the desired functions. The execution assumptions can be acquired and verified directly on-chain if they are facts perceptible by the blockchain itself (as in the case of digital currency transactions between multiple wallets connected to the smart contract); in other cases, in fact, the elements or events that allow the activation of the smart contract are external to the network (off-chain), so respectively they exist and occur only in the real world: in these circumstances, the oracle aims to ensure the connection between what happens on the blockchain and what happens outside, 'certifying' the originality and correctness of the data entered on the blockchain. For this reason, some define oracles as a form of bridge. In this sense, see [94] which define oracles as 'sensors in the physical word'; [95] takes up the concept that 'Oracles act as a bridge that can digest external and non-deterministic information into a format that a blockchain can understand'; similarly, [96]. In particular, in cases of objective oracle, the input provided by the oracle comes from software or a computer tool, which postulates standard, simple and automatic transactions, dependent on the verification of an unquestionable objective fact (whether that fact A actually occurred, which results in event B). In the case of a subjective oracle, on the other hand, the information acquired by the oracle derives from a necessary preliminary assessment of human through his own judgment.

[21] The Smart Contracts Report (pp. 6 and 27) clarifies that 'The code contains the rules that define the conditions under which the smart contract has to act and how it has to behave. Their domain of action is internal to the blockchain that contains them. Although they can also receive information from external sources through the use of oracles [...] there are different limitations for smart contracts, which depend on the perspective that the aspect being researched. For example, the Ethereum blog provides two technical limitations existing in native smart contracts. The first one concerns the inability of smart contracts to evaluate real-world events. This fact should not come as a surprise since blockchains are generally independent and separate environments. The separation is a mechanism to guarantee the network's security on the basis of the consensus algorithm. As a result, data residing in databases ranging from employee information to weather data or football game results are data outside of the blockchain network that are considered potentially dangerous for the network. There are ways to mitigate this limitation, as oracles are a solution for bridging trusted data to the blockchain. In particular, Chainlink's blog points out blockchains' isolation and the potential for hybrid smart contracts to bridge this weakness by using oracles'.

[22] Specifically, DORA is based on five key pillars:

Characteristics of Smart Contracts

For the integration of large-scale smart legal contracts, the Report highlights the challenges posed to consumers by the language used, so it seems necessary to: (i) ensure the usability of the information and identify mechanisms to take into account the legal position of the consumer and (ii) apply strict KYC procedures and AML/CFT controls.

In addition, given the semi-irreversible nature of the data recorded on the blockchain, any error in the code can take time and high costs to correct. From a legal point of view, it is therefore necessary to inform the parties of the contractual meaning of specific legal concepts (e.g. good faith) and leave room for flexibility.

A) ICT Risk Management: In order to achieve a high level of digital operational resilience, financial institutions shall put in place an internal management and control framework that 'ensures effective and prudent management of all IT risks by entrusting the internal management body with the task of defining and approving the implementation of all provisions on the IT risk management framework, as well as overseeing and assuming full responsibility for their implementation.

B) ICT - Related Incident Management: Incident management is a process necessary to avoid or minimise economic and reputational impacts, due to a cyber incident, and thus be able to restore the normal provision of services as soon as possible. It is essentially based on the standardisation of ICT incident classification and reporting activities.

C) Digital Operational Resilience Testing: to ensure adequate ICT risk management, a digital operational resilience testing programme is one of the priorities of financial institutions. For the proper monitoring of the effectiveness of the resilience strategy, digital operational resilience tests must be conducted taking into account the evolution of cyber threats.

D) ICT third-party Risks Management: Third-party risk management in the ICT sector aims to provide all the requirements for financial institutions and ICT service providers to ensure robust monitoring of the risks associated with them. In this context, the ESAs will be responsible for: conducting off-site and on-site inspections, requesting information, issuing recommendations and requests, and imposing sanctions.

E) Information Sharing: One of DORA's objectives is to encourage the exchange of information on financial threats, through the establishment of a voluntary programme to enable financial institutions to establish arrangements for sharing and exchanging information on cyber-threat intelligence.

BOX 1 – Smart contracts and the proposal for a Regulation on Artificial Intelligence

The adoption of innovative technologies in the financial sector is constantly growing. Among the most promising applications, the integration of blockchain and artificial intelligence (AI) assumes a primary role, opening new frontiers for the automation of complex processes and the creation of next-generation systems. In this context, the proposal for a European Regulation on Artificial Intelligence[23] (AI Act) is of fundamental importance.

The AI Act represents a significant step towards regulating AI in the financial sector. While the Regulation does not intend to introduce a comprehensive and detailed standardisation of every aspect related to AI, it provides a solid framework for the assessment and development of innovative technologies such as smart contracts.

The Regulation is based on a risk-based approach, which classifies AI systems according to the level of risk they pose to the fundamental rights and safety of individuals. In particular, AI systems are classified into four risk categories:

- Not acceptable: Prohibited systems (e.g. behaviour-based social scoring).
- High risk: systems subject to specific rules and controls (e.g. staff selection, credit risk assessment).
- Limited risk: systems subject to transparency obligations (e.g. chatbots).
- No risk: not subject to the Regulation.

For high-risk systems, the Regulation introduces a 'horizontal protection clause' to prevent non-harmful systems from being subject to too stringent rules. Specific provisions are also foreseen for General Purpose Artificial Intelligence (GPAI) systems, including large generative AI models. Additional requirements are introduced for models that could pose systemic risks and specific transparency obligations apply to systems that perform a wide range of distinctive tasks, such as generating videos, texts, images, computing data or generating computer codes.

Beyond the specific regulatory provisions, the AI Act offers crucial food for thought for the regulation of smart contracts. First, the proposal highlights the need for a holistic approach that must necessarily be taken into account, given the interconnection between blockchain, AI and smart contracts. Secondly, it stresses the importance of identifying objective criteria to classify the level of risk associated with the different types of applications that could also include the use of smart contracts.

The adoption of the AI Act will have a significant impact on the financial sector, leading to the need to adapt existing practices and technologies to the new requirements. Effective regulation is fundamental in order to ensure the responsible development of financial AI and to maximise the benefits of this technology, while minimising risks. Among the requirements that the AI Act introduces, attention should be paid to data governance, automatic logging, risk management, transparency and human oversight.

Particularly significant, in the context of smart contracts, is the requirement of transparency, functionally linked to the requirement of human supervision. The AI Act requires AI systems to be structured in a way that allows users to 'understand and appropriately use the system'. This objective translates into an obligation for companies to:

- adopt XAI (eXplainable AI) systems to make the logic behind the decisions made by AI understandable;

- carry out a cost-benefit analysis to choose the most suitable AI system based on the context of use;
- ensure the knowledge of the system by a common user.

In cases of processing of personal data, the access rights provided by the General Data Protection Regulation (GDPR) at Articles 12-15 and 22 may complement what is not explicitly provided for by the AI Act on transparency. These rights apply, for example, to scoring models used in the financial sector, which rely to a large extent on the processing of personal data.

The new Consumer Credit Directive extends the rights of the GDPR to AI systems used, for example, for assessing the creditworthiness of consumers. These rights include:

- the right to obtain meaningful information about the evaluation carried out and the functioning of the automated processing;
- the right to express their point of view and to challenge the creditworthiness assessment and the decision;
- The right to human intervention.

In addition to the GDPR and the Consumer Credit Directive, the transparency of AI models used in financial matters is underpinned by the general principles of best interest and adequate information vis-à-vis[24]the customer. As is well known, these principles require that the information provided by financial institutions should be appropriate to the client who receives it and therefore in line with his information needs.

'Appropriate' information on the AI systems used should be instrumental in providing the customer with an understanding of how AI determines the provision of the final financial product or service, thereby increasing customers' awareness of the technologies used, for example, for the determination of the credit rate or the provision of investment advice.

In summary, transparency is a key requirement for the responsible use of AI in smart contracts. The AI Act, the GDPR and the Consumer Credit Directive provide an articulated regulatory framework for the protection of users' rights. However, users' right to information needs to be strengthened to ensure full transparency and effective control over the information rendered by AI models.

Below is a brief overview of the primary and secondary disciplines already adopted (2.1) and under discussion (2.2).

2.1 Speaking of DLT[25] (de iure condito)

On 24 September 2020, the European Commission published a package of proposals on the digitalisation of the financial sector, including on DLT applications, which consists of two strategic

[23] The proposal for a Regulation, presented on 21 April 2021 by the European Commission, aims to establish harmonised rules for the use of artificial intelligence on the territory of the Union in compliance with the principle of technological neutrality and safeguarding the rights of individuals and businesses. Assuming its publication by the first half of 2024, the Regulation would apply two years after its entry into force, i.e. from the second half of 2026.
[24] Under Article 5(3) of the Consumer Code: 'Information to the consumer, from whomever it comes, must be appropriate to the communication technique used and expressed in a clear and comprehensible manner, also taking into account the manner in which the contract was concluded or the characteristics of the sector, such as to ensure consumer awareness'.
[25] On the definition of DLT see footnotes 1 and 2.

communications, the Retail Payments Strategy and the Digital Finance Strategy[26]. As part of the latter, the proposals led to the adoption of the following legislation:

i. Regulation (EU) 2022/858 of 30 May 2022 on a pilot regime for market infrastructures based on distributed ledger technology (DLT Pilot Regime);
ii. Regulation (EU) 2023/1114 of 9 June 2023 on markets in crypto-assets (MiCAR; Markets in Crypto-Assets Regulation);
iii. Regulation (EU) 2022/2554 of 14 December 2022 on Digital Operational Resilence for the Financial Sector (DORA).

The aim of the EU DLT Pilot Regulation is to remove regulatory barriers to the issuance, trading and settlement of financial instruments issued in digital form and to support regulators in gaining experience in the use of DLT.

In so far as is relevant here, the Regulation deals with:

- granting, withdrawing and amending authorisations to operate, including exemptions and compensatory or corrective measures, for the management of DLT market infrastructures (identified by the Regulation as 'DLT multilateral trading facility, DLT settlement system or DLT trading and settlement system', Article 2(5));
- the functioning and supervision of the DLT market infrastructure by competent authorities;
- cooperation between operators of DLT market infrastructures, national authorities and ESMA.

In order to operate under the Regulation, operators of DLT market infrastructures must meet specific requirements and provide appropriate collateral. These conditions are intended to preserve: (i) investor protection, (ii) market integrity and (iii) the financial stability of the system.
On compliance with these requirements and conditions and on the appropriateness of the type of technology used, ESMA gives a non-binding opinion when granting authorisation (Article 8(7); Article 9(7); Article 10(7)).

Infrastructure managers using DLT[27] shall, inter alia:

- establish rules on the use of the technology through clear and detailed business plans and up-to-date publicly available and detailed written documentation (Article 7(1)28). The adequacy of the DLT technology to comply with European legislation would appear to be the main element on the basis of which the operator is allowed to operate;
- provide clear and unambiguous information to participants, issuers and clients;

[26] https://finance.ec.europa.eu/publications/digital-finance-package_en
[27] The Regulation identifies three entities as operators of DLT infrastructures:
(i) the operator of a DLT MTF, i.e. an MTF, which admits only DLT financial instruments to trading. It is subject to the requirements that apply to an MTF under Regulation (EU) 600/2014 of the European Parliament and of the Council of 15 May 2014 on markets in financial instruments and Directive 2014/65/EU of the European Parliament and of the Council of 15 May 2014 on markets in financial instruments;
(ii) the central securities depository (CSD) of a DLT SS, a 'DLT settlement system': settles transactions in DLT financial instruments against payment or delivery. The CSD of a DLT SS shall be subject to the requirements that apply to a CSD operating a securities settlement system in accordance with Regulation (EU) 909/2014;
(iii) the operator of a DLT TSS, a 'DLT trading and settlement system': a DLT TSS or a DLT SS combining the services provided by a DLT MTF and a DLT SS. It is subject to the requirements that apply to an MTF under Regulation (EU) 600/2014 and Directive 2014/65/EU.
[28] 'operators shall [...] establish or document, as appropriate, rules for the operation of the distributed ledger technology they use, including rules on access to the distributed ledger, participation of validation nodes, resolution of potential conflicts of interest and risk management, including any mitigation measures to ensure investor protection, market integrity and financial stability'.

- ensure the security, continuity and continuous transparency, availability, reliability of all IT and cyber devices related to the use of DLT technology, 'including the reliability of smart contracts used in the DLT market infrastructure. Such devices shall also ensure the integrity, security and confidentiality of all data stored by the operators concerned and that such data is available and accessible' (Article 7(4));[29]
- preparing specific operational risk management procedures (Article 7(4)(2));
- segregate funds and provide collateral for the DLT financial instruments. The DLT manager is responsible for any loss of funds and guarantees (Article 7(6));
- Plan a timely and clear 'transition strategy': there is a strategy for the transition or reconversion of distributed ledger technology operations into traditional market infrastructures where: (i) the total value of DLT financial instruments reaches EUR 9 billion (Article 3(3)); (ii) there is a voluntary or involuntary cessation of the operation of the DLT market infrastructure; (iii) a specific authorisation or exemption granted under the DLT Pilot Regime is revoked or otherwise suspended (Article 7(7)).
  With regard to the withdrawal or suspension of authorisation, the case most frequently referred to in the Regulation is that in 'the operation of the distributed ledger technology used or in the services and activities provided by the operator [..] a flaw has been detected which represents a risk to investor protection, market integrity or financial stability and the defect has greater weight than the benefits offered by the services and activities being tested' (see Article 8(12); 9, paragraph 12; 10, par. 12);
- cooperate closely with the competent authorities designated by the Member States of the Union and submit a six-monthly report.

In addition, operators, in order to be authorised under the DLT Pilot Regime, must demonstrate that they:

- comply with sufficient prudential requirements to meet liabilities and compensate clients;
- have taken appropriate measures for the safekeeping of clients' DLT assets;
- have put in place and implemented measures to ensure investor protection and to handle customer complaints and disputes, including through digital media. The International Organization of Financial Market Supervisory Authorities (IOSCO) published a report in January 2021 [25] describing good practices for developing and improving complaint-handling procedures and mechanisms for retail investors. IOSCO prioritises investor protection through access to independent, affordable, fair, accountable, timely and efficient dispute resolution mechanisms.

Another important application of DLT is crypto-assets, which in the context of the European Union, are regulated by MiCAR. Crypto-assets are digital representations of value or rights that can be transferred and stored electronically, using DLT or a similar technology. MiCAR regulates the issuance and trading of e-money tokens (EMTs), asset-referenced tokens[30] (ARTs)[31] and unbacked crypto-assets (also called 'other than'), as well as the provision of crypto-asset services (e.g. custody,

[29] Recital 41 states that: 'DLT market infrastructures should have specific effective IT and cyber arrangements regarding the use of distributed ledger technology. Those tools should be proportionate to the nature, scale and complexity of the business plan of the operator of the DLT market infrastructure. Those tools should also ensure the continuity and continuous transparency, availability, reliability and security of the services provided, including the reliability of any smart contracts used, regardless of whether those smart contracts are created by the DLT market infrastructure itself or by third parties as a result of outsourcing procedures. DLT market infrastructures should also ensure the integrity, security, confidentiality, availability and accessibility of data stored in the distributed ledger. The competent authority of a DLT market infrastructure should be allowed to request a verification to ensure that the general IT and cyber tools of the DLT market infrastructure are fit for purpose'.

[30] A type of crypto-asset that aims to maintain a stable value by referring to the value of an official currency.

[31] A type of crypto-asset that is not an e-money token and that aims to maintain a stable value by reference to another value or right or a combination of the two, including one or more official currencies.

exchange for official currencies, exchange between crypto-assets, trading). In this way, a specific and harmonised framework for markets in crypto-assets is introduced at European Union level, with the aim of defining specific rules for crypto-assets and related services not yet covered by the existing financial services legislation. MiCAR, in addition to providing safeguards to protect users and prudential aspects, contains specific provisions aimed at safeguarding financial stability and the smooth functioning of payment systems and addressing the risks to monetary policy that could arise from crypto-assets. However, MiCAR, while basing the definition of crypto-assets on the concept of DLT, does not regulate aspects related to the underlying technology, which should be described in the white paper, i.e. the document containing the information on each crypto-asset, the entities issuing or offering it, the rights and obligations attached and the related risks.

Crypto-assets already regulated by law that fall under the definitions of financial instruments, deposits, funds (except where they qualify as e-money tokens), securitisation instruments, insurance products and pension products are outside the scope of MiCAR. Tokenised financial instruments are therefore regulated by MiFID 2, as well as by the DLT Pilot Regime Regulation and by the national transposing legislation. The MiCAR will be fully applicable from 30 December 2024 but Titles III and IV (relating respectively to the issuance of ART and EMT) apply from 30 June 2024. The technical standards (RTS and ITS) and guidelines that will complement the MiCAR framework, are currently being defined at EU level. In the meantime, on 21 February 2024, the draft legislative decree was put into public consultation with the provisions for the adaptation of national legislation to MiCAR and which, in particular, identifies the Bank of Italy and CONSOB as the competent national authorities.[32]

2.1.1 Anti-Money Laundering/Countering the Financing of Terrorism (AML/CFT)

Due to the typical characteristics of the blockchain technology they use, smart contracts, under specific conditions, could contribute to the implementation of measures to mitigate the risks of money laundering and terrorist financing. Moreover, the interrelationships between smart contracts and cryptoassets are significant, as is the rapid evolution of the anti-money laundering rules applicable to the latter.

The fifth Anti-Money Laundering Directive (Directive 2018/843, cd. 'AMLD5') included some virtual currency service providers (i.e. exchange and wallet service providers) as obliged entities. The Italian legislator had already introduced similar legislation with the Legislative Decree 90/2017 (implementing the Fourth Anti-Money Laundering Directive). The Legislative Decree 125/2019 (implementing the AMLD5) then extended the scope of the activities of Virtual Asset Service Providers (VASPs) subject to anti-money laundering obligations to[33] include any service functional to the use, exchange and storage of virtual assets. [34]

In 2021, the European Commission presented a package of legislative proposals on the argument (the 'AML Package'), which extends and precisely defines the group of entities subject to AML/CFT obligations, including crypto-asset service providers (CASPs) as defined by the MICA Regulation, and introduces specific rules for the information accompanying transfers of crypto-assets, in order to increase transparency and traceability in line with FATF standards.

[32] https://www.dt.mef.gov.it/en/department/public_consultations/micar.html.
[33] On both occasions, national regulatory measures have set out a broader framework than that provided for in the directives, in order to cover, on the one hand, the risks associated with the use of virtual currencies and, on the other, the Recommendations issued by the FATF (see the legislation on the prevention of money laundering: authorities, rules and controls – Anti-Money Laundering Notebooks – Analysis and Studies No 20, 2023).
[34] See Article 1(2)(ff) and (ff-bis) of Legislative Decree 231/2007.

Specifically, the rule requires crypto-asset service providers to collect and make accessible to the authorities certain information relating to the payer and the beneficiary of transfers, thereby ensuring their traceability and facilitating the identification of suspicious transactions.[35]

However, this approach does not fully exhaust the risk associated with the anonymity of transactions, as virtual currency transfers can take place even without the involvement of a service provider that is the recipient of the AML/CFT obligations. This risk may become even more acute in the context of decentralised finance, in the most extreme forms of which it would be possible to set up service[36] platforms in 'decentralised' virtual assets, which may not be easily traceable to entities subject to AML/CFT obligations. Currently, some platforms do not adopt such control tools (i.e. KYC procedures) and are therefore more attractive for illicit purposes[37] than providers of services in 'centralised' virtual assets.

The European legislator is aware of this and states in recital 9 of the Fifth Anti-Money Laundering Directive:[38] 'the inclusion of service providers whose activity consists in the provision of exchange services between virtual assets and real assets and digital wallet service providers does not fully address the problem of anonymity of virtual currency transactions: in fact, since users can carry out transactions even without having recourse to such providers, a large part of the virtual currency environment will remain characterised by anonymity.'
In other words, the current legislation does not apply to transactions carried out in the absence of intermediaries [85] for which the problem of anonymity clearly emerges; furthermore, it does not consider the differences between different blockchain technologies (i.e. private and public, permissioned and permissionless, etc.).

The investigation into the technical aspects of blockchains and smart contracts therefore leads to considering the potential of these technologies for storing information and tracing transactions. In fact, a transaction carried out through blockchain platforms that have appropriate technical and governance characteristics to achieve these objectives, could be transparent and unchangeable as well as secure and traceable. In this context, smart contracts used in the financial sector on blockchain with the characteristics that allow the identification of the parties to the transaction and their traceability, can also be an opportunity for banks and financial intermediaries to facilitate the fulfilment of AML/CFT obligations. In-depth studies in this direction could allow synergies to be identified for the benefit not only of obliged entities, but also of the activities of the competent authorities for the control and monitoring of transactions.

2.1.2 Experimental initiatives: the European Blockchain Regulatory Sandbox

On 14 February 2023, the European Commission launched a European Blockchain Regulatory Sandbox for innovative use cases involving Distributed Ledger Technologies and/or Blockchain.[39]

---

[35] The proposal for a Regulation of the European Parliament and of the Council on information accompanying transfers of funds and certain crypto-assets (recast) – COM(2021) 422 final – is being finalised. One of the innovative elements of the proposed Regulation is that all information on the originator (asset owner) 'travels' simultaneously with the transfer of the crypto-asset (the travel rule), regardless of the amount of crypto-assets involved in the transaction.
[36] The matter is being brought to the attention of several authorities. In a recent report by the U.S. Treasury Department, it is reported that the level of decentralization pursued by such platforms would be verified on a case-by-case basis. In fact, a certain degree of centralization remains, referring for example to the group of subjects who maintain the code of these applications or who hold the administrative keys.
[37] See the Report of the International Organisation of Financial Market Supervisory Authorities cited [27].
[38] Directive (EU) 2018/843 of the European Parliament and of the Council of 30 May 2018 amending Directive (EU) 2015/849 on the prevention of the use of the financial system for the purposes of money laundering or terrorist financing, and amending Directives 2009/138/EC and 2013/36/EU.
[39] https://digital-strategy.ec.europa.eu/en/news/commission-launches-european-regulatory-sandbox-blockchain.

The initiative builds on the need to overcome the current legal uncertainty caused by complex governance of the process. The new methodology aims to simplify and strengthen the dialogue between regulators and innovators. Specifically, the Commission identifies regulatory barriers to the rollout of solutions, and provides advice, experience and regulatory guidance in a safe and secure environment for participants.

Briefly, the Commission will use operators to investigate the technical aspects of these technologies, while operators will help identify best practices for the market,[40] following the participatory regulation process (see below, **BOX 3 – Participatory regulation**).

2.2 Concerning smart contracts (de iure condendo)

The proposal for a Regulation by the European Parliament and the Council on harmonised rules on fair access to and use of data, published on 23 February 2022 (Data Act), aims to establish a framework of interoperability measures, including procedural measures, for the creation of a single internal market for data, in line with the European Data Strategy. Data governance is the cornerstone of the European Digital Transition Programme[41] and the Green Action Plan.

Among the objectives of the proposal, there is the definition of rules for the 'smart contract', defined as a 'computer program stored in an electronic registry system in which the outcome of the execution of the program is recorded in the electronic registry' (Article 2, point 16 - 'Definitions'), considered a tool potentially able to 'provide data holders and recipients with guarantees of compliance with the conditions for data sharing' (paragraph 1 of the Report - 'Context of the proposal - Reasons and objectives of the proposal').

[40] The criteria for selecting candidates clarify the fact that the initiative is not merely exploratory but has a strong impact on the ground: Priority will be given to more mature use cases for the development of which legal issues of wider relevance arise and are already being brought to the attention of European regulators. Specifically:

- As for the participants in the call (lett. A - 'Identity and eligibility of the applicant'), the Sandbox is open to companies from all industrial sectors and public bodies for projects that go beyond the proof-of-concept phase and are already close to the market or at an early stage of operation;
- as regards thesuitability of the use case (lit. B – 'Elibigility of the use case'), the call requires that this has been validated in a 'relevant environment'; furthermore, it must be determined in the context of EU-funded projects and in particular projects funded under the Horizon 2020 and Horizon Europe framework programmes;
- the level of development of the use case (paragraph D – 'Maturity of the use case'), the call specifies that the use cases closest to commercialisation will obtain a higher score based on the assessment of the maturity of technological solutions in the context of EU-funded projects and in particular projects funded under the Horizon 2020 and Horizon Europe Framework Programmes;
- as regards the link with regulatory issues in all industrial sectors (lit. E – 'Link with novel regulatory issues across industry sectors') and relevance to EU policy priorities (lett. F – 'Relevance with the EU's wider Policy Priorities'), the Commission objectively verifies the continuity of the use case presented with the topics already on the European debate tables and still under development;
- the Commission is interested to know if the participant is assisted by one or more European and/or national regulators and to indicate which in particular (lett. H – 'Regulator support').

[41] See: European Commission, Annexes to the Commission Work Programme 2020 – 'A Union that strives for more' COM(2020) 37, 29 January 2020, p. 4 (' 2.2. A Europe fit for the digital age: A new European data strategy will allow us to make the most of the enormous value of non-personal data, an ever-expanding and reusable resource in the digital economy. [...]
The Commission also intends to present 'a new Digital Services Act [that] will strengthen the single market for digital services and help provide smaller businesses with the legal clarity and level playing field they need' [...], 'review the Directive on security of network and information systems and put forward initiatives to make digital finance more robust against cyber-attacks, including a proposal on crypto-assets'.

The Data Act regulates three central aspects related to each other: (i) access, (ii) use and (iii) interoperability of data. In all three areas, the proposal for a regulation refers to the smart contract as a tool available to the data space operator.[42]

(i) In Chapter III ('Obligations for data holders required by law to make data available'), Article 11 classifies smart contracts among the technical protection measures that the data holder[43] may use to prevent unauthorised access to data and to ensure compliance with Articles 5, 6, 9 and 10 ,[44] as well as with the contractual clauses agreed for making data available.
However, the smart contract, together with any other technical protection measures that may be adopted, must not 'obstruct the user's right to effectively provide data to third parties pursuant to Article 5 or any right of third parties pursuant to Union law or national legislation implementing Union law referred to in Article 8(1)' (Article 11(1), second paragraph).
The Data Act, read together with the working documents that preceded its publication,[45] confirms that, in balancing the different requirements, data sharing (the content of a right for the user, on the one hand, and an obligation for the holder, on the other) prevails over the protection of data against misuse.

(ii) Chapter VIII of the Data Act ('Interoperability')[46] sets out minimum conditions to promote interoperability in smart contracts and identifies some essential requirements that operators have to comply with.

In particular:

- **with regard to interoperability,** Article 28 requires operators drawing up smart contracts to provide 'the means to enable smart contracts to be interoperable within the framework of their services and activities' (Article 28(1) (d)). This provision provides a presumption of conformity for smart contracts that meet the conditions set out in harmonised standards adopted by European standardisation organisations at the request of the Commission, in accordance with the Regulation on European standardisation (Regulation (EU) 1025/2012). Furthermore, in the absence of such harmonised standards, it is provided that the Commission may adopt, by means of implementing acts, common specifications relating to each requirement referred to in paragraph 1;
- **with regard to data sharing,** Article 30 is addressed to the 'seller of applications using smart contracts or, in his absence, the person whose commercial, business or professional activity

[42] See: 2018, White Paper on 'Recommendations for the adoption of common standards in Europe on blockchain and DLT' by CEN (European Committee for Standardisation) and CENELEC (European Committee for Electronic Standardisation); 2018, European Parliament, 'How blockchain technology could change our lives' (in which the economic and social impact of blockchains is analysed and the benefits highlighted); 2018, EBP (European Blockchain Partnership), Declaration signed by 22 countries for the creation of the EBSI (European Blockchain Services Infrastructure) with the aim of ensuring the provision of cross-border digital public services with the highest standards of security and privacy.
[43] The Proposal for a Data Regulation defines the data holder as 'the natural or legal person who has the right, obligation [...] or, in the case of non-personal data and by controlling the technical design of the product and related services, the ability to make certain data available' (Article 2(6)).
[44] These articles are contained both in Chapter II ('BUSINESS TO CONSUMER AND BUSINESS TO BUSINESS DATA SHARING') in which they relate to the obligation of the data holder to make available to third parties the data generated by the use of a product or related service, at the request of the user (Article 5), or to the obligation of the third party receiving the data to use them only for the purposes and under the conditions agreed with the user (Article 6); both in Chapter III, with reference to the fair compensation agreed for the making available of data between the holder and the recipient (Article 9) and to the resolution of disputes through out-of-court bodies (Article 10).
[45] European Parliament resolution of 3 October 2018 on distributed ledger and blockchain technologies 'building trust through disintermediation' (2017/2772 (RSP)); European Parliament resolution of 13 December 2018 on blockchain 'a forward-looking trade policy' (2018/2085(INI)); European Parliament resolution of 25 March 2021 on 'A European strategy for data' (urges the Commission to present a Data Act) (2020/2217/INI)).
[46] 'Interoperability: the ability of two or more data spaces or communication networks, systems, products, applications or components to exchange and use data to perform their functions' (Article 2(19)).

involves the implementation of smart contracts for others in the context of a data provision agreement'. These entities must ensure that the smart contract meets four key characteristics:

a) **robustness**: it must have been designed in such a way as to offer a high degree of robustness in order to avoid functional errors and to withstand manipulation by third parties (Article 30(1)(a));
b) **safe termination and interruption:** it must provide for a mechanism to stop the continuous execution of transactions.[47] In particular, '[the smart contract must] include internal functions that can reset it or instruct it to stop or stop its operation in order to avoid future (accidental) executions' (Article 30(1)(b));
c) **data storage and continuity:** in the event that it is necessary to terminate or deactivate a smart contract, it is necessary to 'provide for the possibility of storing transaction data and the logic and code of the smart contract to keep track of the operations carried out on the data in the past (verification)' (Article 30(1)(c));
d) **access control:** a smart contract must be protected by strict access control mechanisms at the level of governance and of the smart contract itself (Article 30(1)(d)).

(iii) Article 30 requires obliged entities, seller and/or entrepreneur or professional, to carry out an assessment of the compliance of smart contracts with these requirements and to issue an EU declaration of conformity.
Seller and/or entrepreneur or professional shall be responsible for the correspondence of the EU declaration of conformity with the essential requirements of Article 30(1).
The standard also introduces a presumption of conformity for smart contracts that meet harmonised standards adopted in accordance with the rules of the European Standardisation Regulation that impose requirements similar to those of Article 30(1) (Article 30(4)).

Regulatory power in this area is shared between the Commission and the European standardisation organisations (CEN, CENELEC, ETSI). The latter shall develop European standards at the request of the Commission. The rules 'take into account the public interest and policy objectives clearly specified in the Commission's request and are based on consensus. The Commission shall lay down the content requirements to be met by the document and a time limit for its adoption' (Article 30(5)). The Commission intervenes with its own implementing acts containing common specifications only where harmonised standards are lacking or insufficient (Article 30(6)).

3. Italian regulations on smart contracts (de iure condito)

In Italy, the legislator paid attention to the phenomenon and has intervened promptly, compared to other countries, having already partially regulated smart contracts in 2018.

The 'Decreto Semplificazioni' 135/2018, converted with amendments by the Law Nr. 12 of 11 February 2019, allowed the use of distributed ledger technologies and their applications, including smart contracts, and established a framework for regulating the legal effects reagarding their use.

Article 8b(2)[48] of Law Decree Nr. 135/2018 states that:

[47] This is a prescription opposed by the scientific community on the assumption that an interruption of the mechanism of the smart contract imposed by regulation would affect an indispensable requirement of blockchain technology: 'immutability'. The proposed Data Act (..) 'would put smart contract immutability in check, thus challenging technology's survival. Immutability indeed differentiates smart contract from other contractual methods; it creates value' [26].
[48] Furthermore, in paragraph 1, the legislator defines distributed ledger technologies as IT technologies and protocols 'which use a shared, distributed, replicable, simultaneously accessible, architecturally decentralized register on cryptographic bases, such as to allow the recording, validation, updating and storage of data both in the clear and further protected by cryptography verifiable by each participant, not alterable and not modifiable'.

> 'A smart contract is a computer program that operates on distributed ledger technologies and whose execution automatically binds two or more parties based on predefined effects stipulated within the contract itself.'

With reference to the legal effects, Article 8b(2) establishes that smart contracts 'satisfy the requirement of written form', provided that the interested parties have been previously identified electronically. The rule assigns to the Agenzia per l'Italia Digitale (AgID) the task of defining, through guidelines, the requirements for carrying out IT identification. Although a structured contribution is not yet in force, a first step in this direction has been proposed by AgID with the drafting of 'Guidelines for the Modelling of Threats and Identification of Mitigation Actions Conforming to the Principles of Secure/Privacy By Design', and in particular with reference to the 'Best Practices of Secure Design for Architectures Based on Distributed Registers (DLT)'. AgID proposes a high-level analysis of the integrity, availability and confidentiality requirements of a DLT system, with a particular focus on infrastructure components such as the network, data structure and consensus algorithms. Finally, the AgID Guidelines identify smart contracts as the most critical component of DLT; however, they merely suggest a traditional secure-coding approach in software development, without detailing specific threats and vulnerabilities of smart contracts. In fact, the recent provision of 1st June 2023 on surety guarantees,[49] in implementation of Article 26 of the Code of Contracts (Legislative Decree 36/2023), led AgID to issue a decision defining the technical requirements and certification procedures for digital supply platforms.[50] In this decision, the AgID recognzses among the conditions that surety guarantee platforms must meet, the requirement to write surety guarantees using smart contracta. Also on this occasion, the AgID merely provides, for the use of this instrument, a single condition regarding the characteristics the entity must possess to issue guarantees, not those concerning the technology used.[51]

Article 8b (3) also states that, where distributed ledgers comply with the technical standards identified by the Agenzia per l'Italia Digitale, the storage of an IT document using distributed ledger technology 'produces the legal effects of the electronic time stamp referred to in Article 41 of Regulation (EU) 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 domestic market. The reference to the Regulation implies that the data recorded in the blockchain: (a) is assigned a certain date and (b) is recognized as having legal effects and admissibility as evidence in court.

In this regard, the eIDAS Regulation provides the principle of non-discrimination of electronic documents compared to paper documents: the legal effects of the document cannot be denied solely because of its electronic form or because it does not meet the requirements of the advanced electronic signature.[52] By distinguishing the different types of electronic signatures (simple, advanced and qualified),[53] which differ in the degree of security they provide in identifyingì the signatory and providing the document's authenticity, [54] the eIDAS Regulation leaves it to the national regulators of each European country to define the legal effects of the different electronic signatures.

---

[49] Chapter 6 'Guarantee management platforms'.

[50] Ruling 137, in agreement with ANAC and the Prime Minister's Office – Department for Digital Transformation

[51] Paragraph 6.2-5.2 specifies that: 'the writing of the guarantee issued in the distributed ledgers is carried out by means of a smart contract that must ensure that this operation is possible only by a person who is allowed to issue guarantees pursuant to Article 106(3) of the Code is authorised to write in the distributed ledger, subject to electronic identification with a significant or high level of guarantee with reference to the eIDAS Regulation'.

[52] Recital 49 and Article 25(1) 'Legal effects of electronic signatures'. Furthermore, Recital 48 states that 'Although a high level of security is necessary to ensure mutual recognition of electronic signatures, in specific cases such as in the context of Commission Decision 2009/767/EC, electronic signatures with a weaker security guarantee should also be accepted'.

[53] Section 4 ('Electronic signatures'), Articles 25-34 of the Regulation.

[54] Article 8(3) of the eIDAS Regulation ('Guarantee levels of electronic identification schemes'). The degree of trust achievable depends on the complexity and quality of the systems used.

In Italy, qualified electronic signatures are widespread and also known as digital signatures, representing the highest level of security for an electronic signature.[55] It is not, therefore, a simple electronic signature, but a specific type of signature that requires the use of a qualified certificate and employs encryption. According to the provisions of the Code of the Digital Administration, which identifies its characteristics and functioning (Articles 24 to 37), a digital signature[56] grants legal value to documents[57] and is equivalent to the signature of a paper document (handwritten signature)[58] as it guarantees the security requirements required by the legislation.[59] Therefore, the technical-IT process for the creation of this type of electronic signature is based on the concepts of:

- **authenticity:** to ensure that the person who signed the document has also taken responsibility for its content;
- **integrity**: to demonstrate that the document, from the moment it was signed until the moment it was used, has never been modified;
- **non-repudiation**: to ensure that the person who signed the document by means of an electronic signature cannot disregard it; this means that in case of disputes it will not be possible to prove that the owner of the electronic signature did not sign that document.

On a practical level, with reference to the legal effects deriving from the use of electronic signatures, the Code of the Digital Administration[60] states that the legal value of advanced and qualified signatures is that of a handwritten signature, while the other levels of electronic signatures have merely probative value. Specifically, following the use of the simple electronic signature, it is for the court to assess the reliability of the signature because of the security, integrity and immutability characteristics of the system used to affix the electronic signature.
With regard to advanced and qualified electronic signatures, the law establishes the requirements of security, integrity and unchangeability of the systems used and therefore the traceability of the signature to the author. The distinction between the two signatures affects the burden of proof in the event of a dispute of the signature in court: for advanced electronic signatures, if the person against whom the document is produced fails to comply with his handwritten signature, the burden of proof

[55] In Italy, according to the Digital Administration Code (CAD) and the then CNIPA Resolution 45 of 21 May 2009, the three formats allowed for qualified electronic signatures are: Cades (CMS Advanced Electronic Signatures, which creates a cryptographic envelope with an extension of .p7m); Pades (PDF Advanced Electronic Signature, which creates a cryptographic envelope always with an extension .pdf); Xades (XML Advanced Electronic Signatures).
[56] Article 1 of the CAD provides the definition of a digital signature: 'A particular type of qualified signature based on a system of cryptographic keys, one public and one private, interrelated, which allows the holder (electronic signature) via the private key (and a third party) via the public key, respectively, to make manifest and verify the provenance and integrity of an IT document or set of IT documents'.
[57] Art. 24 of the CAD: 'The digital signature must refer unambiguously to a single person and to the document or set of documents to which it is affixed or associated. The affixing of a digital signature complements and replaces the affixing of seals, hallmarks, stamps, markings and marks of any kind for any purpose provided for by the legislation in force'.
[58] The definition of handwritten signature is that provided for in Article 8 of Royal Decree 1669-1933 of the Law on Change: 'Every underwriting of a promissory note must contain the full name or business name of the person who undertakes to do so. However, a subscription in which the name is abbreviated or indicated by the initial alone shall be valid'.
[59] Under Article 35 of the CAD, the devices and procedures used to generate signatures must have security requirements that ensure that the private key (a) is confidential; it cannot be derived and its signature is protected against counterfeiting; be sufficiently protected by the holder against use by third parties. In addition, these devices must meet the requirements set out in Annex II to the eIDAS Regulation.
[60] Art. 20, paragraph 1a: The electronic document meets the requirement of written form and has the effectiveness provided for in Article 2702 of the Civil Code when it is affixed with a digital signature, another type of qualified electronic signature or an advanced electronic signature or, in any case, is formed, after computer identification of its author, through a process having the requirements set by the AgID pursuant to Article 71 in such a way as to guarantee the security, integrity and immutability of the document and, clearly and unequivocally, its traceability to the author. In all other cases, the suitability of the electronic document to meet the requirement of written form and its probative value are freely assessable in court, in relation to the characteristics of security, integrity and immutability. The date and time of the IT document shall be enforceable against third parties if affixed in accordance with the Guidelines'.

lies with the other party; with regard to qualified electronic signatures, the burden of proof lies with those who claim that the signature is not valid.

Moreover, the difference in the effectiveness of strong signatures (advanced and qualified) also concerns the type of acts that can be signed in this way. If, in fact, the qualified electronic signature and the digital signature can be adopted for each of the acts provided for in Article 1350 of the Civil Code 'Acts to be done in writing', the advanced electronic signature can be used only for the acts expressly provided for in point 13 (Article 21(2-bis) of the CAD).

Ultimately, only qualified electronic signatures are legally recognized in all Member States of the European Union.

There are some gaps and ambiguities in Article 8b. The definition of smart contracts is general, and somewhat generic [17]. It refers to the execution of the program, other than the contractual execution, and therefore implicitly presupposes a prior phase of the formation of the agreement. The provision also qualifies the smart contract as legally binding on the parties, leading many to believe that the smart contract itself is source of the legal constraint.

Precisely, these critical issues suggest conducting this work on the characteristics of smart contracts in a holistic way, considering both national and international scientific debate and the regulatory choices made by legislators in other countries.

In this regard, the 'Fintech Decree'[61] by which the Italian legislator transposed the Regulation (EU) 2022/858 (DLT Pilot Regime), which establishes a pilot regime for market infrastructures based on 'Distributed Ledger Technology' and the simplification of Fintech experimentation, is particularly relevant. The provisions of the DLT Pilot[62] introduce the necessary regulatory framework for issuing and trading of tokenised financial instruments. The possibility of tokenising various goods, products or services and thus generating a token in the virtual world and linking it to a real-world good through a smart contract, could have a significant impact in terms of increasing speed and security, but also reducing transaction costs.

Specifically, the DLT Pilot considers smart contracts as one of the elements that can be used by the DLT market infrastructure in carrying out activities, and whose reliability must be guaranteed as much as the continuity, transparency, availability, reliability and security of the services and activities that infrastructure managers offer through IT and cyber devices related to the use of their distributed ledger technology[63] [27, 28, 29].

In this perspective, aimed at 'capturing' the phenomenon of smart contracts in their complexity, the work attributes central importance to the legal doctrine that has addressed the issue of defining the state of the art, to which is added a recognition on the regulatory level, also in a comparative manner. **For further details on this second aspect, please refer to BOX 2 – Comparative analysis at first and second level.**

[61] Decree Law 52 of 17 March 2023, converted into Law 52 of 10 May 2023 establishing urgent provisions on the issuance and circulation of certain financial instruments in digital form and simplifying Fintech experimentation.
[62] More generally, the DLT Pilot aims to enable the use of new technologies, in line with market needs, and to make DLT market infrastructures interoperable with those of the traditional financial system.
[63] Article 7(4) of the DLT Pilot 'Additional requirements for DLT market infrastructures'.

BOX 2 – Comparative analysis of primary and secondary legislation

In the comparative analysis of the legislation of different jurisdictions on Distributed Ledger Technologies and/or blockchain and related software applications commonly referred to as smart contracts, the focus was mainly on jurisdictions of the Member States of the European Union.

The jurisdictions were selected according to a 'snowball' or 'snowball' research methodology, based on the information found online.

The final sampling counts around fifty jurisdictions. For this reason, the results of this survey do not claim to be exhaustive.

For each jurisdiction, the DLT/blockchain and smart contract areas were analysed separately according to the following variables.

With regard to the DLT/blockchain:

- legislation;
- legal definition of DLT/blockchain;
- any additional conditions;
- presence of regulatory acts (i.e. secondary rules).

With regard to smart contracts:

- legislation;
- legal definition of smart contracts;
- qualification of smart contracts as mere software;
- qualification of smart contracts also as a contract;
- possible legal effectiveness of smart contracts.

Each variable was analysed according to a binary approach, verifying for each the presence of data (Y) or its absence (N). For each thematic area, information on the presence or absence of legislation was considered 'preclusive' of all others. In other words, the research on the other variables was based on the positive result for the variable 'legislation'.

Results of the analysis

The analysis highlights the absence of legislation targeting specifically DLT/blockchain in the majority of jurisdictions considered. The jurisdictions for which legislative activity has taken place are mostly European, with a few sporadic exceptions (e.g. Israel where legislation on digital assets is in the process of being approved). Some States of the United States of America have also legislated in this regard (e.g. Wyoming).

As for smart contracts, no specific legislation was found in the jurisdictions considered, except for the legislation of Wyoming, Arizona and Tennessee. Wyoming's legislation broadly encompasses digital assets in general and contains a definition of smart contracts, such as 'an automated transaction, as defined in W.S. 40-21-102(a)(ii), or any substantially similar analogue, which is

comprised of code, script or programming language that executes the terms of an agreement, and which may include taking custody of and transferring an asset, or issuing executable instructions for these actions, based on the occurrence or non-occurrence of specified conditions'. The Arizona legislature (Ariz. Rev. Stat. Ann. § 44-7061 (2018)) defines smart contracts as 'an event-driven program, with state, that runs on a distributed, decentralized, shared and replicated ledger and that Tennessee (Tenn. Code Ann. § 47.10.202 (26 March 2018)) can take custody over and instruct transfer of assets on that ledger'. The Tennessee system further specifies the activities carried out by smart contracts, defined as 'an event-driven computer program, that executes on an electronic, distributed, decentralized, shared, and replicated ledger that is used to automate transactions, including, but not limited to, transactions that:

- Take custody over and instruct transfer of assets on that ledger;
- Create and distribute electronic assets;
- Synchronize information; or
- Manage identity and user access to software applications.'

In addition to the aforementioned American States, in the panorama of the systems considered, the Malta Digital Innovation Authority Act, 2018, must be mentioned, where smart contracts are defined as a 'form of innovative technology' arrangement consisting of::

(a) a computer protocol; and, or

(b) an agreement concluded wholly or partly in an electronic form, which is automatable and enforceable by execution of computer code, although some parts may require human input and control and which may also be enforceable by ordinary legal methods or by a mixture of both.'

In addition, Israel appears to be in the process of approving a general law on smart contracts.

On the international scene, the laws of Vermont must also be mentioned, where a definition of Blockchain is found as 'a cryptographically secured, chronological, and decentralized consensus ledger or consensus database maintained via the Internet, peer-to-peer network, or other interaction' and a discipline on digital records that 'shall considered a record of regularly conducted business activity pursuant to Vermont Rule of Evidence 803(6) unless the source of information or the method or circumstance of preparation indicated lack of trustworthiness' (12 V.S.A. 1913, Title 12, Chap. 081 of 30 May 2018); Ohio, which defines 'electronic record' as 'a record created, generated, sent, communicated, received, or stored by electronic means' and 'electronic agent' as 'a computer program or an electronic or other automated means used independently to initiate an action or respond to electronic records or performances in whole or in part, without review or action by an individual'; and Nevada, where an 'electronic record' is defined as a 'record created, generated, sent, communicated, received or stored by electronic means'. (Nev. Rev. Stat. Ann. § 719.09.0).

In Europe, the Lichtenstein Blockchain Act of January 2020 includes a definition of 'trustworthy technology', understood as 'technologies through which the integrity of Tokens, the clear assignment of Tokens to TT Identifiers and the disposal over Tokens is ensured' and 'trustworthy technologies systems' defined as 'transaction systems which allow for the secure transfer and storage of Tokens and the rendering of services based on this by means of trustworthy technology'. Without entering into the field of smart contracts, the Liechtenstein legislature pays attention to the underlying market infrastructure, laying down general requirements relating to: i) certainty of the allocation of tokens and ii) security of the transfer of tokens.

From the above framework, it can be concluded that the Italian case remains, to date, unique in Europe from the legislative point of view.

Most European legal systems (in particular France, Germany, Luxembourg, Poland, Switzerland) have set up rules on securities tokens, with widely varying guidelines regarding the definition of the regulated subject matter and, therefore, the identification of which securities tokens fall under the new rules. German law, for example, regulates only tokens representing bearer securities, Polish law only unlisted shares, French law all securities not subject to a dematerialisation regime, while Luxembourg and Switzerland seem to open (albeit in different ways given the subordination of the former to supranational legislation, in particular CSDR) to the tokenisation of most securities. From this perspective, these legal systems impose obligations on DLT service providers only to the extent that they facilitate the trading of relevant securities.

Finally, in this work we have adopted an interdisciplinary perspective. Law and technology require a comprehensive overview. Technical aspects supported by the best scientific literature will therefore also be considered, as far as they can help in the legal analysis of the assumptions and consequences of the use of smart contracts.

It is possible to make a methodological choice precisely on the basis of the Italian legislation, which, as has been pointed out, does not clearly distinguish between the execution of the contractual agreement and the execution of the computer programs. Therefore, the analysis will proceed with a firm distinction between smart contracts and smart legal contracts, in order to assess their legal advantages and limitations.

Section II - Doctrine: Themes and Issues

4. The different approaches to the nature of smart legal contracts

Understanding the legal nature of the smart legal contract and proceeding to its qualification is considered essential by the legal doctrine, both to establish the use cases and to identify the applicable regulation. Therefore, assessing whether the smart legal contract has the sole function of automatically execute certain contractual obligations, or whether it can be considered as a real contract is obviously not only a theoretical issue. It determines significant practical consequences regarding the technical characteristics that technology must guarantee to ensure compliance with regulatory requirements and enable the legal effects desired by the parties.
Qualification entails the need of identifying the constraints that regulate the entire lifecycle of the contractual relationship, taking into account the different rules operating in different sectors and different types of customers involved.

That said, there are three main approaches proposed by the legal literature to adress the issue of smart legal contracts.

4.1 Smart legal contracts as a mechanism of fulfilling obligations

According to this approach, the smart legal contract is a computer program used only to execute, in whole or in part, a contract. It serves as a computerized transaction protocol that automatically executes orders (in this case, the terms of an off-chian contract) upon the occurrence of predetermined conditions.

A smart legal contract does not represent, therefore, according to this thesis, t a real contract, but rather the software (or information protocol) developed for its execution [30]. In this sense, a smart legal contract is 'programmed' so that it does not need additional rules to those incorporated in the code and is, therefore, (self-)sufficient [31, 32, 33, 34, 35]. The terms and conditions of the contract, agreed between the parties, written in code and stored in the blockchain, become verifiable, immutable and irrevocable [36]. When the requirements of the agreement are met, the smart legal contract evaluates the terms defined by the agreement (following an 'if-then' logic) and, if possible,automatically enforces the expected effects, such as approving the exchange of a token between the parties.
The automation of the execution of the smart legal contract ensures compliance. The blockchain architecture, in fact, does not allow for voluntary violations of the established conditions. In this way, the degree of enforcability of the agreement derives from the code layer upon which the smart legal contract is executed, rather thanfrom an exogenous (regulatory) source [37]. In the application of smart legal contracts, this interpretation involves a shift in contractual practice, from ex-post authoritative judgement typical of traditional contracts, towards an automated *ex-ante* evaluation [38].

4.2 Smart legal contracts as a technological expression of the business agreement

According to this approach [39, 40, 41, 42, 43, 44, 45], the smart legal contract is a computer program also employed to form (in computer language), in whole [46] or in part [47], the content of the contract which is then executed automatically. The program, therefore, although not without challenges in the application of certain legal institutions [48, 49, 50], is able to integrate some contractual phases (agreement and execution), as already assumed in legal literature concerning standardized contracts [51]; Ultimately, it expresses the legal relationship between the parties as mutually agreed upon. Moreover, by removing the human factor from the performance of the services inferred from the contract due to the 'notarisation' of the clauses on the chain, the smart contract would automatically avoid, or at the very leasr significantly mitigate, the risk of default by one of the contracting party[17].

4.3 Smart legal contracts as 'reinforced contracts'

Within legal literature [52, 53, 54, 55], there are those who believe that the smart legal contract has a negotiating value and can contribute to generate a 'reinforced' contract. This is attibuted to the fact that it is the result of a negotiation process that allows some characteristics of traditional contracts to be combined with the additional ones typical of the blockchain technology. This strengthens their effectiveness in terms of negotiation, self-execution, dispute resolution and negotiation link. This thesis must be distinguished from the more radical positions of those who assert that the smart legal contract is a phenomenon that, due to its degree of autonomy, is destined to replace with algorithms (Rule of Code) the legal rules grounded in the principle of the rule of law (Rule of Law) [57].

4.4 Technical limitations of smart legal contracts

Qualification in one sense or another is not only of theoretical importance but also influences the assessment of the smart legal contract in terms of opportunities and limits.

In ummarizing the doctrinel literature on this subject, it can be noted that several authors [58, 59, 60, 61, 62, 63, 64] focus on the technical limitations of smart legal contracts because the technological guarantees of contract execution would not be sufficient to prevent unjust or unforeseen outcomes [65]. Moreover, while acknowledging the potential of smart legal contracts to foster trust in environments where it is lacking, this perspective does not recognize that they play a decisive role in resolving the problems of bargaining [66]. Broadly speaking, the most widely identified limitations are:

- language: smart legal contracts are inherently limited. The computer, operating on Boolean logic and formal coding language, seeks sharp categorical distinctions in phenomena, unlike the far more indeterminate and ambiguous real world [13, 20, 67, 68, 69]. The need to predetermine a clear and well-defined semantic horizon (which is an advantage in terms of transparency on the agreement's content and leads to a greater degree of certainty concerning future obligations) would, however, lead to a lack of flexibility, compared to real world negotiation ;
- Judgment: some decisions necessitate an assessment of multiple factors, such as industry standards, the regulatory framework and the business relationship between the parties. It is not easy to express in a code, standards and principles such as, for example, 'good faith' [57];
- limited self-determination: contracting parties lack the possibility to make rational choices after the signature of the contract, such as, assessing the appropriateness of compensating damages instead of proceeding with enforcement. Opportunities for 'efficient breach' are lacking [15];
- Vulnerability: susceptibility to attacks, risk of fraud and privacy breaches , all linked to cybersecurity issues. On this point, the European Parliament [23] has proposed technical specifications to prevent such operational risks from operators;
- performance of obligations is not always the sole purpose of a contract: not all agreements are designed for a precise execution. Many open clauses are intentionally drafted to provied parties with discretion, foster long-term relationships in duration contracts and allow the parties to respond flexibly to unforeseen circumstances without the need to redraft the agreement [70, 71];
- the smart contract's ability to represent reality is limited: this is the Digital Twin Problem, according to which the creation through smart contracts of a digital copy of an existing asset (tokenisation of the asset) is not always self-sufficient compared to external reality. In other words, the tokens issued exist on the chain ('digital twin'), while real goods remain 'off-chain', resulting in difficult 'coexistence'. The scenario of digital native goods that do not exist outside the chain is different: in this case, the smart contract enables the issuance of 'native' blockchain tokens, built directly on-chain and residing exclusively on the distributed ledger.[64]

4.5 Smart legal contracts and financial inclusion

A part of the doctrine views the phenomenon of the smart legal contract in the perspective of the potential of the instrument to promote social justice and equity goals [72].
According to some authors [70], smart legal contracts are not designed to accommodate the social complexities involved in traditional contracts, so much so that they are inadequate for social

[64] For further information, see Consob, Tokenisation of shares and token shares, Legal Notebooks, January 2023; see also OECD, Crypto-Asset Reporting Framework and Amendments to the Common Reporting Standard, 2022 [75, 76].

applications. Developers prioritize the technical form of smart contracts, neglecting both the social contexts in which contracts operate and the ways in which parties utilize them. This critical observation is particularly widespread in relation to the financial sector [73]. In this context, the prior assessment of an investor's financial capacity, as the skills necessary to understand the financial markets and their associated risks, is essential for safeguarding less financially informed or more vunerable clients. In the absence of traditional contract law instruments, the application of these contracts could generate undesirable outcomes for investors [74]. On closer examination, this critical finding can be addressed if a smart contract is drafted in natural language and their use in the financial field is accompanied by adequate information and financial education (see below **BOX 4 - Transparency**).
In addition, a significant reduction in processing time and transaction costs could contribute to enhancing financial inclusion [71].

5. Legal issues raised in the legal literature

The legal distinction between traditional electronic agreements and smart legal contracts on blockchain originate from the different methods by which contracts are executed and enforced.

5.1 With reference to the smart contract code

In the analysis of smart contracts as code, 'translational' issues [13] arise from the process of creating computer programs. The main critical issues are focused on the transition from natural language to two other languages comprehensible by machines: programming language and machine language (usually distinguished as high-level and low-level languages, respectively).

'Programming language' refers to a language expressed through words, numbers, punctuation symbols and other graphic symbols[65] and that is expressed in a plurality (hundreds) of languages and even 'dialects' that vary in syntax and semantics. This language is intended for the processing instructions to be translated into machine language for execution by computers. 'Machine language', on the other hand, is a language composed of bits, conventionally represented by the numbers 0 and 1 (alphabet or binary bitcode). It also manifests itself in a plurality of 'languages' or codes, diversified according to the characteristics of the computers (hardware architecture).[66]

In order to transfer to the computer the instructions originally conceived and expressed in natural language, it is necessary to translate them into a programming language (first translation) and then from the programming language into a machine language (second translation). Programmers are responsible for the first phase, while the second phase is handled by the computer automatically through specialized programs designed for this purpose (compilers).[67]

Therefore a problem of adaptability arises, due to the need to transpose the contractual semantics into an algorithmic framework.: human language is 'converted' into programming code, replacing the

[65] The programming language is categorized in high-level programming language, which is normally the first language used by programmers, and low-level programming language, called assembly, whose syntax and semantics ccloasely resemble to machine language. [97] notes : 'The transition from semantic to computer language, on the one hand, involves several iterations, before reaching the final version of the contract rewritten with the criteria of computer logic, on the other hand, assumes an important 'filter function', since it also appears decisive for discerning which clauses, conceived in the legal world, are or are not translatable into binary language and, therefore, are or are not exportable in the computer context. In addition to having to reconstruct the meaning of the clauses in a methodical, logical and sequential manner, using simple and unambiguous phrases, reducing, to a good approximation, the traditional negotiating rules within theframework of the instructions to be carried out, the court must, with the help of the computer, draw up the pseudo-code and the flowchart, depending on the competence'.

[66] A program written in a programming language is referred to as 'source code', whereas a program written in machine language, which is executed by a computer, is called 'machine code' (or 'object code').

[67] In fact, and in almost all cases, there are three translations, as two levels of programming language are used before proceeding with the machine language translation.

comprehensibility, flexibility and ductility of natural language with the binary dialectical rigidity of 0a and 1s [17, 77].

5.2 With reference to smart legal contracts

Lawyers consider the smart legal contract an innovative instrument to articulate, verify and enforce an agreement between the parties. However, in the absence of a clear and unambiguous definition of the phenomenon, scholars have put forward various logical and legal arguments to legitimise the use of smart legal contracts in the practice of the traditional transactions.

According to part of the doctrine, it would be superfluous to investigate the discipline applicable to smart legal contracts: because of their inherent ability to survive outside any legal system, they constitute a genuine alternative to contract law [77]. For example, by virtue of their functional link with distributed ledger systems, the assumption of non-execution of the agreement would have no relevance in terms of application. Moreover, even if there were malice or violence in the formation of the contract, or hypotheses of its invalidity, it would still be impossible to modify the blockchain database *ex post*. On the basis of this reconstruction, any actions for compensation and/or refunds: a) would be unlikely, given the difficulty of identifying the parties, and b) could never affect the functioning of the blockchain.

Another part of the doctrine considers that the agreement between the parties is both essential and sufficient for a valid contract to be concluded. This reconstruction derives, on the one hand, from the comparative analysis that identifies the lowest common denominator of the contract in the agreement, and, on the other (as regards Italian law), from the reading of Article 1321 of the Civil Code, which defines the contract as 'the agreement of two or more parties to establish or terminate a property legal relationship between them'. From this point of view, the elements identified in Article 1325 of the Civil Code[68] must be considered additional to the essential one: the Agreement [13]. In other words, the smart legal contract can be conceptually assimilated to the traditional contract if it is suitable to achieve the effects desired by the parties.

Within this approach, some have enhanced the agreement not only from a substantive point of view, with specific regard to the agreements covered by the contract, but also from a functional point of view. In this perspective, the material action of 'starting' the programme will be evidence of a party's willingness to accept the instructions contained therein; it will therefore be the joint launch of the programme by the interested parties that will document its agreement [77].

According to others [30, 78], however, the smart legal contract designates, in a more general way, a contractual structure, the nature of which lies not in the contents, but in a particular architecture of the contract. What legitimizes this technique of contract construction is the agreement of those who program it, choose it and decide to use it: it is the agreement on the technique (i.e. on the specific structure to be given to the contract) that preserves the negotiating nature of the 'product' of the technique itself. Ultimately, the smart legal contract is a 'contract on the contract' [79] and the valid conclusion of the relationship is determined on the basis of the theory of reliance on the IT instrument [80].

Consequently, those authors recognize that smart legal contracts have the function of producing legal effects that are appreciable for the legal system by reason of their ability to express the will of the parties, even if only on the choice of the contractual architecture to be used to bind themselves to the performance of certain services. From this perspective, the use of the Italian legislator's expression 'effects predefined by the parties' in Article 8b of the 'Decreto Semplificazioni' does not

[68] Article 1325 of the Civil Code: 'The requirements of the contract are: 1) the agreement of the parties; 2) the cause; 3) the object; 4) the form, when it appears that it is prescribed by law under penalty of nullity'.

demonstrate the absence of the agreement, but, on the contrary, the existence of the willingness of the parties to implement (automated) a negotiating agreement, establishing a sort of reality that does not take the form of a *traditio,* but rather through the execution of the contractual terms agreed.

A part of the doctrine [48], on the other hand, starts from the assumption that the digital world cannot be considered detached from the real world, and it is, therefore, necessary to verify what critical issues may arise in the application of the rules designed for traditional transactions to the new phenomenon of smart legal contracts understood as contracts between two or more parties. In other words, for a smart legal contract in order to produce legal effects that are relevant to the law and therefore binding on the parties, it is necessary that the essential elements of this new contractual instrument and the relevant applicable rules are compatible, as far as possible, with the civil law framework governing traditional contracts [20, 36, 81, 82, 83]. In this regard, the issues most concerning by the doctrinal debate concern:

1. **Recognition of the actors** involved in the negotiating agreement: the question arises with regard to the need to verify, in the first place, the legal capacity and capacity to act of the parties, as required by Article 2 of the Civil Code; secondly, to identify with certainty the party that may be sued in the event of a dispute;[69]
2. the **conclusion of the contract**, which is relevant to the application of Article 1326 of the Civil Code ('Conclusion of the contract'), on the basis of which the contract can be considered concluded when 'the person making the proposal is aware of the acceptance of the other party'. The complexity here arises from the nature of the proposal and the acceptance: as declarations of intent, expressed in the smart contract in a programming language, the intention must be perceptible and comprehensible to all the parties. The issue is treated differently in relation to a negotiated agreement or a standard-form contract.[70]
3. the **form of the contract,** relevant in cases where the law requires under penalty of nullity the written form pursuant to Article 1350 of the Civil Code or in any case when the form 'of protection' is required, for example, in the case of investment or banking contracts: here the issue does not concern the suitability of the smart legal contract to guarantee the origin of the declarations, but the possible need to reconstruct the intention of the parties *ex post*, its knowability and enforceability against third parties, or the cases in which the form is established to protect the weak party, in order to guarantee correct and adequate information to ensure their awareness [84, 85];
4. The **application of general principles of law** such as the duty to perform the contract in good faith pursuant to Article 1375 of the Civil Code ('good faith performance'), compliance with which may be assessed after the conclusion of the agreement, since it is linked to the conduct of the parties in the course of its performance;
5. the **nullity of the clauses** included in the contract on the ground that they are contrary to mandatory rules or in the event that, even if there is a valid clause, the smart legal contract code specifically determines their application contrary to the law;
6. **contingencies**, which are of fundamental importance in long-term relationships: the doctrine in this regard questions the ability of the smart legal contract to guarantee certainty in the performance of the service referred to in the contract even in the event of events not originally foreseen by the parties;

[69] This issue has already been addressed by the legal literature with the dissemination of contracts concluded electronically, for the security and validity of which electronic identification techniques were required. However, the solutions identified there, based on centralised systems, are not compatible with DLT systems, whose added value is represented by decentralisation.

[70] With specific reference to the way in which the contract is concluded, there are those who argue that a smart contract is not conceptually different from an advertisement [86].

7. from a broader perspective, reflection on the implementation of the **concept of substantive justice** which is expressed in the mandatory duty of solidarity under Article 2 of the Constitution which, with reference to contract law, (i) complements its content or effects (Article 1174 of the Civil Code); (ii) guides its interpretation (Article 1366 of the Civil Code) and (iii) directs its execution (Article 1375 of the Civil Code);
8. the **resolution of disputes,** which are reduced to a greater or lesser extent by reason of the characteristic of the smart contract code used and which can be resolved within the smart legal contract by means of a set of instruments (relating to the concept and function of 'oracle') again, more or less automatically depending on the smart contract code used.[71] Some of these profiles are also connected to the issue of banking transparency, understood as the comprehensibility of the content and effects of the contract, also in the perspective of assessing the correctness of the intermediary, which is also linked to the identification of the remedies recognized to the customer and / or consumer for the violation of their rights in the execution of the contract. On this aspect see **BOX 4 - Transparency**

More generally, some authors [15] point out that the use of the code leads to uncertainties regarding the protection of the integrity of the parties' wishes [72] as well as the intelligibility of the agreement by the parties and, consequently, the validity of the consent given by them, especially if they lack special IT skills and/or belong to protected categories [97].

That process of adaptation is entrusted to the judgment of the programmer who, if not assisted by a lawyer, may produce a negotiating result, which does not correspond to the will of the contracting parties as they understand and express it, and may give rise to questionable or distorted results.[72]

Moreover, in regulated markets (and in this study mainly those in the banking, insurance and financial markets were assessed), the relationship between the programmer and the jurist may not be sufficient. In many cases, regulatory and/or supervisory authorities must be involved from the beginning in order to cooperate in verifying that the minimum safeguards provided for by the rules are actually in place. The relationship between these entities is defined in the doctrine as 'participatory regulation' [52, 53, 54, 55].

---

[71] From the characteristics of the blockchain technology of reference derive, on a technical level, different methods of dispute resolution: the parties will be able to identify by mutual agreement the one chosen on the basis of the criterion of convenience, in relation to the characteristics of the contract itself. Within this perimeter moves that part of the doctrine that embraces all the limitations and advantages of the different approaches in a coherent and inclusive system, which contains all the possible methods of dispute resolution. In this sense we see: [98], [99].
At present, there are already Decentralized Dispute Resolution (DDR) projects on the market that adopt solutions based on human oracles, which allows to overcome (in part) the limits of traditional dispute resolution systems using: (i) the traceability and immutability characteristics of the blockchain; (ii) self-execution typical of smart contracts; (iii) the flexibility of the intervention of a human agent. However, the efficient use of DDR is limited in these trials to medium-low-value disputes, which do not require complex technical and legal investigations and which have as their object the mere transfer of sums of money. For a detailed analysis of the functioning of some of the current GDRs, see: [100], [101]. For a broader insight into the role of blockchain technology compared to traditional justice see: On the one hand, online dispute resolution based on blockchain technologies seems to be a growing reality, meeting a demand for adjudication that cannot be satisfied by state courts due to the excessive costs of litigation. On the other hand, however, traditional avenues of court litigation could become more attractive for users if blockchain technologies were to become relevant on a wide scale for high-value transactions. In the light of this, claims that blockchains are radically disrupting the way justice is administered by courts are excessive'.

[72] It is possible that: (a) thsre is an unjustified exclusion from the computer program of some instructions implicitly or explicitly contained in the text to be translated (because, for example, some instructions are not recognised as such by the programmer or are deemed by the programmer to be irreducible to the logic of if → then, or because the programmer, while acknowledging that a part of the text contains an instruction reducible to the logic of if → then, nevertheless considers that one or more of the elements of that instruction is too ambiguous and it is therefore not possible to proceed with certainty to identify it), or (b) there is an incorrect translation into programming language due to a misunderstanding or a voluntary deviation from the meaning that is textually recognisable or otherwise known by the programmer and recognisable on the basis of extra-textual elements.

BOX 3 – Participatory regulation

In DLT and blockchain, the rules are embedded in the technology. The smart contract can orient economic flows, but also compress fundamental rights. It is therefore impossible for the supervisory or regulatory authority to intervene *ex post*, imposing not only obligations of transparency and non-discrimination, but also technical requirements that guarantee a minimum level of rights and protections provided for by law. It is therefore necessary that the authorities intervene from the outset, cooperating with the operators, allowing the development of the technology according to shared minimum conditions and requirements, compliant and consistent with the welfare that corresponds to the legal tradition that is intended to be defended.

This regulatory procedure is defined in the legal literature as 'participatory regulation' [52, 53, 54, 55] in order to differentiate it from other forms of cooperation between authorities and the market, such as consultation procedures on measures taken by the authorities, or the definition of behavioural or structural commitments of undertakings in a dominant position (competition law) or holding a significant market position (regulation of electronic communications).

5.3 Translation risks in smart code and smart legal code

Taking up the language theme downstream of this analysis, the 'translation risks' are greater when the original text of the natural language to be translated into programming language is technically connoted [87]. In this case, the programmer is required to understand the technical language used in the natural language text to be translated, or to have or use the means of an organisation that has sufficient resources to hire, manage and assimilate a consultant that is itself sufficiently qualified to fill the knowledge gap of the programmer for the purposes of interpreting the text and translating it into the chosen programming language. This is also the most critical aspect in terms of safeguarding transparency.

According to others [88, 89] instead, computer language has the power to eliminate the vagueness and ambiguity typical of human language. The code structured according to the scheme 'if X then Y' does not know the polysemy, but is characterized by a uniqueness aimed at achieving a message for the machine. The variables and conditions have only one meaning, and deliver to the parties a contract free of ambiguity and ambivalence, both interpretative and executive. The smart contract therefore ensures certainty: (i) in its intrinsic semantic meaning; (ii) in its formal extrinsic meaning; (iii) in the performance of the obligations referred to therein. The smart contract is thus subtracted from possible discordant interpretations of its content. Otherwise, there could be an 'endemic situation of precariousness', given that each of the parties could, in fact, refrain from performing the contract as long as the performance of the contract is not imposed on it by the authorities, where and when this actually happens [89, 90].

BOX 4 – Transparency

Irrespective of the legal classification of the smart legal contract that is assumed to be correct, the latter, if used for banking and financial transactions and services, must also ensure compliance with the provisions on the transparency of contractual conditions and relations with customers, provided for, as regards the specific sector, which is assumed here as a paradigmatic hypothesis even if certainly not exhaustive, in Title VI of the Testo unico delle leggi in materia bancaria e creditizia (TUB, as well as in ad hoc regulations, such as those on the rights and obligations of the parties in payment services pursuant to Legislative Decree 11/2010) and in the provisions on the transparency of banking and financial transactions and services,[73]which regulate relations between intermediaries and their customers. It is therefore necessary that the smart legal contract should be formulated in a manner consistent with the current regulatory system also as regards the profiles of protection of transparency, an area regulated by a transversal corpus that affects different sectors of our legal system (e.g. Bank of Italy, but also CONSOB and IVASS, to remain in the financial and insurance sector).

Transparency and information are essential instruments for protecting customers, especially retail customers, who must be able to make informed contractual choices functional to their needs. At each stage of the contractual relationship (advertising, pre-contractual, contractual, in the course of the relationship and upon its termination) the legislation, including European legislation, establishes rules of conduct for intermediaries, based on compliance with specific requirements as well as a general duty of correctness, which are based, in a nutshell, on: (i) assistance and information obligations to allow the customer to understand the characteristics and costs of the service and / or product, compare them easily with those offered on the market, and make thoughtful decisions; (ii) standardisation of certain information documents; (iii) requirements regarding the form and content of contracts; (iv) remedies for non-compliance with the rules. The scheme envisaged is then 'variable geometry', in view of the principle of proportionality: the rules, in fact, are articulated according to different methods in relation to the needs of the different customer groups and the characteristics of the services.

The use of the smart legal contract instrument to provide financial services or to conclude banking contracts, does not seem in abstract to undermine the objectives of transparency by being able, on the contrary, where possible, to comply with regulatory requirements and managed by conscious customers, for some profiles, to promote clear, certain and transparent relationships. Blockchain technology allows the market to use new tools to efficiently offer financial and banking services, in accordance with the regulatory protection goals. However, this is a hypothesis that will have to be verified, given that to date there is still no such experimentation to guarantee the stability of the mechanism with respect to the rules in force.

On this issue, the fact that supervisors must be able to verify in practice there is no harm for customers and possible gaps in respect of compliance with transparency rules is important; It should be added that, in addition to strict compliance with the rules, the authorities must also be able to ascertain the correctness of the conduct, which depends not only on the way in which the rules themselves have been translated into code for the creation of the smart legal contract, but also on the representation of the contractual conditions to the customer.

As mentioned above, the legislator, who recognised the legal value of smart contracts by means of 'Decreto Semplificazioni' 135/2018, converted with amendments by Law 12 of 11 February 2019, expressly gave smart contracts the same probative value as written contracts, provided that the parties concerned were identified electronically.

Article 8 ter (2) provides that 'smart contracts shall meet the requirement of written form after computer identification of the parties concerned, through a process having the requirements set by the Agenzia per l'Italia digitale with guidelines to be adopted within ninety days from the date of entry into force of the law converting this Decree'.

The provision does not expressly refer to Regulation (EU) 910/2014, known as eIDAS, and to Legislative Decree 82/2005, establishing the Code of Digital Administration (CAD), which, in Article 20 (1-bis) of the specification, provides that the electronic document 'satisfies the requirement of written form and has the effectiveness provided for in Article 2702 of the Civil Code when it is affixed with a digital signature, another type of qualified electronic signature or an advanced electronic signature or, in any case, is formed, after computer identification of its author, through a process meeting the requirements laid down by the AgID pursuant to Article 71 of the Code, in such a way as to guarantee the security, integrity and immutability of the document, as well as its traceability to the author. In all other cases, the suitability of the electronic document to meet the requirement of written form and its probative value are freely assessable in court, in relation to the characteristics of security, integrity and immutability.'

This rule, laid down for the electronic document, may also be useful for interpreting Article 8 ter in a way that is functional to the protected interests. For the purposes of the probative value provided for in Article 2702 of the Civil Code, the characteristics of safety, integrity and immutability are relevant. These prerogatives can be guaranteed by the use of smart legal contracts on blockchain.

On the basis of this consideration, even in the banking sector it can be considered that the smart contract can satisfy the formal requirement laid down in Article 117(1) of T.U.B., according to which 'contracts shall be drawn up in writing and a copy shall be delivered to customers',[74] failing which they will be null and void. Also from this perspective, the immutability and traceability of contractually relevant actions make it possible to determine with certainty when the agreement is finalised and recognise the smart contract as having the same effect as a written contract. The system that distinguishes the transparency regime also provides for a formal invalidity regime that concerns the necessary content of the contract, the contractual conditions and, more generally, the completeness and consistency of the contractual terms, also in relation to the information publicised and provided in the pre-contractual context (for example, those that provide for non-publicised fees). The smart contract must therefore be aligned with the requirements laid down by the rules or by the operators themselves, including giving the customer the opportunity to exercise all legal rights, especially for the purposes of withdrawal from the negotiation relationship. It remains clear that these potentialities of the smart contract are still to be verified in the experimentation phase and presuppose that the problem of language is solved.

More generally, the hopefully positive impact on the degree of transparency of the use of this technology would depend on the characteristics of the smart legal contract of traceability and immutability, which could provide greater certainty to the relationship between intermediaries and customers, for the benefit of the parties to the relationship themselves and, in the future, of the supervisory authorities in carrying out supervisory tasks. If well structured, the smart legal contract could also help to handle complaints and disputes that may arise between the parties.

Once again, the issue of language and translation is of central importance, highlighted by many as the most complex aspect, and is also reflected in the way in which the Authority can carry out the

[73] Reference is made to Banca d'Italia's measure of 18 June 2019 laying down provisions on the transparency of banking and financial transactions and services [91], which concerns the correctness of relations between intermediaries and thier clients. The measure implements Directive 2014/92/EU (Payment Account Directive, or PAD) and Chapter II-ter, Title VI, of the Consolidated Banking Act.

[74] For the sake of completeness, paragraph 2 gives to the CIRC (Inter-Ministerial Committee for Credit and Savings) the power to provide that, for justified technical reasons, particular contracts may be concluded in another form.

supervisory tasks for which it is responsible. In the long run, it seems that these critical issues can be overcome through the possibility (to be technically verified) of making the smart contract readable in natural language.

A delicate aspect, which is relevant for a broader consideration on the consistency of smart legal contracts with the protection rules, is that of the meaning of transparency in the substantive sense, as comprehensibility of the content and effects of the contract, to protect above all the consumer. However, the theme, by its complex nature, does not seem to be placed in too different terms for the on-chain contract and for the traditional off-chain one.

Moreover, the application must make it possible to protect the customer, but also to the authorities, not only an understanding of the contract expressed in natural language, but also that the translation into computer language accurately reflects what is expressed in natural language.

Or rather, the problem of poor comprehensibility arises if in the smart contract the agreement is 'translated' into computer language that cannot be read or understood, if not by cryptographers. In this regard, however, part of the legal literature has objected that the smart contract is a summary of the agreement and that the rules are 'pre-set' by the parties and therefore 'pre-understood' before being codified. This conclusion is based on the assumption that the computer language limits (even if it does not prevent) the comprehensibility of the agreement once it has been uploaded to the chain. A different approach, which can be pursued in the abstract, is instead to articulate the process of creating the smart legal contract by providing that, in the end, it is, in some way, readable in natural language, or its consistency with the text in natural language is ensured, so as to ensure clarity and comprehensibility for customers of the terms of the agreement.

In any case, must be guaranteed to the contracting party the right to the conformity of the computer language version with the natural language version, and appropriate information should be provided by the intermediary on how to exercise this right and on the characteristics of the service or product offered. More generally, intermediaries wishing to use the smart legal contract instrument will have to correctly identify the groups of customers willing to use this instrument (also in order to assess a certain gradualness in its dissemination) and provide adequate information. In other words, there is a question of assessing the correctness of the intermediary in order to protect transparency.

In addition, in order to ensure substantial as well as formal transparency, it is possible to envisage the use of automation systems (**for further details see Box 1 – Smart contracts and the proposal for a Regulation on Artificial Intelligence**) in the execution of the smart contract, for example, by providing, when concluding the agreement, for questionnaires to be submitted to customers aimed at verifying their actual level of technical awareness of the agreement they are negotiating, concluding and/or executing. This would guarantee an additional instrument of protection with respect to those imposed by the legislation in force. Moreover, the smart legal contract instrument has the purpose and the capacity to substantially reduce cases of breach of contract.[75]

It is therefore highly desirable that the smart legal contract should, for this purpose, lay down by design, not only in legal terms but also in technological terms, the appropriate rules to ensure the minimum protections provided for by sectoral legislation [93].

[75] See the survey conducted by Banca d'Italia in 2017 on Fintech in Italy. Fact-finding survey on the adoption of technological innovations applied to financial services, www.bancaditalia.it., [92] according to which: these are contracts written in a computer language that is intelligible by special software, and is able to enter into execution and enforce the clauses contained therein automatically, once the predefined conditions are met.

Again, as we have seen, the smart legal contract does not pose a problem of non-compliance with the written form, if it meets the above requirements, without prejudice to compliance with the rules on unfair clauses that should be signed independently.

In addition, the withdrawal can be exercised with a so-called Kill function at the occurrence of regulatory preconditions.

Finally, the rules governing the remedies available to the customer/consumer upon the occurrence of events that prevent the proper execution of the agreement concluded can also be complied with: remedial protection is reproducible in the way smart contracts are 'created' and applicable. The market should look to this standard as a benchmark.

It is agreed that, and assumes central importance also for the purposes considered here of the protection of transparency, if the contractual clause written in natural language is ambiguous or unclear, the smart legal contract could provide for the possibility of contacting a third party (an oracle); hypotheses that could already be 'planned' during the creation of the smart legal contract. The profiles related to the management (possibly automated) internal and / or external (also with possible attachment to the competence of already existing ADRs, e.g. Banking Financial Arbitrator) is an open front from a conceptual and operational point of view, therefore it will require further monitoring and deepening.

5.4 Outline of private international law

A particularly debated issue in legal doctrine concerns the identification of the rules applicable to agreements that are based on the use of blockchains and smart contracts and that have elements of extraneousness with respect to a given order. The issue does not arise to determine the validity of an agreement that, although concluded through the use of new technologies, concerns national counterparts and must be executed in the national territory: in this case, the relevant national law would be applicable, both substantively and procedurally. The case is different instead of a contract on the blockchain that regulates legal relationships that have elements of internationality and for which one wonders whether or not the traditional connecting criteria established by private international law are effective. In fact, some [106] believe that regulators should not adopt specific rules for blockchain; applying the principle of proportionality, this would be a last resort measure, to be used only when a careful and extensive interpretation of the existing regulatory framework does not prove to be sufficiently elastic and flexible. The applicability of existing contractual and commercial rules should not be excluded *a priori*.

Another part of the doctrine, on the other hand, argues that the solution is to be found at international level, and consists in identifying common principles, suited to the nature of the technology:[76] Blockchains, while operating on a technically similar basis, face fragmentation of the applicable rules,

[76] Thus [104], according to which 'In the opinion of the writer, while assuming that it is desirable to regulate the subject, the preferable solution should be sought at international level (for example within the European Union) and, in particular, through the preparation of common criteria, even flexible, adapted to the nature of the technology in question. The choice to promote competition between the legal systems in regulating the matter risks, in this specific field, favouring the adoption of hasty, and sometimes poor-quality, measures'. In the same sense also [106]: 'The decision to promote regulatory state competition as well as any passive approach with a similar outcome is highly risky in a context where there are no parameters, in terms of both regulation and case law. In such a context, the competition among state may lead to a 'race to the bottom'; [107] argues that 'What many considered to be a "vehicle" of part of the civil law doctrine to create common principles (the *lex mercatoria*) applicable to international contracts, the subject of at most volumes intended to enrich some legal library, becomes a real economic necessity in the common awareness, on the one hand, that national private law cannot regulate contracts for "a-territorial" definitions and, on the other, that a "third" right can constitute a common legal basis to prevent the "void" from being occupied by technology and, therefore, by the "Code as Law"'.

due to the multiplicity of legal systems.[77] It follows that the possibility of applying the rules, including the criteria for resolving the conflict of laws, also to the blockchain is an essential element. Indeed, some authors argue that the smart legal contract would not be contrary to the contract law of individual States and that, on the contrary, the connecting factors on which private international law is based, in particular in the European Union the Rome I Regulation[78], would make it possible to define the national law applicable to the relevant smart legal contract even in the absence of an express indication of that effect by the parties.[79] Moreover, the principle of contractual autonomy[80] allows the parties to determine their preferred national law by settling any disputes relating to the agreement they wish. This would also be possible by considering the applicability of the law 'in an algorithmic way' (representing the choice of law using the technology that underpins the smart legal contract) and allowing the parties to adopt a supplementary agreement to the smart contract specifying the applicable national law(s).[81] On the other hand, the situation is different where the parties have not expressly chosen the applicable law and it is not possible to determine it implicitly on the basis of the content of the contract. In this case, in the European Union, the Rome I Regulation,[82] with the principle of characteristic performance based on the identification of the domicile or residence of the contractual parties (and which depends on the specific nature of the obligation relied on in the contract), presents significant problems of adaptability to the blockchain, characterised on the one hand by decentralisation and on the other by the possible 'pseudo anonymity' of users.[83]
With regard to this approach, some consider that the only way forward is through the explicit decision of the parties both to the law governing their relationship and to the court having jurisdiction to give

[77] [105] identifies, as part of the various national initiatives to regulate the phenomenon, an important risk, capable of undermining those advantages that the use of blockchain is intended to achieve: 'there is a strong risk that the blockchain will be made subject to diverging legal rules [...] Unless some degree of uniformity is sought, the result will be tremendous legal fragmentation around the world. It would become impossible, or at least significantly difficult, to trade on various types of blockchains at the same time. That is because the diverging governing laws will necessarily have an effect on their functioning. Different interfaces will be required to trade in financial instruments on a French DEEP or a Liechtenstein TT. In the absence of common rules, these interfaces will not be interoperable among each other. The inevitable consequence is a rise in transaction costs. Economically beneficial diversification into multiple crypto assets will be made more difficult. Some national markets will be too small and lack sufficient liquidity for trading, with the result that they may eventually be extinguished altogether'.
[78] Regulation (EC) 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, 4 July 2008, 6-16. This Regulation therefore applies, in circumstances involving a conflict of laws, to contractual obligations in civil and commercial matters.
[79] [108] concludes that 'Blockchain technology is a novel phenomenon, but it does – in most cases – not necessitate new connecting factors or conflict rules [...] technical developments may and should act as an impetus to legislators for legislative amendments but should not prevail over the existing rules of law. Those who desire legal advantages – such as a limitation of liability or even a uniform statute – must in exchange fulfil and adhere to the laws' requirements'.
[80] Article 3.1 of the Rome I Regulation (entitled 'Freedom of choice') provides that 'The contract shall be governed by the law chosen by the parties. The choice is expressed or is clear from the provisions of the contract or the circumstances of the case. The parties may designate the law applicable to all or only part of the contract'.
[81] It must be a smart legal contract, therefore understood as an agreement between the parties in the legal sense, regardless of whether it has been concluded on the blockchain or elsewhere: the application of the Rome I Regulation is possible in the event that the smart contract is able to (i) represent the content and conditions of the contract (ii) execute bilateral or multilateral contracts concluded completely or partially outside the blockchain. It does not, therefore, contemplate the case in which the smart contract represents only a mere computer program. In the latter case, the smart contract would still be subject to applicable national and international laws, although not necessarily to the rules of contract law. See: [109]; [110]
[82] Article 4 of the Rome I Regulation ('Law applicable in the absence of choice'), which is based on the search for the criterion of the 'closest connection' between the law and the contract in question.
[83] In this regard, [104]; [111] points out that it is also 'curious to imagine that a party intends to conclude a commercial agreement with an anonymous party. In any case, where the parties want to conclude a legally binding agreement, it is obviously desirable to choose to operate on a blockchain in which the participants are identified or identifiable in the real world, for example by means of a reference to the place of residence, or the registered office of the company'.

a ruling.[84] There is no alternative solution capable of ensuring an appreciable degree of certainty as to the definition of the applicable law.[85]

Moving from the application of international law to contractual dynamics involving the use of blockchain, some authors consider it necessary, but also decisive, to create a uniform private law of blockchain, understood as a set of rules reasoned and created *ad hoc,* for contracts using blockchain and smart contracts. Doctrine proposes two solutions in this direction. On the one hand, someone[86] pointed out that the reference by rules of private international law to rules of national law still leaves the question of 'which law applies' unresolved, since by definition blockchain is linked to multiple legal systems and jurisdictions. The lack of clarity on the identification of the applicable law and the diverging conflict-of-law rules would give rise to disputes that would last for three reasons: (i) each jurisdiction may have different rules of choice of law;[87] (ii) the conflict-of-law rules identified suggest that the courts of the reference countries will not easily accept the 'choice' of the law of another State; (iii) there is also uncertainty about the content: investors cannot be expected to be familiar with the variety of blockchain laws, without considering that the costs of such information would become significant.

On the other hand, there are those who, with specific reference to business-to-consumer contracts, would appreciate an international convention 'drawn up by the highest number of States', capable of overcoming the limits of the extraterritoriality and a-territoriality of the blockchain. From that point of view, the smart legal contract cannot be regarded as self-sufficient.[88]

[84] Moreover, the limits that the autonomy of the parties would encounter when one of them is a worker or a weak contractor, protected by a necessary and mandatory regulation, are clear. See [104]. The same view is also expressed in [107], according to which 'where the smart contract is concluded on the blockchain between a trader and a consumer, the conclusion protocol and the implementation protocol must be set up in compliance with consumer protection legislation'. From an even broader perspective, there are those who have even gone so far as to argue that 'what is needed, at least in our reality, is not (only) yet another European Directive (although necessary, at least for some of the issues described above), but rather a new "Vienna Convention", to which the construction of a real world code of the algorithmic contract is to be entrusted. Oriented not (only) to "legitimate" a technique, but to make it compatible with the higher values of equality and solidarity, to which it must conform, whatever the technique to which its formation or execution is entrusted' [112]

[85] [104]. In the same way [109]

[86] [105]

[87] Think of the Wyoming draft law that provides for its applicability in the event that a digital asset is in that state, which is considered as such when this asset is held by a custodian resident there, if the holder or the guaranteed part is incorporated or organized in Wyoming. Other jurisdictions may have different choice of law rules and therefore not apply Wyoming law in these same circumstances. Thus [105] 'Given the division of the world into different states with diverging legal systems, each and every form of property exists by virtue of its recognition under some applicable national law. It is first necessary to identify this law through the mechanics of conflicts of law before it can be applied to any phenomenon of the real or virtual world'. See [113].

[88] [107] argues that 'If it is true that mathematics uses a universal language, the attentive observer cannot fail to note that in smart contracts the mathematical language presupposes the "translation" of natural language: it is, in fact, the computer scientist, who creates the program, who translates the concepts taken from natural language into algorithms. It follows that IT would become a legislator if, in creating the algorithms, it did not follow the legal rules'.

Bibliography

[1] Banca d'Italia, 'Communication from the Bank of Italy on decentralised technologies in finance and crypto-assets' (2022)
[2] Banca d'Italia, 'No. 26 - Integrating DLTs with market infrastructures: analysis and proof-of-concept for secure DvP between TIPS and DLT platforms' (2022)
[3] Cipollone, P., keynote speech at the 'Conference on Digital Platforms and Global Law', URL: https://www.bancaditalia.it/pubblicazioni/interventi-direttorio/int-dir-2022/en_CIPOLLONE_29_aprile_2022.pdf?language_id=1 (2022)
[4] OECD, 'The Tokenisation of Assets and Potential Implications for Financial Markets' (2020). URL: https://www.oecd.org/finance/The-Tokenisation-of-Assets-and-Potential-Implications-for-Financial-Markets.pdf
[5] European Law Institute 'Principles on Blockchain Technology, Smart Contract and Consumer Protection' (2022)
[6] Szabo, N. 'Smart Contract' (1994)
[7] Szabo, N. 'Smart Contract Glossary' (1995)
[8] Szabo, N. 'Smart Contract: Building Blocks for Digital Markets' (1996)
[9] Szabo, N. 'The Idea of Smart Contract; Formalizing Securing Relationships on Public Networks' (1997)
[10] Amato, C. 'Computerisation of contracts (Smart, data oriented, computable and self-driving contracts. An overview)', Europe and Private Law, fasc.4, 1 December 2020, p. 1259 et seq.
[11] Buterin, V., 'A Next-Generation Smart Contract and Decentralized Application Platform' (2014) available at http://ethereum.org
[12] Belloni, D and Vasoli, F. 'Blockchain, Smart Contract and Simplification Decree, in Cammino Diritto' (2020)
[13] Orlando, S. 'Smart contracts as software products', 2021 Yearbook – Legal Observatory on Digital Innovation (2021)
[14] Stark, J. 'Making Sense of Blockchain Smart Contract', in Coindesk (2016)
[15] Patti, F. P. and Janssen A. U. 'Demystifying smart contracts', ODCC (2020), pp.31-50
[16] Carron, B. and Botteron, V. 'How smart can a contract be?', in D. Kraus, T. Obrist, O. Hari (eds.), Blockchains, Smart Contract, Decentralised Autonomous Organisations and the Law, Cheltenham, UK-Northampton, MA, USA (2019) p. 101 et seq., spec. pp. 111-114
[17] Rinaldi, G. 'Smart contract: mechanisation of the contract in the blockchain paradigm', in G. Alpa (cur.), Law and artificial intelligence (2020), pp. 353-354
[18] Davola, A. 'Blockchain and Smart Contract as a Service: market prospects with critical regulatory aspects of BaaS and SCaaS services in the light of an uncertain legal classification', Industrial Law (2020)
[19] Ante, L. 'Smart Contract on the Blockchain—A Bibliometric Analysis and Review', BRL Working Paper Series No. 10 (2020)
[20] Gentili, A. 'Will in the digital context: market interests and people's rights', Rivista Trimestrale di Diritto e Procedura Civile, fasc.3 (2022), p. 701
[21] Pardolesi, R. – Davola, A. 'What is wrong in the debate about smart contract', Book of Short papaers, Società Italiana Statistica (2019), p. 481
[22] Cappai, M. 'The role of private and public regulation in the case study of crypto-assets: The Italian move towards participatory regulation', Computer Law & Security Review, 49 (2023)
[23] EU Blockchain Observatory Forum, Smart Contract Report (2022), pp. 10-13, URL: https://www.eublockchainforum.eu/sites/default/files/reports/SmartContractsReport_Final.pdf
[24] Proposal for a Regulation of the European Parliament and of the Council on artificial intelligence and amending certain Union legislative acts (COM/2021/205 final)
Bertuzzi, L. 'AI Act Enters Final Phase of EU Legislative Process' (2023) https://www.euractiv.com/section/artificial-intelligence/news/ai-act-enters-final-phase-of-eu-legislative-process/.

[25] IOSCO 'Compliant Handling and Redress System for Retail Investor' Final Report (2021)
[26] Schrepel, T 'Law+Technology', Stanford CodeX Working Paper, p. 6, (2022) http://dx.doi.org/10.2139/ssrn.4115666
[27] Malvagna, U. 'Digital securities: First notes on the Decree implementing the DLT Pilot', Journal of Banking Law (2023)
[28] Annunziata, F. 'The new EU Distributed Ledger Regulation', Journal of Banking Law (2022)
[29] Annunziata, F., Chisari A. C. and Amendola, P. R. 'DLT-Based Trading Venues and EU Capital Markets Legislation: State of the Art and Perspectives under the DLT Pilot Regime', Bocconi Legal Studies Research Paper No. 4344803 (2023)
[30] Di Ciommo, F. 'Smart contract and (non-) law. the case of financial markets', Nuovo Diritto Civile, IV Anno (2019), p. 257 et seq.
[31] Cuccuru, P. 'Beyond bitcoin: an early overview on smart contract', International Journal of Law and Information Technology, vol. XXV (2017), 179 et seq.
[32] Cutts, T. 'Smart Contract and Consumers', West Virginia University (2019)
[33] Karamanlioğlu, A. 'Concept of Smart Contract. A Legal Perspective', Kocaeli Üniversitesi Sosyal Bilimler Dergisi (2018), p. 29 et seq.
[34] Kasprzyk, K. 'The Concept of Smart Contract from the Legal Perspective', Review of Comparative Law Vol. XXXIV (2018)
[35] Raskin, M. 'The Law and Legality of Smart Contract', Geo. L. Tech. Rev.305 (2017), p. 312
[36] Parola, L., Merati, P. and Gavotti G. 'Blockchain and smart contract: open legal questions', Contr. (2018), p. 681 et seq., spec. 685.
[37] Pellegrini, T. 'Self-executive performance: smart contract and surroundings', Comp. dir. civ. (2019), pp. 27-28.
[38] Jerry, I. – Hsiao, H. 'Smart Contract on the Blockchain – Paradigm Shift for Contract Law?', US – China Contract Law Review, Vol. 14 (2017), p. 686 et seq.
[39] Giaccaglia, M. 'Considerations on blockchain and smart contracts', Contract and enterprise (2019), p. 951
[40] Giuliano M., 'Blockchain and smart contracts in law innovation in the third millennium', Diritto dell'Informazione e dell'Informatica (Il), fasc.6 (2018), p. 989 et seq.
[41] Catchlove, P. 'Smart Contract: A New Era of Contract Use', available on SSRN (3090226), p. 15
[42] Clack, C. D. et al., 'Smart contract templates: foundations, design landscape and research directions 2' (last revised 2017)
[43] Durovic, M. and Lech, F. 'The Enforceability of Smart Contract', Italian Law Journal, 2 (2019), p. 504 et seq.
[44] Eenmaa-Dimitrieva, H. and Schimdt – Kessen, M. J. 'Creating Markets in No-trust Environments: the Law and Economics of Smart Contract', 35 C.L.S. Rev. (2019), pp. 69-88
[45] Stazi, A. 'Smart Contract: Elements, Pathologies and Remedies', European University of Rome, National University of Singapore (Forthcoming in: Law and Change: An Asian Perspective, edited by J. Loo & N. Remolina Leon, SMU) 2022
[46] Werbach, K. and Cornell, N. 'Contract ex machina', Duke Law Journal (2017) p. 338 et seq.
[47] Caggiano, I. A. 'Il contratto nel mondo digitale', (edited by Lucilla Gatt) Il Contratto del Terzo Millennio - Dialogando con Guido Alpa, Editoriale Scientifica Napoli (2018), p. 61 et seq.
[48] Maugeri, M. 'Smart contract and contract regulation', Il Mulino, (2021)
[49] Finocchiaro, G. 'The contract in the era of artificial intelligence', Quarterly Journal of Civil Law and Procedure, fasc.2 (2018), p. 441 et seq.
[50] Carriero, V. 'Smart Contract In The Blockchain Context: What Happens?', European Journal of Privacy Law & Technologies (2019)
[51] Irti, No 'Exchanges without agreement', in rev. trim. dir. proc. civ. (1998), p. 350 et seq.

[52] Bassan, F. 'Digital Platforms and Blockchains: The Age of Participatory Regulation', European Business Law Review (2023), pp. 1103-1132
[53] Bassan, F. 'Digital Platforms and Global Law, Edward Elgar Publishing' (2021)
[54] Bassan, F. 'Algorithm power and market resistance in Italy. Lost sovereignty over services', Rubbettino (2019)
[55] Bassan, F. - Rabitti, M., Recent evolutions of contracts on the blockchain. From smart legal contracts to 'contracts on chain', Rivista di Diritto Bancario, 2023, pp. 561-639.
[56] Rabitti, M. 'Financial products between rules of conduct and organisation. The limits of MiFID II', Journal of Banking Law, fasc. I (2020), pp. 145-177
[57] De Filippi, P. and Wright, A. 'Blockchain and the Law. The Rule of Code', Harvard University Press (2018), p. 77 et seq.
[58] Giancaspro, M. 'Is a smart contract really a smart idea? Insights from a Legal Perspective', Computer Law & Security Review (2017), pp. 7-8
[59] Lauslahti, K., Mattila, J. and Seppälä, T. 'Smart Contract - How will Blockchain Technology Affect Contractual Practices?', ETLA Report, No 68, Elinkeinoelämän Tutkimuslaitos (2017)
[60] Levi, S.D. and Lipton, A. B., 'An Introduction to Smart Contract and Their Potential and Inherent Limitations' (2018)
[61] Howell, B. E. and Potgieter, P. H. 'Uncertainty and dispute resolution for blockchain and smart contract institutions', Cambridge University Press (2021)
[62] Lim, A. '502 Bad Gateway: Rebooting Smart Contract, Legal Information Management', Volume 20, Issue 2 (2020), pp. 106 – 107
[63] Mik, E. 'Smart contract: Terminology, Technical Limitations and Real World Complexity', Law, Innovation and Technology, 9 (2017) pp. 269-300
[64] Savelyev, A. 'Contract law 2.0: Smart Contract as the Beginning of the End of Classic Contract Law', Information & Communications Technology Law (2017), p. 124 et seq.
[65] Faini, F. 'Blockchain and law: the 'value chain' between IT documents, smart contracts and data protection, Civil Liability and Social Security, fasc.1 (2020), p. 307 et seq.
[66] Fauceglia, D. 'The problem of smart contract integration', I Contracts (2020)
[67] Crisci, S. 'Artificial intelligence and algorithm ethics', Hole admin. (2018), p. 1803 et seq.
[68] Gentili, A. 'Crisis of categories and crisis of interpreters', Journal of Civil Law (2021)
[69] Caldana, F. G. and Colosio, C. 'Smart Contract: problems of interpretation. Overview of a systematic nature' (2021)
[70] Levy, K.E.C. 'Book Smart, not Street-Smart: Blockchain-Based Smart Contract and the Social Workings of Law', Engaging Science, Technology and Society (2017)
[71] Casey, M. 'Could Blockchain Technology Help the World's Poor?', World Economic Forum Agenda (2016)
[72] Gitti, G. 'Digital technologies, people and institutions', Journal of Civil Law (2020)
[73] Lusardi, A – Mitechell, O. 'Financial Literacy and Retirement Preparedness: Evidence and Implications for Financial Education', Business Economics (2007)
[74] BIS Working Papers, No. 1061 'Cryptocurrencies and Decentralized Finance (2022), pp. 11-19
[75] Consob, 'Tokenisation of shares and token shares', Legal Notebooks (2023)
[76] OECD, 'Crypto-asset Reporting Framework and Amendments to the Common Reporting Standard' (2022)
[77] Sklaroff, J. M. 'Smart Contract and the Cost of Inflexibility, University of Pennsylvania Law Review, Vol. 166, (2017), p. 287 et seq.
[78] Manente, M. 'L. 12/2019 – Smart contracts and technologies based on distributed ledgers – Prime notes', National Council of Notaries, approved by the IT Commission on 4 April 1029, pp. 6-7

[79] Benedetti, A. M. 'Contract, algorithms and transnational civil law: Five Questions and Two Scenarios', Journal of Civil Law, fasc. 3 (2021), pp. 415-417
[80] Lemme, G. 'Smart contracts and the three laws of robotics', Economic Legal Analysis, fasc. 1 (2019)
[81] Nuzzo, G. 'Smart contracts between calculability needs and contingency management' Content and limits of private autonomy in Germany and Italy, by Bordiga and Wais, Turin (2021)
[82] Cerrato, S. A. 'Notes on smart contracts and contract law', Banca Borsa Titoli di Credito, fasc. 3 (2020) p. 370 ff
[83] Sirena, P. and Patti, F. P. 'Smart Contract and Automation of Private Relationships, Bocconi Legal Studies Research Paper Series, No. 3662402 (2020)
[84] Cappiello, B. 'From smart contract computer code to smart (legal) contract. The new (para) legal instruments in the light of national legislation and European private international law: Perspectives de jure condendo', International Trade Law, fasc. 2 (2020), p. 477 et seq.
[85] Rabitti, M. and Paglietti M. C. 'A matter of Time. Digital-Financial Consumers Vulnerability in the Retail Payments Market', European Business Law Review, 33, no. 4 (2022), pp. 581-606
[86] Durovic, M. and Janssen, A. 'The Formation of Blockchain-based Smart Contracts in the Light of Contract Law, European Review of Private Law, 6, pp. 753-772 (2019)
[87] Musio, A. 'The unfinished story of the evolution of the contract between technological novelties and consequent regulatory requirements', NGCC, fasc. 1 (2021), p. 234 et seq.
[88] Remotti, 'Blockchain smart contract. A first framework', Observatory on Civil and Commercial Law, fasc. 1 (2020), p. 200 ff
[89] Cuccuru, P. 'Blockchain and contract automation. Reflections on smart contracts', NGCC 2017, II, p. 111 et seq.
[90] Werbach, K. 'Trust, but Verify: Why the Blockchain Needs the Law', 33 Berkeley Tech. L.J. 489 (2018), pp. 545-546
[91] Banca d'Italia, 'Disposizioni in materia di trasparenza delle operazioni e dei servizi bancari e finanziari' (2019)
[92] Banca d'Italia, 'Fintech in Italia. Indagine conoscitiva sull'adozione delle innovazioni tecnologiche applicate ai servizi finanziari' (2017)
Fact-finding survey on the uptake of technological innovations applied to financial services' (2017)
[93] Sirgiovanni, B. 'Smart contracts and consumer protection: the translation of natural language into computer language through legal design', Le Nuove Leggi Civili Commentate, p. 214 et seq. (2023)
[94] Carron, B. and Botteron, V. 'How smart can a contract be?', in D. Kraus, T. Obrist, O. Hari (eds.), Blockchains, Smart Contract, Decentralised Autonomous Organisations and the Law, Cheltenham, UK-Northampton, MA, USA (2019)
[95] Cardarelli, G. 'Beyond Oracles – A Critical Look at real-world Blockchains', Future Internet (2022) p. 175
[96] Curran, B. 'What are Oracles? Smart Contracts, Chainlink & 'The Oracle Problem', available on medium.com (2019)
[97] Longobucco F., Utopia of an autonomous Lex Criptographi(c)a and responsibility of the jurist, Edizioni Scientifiche Italiane (2023), p. 31 et seq.
[98] Allen, D., Lane A., Poblet, A. 'The Governance of Blockchain Dispute Resolution', Harv. Negot. L. Rev. (2019), pp. 75 et seq.
[99] Howell, B. and Potgieter, P. 'Uncertainty and dispute resolution for blockchain and smart contract institutions', Journal of Institutional Economics (2021), pp. 545–559, spec. p. 547
[100] Battaglini, R. 'Dispute resolution and decentralized platforms', Theory and Criticism of Social Regulation (2021) pp. 77-91
[101] Gabuthy, Y. 'Blockchain-Based Dispute Resolution: Insights and Challenge, in Games' (2023), pp. 14 et seq., available at https://doi.org/10.3390/g14030034

[102] Ortolani, P. 'The Judicialisation of the Blockchain', available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3230880 (2018), pp. 1-39, spec. pp. 30-31
[103] ESMA, 'The Distributed Ledger Technology Applied or Securities Markets' (2017) https://www.esma.europa.ew/sites/default/files/library/dlt_report_-_esma50-1121423017-285.pdf
[104] Poncibò, C. 'Smart contract. Applicable law profiles and choice of court', Blockchain and Smart contract (eds.) R. Battaglini and M. T. Giordano (2019), pp. 349-350
[105] Lehemann, M. 'National Blockchain Laws as a Threat to Capital Markets Integration', Uniform Law Review, Volume 26, Issue 1 (2021), pp. 165 et seq.
[106] Dell'Erba, M. 'Demystifying Technology, Do Smart Contracts Require a New Legal Framework?', available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3228445 (2018) p. 42
[107] Sirgiovanni, B. 'Smart contracts and consumer protection: the translation of natural language into computer language through legal design', Rivista Le Nuove Leggi Civili Commentate, No 1 (2023), p. 214 et seq.
[108] Rühl, G. 'Blockchain Networks and European Private International Law', available at https://conflictoflaws.net/2018/blockchain-networks-and-european-private-internationale-law/?print=print (2018)
[109] Guillaume F., 'Aspects of Private International Law Related to Blockchain Transactions', in Kraus, Obrist and Hari (eds.), Blockchain, Smart Contracts, Decentralised Autonomous Organization and the Law, Cheltenham/Northampton (2019), pp. 49-82
[110] Vahabava M., 'Smart Legal Contracts. Comparative reflection points', Journal of Communication Sciences and Legal Argumentation – A. XIV, No 2 (2022), pp. 114-127.
[111] Poncibò C., 'Il Diritto Comparato e la Blockchain', Edizioni scientifica Italiane (2020)
[112] Benedetti A.M., 'Contract, algorithms and transnational civil law: five questions and two scenarios', Rivista di Diritto Civile, No 3 (2021), pp. 411-326, spec. p. 426
[113] Lehemann M., 'Who Owns Bitcoin? Private Law Facing the Blockchain, Private Law Facing the Blockchain, 21 Minn. J.L. Ski. & Tech. 93 (2019), p. 102
[114] Delgado De Molina Rius, A. 'Smart Contracts: Taxonomy, Transaction Costs, and Design Trade-offs', in J. G. Allen and P. Hunn (eds.), Smart Legal Contracts: Computable Law in Theory and Practice, Oxford (2022), p. 107 et seq. and, spec., p. 121 et seq.

PART II – SMART CONTRACTS: TECHNOLOGICAL PROFILE

Introduction

In order to address the legal questions surrounding smart contracts and smart legal contracts, it is essential to examine the solutions that technology provides and the problems it solves. This analysis can then inform the identification of minimum standards that guarantee the safeguards imposed by current legislation and the protections that technology allows to be strengthened, for example, in terms of certainty and transparency. In fact, technology can perform regulatory functions (*ex ante*) or control and supervision functions (ongoing and ex post) by integrating the activity of the regulator. These functions are defined in regulation by technology [74, 75] or rule of code [76, 77] doctrine.

This second part of the document focuses on the technological profiles, respectively of blockchain (Section I) and smart contracts (Section II).

In particular, in Section I, the elements (or properties) of the technology that are considered relevant for the development and execution of smart contracts are illustrated from the perspective of blockchains.

Section I – Taxonomy for the analysis of blockchain platforms

1. Introduction

Blockchain technology enables the decentralisation of transactional systems through the application of a decentralised protocol executed on a peer-to-peer network.[89] The protocol associated with the blockchain defines the rules for updating the shared ledger and ensures the security and integrity of the data stored without the need for the intervention of a trusted third party.

The data stored in the register can represent any change in status and thus they are organised into transactions. Transactions are ordered chronologically and grouped into blocks.[90] Each block contains a group of transactions that have been cryptographically signed by one or more participants in the protocol. The organization of the block and the structure of the transaction are dependent on the specific blockchain technology in question. A transaction is typically constituted by the recording of an event (for instance, a transfer of value between two or more users, the invocation of a smart contract) that results in a change in the status recorded in the ledger. In order to guarantee the immutability property, each block is connected to the previous one in a cryptographically secure manner. This is technically accomplished by means of a pointer to the previous block, which is calculated by applying a hash function[91] to the contents of the current block (which includes the pointer, hash, to the previous block). Consequently, the result of the hash function is an integral part

[89] Networking type where each node can communicate directly with another node without going through a centralized point. (Source NISTIR 8202).

[90] The organisation of the transaction ledger in the form of cryptographically linked blocks characterises blockchains within the broader context of 'Distributed Ledger Technologies' (DLT). Although in this document we focus on blockchains and explicitly refer mainly to them, net of this difference, what is described has general value also for DLTs. Exceptions, if not easily deduced from the context, are reported.

[91] A hash function is equivalent to the application of a mathematical algorithm to input data (of any type and size: files, texts, images, etc.) able to produce in output a synthetic and relatively unique representation of the input data. (Source NISTIR 8202).

of the block. As each block is cryptographically linked to the previous block, by means of the hash pointer, it follows that a change to one block invalidates the integrity of all subsequent blocks.

Blockchain technology represents a fundamental shift in the way information and infrastructure are managed. It is therefore essential, that the protocol adopted is able to ensure the security, integrity and efficiency of the system. In the event that a blockchain is required to operate on a large scale and process a large number of transactions, it is of the utmost importance that this is done without compromising the availability and integrity of information.

In recent years, blockchain has been applied in numerous sectors [1, 2, 3], as evidenced by the numerous studies and report that have been published on the subject. The interest of industry and academia has fuelled the offer of new layer 1 protocols and platforms.[92] To date, there are dozens of blockchains, each with its own characteristics and trade-offs [8]: some encourage scalability, while others aim to optimize the development of decentralized applications or provide advanced privacy mechanisms.

Despite the potential for process improvement that blockchain technology and smart contracts have proven to guarantee in different domains (for example, supply chain management [62], finance [63], health care [64]), their application in regulated contexts is still limited and has ample room for improvement, both in technical terms (for smart contract codes) and in transparency (for smart legal contracts). Indeed, the current regulation framework is fragmented, which makes it challenging to adopt the technology on a large scale, particularly in the context of government and institutional applications [4].

The introduction of new technologies, particularly those that facilitate advanced forms of programmability, has enabled the incorporation of rules. For instance, those developing smart contracts may direct financial flows, restrict fundamental rights, guarantee the immutable execution of contracts on time, offer simultaneous performance and counter-performance, utilise a national jurisdiction, or even employ preventive forms of conciliation, such as oracles [5]. Consequently, the supervisor is unable to limit thier intervention to a *posteriori* analysis and must be able to verify the process at the time it is being drawn up. Furthermore, the characteristics of smart contracts can also be influenced by the particular blockchain technology they use. Therefore, it is considered useful to preface the analysis of the taxonomy of smart contracts with an analysis of some characteristics of blockchains.

Moreover, blockchain technology remains a tool of complex understanding to this day; as discussed in BOX 5 on the state of the art (see below); defining strategies for analysis and comparison between the different platforms is an open topic in the scientific field.

[92] It means a basic blockchain network that defines the essential functionalities of a blockchain protocol: executing transactions on the blockchain and updating a shared ledger. Different layer-1 networks differ in consensus protocol, degree of decentralization, security properties, efficiency, interoperability, technical functionality, ledger structure, and scalability. Instead, we speak of a 'layer 2' framework or secondary protocol in the case of technological solutions built on top of an existing blockchain; this is often done in order to change some of its characteristics (e.g. scalability, decentralisation, visibility of information).

BOX 5 - The state of the art

To date, following the analysis conducted on the state of the art, it would appear that there is no single shared tool available to compare different blockchains and analyse their peculiarities in terms of, for example, the security and efficiency of platforms. Indeed, a significant proportion of the studies conducted to date have focused on specific characteristics, with the objective of identifying the advantages and disadvantages of a blockchain in comparison to others or to illustrate specific use cases on an experimental basis. For example, [79] presents a taxonomy of blockchains and an ontology of key features, yet it lacks an analysis of the security aspects of key components such as the consensus protocol. Other studies [8] propose a classification of the fundamental properties of the blockchain and their trade-offs, focusing on general security and performance features. A qualitative approach has also been employed by those who, in evaluating blockchain consensus protocols, have concentrated on the security and performance aspects that they can guarantee, without however assessing the impacts on the infrastructure, such as decentralisation and privacy [9]. Others propose [10] a qualitative analysis of the consensus protocols used by permissioned blockchains.[93] Some researchers [11, 12] have conducted a fairness/adequacy analysis with respect to safety and liveness properties, focusing their study on a small set of permissioned blockchains. In contrast, those who [13] have proposed a quantitative analysis have offered experimental benchmarking analyses of the performance of the Ethereum and Hyperledger Fabric blockchains, without evaluating other fundamental properties such as security and decentralization. Finally, others [14] analysed the efficiency of permissioned blockchains in financial use cases, but did not assess their requirements in terms of, for example, security and reliability.

The objective of the subsequent in-depth analysis is to identify a taxonomy of parameters useful for answering the following question:

What are the principal parameters or requirements that should be taken into account in order to analyse the characteristics of a blockchain platform?

The identification of a set of main parameters enables the analysis of blockchain platforms, which could prove to be a useful tool in the development of regulation for this technology. This will enable the construction of a taxonomy of smart contracts in terms of technology (Part II, Section II), which will in turn be a prerequisite for the subsequent definition of guidelines.

2. Methodology

The functioning of a blockchain relies on the convergence of multiple domains within the field of computing and mathematics, including cryptography, distributed systems, and telecommunications. The objective of blockchain technology is to guarantee the authenticity and treaceability of transactions even in the absence of mutual trust between the parties involved, through the implementation of a shared protocol and a decentralised technological infrastructure.

Despite the promised benefits of the technology, the implementation of specific applications has revealed dependencies among the typical properties of blockchain technology that lead to necessary compromises [8, 15]. In summary, the improvement of one property can lead to the worsening of another property. For example, there is a trade-off between the availability of stored information and the consistency guaranteed within distributed ledgers [8]. Consequently, it is crucial when developing applications based on blockchain (or smart contracts) to keep these trade-offs in mind.

[93] Permitted blockchains are a particular type of blockchain where the nodes that can upgrade the blockchain are limited, known, and authorised. Please refer to section 3.1 'Network architecture' for further information.

Characteristics of Smart Contracts

This document outlines the main qualitative characteristics (or parameters) that are fundamental for the analysis of blockchains. The choice of parameters was based on the state of the art (for example, [8, 9, 10, 12, 16, 17, 18, 19]) and the qualitative aspects deemed useful to represent blockchain technology. This document presents an analysis of three main verticals:

1. technical characteristics;
2. economic model;
3. ecosystem and on-chain data.

The following sections present a more detailed examination of the individual verticals, which serves to reinforce the rationale behind the selection of quality parameters. Additionally, a methodological approach is proposed as a potential framework for the acquisition of information necessary for the analysis of a specific blockchain technology in relation to the aforementioned parameters.

3. Technical characteristics

This section considers the architectural aspects of blockchain networks, the parameters of efficiency – understood as scalability and decentralisation – and security, as well as flexibility – understood as computational capacity (programmability) obtained through the use of smart contracts, system configurability and interoperability with other platforms.

3.1 Network architecture

A blockchain network is a decentralized system of nodes connected through the network in order to:

- maintain a shared ledger on each node of the network;
- create and validate transactions and blocks;
- update the ledger through specific rules (or protocol).

In order to analyse the robustness of the network model proposed by a blockchain, it is necessary to hypothesize unsafe communication channels, which can be corrupted or prone to failure. A partially synchronous network [20] is considered to provide for asynchronous communications where messages are delivered within a certain time limit. This model is recognized by distributed systems theory as a reference model for the simulation of an Internet network.

The network nodes should be separated between client nodes and validator nodes. The role of validating nodes is to verify the validity of transactions and blocks and to participate in the mechanism for selecting the next block to be added to the chain (consensus protocol). Client nodes are responsible for maintaining the state of the blockchain and for providing users with interfaces through which they can interact with the blockchain protocol functionalities, such as sending transactions or invoking smart contracts.

It should be noted that client and validator nodes could operate on networks with different levels of visibility [21, 22, 23, 24]. These levels are defined as follows:

1. Public network: there is no restriction on the visibility of the information stored on the ledger*;*
2. Private network: only a small group of authenticated participants can access the information stored in the ledger.

Each network model is also characterized by two distinct permission levels:

1. Permissionless**:** nodes are not required to have permission to join the network and participate in the consensus;
2. Permissioned**:** only authorised nodes are enabled; they participate in the consensus and, therefore, in the validation and writing of the information on the ledger.

The network architecture has implications for the activities a blockchain can guarantee. Public blockchains facilitate the transparency of transactions and data, which are inherently non-confidential. This is a fundamental aspect of the design of public blockchains, even in the presence of technical solutions such as data masking. In contrast, permissioned blockchains are characterised by a restricted consortium of actors managing their infrastructure and the underlying ledger. In permissioned blockchains a degree of trust is required with respect to the security of the system. If the core group were corrupted the entire blockchain would be corrupted. [8, 15, 32].

3.2 Security, scalability, decentralization

The three fundamental properties of blockchains are commonly described as security, scalability and decentralisation. Vitalik Buterin, the creator of Ethereum, developed the 'Scalability Trilemma' [25] to illustrate the need to balance the trade-off conditions of security, scalability and decentralisation in a blockchain network. The trilemma posits that it is not possible to achieve an equal level of priority for all three properties simultaneously: a compromise must be reached at the expense of one of the three dimensions.

The subsequent sections present the three properties and the characteristics of the blockchains that influence them in detail.

3.2.1 Safety Parameters

This section outlines the principal profiles to be analysed in order to assess the security aspects of a blockchain. The fundamental security properties of confidentiality, integrity, availability and consistency are referenced [8, 26]. Confidentiality is defined as the ability of a system to prevent the disclosure of information to unauthorised users. Integrity is the ability of a system to maintain the transactions and information contained in the ledger. Availability is the probability that a system is able to receive update requests (and therefore process transactions) at any time. Consistency is defined as the absence of divergent ledger states between the blockchain nodes. The following sections will analyse in detail the characteristics of blockchains that affect security properties.

3.2.1.1 *Confidentiality*

The prevailing narrative (37) posits that blockchain, by virtue of its immutable and decentralised nature, is incompatible with confidentiality requirements and the European regulation on the protection of personal data (GDPR). Actually, the GDPR stipulates that: (i) The data must be under the control of an individual who is responsible for them; and (ii) at any time these data can be modified or removed from the reference information system. It is not the case that any type of blockchain is guaranteed to possess these properties. However, it is necessary to consider the potential trade-offs that may arise when addressing the issue of personal data protection in the context of blockchain technology.

Permissioned blockchains offer a controlled system where only authorised and authenticated parties have access to information. This type of network is distinguished by a defined and limited set of validators who exercise control over the network, the shared ledger, and the information contained therein. In a permissioned system validators are able to modify and, in certain circumstances, even

delete transactions from the ledger as they effectively control the infrastructure. This has an impact on the security of the system. On the one hand, it allows a small group of participants to invalidate the immutability property of data in the blockchain. On the other hand, it makes it easier to correct any clerical errors. Consequently, the trust model in permissioned blockchains must take these characteristics into account and, where appropriate, be more stringent than in permissionless networks where trust is distributed over the entirety of the decentralised network [15].

In public networks (permissionless or permissioned), any entity connected to the network is able to read the information written in the ledger. In these systems, the immutability of transactions is ensured by a larger decentralised set of validators. In particular, in permissionless cases, the deletion or modification of data from the ledger would require a global consensus and the implementation of a 'hard fork' (see section 3.2.1.4.2). An alternative to the implementation of hard forks is the use of application approaches, in which an entity is able to 'invalidate' a data item by providing an 'updated' version of it. However, given the immutability of the ledger, that information would remain tracked and thus visible.

Compliance with regulations on the protection of personal data in the context of public blockchains can be ensured through data-masking and information anonymization mechanisms. This would result in the creation of an encrypted trail of the data on the ledger without the necessity of verifying its extent. Some platforms have proposed alternative solutions including the utilisation of advanced cryptographic techniques for the right to be forgotten on-chain (RTBF), zero-knowledge mechanisms [38], implementation of 'controlled access platforms' or private sub-networks.

3.2.1.2 *Integrity*

The integrity property of a blockchain indicates that transactions and blocks adhere to the validity rules defined by the blockchain protocol. For example, the protocol may require that a transaction transferring funds from one user to another is only valid if it spends a specified amount of available funds (either from a previous transaction or from the user's balance sheet) and is signed by the user who has the authority to spend those funds.

Typically, the validity rules defined by the blockchain protocol can make use of cryptographic functions, such as hash functions and digital signatures. Hash functions permit the derivation of control values from data (typically an entire block), thereby facilitating the identification of any unauthorised alterations to the same. In contrast, digital signatures permit the authentication of data from a sender, thereby providing a guarantee of non-repudiation. The maintenance of integrity is of paramount importance in ensuring the security of a blockchain [8].

3.2.1.3 *Availability*

In general, the availability of a system is defined in terms of probability that the system will operate correctly at every moment in time. Given the replicated nature of the blockchain, each node holds a copy of the entire ledger and, therefore, the data is available as long as there is at least one active node with an updated copy of the ledger. However, in order for a blockchain to be considered available, it is also necessary for the system to be able to process new transactions to be added to the ledger. This is guaranteed by the underlying consensus algorithm which assumes the existence of an adequate number of 'validator' nodes with different requirements depending on the specifics of the algorithm. In protocols such as Proof-of-Work (PoW), the presence of a single node capable of producing blocks is potentially sufficient, although in this limit scenario no decentralisation would be guaranteed. In other Proof of Stake (PoS) based protocols, consensus sets limits within which the protocol ensures availability.

3.2.1.4 *Consistency*

All nodes of a distributed ledger (in this case, a blockchain, maintain a local replica of the ledger; consequently, all nodes must be synchronised and must agree on the changes to be applied to the stored data (i.e. status) in order to ensure the consistency of the ledger. A consensus protocol is used for this purpose.

In general, a consensus protocol enables the managment of negotiations between the nodes and the achivement, within a certain period, of an agreement on how to update the shared state, even in the presence of node or network failures [10, 18, 27]. Specifically, in blockchains, consensus is used to define the order of transactions to be added to the ledger and to ensure that network nodes converge on the same state by solving (probabilistically or deterministically) problems such as double-spending [10].

In literature [11, 27, 29, 30], the correctness of a consensus protocol is evaluated in terms of its safety and liveness properties. Safety represents the ability of the protocol to execute correctly even in the presence of adverse conditions (failures). Liveness represents the ability of the protocol to progress and terminate correctly, even in the presence of failures.

Some authors [59] argue that the properties of safety and liveness are related to the CAP theorem[94], which is defined in distributed systems [28]. For the purposes of this work, the CAP theorem can be expressed as follows: it is impossible for a distributed, shared, replicated and synchronized ledger between multiple nodes to guarantee, in the presence of communication errors between the nodes, the simultaneous execution of both reading and writing atomic data and the availability of the most recent data writings. In CAP theorem, consistency is linked to the property of safety: any replication of the data will always be correct (i.e. updated). Availability, on the other hand, is linked to the property of liveness: at the end, every request to update or read the data will eventually receive a response. Consequently, the CAP theorem posits that it is not possible for any protocol that implements a log with atomic read and write operations to simultaneously guarantee safety and liveness in a partitioned network.

The following section will examine two concepts related to consistency in blockchain technology, namely finality and forks.

3.2.1.4.1 *Finality*

The concept of finality (or purpose) is a fundamental property that is introduced in order to determine the differences between various blockchain consensus algorithms. This property is applied to ledger blocks (and transactions) and defines the ability of a system to consider an immutable and irreversible transaction within the ledger. Two types of finality are distinguished [23]:

> ***Probabilistic finality:*** indicates that the probability of a transaction being valid and, therefore, irreversibly part of the ledger, increases with the number of new blocks following the block containing the transaction. Proof-of-Work (PoW) is a consensus algorithm in which the computational work required to modify or remove a transaction from the ledger is inversely proportional to the number of blocks that follow the transaction. In contrast, in some versions of Proof of Stake (PoS), the objective is pursued by economically discouraging the behaviour of

[94] The CAP theorem was introduced as a trade-off between consistency, availability and partition tolerance in distributed systems where a service is implemented and provided by multiple nodes. Consistency refers to the ability of each node to provide a correct response to each request made. Availability means the ability of the service to always provide an answer for each request made. Tolerance to network partitions means that the service continues to operate even in the face of failures in communication between nodes.

malicious actors. However, the probabilistic purpose does not offer strong guarantees of consistency, as the transactions of a ledger could be replaced with others in the event of conflicts.

Deterministic finality: ensures that a transaction is valid immediately after it has been inserted into a block of the blockchain. The Practical Byzantine Fault-Tolerant (PBFT) protocol and its variants [18, 33] are part of the consensus algorithms with deterministic purposes. In such protocols, validators vote on the next block to be added to the chain; when a certain number of votes is reached (quorum), each node in the network updates its ledger simultaneously. A transaction, if approved by a quorum of validators, is immediately part of the ledger irreversibly. Assuming an adversarial system where nodes can be faulty or corrupted (i.e., prone to Byzantine failures), distributed systems theory sets the minimum value of the approval quorum at 2/3 of the total number of validating nodes in the network [33].

It is important to note that additional dimensions are employed in the literature to classify consensus algorithms. For instance, in [24], a distinction is made between lottery-based and vote-based consensus algorithms. The former involves the selection, based on a specific criterion of the protocol itself, of a validator node to propose the next block to be added to the chain. In contrast, the latter employs a more traditional approach based on rounds of votes. PoW and PoS are representative examples of lottery-based consensus algorithms, whereas the proposed by the forking technique (PBFT) is a representative example of a vote-based consensus algorithm.

3.2.1.4.2 *Fork*

The consistency property is lost in a blockchain system in the event that several versions of the shared ledger exist at the same time. Consequently, the different nodes of the blockchain network are not synchronized on the same state as the ledger in a certain instant of time. This scenario gives rise to the phenomenon of forks. In general, forks can be accidental or intentional.

In the case of accidental forks, the consensus mechanism employed by the specific blockchain technology is typically the cause. In the event that two or more new, valid blocks are created almost simultaneously by validator nodes, the blockchain will be divided into two or more valid branches. In this instance, the consistency of the blockchain is temporarily breached [31], rendering it susceptible to issues such as double-spending [10, 31]. Typically, conflicts between the different versions of the chain are resolved within a certain time window. This is achieved through the selection and consequent extension by the nodes of the network of a single branch formed from the chain of blocks [32]. The selection of the branch is conducted in accordance with the specific criteria inherent to the particular blockchain technology. For instance, in the case of Bitcoin, each node is required to select and extend the branch that leads to the definition of the blockchain for which the most work has been done. This criterion is known as the 'longest chain criterion' [24]. In this scenario, the consistency is referred to as final (or eventual consistency) because, over time and with the occurrence of updates, all the replicas of the register will converge towards the state of the register defined in the longest chain of the blockchain. In the event that a blockchain is unable to resolve potential fork events, the security of the blockchain is compromised. This is evidenced by the findings of [10, 32, 60].

The consensus protocols are capable of maintaining the property of safety at any given moment in time, thereby preventing accidental forks even during periods of adverse network activity. This property is secured at the cost of grid stall periods, as outlined in references [33, 34]. Conversely, there are consensus protocols that prioritize liveness, thus preventing stalemates by adopting models of final consistency simultaneously, though this may result in the occurrence of accidental fork events [16, 31, 35].

Intentional forks, however, occur during planned updates of the blockchain protocol (software). In particular, the upgrade process necessitates that the network nodes update the software

asynchronously and autonomously, which can result in a network fork (for instance, in the event that a protocol parameter or block structure is altered). In such instances, two distinct types of forks are identified, namely soft forks and hard forks [36]. In the case of soft forks, the protocol update does not result in the loss of backward compatibility, and therefore is not considered a mandatory update. Network nodes continue to interact with each other, even with different versions of the protocol. In contrast, in the case of hard forks, updating the protocol results in the loss of backward compatibility, which is why the update is considered mandatory. Nodes that fail to upgrade to the new version will be unable to interact with the rest of the network, but will continue to operate on an outdated ledger.

3.2.1.5 *Quantum Resistance*

Quantum computers are expected to l be capable of solving complex computational problems in a significantly shorter time than traditional computers. This represents a significant challenge to the foundations of modern cryptography and blockchain security. For instance, a quantum computer could be employed to derive a user's private key or even to forge blocks arbitrarily with the intention of manipulating the shared ledger.

The asymmetric cryptography schemes currently in use for computing digital signatures are insufficiently secure to withstand attacks from quantum computers. Nevertheless, the NIST is currently engaged in the development of new standards for the definition of post-quantum cryptographic algorithms, which will facilitate the definition of digital signatures that are resistant to attacks from quantum computers [65]. The majority of the proposals currently under evaluation by NIST are based on lattices, a mathematical abstraction that appears to be capable of defining computationally complex problems even for quantum computers [39].

Quantum computers represent a pioneering technology that is not yet ready for large-scale deployment. Consequently, they do not pose a significant threat at the writing time of this document. Nevertheless, the parameter of quantum resistance is a useful tool for evaluating the robustness and security of a broad-spectrum blockchain platform over the medium to long term[95].

3.2.2 Scalability parameters

In distributed system theory, scalability is defined as the ability of a system to maintain an adequate level of performance as network complexity (number of nodes) or workload (number of operations required) increases [40]. A number of metrics can be employed to assess the performance of a system. These include throughput and latency [41].

Throughput is a metric that quantifies the number of operations completed by the system within a specified time interval. In the context of blockchains, throughput is defined as the maximum number of transactions per second (TPS) that the blockchain is able to confirm[96].

Latency measures the time required by the system to complete an operation. For blockchains, latency is measured as the average time required by the protocol to process and finalize transactions. This is

[95] Crypto agility is often referred to as a possible mitigation approach to the threats posed by quantum computing. This term indicates the ability of a computer system to implement alternative encryption methods, thus making them able to react quickly to any cryptographic threats. In this sense, a 'crypto-agile' blockchain should make it possible to modify signature and/or hash algorithms in an easy way, also outlining a change management path that makes it possible to manage transitional periods in which different categories of algorithms coexist within the same protocol.

[96] A transaction is considered confirmed when 'purpose is achieved', i.e. the property that guarantees its immutability and irreversibility on the shared ledger. Please refer to section 3.5 for further details on the purpose.

usually measured as the difference between the time when a transaction is considered finalised and the time when the transaction was sent by a user to the client node ('time to finality').

In recent years, platforms called 'layer-2' [80] have been proposed, alongside a blockchain with the aim of maximising transaction throughput and latency, and reducing costs. These solutions offer blockchains parallel to the already known 'layer-1' blockchains (e.g. Bitcoin and Ethereum) which constitute a fundamental infrastructure. The most well-known 'layer-2' approaches are those based on payment channels (i.e. Bitcoin's Lightning Network [81]), those based on 'optimistic rollups' in which transactions are considered valid and possibly verified using the underlying layer-1, or 'zero-knowledge rollups' based on zero-knowledge evidence [38].

3.2.3 Parameters of decentralization

A decentralized system is defined in the literature as a system in which the nodes of the network are not managed by a single authority but by multiple independent actors [15].

The number of validating nodes participating in the consensus helps determine the level of decentralization of the network. In general, permissioned networks have a lower level of decentralization than permissionless networks because they generally use a predefined and limited set of validators. In contrast, in a permissionless network, all network nodes can act as validators, thereby increasing the level of decentralization of the network [8, 15].

As described in the Scalability Trilemma [25], ensuring scalability and decentralisation in a blockchain network is a challenging task. Increasing the number of validators makes it more complex to reach consensus and ensure tolerance to Byzantine failures simultaneously [10, 18, 33].

In the PoW consensus, the consensus is reached probabilistically, utilizing the computational resources available to the network nodes for the resolution of a complex mathematical puzzle. Validators with greater computational power are more likely to be selected as proposers of the next block [31, 42]. In other systems, based on PoS, validators must deposit money (stakes) to be selected to propose a block. In this case, the block proposer can be elected in two ways:

1. A deterministic approach is employed, whereby the proposing validator is selected based on the proportion of stakes deposited.
2. A Probabilistic approach is used, whereby the proposing validator is chosen at random.

In both solutions, the deposited stake plays a fundamental role as it increases the chances of being selected as a proposing validator. The degree of decentralization of a blockchain system is then evaluated according to two dimensions [8]: (i) the number of validating nodes participating in the consensus, (ii) the distribution of validation power, i.e. the distribution of resources (e.g., computational as in PoW). Dimension (ii) is relevant in the context of lottery-based protocols of which PoW and PoS are part. In the following discussion, we adopt this perspective.

3.2.3.1 *Number of validator nodes*

In permissioned blockchain networks, the number of validator nodes is typically a predefined set of nodes. In contrast, in permissionless networks all nodes must be able to update the shared ledger by participating (explicitly or not) in the consensus protocol. In PoW, validators are those nodes that participate in the process of mining, which is solving the cryptographic puzzle. In PoS, validators are all those nodes that are registered to participate in the consensus. The greater the number of validating nodes in a blockchain network, the more decentralized the system can be.

3.2.3.2 *Distribution of validation power*

It is essential to identify within the system how the validation power is distributed between validator nodes. In PoW, for example, miners with greater computational resources are more likely to solve the mathematical puzzle and thus more likely to propose a block. Consequently, PoW is conditioned by a mechanism called pooling, whereby multiple miners tend to combine their computational resources in order to enhance the probability of producing new blocks. This results in a more centralised network, with the production of blocks entrusted to a few miners. This exposes the network to the 51% attack, whereby the majority of the network's computational power is in the hands of a group of miners [43].

A comparable process can be observed in PoS, whereby validators and other stakeholders can combine their stakes to enhance the probability of producing blocks. This is known as 'pooled staking'. [6]. In PoS, some penalty rules can be defined, whereby a portion of the stake deposited by the validator (or the entire amount) is deducted in the event of malfunctions or malicious behaviour. This mechanism is referred to as 'slashing'. It is, therefore, important for a staking pool to minimise the risk of slashing. This can be achieved by offering an efficient and reliable validation service in order to attract more stakes and thus maximise the chances of producing blocks. As in mining, pooling introduces risks of centralisation. As the stake is distributed among a few groups of validators, security risks such as the attack increase by 51%.

BOX 6 – PoS systems

It should be noted that not all PoS systems are identical, and in certain implementations, the risk of attack at 51% of the stake can be mitigated. In particular, PoS-based blockchains can adopt different election approaches, including:

- Bonded Proof of Stake: a set of nodes is designated as validators, with stakeholders depositing their stake in the protocol to acquire voting rights and indicating one or more validators. Nodes with multiple votes are elected as validators and then allowed to produce new blocks. The distribution of the stake is concentrated on a group of validators who are willing to deposit the required stake.

- Delegated Proof of Stake: the validators are elected through a stake-based voting system. The stakeholders express their preference by delegating the stake to a validator. The validators with multiple delegated stakes are elected. A relatively small number of validators are usually chosen in order to maintain a balanced trade-off between security and scalability. The distribution of the stake is limited to the number of validators elected [60].

- Pure Proof of Stake: each stakeholder has the authority to participate in the consensus as a validator and thus, be elected as a proposer of a new block. Validators are elected randomly using a cryptographic mechanism that does not require high computational costs. This type of PoS does not provide for mechanisms of aggregation of the stake to increase the chances of being elected as proposers of a block. Furthermore, it does not pose economic or computational barriers to participation in the consensus. The distribution of the stake is broad across all stakeholders [60].

3.2.3.3 *Fairness*

In numerous blockchain platforms, access barriers for validators could potentially impede their impartiality. To illustrate, in PoW-based systems, the access barrier is the cost of the hardware and energy required to participate in the consensus. In contrast, in some PoS protocols, the barriers are represented by the minimum amount of stakes that must be deposited in order for a node to be considered a validator. To enhance fairness, some PoS solutions provide 'liquid staking' features, wherein validators receive an alternative asset representing a one-to-one *ratio* of the staking tokens deposited. This provides a form of 'liquid' staking in which assets are locked, but their counter-value remains available for use as collateral.

The lower the access barriers, the greater the possibility of having a highly decentralised network.

3.3. Flexibility

In the previous sections, we presented the infrastructure features and the security, scalability and decentralization parameters that differentiate blockchains. This section will analyse the differences between them from an application and usability perspective. The former will be considered in terms of the ability to build applications on the blockchain, while the latter will be evaluated in terms of ease of use and interaction between the blockchain and other systems. In particular, we identify three key characteristics of blockchain flexibility: programmability, network configuration, and interoperability.

Blockchain programmability refers to the ability to implement and run decentralized applications through the development of smart contracts.

Blockchain network configuration refers to the ability to customize the network according to specific rules.

Blockchain interoperability refers to the techniques available to facilitate interaction between different systems.

3.3.1 Programmability

Programmability is defined as the capacity of a blockchain to interpret and execute programs (smart contracts) in a decentralized manner. The execution of a smart contract is deterministic, and the result of the execution, stored immutably on the blockchain, depends on the input parameters and the state of the blockchain. Smart contracts permit the definition of business logic for the approval of transactions or for updating the status of the ledger itself. They are adopted for the creation of decentralised applications. A more detailed examination of smart contracts is provided in Section II.

3.3.2 Configurability

Each blockchain system is characterised by a set of parameters that influence its operational efficiency, usability and security. For instance, it may be necessary for a validator node to be able to determine the memory space made available to manage the queues of incoming transactions (for example, in the case of limited memory of the node), or even configure protocol parameters such as the size of the blocks produced and the frequency. In this case, flexibility is impacted, and it is necessary to distinguish the trade-offs between blockchains that provide a configurable execution environment, and therefore more flexible, compared to systems that do not allow 'custom' configurations of the node, in favour of a more stable and consolidated system [8].

3.3.3 Interoperability

Several blockchain platforms are emerging and gaining a significant presence in the market, attracting users and new decentralized applications. It is crucial to enable interoperability mechanisms that permit users to operate on different blockchain platforms (e.g. transferring assets from blockchain A to blockchain B, or invoking a smart contract from blockchain C) [49].

BOX 7 - Interoperability and Bridges

To date, the objective of interoperability is typically achieved through intermediaries (known as notaries*)* operating cross-chain atomic exchanges. Intermediaries are typically represented by smart contracts, or pairs of smart contracts defined on the relevant blockchains [50]. These components act as centralised or semi-centralised oracles and are commonly referred to as 'Bridges'. Given that blockchains are typically incompatible[97]with one another, the problem of exchanging an asset or information from one platform to another is not straightforward to solve. In order to enable this type of operation, Bridges are in charge of 'deleting' (burning) information from the sending blockchain and creating (minting) the same information in the destination blockchain. This process is typically conducted through the use of smart contracts for burning and minting operations, in conjunction with a software component (middleware) between the two blockchains to certify the correctness of the process itself. Given their inherent complexity, Bridges represent a significant vulnerability in the security of cross-chain exchanges. Both the smart contracts and the middleware component may be susceptible to compromise. In 2022, Bridges were among the primary targets of blockchain attacks that resulted in the theft of hundreds of millions of dollars [51].

An alternative to notary bridges are interoperability solutions based on relays, i.e. systems run directly within a blockchain, capable of validating their status through the verification of validators' signatures, and communicating the latter to a second blockchain [52]. Relay-based mechanisms are convenient in extremely efficient networks, but verification can become a bottleneck if this involves one blockchain with longer confirmation times than another. This represents a limitation in the context of cross-chain market exchanges where transaction validation time is a key element [50]. However, alternative solutions exist in the literature that mitigate the problem of verifying the validation signatures of a blockchain using compact encrypted certificates [53]. These cryptographic objects can be readily exported from one blockchain to another, thus enabling the state to be verified directly on the blockchain (for instance, through a smart contract that interprets the certificate) in an efficient and reliable manner, without the need for external bridges.

3.4 Energy impact

As with any IT infrastructure, blockchains require a certain energy consumption in order to operate efficiently and securely. The system performs transaction processing and smart contract execution operations, necessitating the synchronisation of network nodes via the consensus protocol characteristic of the blockchain. It is evident that these operations necessitate the utilisation of computational resources, which subsequently results in an associated energy expenditure.

The Crypto Carbon Ratings Institute has developed a model for evaluating the energy consumption of blockchains. The study indicates that systems based on the PoW protocol (e.g., Bitcoin) consume

[97] There are blockchains that share native protocols and make cross-chain interoperability easier; however, these are currently exceptions in the market landscape. An example of this is Polkadot, a native multichain blockchain that allows cross-chain exchanges of data and assets between the 'specialised' chains that make up the Polkadot ecosystem.

approximately 120,000 [Gwh/year], require approximately 1,700 [kWh] for transactions validation[98] [78]. The PoW consensus protocol is inherently computationally intensive, necessitating the resolution of complex mathematical problems. This, in turn, results in a significant energy consumption. The study also indicates that other protocols, in particular those based on PoS, can be defined as more environmentally friendly because they do not involve a high computational load on the validators to perform the consent. The study demonstrates, for instance, that a PoS network (e.g. Ethereum) consumes approximately **2.7** [GWh/year] [54].

4. Economic model

In order to guarantee the optimal functioning of permissionless blockchains, it is essential to utilise a native token that enables the definition of financial incentives, thereby encouraging validators to act in an honest and reliable manner and to maintain the integrity of the network. Furthermore, the native token is employed to cover transaction execution costs, which are commonly referred to as commissions or fees. In other words, while the underlying economic model of permissioned blockchains is traditional and extrinsic in comparison to blockchain, the decentralised nature of permissionless blockchains necessitates the existence of an intrinsic incentive mechanism, which guarantees those who accept the role of validator a remuneration that makes it economically convenient to continue to perform it. This economic incentive is constituted precisely by the native tokens, which remunerate the validation activities.

4.1 Distribution of the native token

The technique of distributing the native token (if applicable) determines the initial value of the token, the role and incentives of the main actors involved in the project, and the methods chosen to sustainably incentivise the growth of the ecosystem over time. The initial distribution of tokens must be evaluated in relation to certain fundamental quantities, such as the theoretical maximum number of native tokens and the number of tokens actually in circulation at a given time.[99]

In PoS-based blockchains, the allocation of native tokens also determines the major stakeholders and, consequently, the individuals or entities able to exercise greater control over the network. To evaluate a blockchain, it is therefore necessary to analyse the methods adopted for the initial release of the token, the way in which the stakeholders were selected and the resulting allocation. For example, a blockchain that has entrusted the majority of tokens to an entity or a small group of beneficiaries is exposed to increased risks of stake concentration in a few entities, which can affect its security and reliability [55].

The most common initial distribution techniques – usually combined with each other with different weights – are:[100]

- *Allocation to insiders:* the distribution of tokens to internal beneficiaries (e.g. developers) that takes place before the distribution or public sale, may be associated with constraints that limit the transfer of the tokens received for a certain period.
- *Non-profit foundations:* the issuance of tokens to foundations in charge of promoting the platform and incentivising new projects in the ecosystem.

[98] 1700 kWh represent about the average annual energy needs of an Italian family of 2-3 people.
[99] In case of total issuance of the token at the time of launch, the two dimensions may coincide.
[100] The techniques described are usually used for non-native tokens. In the case of native tokens, it also includes the case in which they are not distributed at all, as they are assigned exclusively as a result of the mining / staking process (this is the case with Bitcoin).

- *Private sale:* the transfer to institutional investors (e.g. venture capital) based on bilateral agreements.
- *Airdrop*: the free distribution to the public (e.g. to users who have interacted with the blockchain during the testing phase).
- *Public sale*: sale through public auctions or ICOs.

4.2 Capitalisation

The term 'capitalisation' is used to indicate the total market value of all native tokens that have been put into circulation. This value is calculated based on the market price at a given time. This parameter can be employed in the analysis of the robustness and security of a blockchain, as it can influence the economic effort required to control the network. However, there exist diverse incentive models, which can result in varying effects on the capitalisation of the native token, depending upon the specific case in question.

For instance, in PoW-based blockchains, an adversary is less motivated to attack the network if the market value of the token is low. In this case, the costs of carrying out the attack (which do not depend on the value of the token but for example, on the cost of electricity) could be higher than any gains. Conversely, this line of reasoning may be overly simplistic if it is not also accompanied by considerations of the opposite direction, namely the attractiveness of a blockchain for potential miners. In fact, in the presence of an entry barrier to the 'mining market' consisting of the investments in hardware necessary to participate in the validation mechanism, a high value of the token constitutes an incentive to invest and therefore tends to contribute positively to the growth in the number of validators, a phenomenon that makes the execution of attacks more difficult. In the case of PoS consensus mechanisms, an attacker could exploit low market prices of the token to increase their stake and thus gain decision-making power on the network. This would allow them to act maliciously without suffering significant economic penalties (for example, in the case of slashing) [43].

4.3 Transaction Costs

The operation of blockchain platforms necessitates the payment of transaction execution costs by users. These costs are levied in order to compensate validators for carrying out the validation process. This process helps to make the network secure, preventing potential attacks or spam. The cost of transactions is divided into two categories: flat cost and dynamic cost.

- The flat cost is a fixed fee that applies to all transactions, regardless of their type (e.g. payment, execution of smart contracts).
- The dynamic cost is a variable fee that is determined by the type of transaction to be executed or by network congestion. In the event of high network usage, the costs can increase exponentially.

Some blockchains also allow users to incentivize the level of priority with which to request the registration of their transactions, adding an additional reward to the validators. This functionality, designed to enhance the overall efficiency of the system by enabling the execution of transactions at reduced costs (i.e. where the user does not have a pressing need for the transaction to be completed), could, however, result in platforms becoming less inclusive [29], as it discourages validators from approving transactions with low rewards.

5. Ecosystem and on-chain data

5.1 Governance

The governance of a blockchain platform is another useful parameter to evaluate its sustainability and durability. It is important to note that blockchain systems, as they are decentralised, may not have central authorities making governance decisions with respect, for example, to protocol updates or the management of non-distributed native tokens. Consequently, it is necessary to identify which governance mechanisms can be adopted depending on the context, in order to enable the relevant actors to participate in the decision-making process.

In a blockchain permissioned system, where participants can only operate following specific authorisation and with well-defined roles, traditional governance mechanisms, that is, centralised governance, are more easily applicable. This is because they can be delegated directly to the consortium managing the infrastructure. In fact, the controlling entities are able to modify the software of the network nodes or make decisions regarding the permissions and roles of the participants. Conversely, in a permissionless blockchain, it is not possible to identify the entities responsible for the provision of the system's services with certainty, due to the decentralised nature of the system. Consequently, different governance models can be applied.

In general, the design of the protocol, the development of the infrastructure and the tools for its use necessarily require the involvement of a group of people. In certain instances, the development team is responsible for protocol maintenance and update management. This may give rise to questions regarding the extent of the system's actual decentralisation. In a context where there is no central authority, the team must necessarily share the choices regarding network management with a community of users and external collaborators. This is particularly relevant in the case of decisions regarding the upgrade of the protocol.

These governance mechanisms, which may be defined as 'off-chain' and which sometimes also make use of unconventional coordination mechanisms (for example, forums), are then flanked by the platform's native IT protocols, which regulate, for example, the process of validating transactions and executing smart contracts. These mechanisms, together with other more sophisticated ones (e.g. based on smart contracts and/or governance tokens), are referred to as 'on-chain'. Furthermore, the objective is to automate the processes by which participants cooperate to make decisions, for example on protocol updates and the definition of new functionalities, but also on ecosystem incentive policies or internal organisation (e.g. definition of new development or management teams) [57, 58, 66, 67, 68, 69, 70, 71, 72, 73].

5.2 Use of the platform

Another useful parameter for characterising a blockchain is the actual use of the platform. A system that is not utilised to its full potential is unlikely to gain value. One parameter for assessing the utilisation of a platform is the volume of transactions executed 'on-chain'. If the maximum TPS is identified as a fundamental metric for measuring the scalability of the system, the actual TPS can be considered an indicator for measuring the extent to which the blockchain platform is utilised at any given moment in time. The actual TPS depends on two factors: the number of confirmed transactions within the ledger blocks during a given period of time and the maximum TPS.

6. Application of taxonomy

This section presents a potential methodology for analysing a blockchain platform in comparison to the parameters outlined in the preceding sections. In particular, the diagram below (Diagram 1) presents a methodological approach for acquiring the information necessary to map the qualitative parameters with the blockchain solution under analysis. Subsequently, in the case of a blockchain, the process involves the collection and analysis of data in three distinct phases.

The initial stage of the process entails the assessment of technical qualitative indicators:

1. Evaluate peer-reviewed publications as these should be considered as the primary source of information for data pertaining to the technical characteristics of the platform.
2. Integrate technical information with the platform white paper (if any).
3. In the event of a lack of technical assessments (for instance, in the absence of information regarding the type and functionality of the specific consent mechanism), consult the online documentation of the platform.

The second phase is characterized by an economic analysis with respect to the tokenomics of the project and the data relating to the reference market. This information can be obtained through specialized data providers; however, in the current context assessing the credibility and reliability of sources can be challenging.

Finally, the third and final step of the methodology is to collect data on the ecosystem and direct 'on-chain' use.

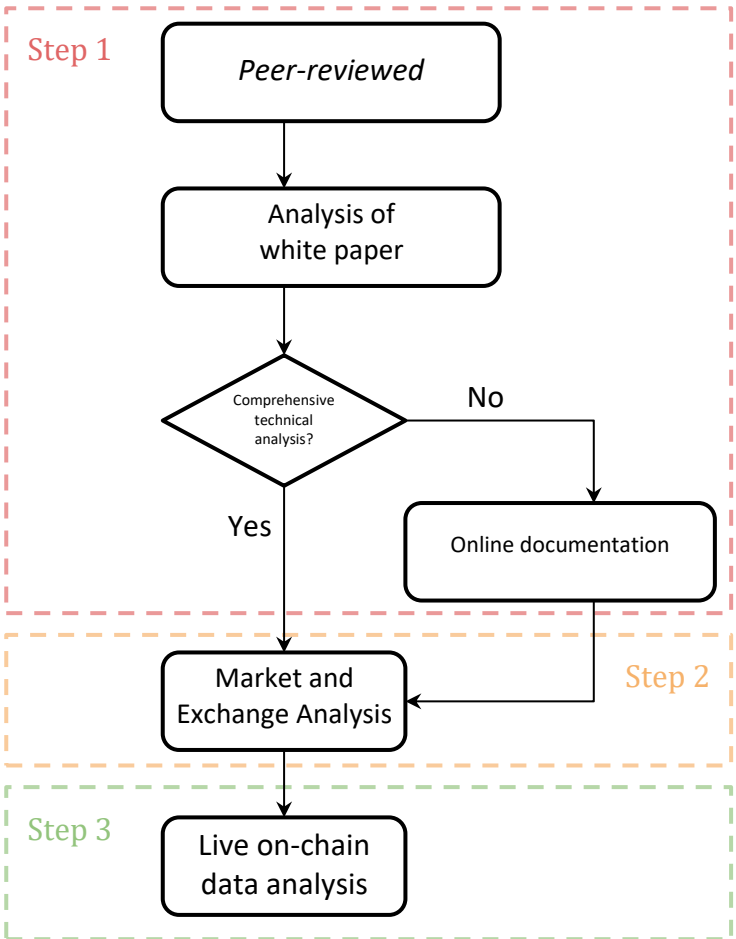


Diagram 1. Application of the analysis process to a new blockchain. Data acquisition phase.

6.1 Conclusions

The development of applications based on blockchain (or smart contracts) is strongly conditioned by the characteristics of the specific technology used. In order to achieve this object

tive, this document commences with an analysis of the state of the art, which is then followed by a list and description of the main characteristics of blockchain technology. The document also proposes a methodological approach that could be used to capture the information needed to describe and analyse blockchain platforms in relation to the parameters identified. In this context, the proposed analysis process can be applied as an experimental exercise to a group of blockchains with the objective of evaluating their fundamental characteristics and analysing their similarities and differences.

Bibliography

[1] Italian Blockchain Service Infrastructure (IBSI). URL: https://progettoibsi.org/; Access: 30/01/2023
[2] ABI Lab, 'Spunta Banca DLT', (2020). URL: https://www.abilab.it/research-areas/blockchain-dlt/spunta-banca-dlt; Access: 30/01/2023
[3] European Blockchain Service Infrastructure (EBSI). URL: https://ec.europa.eu/digital-building-blocks/wikis/display/EBSI/Home; Access: 30/01/2023
[4] World Economic Forum, 'Blockchain Toolkit - Legal and Regulatory Compliance'. URL: https://widgets.weforum.org/blockchain-toolkit/legal-and-regulatory-compliance/index.html; Access: 20/12/2022
[5] Bassan, F. 'Digital Platforms and Global Law', Edward Elgar Publishing (2021)
[6] Bassan, F. 'Digital Platforms and Blockchains: the AGE of Participated Regulation', European Business Law Review, 2022
[7] Bassan, F. 'Algorithm power and market resistance in Italy: lost sovereignty over services', Rubbettino Editore, 2018
[8] Kannengießer N. *et al.*, 'Trade-offs between Distributed Ledger Technology Characteristics', ACM Comput. Surv. 53.2, (May 2020)
[9] Mingxiao, D. *et al.*, 'A Review on Consensus Algorithm of Blockchain', *IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 2017, pp. 2567-2572, doi: 10.1109/SMC.2017.8123011*
[10] Cachin, C. and Vukolic, M. 'Blockchain Consensus Protocols in the Wild', CoRR, 2017
[11] Lamport, L. 'Proving the Correctness of Multiprocess Programs, IEEE Transactions on Software Engineering SE-3.2, 1977, pp. 125–143
[12] Dinh, A. *et al.*, 'BLOCKBENCH: A Framework for Analyzing Private Blockchains', SIGMOD. ACM. 2017, pp. 1085–1100
[13] Schäffer, M., Di Angelo M. and Salzer, G. 'Performance and Scalability of Private Ethereum Blockchains', Business Process Management: Blockchain and Central and Eastern Europe Forum. BPM, 2019. Lecture Notes in Business Information Processing, vol 361. Springer, Cham. https://doi.org/10.1007/978-3-030-30429-4_8
[14] Mazzoni, M., Corradi, A. and Di Nicola, V. 'Performance evaluation of permissioned blockchains for financial applications: The ConsenSys Quorum case study', Blockchain: Research and Applications, 2021
[15] Troncoso C. *et al.*, 'Systematizing Decentralization and Privacy: Lessons from 15 Years of Research and Deployments', Proceedings on Privacy Enhancing Technologies, 2017
[16] Pass, R. Seeman L, and Shelat, A. 'Analysis of the Blockchain Protocol in Asynchronous Networks', Coron, JS., Nielsen, J. (eds) Advances in Cryptology – EUROCRYPT 2017. EUROCRYPT 2017. Lecture Notes in Computer Science, vol 10211. Springer, Cham. https://doi.org/10.1007/978-3-319-56614-6_22

[17] Reis Furtado F. et al., 'Towards characterising architecture and performance in blockchain: a survey', International Journal of Blockchains and Cryptocurrencies, 2020 Vol.1 No.2, pp.121 - 153
[18] The Performance of Byzantine Fault Tolerant Blockchains'. In: 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), IEEE 19th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 2020, pp. 1-8, doi: 10.1109/NCA51143.2020.9306742
[19] Sorensen, D. 'Establishing Standards for Consensus on Blockchains', Blockchain Second International Conference
[20] Dwork, C., Lynch N. and Stockmeyer, L. 'Consensus in the presence of partial synchrony', Journal of the ACM, 1988
[21] BitFury Group and Garzik, J. 'Public versus Private Blockchains Part 1: Permissioned Blockchains', URL: https://bitfury.com/content/downloads/public-vs-private-pt1-1.pdf
[22] BitFury Group and Garzik, J. 'Public versus Private Blockchains Part 2: Permissionless Blockchains', URL: https://bitfury.com/content/downloads/public-vs-private-pt2-1.pdf
[23] Lorne, L. and Cawrey, D. 'Mastering Blockchain'. O'Reilly Media, 2020
[24] Bashir, I. 'Mastering blockchain'. Packt Publishing Ltd, 2017
[25] Buterin, V. 'The Scalability Trilemma' URL: https://vitalik.ca/general/2021/04/07/sharding.html; Access: 20/12/2022
[26] Avizienis A. et al., 'Basic Concepts and Taxonomy of Dependable and Secure Computing'. In: IEEE Trans. Dependable security. Comput
[27] Cachin, C., Guerraoui R. and Rodrigues, L. 'Introduction to Reliable and Secure Distributed Programming', Springer, 2011, XIX, 320 pages
[28] Gilbert S. and Lynch, N. 'Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-tolerant Web Services', SIGACT News 33, 2 (June 2002), 51–59. https://doi.org/10.1145/564585.564601
[29] De Angelis, S. *et al.*, 'Evaluating Blockchain Systems: A Comprehensive Study of Security and Dependability Attributes' Proceedings http://ceur-ws. org ISSN, (2022)
[30] Van Steen, Maarten, and A. Tanenbaum. 'Distributed systems principles and paradigms.' Network 2 (2002), URL: https://www.distributed-systems.net/index.php/books/ds3/
[31] Vukolić, M. 'The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication'. In: Open Problems in Network Security
[32] Vukolic, M. 'Eventually Returning to Strong Consistency', IEEE Data Eng. Bull. 39, 2016
[33] Castro M. and Liskov, B. 'Practical Byzantine Fault Tolerance', Proceedings of the Third Symposium on Operating Systems Design and Implementation
[34] Gilad, Y., Hemo, R., Micali, S., Vlachos, G. and Zeldovich, N. 'Algorand: Scaling Byzantine Agreements for Cryptocurrencies'. In Proceedings of the 26th Symposium on Operating Systems Principles (SOSP '17). Association for Computing Machinery, New York, NY, USA, 51–68. https://doi.org/10.1145/3132747.3132757
[35] Buterin, V. & Griffith, V. 'Casper the Friendly Finality Gadget', 2017
[36] Schär, F. 'Blockchain Forks: A Formal Classification Framework and Persistency Analysis', The Singapore Economic Review, 2020
[37] European Parliamentary Research Service, 'Can distributed ledgers be squared with European data protection law?'. Blockchain and the General Data Protection Regulation
[38] Goldwasser, S., Micali, S. and Rackoff, C. 'The knowledge complexity of interactive proof-systems'. In Proceedings of the seventeenth annual ACM symposium on Theory of computing (STOC '85). Association for Computing Machinery, New York, NY, USA, 291–304. https://doi.org/10.1145/22145.22178
[39] Micciancio, D. & Regev, O. 'Lattice-based Cryptography', Bernstein, D.J., Buchmann, J., Dahmen, E. (eds) Post-Quantum Cryptography. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-88702-7_5
[40] Bondi, A. B. 'Characteristics of Scalability and Their Impact on Performance', Proceedings of the 2nd International Workshop on Software and Performance
[41] Braddock, R. L. Claunch M. R and Walter Rainbolt, J. 'Operational Performance Metrics in a Distributed System. Part II.: Metrics and Interpretation', Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing: Technological Challenges of the 1990's
[42] Nakamoto, S. 'Bitcoin: A peer-to-peer electronic cash system'. 2008. URL: https://bitcoin.org/bitcoin.pdf.

[43] Eyal, I. and Emin Gün Sirer. 2018. Majority is not enough: Bitcoin mining is vulnerable. Commun. ACM 61, 7 (July 2018), 95–102. https://doi.org/10.1145/3212998
[44] Micali, S., Vadhan, S. and Rabin, M. 'Verifiable Random Functions', Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS '99). IEEE Computer Society, USA, 120.
[45] Marc, J. & Farouk, H. & Ramy, G. & Ziyaad, Q. 'Do Smart Contract Languages Need to be Turing Complete?', Prieto, J., Das, A., Ferretti, S., Pinto, A., Corchado, J. (eds) Blockchain and Applications. BLOCKCHAIN 2019. Advances in Intelligent Systems and Computing, vol 1010. Springer, Cham. https://doi.org/10.1007/978-3-030-23813-1_3
[46] Zheng, Z., Xie, S., Dai, H.N., Chen, W., Chen, X., Weng, J. Imran, M. 'An overview on smart contracts: Challenges, advances and platforms' FGCS, 2020
[47] Atzei, N., Bartoletti, M., Cimoli, T. (2017). A Survey of Attacks on Ethereum Smart Contracts (SoK). In: Maffei, M., Ryan, M. (eds) Principles of Security and Trust. POST 2017. Lecture Notes in Computer Science, vol 10204
[48] Destefanis, G., Marchesi, M., Ortu, M., Tonelli, R., Bracciali A. and Hierons, R. 'Smart contracts vulnerabilities: a call for blockchain software engineering?' 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE), Campobasso, Italy, 2018, pp. 19-25, doi: 10.1109/IWBOSE.2018.8327567
[49] Johnson, S., Robinson, P., and Brainard, J. 'Sidechains and interoperability', eprint arXiv:1903.04077, 2019
[50] Buterin, V. 'Chain Interoperability' (2016), URL: https://allquantor.at/blockchainbib/pdf/vitalik2016chain.pdf
[51] Chainalysis, '2022 Crypto Crime Report.'. URL: https://go.chainalysis.com/2023-crypto-crime-report.htmAccess: 02/03/2023
[52] Goes, C. 'The Interblockchain Communication Protocol: An Overview' (2020), eprint arXiv:2006.15918
[53] Micali, S., Reyzin, L., Vlachos, G., Wahby, R. S. and Zeldovich, N. 'Compact Certificates of Collective Knowledge' (2021), IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2021, pp. 626-641, doi: 10.1109/SP40001.2021.00096
[54] CCRI, 'Crypto Sustainability Indices'. URL:https://indices.carbon-ratings.com/
[55] Bucko, J. & Palová, D. & Vejačka, M., 'Security and Trust in Cryptocurrencies', Central European Conference in Finance and Economics, 2015
[56] Sayeed, Sarwar & Marco-Gisbert, Hector, 'Assessing Blockchain Consensus and Security Mechanisms against the 51% Attack'. Applied Sciences. 9, (2019)
[57] Reijers, W., O'Brolcháin, F., & Haynes, P. 'Governance in Blockchain Technologies & Social Contract Theories. Ledger', 1, 134–151. https://doi.org/10.5195/ledger.2016.62, Ledger, 1, 134–151. https://doi.org/10.5195/ledger.2016.62, 2016
[58] Atzori, M. 'Blockchain Technology and Decentralized Governance: Is the State Still Necessary?' Available at SSRN: https://ssrn.com/abstract=2709713 or http://dx.doi.org/10.2139/ssrn.2709713, 2015
[59] Gilbert S. and Lynch, N. 'Perspectives on the CAP Theorem,' in *Computer*, vol. 45, no. 2, pp. 30-36, Feb. 2012, doi: 10.1109/MC.2011.389
[60] Xiao, Y., Zhang, N. Lou W. and Hou. Y. T. *'A Survey of Distributed Consensus Protocols for Blockchain Networks'* in IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 1432-1465, (2020), doi: 10.1109/COMST.2020.2969706
[61] He, Ping and Tang, Dunzhe and Wang, Jingwen. 'Staking Pool Centralization in Proof-of-Stake Blockchain Network' (May 25, 2020). Available at SSRN: https://ssrn.com/abstract=3609817
[62] Feng Tian. 2016. An agri-food supply chain traceability system for China based on RFID & blockchain technology. In Proceedings of the 13th International Conference on Service Systems and Service Management. 1–6.
[63] Filip Caron. 2018. The evolving payments landscape: Technological innovation in payment systems. IT Profess. 20, 2 (2018), 53–61.
[64] Gaby G. Dagher, Jordan Mohler, Matea Milojkovic, and Praneeth Babu Marella. 2018. Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. Sustain. Cities Soc. 39 (2018), 283–297.

[65] NIST Computer Security Resource Center, Post-Quantum Cryptography
https://csrc.nist.gov/projects/post-quantum-cryptography
[66] Accenture. 2019. Governing DLT Networks. Distributed Ledger Technology Governance for Permissioned Networks.
[67] Allen, D. W. E., Berg, C. 2020. Blockchain Governance: what we can learn from the Economics of Corporate Governance, The JBBA, Volume 3, Issue 1.
[68] Astri. 2016. Whitepaper on Distributed Ledger Technology, Hong Kong Applied Science and Technology Research Institute and Hong Kong Monetary Authority.
[69] Hofman, D., et al. 2021. Blockchain Governance: De Facto (x)or Designed? in Lemieux, V.L., Feng, C. (eds.), Building Decentralized Trust, Chapter 2.
[70] Liu, Y., et al. 2022. Defining Blockchain Governance Principles: A Comprehensive Framework, University of New South Wales, Australia.
[71] Naudts, E., et al. 2022. Governance in systems based on distributed ledger technology (DLT): A comparative study, AFM.
[72] van Pelt, R. et al. 2020. Defining Blockchain Governance: A Framework for Analysis and Comparison, Information Systems Management, 38:1, 21-41.
[73] Wang S. et al. 2019. Decentralized Autonomous Organizations: Concept, Model, and Applications, IEEE Transactions on Computational Social Systems, Vol. 6, No. 5.
[74] Bassan, F. 'Web 3 Regulation in Transition, Competition Policy International' in TechREG CHRONICLE, 2023
[75] Bassan, F. 'Digital Platforms and Blockchains: The Age of Participated Regulation', in European Business Law Review, 2022
[76] De Filippi, P., Mannan, M. and Reijers, W. 'Blockchain Technology and the Rule of Code: Regulation via Governance', available on SSRN: https://ssrn.com/abstract=4292265 (2022)
[77] De Filippi, P., Mannan, M., Henderson, J., Merk, T., Cossar, S. and Nabben, K., Report on blockchain technology & legitimacy, European University Institute (2022)
[78] Varun Kohli and Sombuddha Chakravarty and Vinay Chamola and Kuldip Singh Sangwan and Sherali Zeadally, 'An analysis of energy consumption and carbon footprints of cryptocurrencies and possible solutions'. Digital Communications and Networks, Vol. 9, No 1. https://doi.org/10.1016/j.dcan.2022.06.017 (2023)
[79] ISO/TS 23258:2021. Blockchain and distributed ledger technologies — Taxonomy and Ontology. URL: https://www.iso.org/standard/75094.html
[80] Gudgeon, L., Moreno-Sanchez, P., Roos, S., McCorry, P., Gervais, A., 'SoK: Layer-Two Blockchain Protocols', Bonneau, J., Heninger, N. (eds) Financial Cryptography and Data Security. FC 2020. Lecture Notes in Computer Science, vol 12059. Springer, Cham. https://doi.org/10.1007/978-3-030-51280-4_12 (2020)
[81] Poon, J, Dryja, T., 'The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments', https://lightning.network/lightning-network-paper.pdf, January 2016.

Section II - Taxonomy of the technical characteristics of smart contracts

1. Introduction

The purpose of this section is to describe, from a technical point of view, the phenomenon of smart contracts and to identify their peculiarities (see the box below on the methodological approach – Methodological Note 1). The section is organised as follows: Paragraph 2 introduces the technical definition of smart contracts, their main functionalities and their life cycle. Paragraph 3 presents a classification of key characteristics, distinguishing between technological characteristics and high-level characteristics. In paragraph 4, the security issues of the smart contracts are addressed, with an introduction to the main challenges for the secure development of decentralised applications (DApps) and an initial analysis of the possible vulnerabilities that the technology introduces.

Methodological note 1

The section describes the technological and high-level characteristics of smart contracts and their main features. In particular, the section's contribution is based on literature analysis and selective research of the methodological work. Existing works in the literature dealing with topics such as the definition, comparison and regulation of smart contracts were analysed. In order to provide the generic model presented in paragraph 3 ('Smart Contract Overview'), systematisation work and surveys on smart contracts [1, 2, 3, 4, 5] were examined. The technical analysis and benchmarking work [2, 5, 6, 7, 8] was used to (i) provide a systematization of the technical characteristics of smart contracts; (ii)analyse differences and trade-offs between different execution environments; (iii) investigate existing programming languages.

Finally, an evaluation of the security parameters to be considered in the development of decentralized applications and related smart contracts is proposed. Specifically, the set of known vulnerabilities in the sector is reported [8, 9, 10, 11]. The result obtained was partly inspired by known works in the literature such as DASP - TOP 10 [12], the Smart Contract Security Verification Standard [13] and the SWC registry [14] of known vulnerabilities.

2. Smart Contract Overview

In computer science, the notion of smart contracts was first introduced in 1990 as a digital computing protocol capable of autonomously executing the terms of a contract defined in a transactional system [15]. With the advent of blockchain, smart contracts have been the subject of renewed interest also thanks to the significant increase in the number of decentralised applications (or 'DApps') developed [57].

In general, the term DApp is frequently used to refer to any decentralized blockchain-based application that includes, in addition to the underlying blockchain technology, the user interface (usually a web interface) and smart contracts [19, 20]. As DApps are decentralised, they do not require a service provider or, in general, a trusted party to manage their infrastructure. One of the main benefits of using decentralised applications is to optimise inefficient execution flows.

Considering only the level of smart contracts, this section provides an analysis of them from a technological point of view.

2.1 What are smart contracts?

In general, smart contracts are software programs developed in a specific programming language. Bitcoin, for example, allows, using a non-comprehensive Turing scripting language, the development

of smart contracts to manage asset transferability [16, 17]. Ethereum [18], on the other hand, through the computational abstraction of the Ethereum Virtual Machine (EVM), allows the execution of smart contracts developed in complete Turing programming languages[101].

Since smart contract programs are based on blockchain and are therefore executed in a collective and decentralized manner by the nodes of the network [1, 18], their execution is validated by the underlying blockchain network and their reliability is connected to that of the blockchain [8, 11]. The execution of a smart contract is deterministic and is based exclusively on data available on-chain. This ensures that, during the distributed execution of the smart contract code, each node in the network gets the same result (or output) given a set of input parameters and a certain state of the blockchain[102] [21, 22]. Examples of execution of a smart contract are: conditional approval of a payment between two users (for example, you can approve the payment transaction to a beneficiary user if and only if a certain time frame has passed); the exchange of an asset (for example, a smart contract that implements a market place of collectible assets that can be exchanged between users) [3, 6].

The execution of a smart contract is typically triggered by a transaction known as a 'call'. For example, a smart contract can be invoked through a transaction that implements the sum transaction by passing the transaction addends as transaction arguments (i.e. input parameters). Transactions invoking a smart contract are approved and the code of the smart contract executed only if the protocol rules of the underlying blockchain that determine the validity of a transaction are met.

Smart contracts inherit the traceability and immutability properties of the underlying blockchain. In particular, the code of smart contracts is registered within the blockchain itself and therefore by its nature cannot be changed. This feature is crucial to define security models based on decentralized applications in which users are certain that the code is not altered. In this regard, the blockchain maintains a unique track of both creation and all invocations to a smart contract. Moreover, even with complex computational patterns that alter the execution of the smart contract and make it possible to create new behaviours of the smart contract itself (for example, this happens in the 'Proxy Pattern' update mechanism [23, 24]), it is always possible to reconstruct the trace of the smart contract by observing the history of the ledger [8, 23].

Smart contracts that, as anticipated, are written in a specific high-level programming language (for example, Solidity for the Ethereum blockchain [25]) or in a scripting language (for example, Bitcoin Scripting [16, 17] for the Bitcoin blockchain), are compiled into a set of instructions directly executable from the computational model implemented by the specific blockchain (bytecode). The compiled bytecode of a smart contract is installed and executed within an execution environment.

There are several execution environments that generally fall into two categories [8, 26]: (i) memory stack-based interpreters (e.g., Bitcoin Script interpreter [16, 17]); (ii) virtual machine-based interpreters (e.g., the Ethereum Virtual Machine - EVM [6, 18]).

Execution environments differ in the programming language they support, the type of interpretable commands (e.g. instructions, functions, computational loops, jumps) and the type of memory used [2, 7, 8, 27]. In paragraph 3 ('Fundamental Characteristics'), we will present the different characteristics of the execution environments, which will be followed by an assessment of the respective trade-offs.

[101] Ability to execute recursive code and loops within the program. Please refer to section 4.1.4 to learn more about the differences between complete Turing languages and non-complete Turing languages.
[102] Blockchain status means the latest version of the ledger – the set of blocks and transactions – validated and approved by the majority of the blockchain network at any given time [18].

2.2 Interacting with smart contracts

To interact with a smart contract, blockchain users typically send a cryptographically signed transaction with their private key. The blockchain keeps track of both the creation and all invocations to a smart contract by storing them in the state maintained by the ledger.

There are two main state models on which blockchains are based: account-based and token-based.

An account is, in general, an entity that is able to send transactions to the blockchain and has a balance sheet. In the account-based model, each account is allocated a memory space to save the status information associated with the account (e.g. the balance of tokens held by the account) [6, 18]. The local state of each account contributes to the definition of the global state of the blockchain [6, 8, 18]. Generally, this blockchain status model distinguishes two types of accounts: (i) standard accounts, which are associated with a public/private key pair and controlled by anyone who owns the private key; (ii) contract account controlled by the logic of the specific associated smart contract [6, 18].

An alternative to the account-based model is the token-based model, of which the Unspent Transaction Output (UTXO) used by Bitcoin [16, 17, 28] and the Extended UTXO [29] are particular examples. In this model, the blockchain traces the available tokens through the transactions stored in the ledger. As a result, a user's token balance can be spread across hundreds of transactions and blocks. In a token-based model, there is no concept of account and associated balance sheet; there are only tokens locked to be used by specific users. As a result, at a given time, the status of the ledger is defined by all the tokens still available.

In this document, without claiming to be exhaustive, we mainly consider blockchain technologies that adopt an account-based state model, which are currently the most widespread in the financial sector.

Therefore, in a blockchain technology with an account-based state model, smart contracts are a particular type of account; As a result, they have an associated balance sheet and can be invoked through transactions. However, they are not controlled by a user and are executed by the blockchain's network of nodes. Each invoking transaction produces the execution of the logic contained therein and the updating of the global state of the blockchain. In particular, given a user of a smart contract, the execution steps can be summarized as follows [2, 31, 32]:

1. the user signs and sends an invoking transaction on the blockchain specifying a function of the smart contract;
2. the blockchain software assesses the validity of the transaction, verifies the existence of the requested smart contract and extracts the input parameters (if specified);
3. the required smart contract function is invoked and the coded logic is executed by passing the input parameters (if specified);
4. Following execution, the resulting transactions and new status information are confirmed and stored in the ledger in accordance with the protocol rules of the underlying blockchain.

These steps define the execution cycle of a smart contract.

2.3 Life cycle

This document describes a state model of account-based blockchains. In addition, smart contracts executed in a virtual machine (VM)-based environment are considered (see paragraph 3.1.1). In this context, it is possible to articulate the life cycle of smart contracts in six fundamental phases [2, 32]:

- Development: translating the functional requirements of a specific contract or process into the smart contract programming language. This phase follows the same traditional programming principles, such as modular development and testing of programme functions;
- Compilation: generation of the smart contract bytecode for running on VMs; the source code is processed by a compiler;
- Deployment: the bytecode is validated by the blockchain software and instantiated within the ledger; the smart contract account and its status are generated in the execution environment; users can send invoking transactions to execute the smart contract logic;
- Execution: following an invocation, the VM executes the smart contract logic by applying the inputs and generating a unique output; at the end of the execution, the VM updates the status of the smart contract with the result of the output;
- Update: modification of the smart contract to a new version of the compiled bytecode. For example, developers may decide to change approval conditions, troubleshoot code/bugs or add new features; the updating procedure can be made arbitrarily complex (e.g. for Ethereum smart contracts there are update patterns that developers can perform (e.g. Contract Migration, Proxy pattern etc. [33]) [23, 24]. However, these mechanisms may vary depending on the underlying blockchain; in general, this procedure requires the implementation of governance permissions to avoid unexpected manipulations by actors not authorised to change the execution logic [24];
- Cancellation: disabling the functions of the contract; the smart contract is inaccessible and therefore not usable. This procedure requires the implementation of permits to avoid unexpected manipulations.

2.4 Updating and Governance

In the previous section, the fundamental properties of immutability and irreversibility of smart contracts have been highlighted, which offer guarantees of security both on the content of the contract and on the operations envisaged during execution. However, it is possible to provide for an update of a smart contract in the circumstance in which, for example, it becomes necessary to add, change or remove a clause, or possibly solve technical problems related to unexpected execution patterns, or vulnerabilities in the code. Updating a smart contract involves modifying the logic performed while preserving the status of the contract. It is important to clarify that mutability and updateability are not synonymous, especially in the context of smart contracts. It is not possible to change the code: however, it is possible to change the logic performed when invoking a smart contract, provided that appropriate precautions are taken.

In this section, we propose a focus on the techniques that can be used to manage the updating of a smart contract. There are different patterns of updating a smart contract in the literature [23, 24], which rely on different levels of governance and are optimized on individual blockchains [33]. Below is a high-level description of possible upgrade patterns within two categories: (i) upgrade of individual functions and (ii) upgrade of the smart contract. Finally, the governance mechanisms that the actors involved can adopt to ensure a safe, efficient and democratic update of the code are highlighted.

- Updating individual smart contract functions: update pattern of a limited part of the smart contract. In this scenario, some parameters of the smart contract can be modified or updated (e.g. by sending new inputs) which consequently modify the execution of it (an example are smart contracts that rely on a parameter passed as an input to influence the execution of the business logic); in more complex scenarios, the execution of entire functions of the smart contract is delegated to functions of other smart contracts [34] (for example, in Ethereum

this is done through the 'Proxy' and 'Diamond' patterns [33] - Modification through input/modification through modularity);
- Updating the smart contract: the entire smart contract code is moved to a new version; this procedure totally replaces the code of a smart contract with a new version; in platforms such as Ethereum, this can be done through the cancellation of the smart contract [35] and the subsequent deployment of a new version, or through proxy mechanisms that manage the addressing of requests to new versions of the contract [33, 34].

Updating a smart contract can drastically change its functioning and therefore the business logics performed. It is therefore essential to manage the permissions of the actors authorised to approve these changes. In this sense, there are different approaches to managing the actors in charge of updating [34] that are based on three directives:

- individual administrator: entity known to the smart contract (identified with the blockchain public key identity) in charge of performing the update functionalities; this entity can be written immutably in the code of the smart contract or can be made explicit programmatically (for example by the creator of the contract itself);
- group of administrators: group of entities authorised to update the smart contract, which are usually identified by a 'multiple' blockchain identity (called a multi-signature wallet [36]) where the update is approved if and only if all members of the group of administrators sign the transaction;
- decentralised organisation: a decentralized system (usually based on smart contracts) in which a set of participants, not necessarily administrators, agree through a democratic vote on the approval or refusal of the update of the smart contract.

3. Key features

This section describes the fundamental characteristics of smart contracts. On the one hand, the technological characteristics are illustrated, also showing the trade-offs between the implementing alternatives adopted to date by blockchain platforms, and on the other hand the high-level characteristics that identify general parameters to be considered for the evaluation of smart contracts.

3.1 Technological features

3.1.1 Execution environment

Smart contracts run in a computational environment distributed across all nodes of the blockchain. In general, the bytecode instructions of a smart contract are executed by an interpreter based on dynamic memory allocation (stack-based) [7, 8, 26]. VM is an implementation of an abstract computer model that virtualizes components such as CPU, memory and storage, and is often used to simulate an isolated computational environment within a physical computer. In the field of blockchain technologies, the VM used keeps track of the state of the blockchain, including the accounts, the bytecode of the smart contracts and the memory used by them. Execution of the code is done by evaluating the bytecode instructions sequentially if no error conditions occur.

The VM architecture may generally consist of the following components [7, 18]:

- stack: dynamic portion of memory that executes a set of instructions in a push-down manner (towards decreasing memory addresses); usually has a limited length and is compatible with basic data types such as bytes and integers; the stack is reset with each execution;
- memory: volatile (or temporary) memory space dedicated to a single execution that is not saved to global state storage;
- storage memory: memory space saved on the global VM state, usually represented with a 'value key' data structure. Some blockchains offer limited storage space for smart contracts, while others define memory limits in order to optimize the execution and reading of information [6, 11, 18, 37, 38].

3.1.2 Tradeoff between Stateful and Stateless execution environments

The complexity of the execution environment and state management have a direct impact on the performance of the system.

There are, in fact, trade-offs between the implementation of stateless execution environments and stateful execution environments. In the first case, the execution environment does not store information or references due to past executions [39, 40, 41]. The interpreter used by Bitcoin for executing scripts is an example of a stateless execution environment [17, 41]. On the contrary, execution environments that track information from past executions are called stateful [6, 18, 39]. The VM-based execution environment used in Ethereum is an example of a stateful execution environment.

Thus, in the stateless case, the execution environment is restricted and limited, as it cannot rely on conditions and information stored in memory. Stateless execution environments, however, enable secure, efficient and cost-effective execution [7, 40, 41, 42]. On the contrary, in stateful environments it is possible to rely on memory to maintain a state that can also be consulted in different executions of the smart contract. This makes the system more flexible, as it allows the development of more complex applications, but with higher management costs. In this regard, it is necessary to evaluate the trade-offs on memory management. In the event that there are no memory limits, it is the

developers who have to manage the memory of the smart contract in an optimal way, possibly introducing risks related to possible bugs or inefficient implementations. On the contrary, having limited memory helps to mitigate possible errors, at the cost of reduced flexibility [42].

3.1.3 Programming language

In stateful execution models based on VMs, a set of basic instructions (opcodes) [43] related to logical, arithmetic and specific operations are executed on the VM stack to interact with the state of the blockchain.

Bytecode is generated from a compiler and source code usually written in a high-level language. There are different types of languages with which it is possible to develop smart contracts, usually dependent on the blockchain and the reference VM. For example, for Ethereum the two most popular languages are Solidity [25] and Vyper [44].

Programming languages offer many features such as [7, 25, 40, 43]:

- extension of the types natively supported by the VM (bytes and integers) for more complex use cases;
- use of libraries: definition of functions that can be reused in the smart contract or invoked by other smart contracts;
- support functions for manipulating the state of the VM.

These features determine the flexibility of a programming language, and in particular the ability to develop complex applications through the tools offered such as types, libraries and functions.

3.1.4 Turing-Complete vs Non-Turing-Complete tradeoff

Smart contract programming languages differ mainly between full Turing and non-complete Turing. In general, literature defines a language as complete Turing if it is able to express a complex computational problem solvable with a Turing machine (a computer) [45, 46].

In the field of smart contracts, Ethereum was the first platform to offer a VM capable of running a complete Turing language (Solidity [25]). The advantage of Turing completeness is that smart contracts are extremely flexible, allowing developers to reproduce any type of calculation directly on the blockchain [8]. For example, a complete Turing language allows the development of programs, even complex ones, that use loops or recursive functions. However, this introduces greater complexity and thus makes the system itself more prone to errors and vulnerabilities (e.g., endless code executions can occur) [7, 42, 47]. On the contrary, a complete non-Turing language sacrifices flexibility in favour of a simpler language and less prone to errors / bugs. For example, the scripting language adopted in Bitcoin is not Turing complete [17, 41, 47].

3.2 High-level features

This section describes the high-level features for evaluating smart contracts, such as deployment costs and execution costs. These differ from the underlying blockchain and are described as follows:

- Deployment costs: the deployment of a smart contract takes place by sending a deployment transaction on the blockchain. This transaction must contain the bytecode of the smart contract that will be written on the ledger and instantiated in the context of the VM. In particular, the VM will have to dedicate to the smart contract a memory space for the management of the stack, memory and storage memory. For the execution of smart contracts, costs are determined by the size of the bytecode and the memory space it requires to operate. These costs are also

added to the transaction costs, i.e. the fees that users have to pay to deploy a smart contract on the blockchain itself. [48, 49]

- Execution costs: In addition to deployment costs, there are execution costs. Every smart contract, in fact, is invoked through a transaction that executes its logic. This involves the consumption of resources, which on the blockchain represent a common and shared good. To perform the operations of the smart contract it is therefore necessary to cover the execution costs. For example, in Ethereum, this is done via the concept of gas [50].

4. Security considerations

We now illustrate security considerations to be evaluated for the development of DApp. Although the DApp architecture may involve a large attack surface, determined by the entire application stack (e.g. a backend that interacts with the smart contract, a frontend for the use of UI/UX, databases, etc.), this section proposes an exclusive focus on the smart contract execution domain. The proposed analysis is therefore divided into two parts. The first highlights the challenges related to the secure development of applications based on smart contracts. In the second part, we illustrate the technical and technological components of smart contracts that can make the system vulnerable to attacks and therefore unreliable. In particular, it is already clear that there is a need for guidelines setting out in detail how to prevent and react to any attacks on smart contracts allowed by vulnerabilities (i) in the VM, (ii) in the programming language and (iii) in the code.

4.1 Challenges for developing secure DApps

Smart contract programming paradigms for DApps require a different development process than traditional software engineering approaches. Technology, on the one hand, provides opportunities and, on the other, creates new risks due, for example, to cyber-attacks. In general, a failure of a smart contract has a significant impact on the costs, as the value it controls or represents could be compromised, lost or stolen. The main challenges that developers face in approaching secure development of decentralized applications are defined below.

1. Accessibility: smart contracts are executed in a decentralized environment, and each phase of a smart contract's lifecycle brings security challenges;

2. Modularity: support for comprehensible and robust applications requires architectural models that consider all possible interactions within the VM (e.g. from other smart contracts) and outside the smart contract (called by malicious actors through invoking transactions) [52, 53];

3. Complexity: the smart contract compilation must generate simple bytecode to limit possible unexpected execution patterns [8];

4. Test: to execute deployment in production, the smart contract must be tested and validated from multiple sources; it is preferable to use, where possible, open-source components developed in projects that have already been extensively tested (and preferably have passed security audits). Once running, it must be ensured that all the functionalities of the smart contract follow the contractual conditions [35, 54];

5. Recovery: since smart contracts are programs written in machine language and executed in a decentralized manner, they could be exposed to malfunctions, errors or even external manipulations. In case of business logics for the management of value, it is therefore essential to define recovery/cancellation models of operations with which developers (or entities identified to the management) can update the smart contract and in extreme cases block its

execution (or cancel it) and mitigate the risk of theft; for example, some state-of-the-art solutions propose recovery patterns of tokenised assets [9];

6. Secure update: Paragraph 3.4 ('Update and Governance') introduced the mechanisms for updating a smart contract and the relevant authorised actors; to ensure the security of a smart contract, it is essential to properly manage the governance of the update patterns, defining the permissions and roles of the actors involved according to the use case;

7. Interoperability: smart contracts are installed and executed in the context of a single blockchain platform; this places limits on the interoperability functions between blockchains [55]. As highlighted in **BOX 7** of Part II, Bridges are application solutions that provide for the composition of smart contracts between different blockchains; these are vulnerabilities for Dapps;

8. External inputs: smart contracts, and blockchains in general, do not have access to external data. In particular, oracles [10] are used to access specific information, which provide, depending on the use case, the information to be used in the business logic of the smart contract; it is crucial for the security of DApps to validate inputs from oracles and limit their use to avoid possible external manipulation.

4.2 Possible vulnerabilities

Smart contracts are executed on the blockchain in a decentralized manner and in the absence of trust. To ensure fairness, we have seen in the previous sections how the use of complex systems is necessary, starting from the underlying blockchain to the smart contract execution environment. This section analyses the components of smart contracts that can hide security threats due to vulnerabilities or code errors.

The areas identified were obtained from a review of the existing work in the literature on smart contract vulnerabilities (e.g., [10, 11, 51, 56]):

1. the execution environment: the VM where smart contract instructions are executed is a critical component for security. Appropriate tools are needed to prevent and repress issues related to the enforcement environment. For example: a VM with too strict computational constraints could cause unexpected failures in execution or bugs in the implementation of opcodes could generate unexpected results. To develop or use a smart contract it is therefore important to assess the vulnerabilities of the underlying VM;
2. the programming language: the smart contract bytecode executed on the VM is compiled from a program written in high-level language. When compiling, it must be ensured that the language itself does not generate errors or vulnerabilities. For example, it is crucial that the high-level language offers controlled access method writing capabilities to avoid manipulation by unauthorised users or that it ensures secure arithmetic libraries;
3. the code: although smart contracts are executed in a secure and reliable environment guaranteed by blockchain technology, they are programs that can present errors due to incorrect code or unexpected executions. A code is safe if it takes into account that, in an execution context such as a blockchain, anyone can access the data written in the ledger or interact with smart contracts through invoking transactions. Blockchain is a trustless environment, by its nature, and smart contracts must be structured on this assumption.

5. Conclusions

In this section, smart contracts have been introduced on a technical and technological level.

In the first part, the main components of smart contracts were analysed, presenting the account-based and token-based status models and discussing their peculiarities and differences. In the context of account-based models, the life cycle of smart contracts and the methodologies known for updating and for governance have also been described. Subsequently, a taxonomy of the fundamental characteristics of smart contracts was proposed, highlighting the trade-offs between stateful and stateless execution environments and between complete and non-complete Turing programming languages. The section concludes with high-level features focusing the study on the costs of deployment and execution of smart contracts.

In the second part, an in-depth analysis on security was proposed, identifying the security challenges that developers face for the creation of safe and reliable DApps as well as a classification of the possible vulnerabilities that can affect smart contracts such as VM, programming language and code.

Bibliography

[1] Mik, E. 'Smart Contracts: Terminology, Technical Limitations and Real World Complexity' SSRN: https://ssrn.com/abstract=3038406 or http://dx.doi.org/10.2139/ssrn.3038406 (2017)
[2] Zheng, Z., Xie S., Dai H-N., Chen, W., Chen X., Weng J., Imran M. 'An Overview in Smart Contracts: Challenges, advances and platforms', FGCS (2020)
[3] Mohanta B. K., Panda S. S. and Jena, D. 'An Overview of Smart Contract and Use Cases in Blockchain Technology,' 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT 2018)
[4] Khan, S.N., Loukil, F., Ghedira-Guegan, C. et al. 'Blockchain smart contracts: Applications, challenges, and future trends', Peer-to-Peer Netw. Appl. 14, 2901–2925 (2021)
[5] Wang, S., Ouyang, L., Yuan, Y., Ni, X. and Han, X. 'Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends', IEEE Transactions on Systems, Man, and Cybernetics (2019)
[6] Buterin, V. 'A Next Generation Smart Contract & Decentralized Application Platform' (2015)
[7] Zheng, S., Wang, H., Wu, L., Huang, G., & amp; Liu, X, 'VM Matters: A Comparison of WASM VMs and EVMs in the Performance of Blockchain Smart Contracts', ArXiv, abs/2012.01032 (2020)
[8] Kannengießer, N., Lins S., Sander C., Winter K., Frey H. and Sunyaev A. 'Challenges and Common Solutions in Smart Contract Development', IEEE Transactions on Software Engineering, Vol. 48, No. 11, pp. 4291-4318, 1, doi: 10.1109/TSE.2021.3116808 (2022)
[9] López Vivar A, Castedo AT, Sandoval Orozco AL, García Villalba LJ. An Analysis of Smart Contracts Security Threats Alongside Existing Solutions. Entropy (Basel); 22(2):203 (2020)
[10] Atzei, N., Bartoletti, M., Cimoli, T. (2017). A Survey of Attacks on Ethereum Smart Contracts (SoK). In: Maffei, M., Ryan, M. (eds) Principles of Security and Trust. POST 2017. Lecture Notes in Computer Science, vol 10204.
[11] De Angelis, S. and Zanfino, G., Aniello, L., Lombardi, F., Sassone, V. 'Evaluating Blockchain Systems: A Comprehensive Study of Security and Dependability Attributes', Proceedings http://ceur-ws. org ISSN, (2022)
[12] NCC Group, 'Decentralised Application Security Project – DASP Top 10' (2018). URL: https://dasp.co/index.html
[13] Securing, 'Smart Contract Security Verification Standard' (2020). URL: https://github.com/securing/SCSVS
[14] Smart Contract Weakness Classification Registry (2019), URL: https://swcregistry.io/
[15] Szabo, N. 'Secure Property Titles with Owner Authority' (1998), https://web.archive.org/web/20140115142013/http://szabo.best.vwh.net/securetitle.html
[16] Nakamoto, S. 'Bitcoin: A peer-to-peer electronic cash system' (2008) URL: https://bitcoin.org/bitcoin.pdf.

[17] Bistarelli, S. 'Mercanti Ivan, Santini Francesco, 'An Analysis of Non-standard Transactions', Crypto Valley Conference on Blockchain Technology (CVCBT 2018), pp. 93-96, doi: 10.1109/CVCBT.2018.00016 (2018)
[18] Wood, G. 'Ethereum: A secure Decentralised Generalised Transaction Ledger', EIP-150 REVISION (2014)
[19] Cai, W, Wang, Z., Ernst J. B., Hong, Z., Feng C. and Leung V. C. M. 'Decentralized Applications: The Blockchain-Empowered Software System,' *IEEE Access*, vol. 6, pp. 53019-53033, (2018), doi: 10.1109/ACCESS.2018.2870644
[20] Chibuzor, U., Anyanka, H. and Norta. A. 'Evaluation of Approaches for Designing and Developing Decentralized Applications on Blockchain. In Proceedings of the 4th International Conference on Algorithms, Computing and Systems (ICACS '20)'. Association for Computing Machinery, New York, NY, USA, 55–62. https://doi.org/10.1145/3423390.3426724 (2020)
[21] Morabito, V. 'Smart Contracts and Licensing', Business Innovation Through Blockchain (pp.101-124) (2017)
[22] Alharby M., Aldweesh A. and Moorsel A. V., 'Blockchain-based Smart Contracts: A Systematic Mapping Study of Academic Research' International Conference on Cloud Computing, Big Data and Blockchain (ICCBB 2018), pp. 1-6, doi: 10.1109/ICCBB.2018.875639
[23] Bloo F.W.C., 'Towards updatable smart contracts', http://essay.utwente.nl/76769/(2018)
[24] Bui V. C., Wen S., Yu J., Xia X., Haghighi M. S. and Xiang Y., 'Evaluating Upgradable Smart Contract,' IEEE International Conference on Blockchain (Blockchain 2021) pp. 252-256, doi: 10.1109/Blockchain53845.2021.00041
[25] 'Solidity: Ethereum Smart Contracts Programming Language', URL: https://soliditylang.org/
[26] Zou W. et al. 'Smart Contract Development: Challenges and Opportunities,', IEEE Transactions on Software Engineering, vol. 47, no. 10, pp. 2084-2106, 1 (2021), doi: 10.1109/TSE.2019.2942301
[27] Connors, C. and Sarkar, D. 'Comparative Study of Blockchain Development Platforms: Features and Applications', eprint: arXiv:2210.01913 (2022)
[28] Konrad, R. and Stephen P. 'Bitcoin UTXO Lifespan Prediction.', Available at: https://cs229.stanford.edu/proj2015/225_poster.pdf (2015)
[29] Chakravarty, M., Chapman, J., MacKenzie, K., Melkonian, O., Jones, M. and Wadler, P. 'The Extended UTXO Model', Workshop on Trusted Smart Contracts (Financial Cryptography (2020)
[30] Brünjes, L., Gabbay, M.J. 'UTxO- vs Account-Based Smart Contract Blockchain Programming Paradigms', in Margaria, T., Steffen, B. (eds) Leveraging Applications of Formal Methods, Verification and Validation: Applications. IsoLA, Lecture Notes in Computer Science, vol 12478. Springer, Cham. https://doi.org/10.1007/978-3-030-61467-6_6 (2020)
[31] EU Blockchain Observatory and Forum, 'Smart Contracts' (2022)
[32] Sillaber, C., Waltl, B. 'Life Cycle of Smart Contracts in Blockchain Ecosystems.' Datenschutz Datensich 41, 497–500 (2017)
[33] Upgrading Ethereum Smart Contracts. URL:
https://ethereum.org/en/developers/docs/smart-contracts/upgrading/#proxy-patterns
[34] Salehi, M., Clark, J. and Mannan, M. 'Not so immutable: Upgradeability of Smart Contracts on Ethereum' (2022) 10.48550/arXiv.2206.00716
[35] Chen, J., Xia, X. and Lo, D. and Grundy, J. 'Why Do Smart Contracts Self-Destruct? Investigating the Selfdestruct Function on Ethereum', ACM Trans. Softw. Eng. Methodol., Vol. 1, No. 1, Article 1 (2020)
[36] Chatterjee A. and Hansdah R. C. 'Deploying Transactional Smart Contracts using Multisignature Boolean Formulas. In Proceedings of the 23rd International Conference on Distributed Computing and Networking (ICDCN '22). Association for Computing Machinery 170–174. https://doi.org/10.1145/3491003.3491014 (2022)

[37] Chakravarty, M., Chapman, J., MacKenzie, K., Melkonian, O., Müller, J., Jones, M., Vinogradova, P. and Wadler, P. 'Native Custom Tokens in the Extended UTXO Model', ISoLA (2020)
[38] Chaudhury, A. and Haney, B. 'Smart Contracts on Algorand', IUP Journal of Knowledge Management, Available at SSRN: https://ssrn.com/abstract=3887719 or http://dx.doi.org/10.2139/ssrn.3887719 (2021)
[39] Johnson S, Hyland-Wood D., Madsen A.L. and Mengersen K., 'Stateful to Stateless: Modelling Stateless Ethereum', Electronic Proceedings in Theoretical Computer Science (2022)
[40] Bartoletti, M., Bracciali, A., Lepore, C., Scalas, A. and Zunino, R. 'A formal model of Algorand smart contracts', eprint: arXiv:2009.12140 (2020)
[41] Antonopoulos A.M., 'Mastering Bitcoin', O'Reilly Media, Inc. (2014)
[42] Jansen, M., Hdhili, F., Gouiaa, R., Qasem, Z. (2020). Do Smart Contract Languages Need to Be Turing Complete? In: Prieto, J., Das, A., Ferretti, S., Pinto, A., Corchado, J. (eds) Blockchain and Applications. BLOCKCHAIN 2019. Advances in Intelligent Systems and Computing, vol 1010. Springer, Cham. https://doi.org/10.1007/978-3-030-23813-1_3
[43] Bistarelli, S., Mazzante, G., Micheletti, M., Mostarda, L., Tiezzi, F. (2020). 'Analysis of Ethereum Smart Contracts and Opcodes'. In: Barolli, L., Takizawa, M., Xhafa, F., Enokido, T. (eds) Advanced Information Networking and Applications. AINA 2019. Advances in Intelligent Systems and Computing, vol 926. Springer, Cham. https://doi.org/10.1007/978-3-030-15032-7_46
[44] Vyper: A Smart Contract Programming Language for the EVM. URL https://docs.vyperlang.org/en/latest/
[45] Hopcroft, J., and Ullman, J. 'Introduction to Automata Theory, Languages, and Computation', Addison-Wesley (1979).
[46] Teller, A. 'Turing completeness in the language of genetic programming with indexed memory,' Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, Orlando, FL, USA, (1994)
[47] Lantz, L., Cawrey, D. 'Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications', O'Reilly Media 1st edition (2020)
[58] Suvitha, M. and Subha, R. 'A Survey on Smart Contract Platforms and Features,' 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India (2021) pp. 1536-1539, doi: 10.1109/ICACCS51430.2021.9441970.
[48] Baird, K., Jeong S., Kim, Y., Burgstaller, B. and Scholz, B. 'The Economics of Smart Contracts', CoRR (2019)
[49] Chakravarty, M., Chapman, J., MacKenzie, K., Melkonian, O., Müller, J., Jones, M., Vinogradova, P. and Wadler, P. 'Native Custom Tokens in the Extended UTXO Model', ISoLA (2020)
[50] Ethereum Gas. URL: https://ethereum.org/en/developers/docs/gas/
[51] Destefanis, G., Marchesi, M., Ortu, M., Tonelli, R., Bracciali A. and Hierons, R. 'Smart contracts vulnerabilities: a call for blockchain software engineering?' 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE 2018), pp. 19-25, doi: 10.1109/IWBOSE.2018.8327567
[52] Wöhrer M. and Zdun, U. 'Design Patterns for Smart Contracts in the Ethereum Ecosystem', IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) (2018)
[53] Chen, J., Xia, X., Lo, D., Grundy, J., Luo X. and Chen, T. 'Defining Smart Contract Defects on Ethereum,' IEEE Transactions on Software Engineering, vol. 48, no. 1, pp. 327-345, 1 Jan. 2022, doi: 10.1109/TSE.2020.2989002
[54] Akca, S., Peng, C. and Rajan, A. 'Testing Smart Contracts: Which Technique Performs Best?', Proceedings of the 15th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM) (ESEM 2021)

[55] Beniiche, A. 'A Study of Blockchain Oracles', arxiv.org/pdf/2004.07140, (2020)
[56] Watanabe H., Fujimura S., Nakadaira A., Miyazaki Y., Akutsu A. and Kishigami J. 'Blockchain contract: Securing a blockchain applied to smart contracts', IEEE International Conference on Consumer Electronics (ICCE 2016) pp. 467-468, doi:10.1109/ICCE.2016.7430693
[57] Kannengießer N. et al., 'Trade-offs between Distributed Ledger Technology Characteristics', ACM Comput. Surv. 53.2, (May 2020)
[58] ISO/TR 23576:2020. Blockchain and distributed ledger technologies — Security management of digital asset custodians. URL: https://www.iso.org/standard/76072.html.

Appendix

For a better understanding of the text, please find attached this summary table of definitions[103].

Term	**Explanation**
ASSET	Anything of value to a stakeholder.
BLOCKCHAIN(S)	A method to manage a distributed digital ledger in which data is stored in blocks organized according to an *append-only* sequential chain, which uses cryptographic tools to validate the integrity of the data history through an algorithmic validation of the logic of a TRANSACTION and the confirmation of its registration by a pre-established consensus mechanism between the NODES that process the aforementioned TRANSACTIONS.
CODE	The language provides instructions to the computer. A distinction can be made between source code and *bytecode*. The source code is readable by those who know the programming language. Otherwise, *bytecode* is generally not readable by humans.
CRYPTOGRAPHY	Discipline that embodies the principles, means and methods for the representation of a message or, more generally, of data in a form that hides its semantic content, prevents its unauthorized use or prevents its non-detection, as well as its modification by third parties.
CRYPTO-ASSET	A digital representation of value (or, possibly, rights) that can be transferred and stored electronically, using DISTRIBUTED LEDGER TECHNOLOGY or a similar technology.
DIGITAL ASSET	See CRYPTO-ASSET.

[103] This table reproduces and adapts in the context of the document the definitions contained on pages 19 - 20 of the ELI Principles on Blockchain Technology, Smart Contracts and Consumer Protection, available at https://www.europeanlawinstitute.eu/fileadmin/user_upload/p_eli/Publications/ELI_Principles_on_Blockchain_Technology__Smart_Contracts_and_Consumer_Protection.pdf

Characteristics of Smart Contracts

DISTRIBUTED LEDGER TECHNOLOGY	A type of technology that supports distributed recording of encrypted data. A BLOCKCHAIN is a subcategory of the DISTRIBUTED LEDGER TECHNOLOGY.
NATURAL LANGUAGE	Language that people use to communicate with each other (e.g. Italian, English, French, German, etc.).
NODE	Device or system participating in a DISTRIBUTED LEDGER. Nodes can store a complete or partial copy of the DISTRIBUTED LEDGER.
OFF-CHAIN	Which is located, performed or performed outside of a BLOCKCHAIN system.
ON-CHAIN	Which is located, performed or performed within a BLOCKCHAIN system.
ORACLE	Device that updates a DISTRIBUTED LEDGER (e.g. a BLOCKCHAIN) using data from outside a DISTRIBUTED LEDGER (or outside a BLOCKCHAIN). An ORACLE transmits OFF-CHAIN information in a format readable by computers participating in the network.
PERMISSIONED BLOCKCHAIN	DISTRIBUTED LEDGER system where its NODES require an authorisation to carry out certain actions or activities, in particular the processing of TRANSACTIONS.
PERMISSIONLESS BLOCKCHAIN	DISTRIBUTED LEDGER system where its NODES do not require an authorisation to carry out certain actions or activities, in particular the processing of TRANSACTIONS.
PRIVATE BLOCKCHAIN	DISTRIBUTED LEDGER system in which a controlled and limited set of NODES participates in the operation of the system itself.
PUBLIC BLOCKCHAIN	DISTRIBUTED LEDGER system in which participation (as a NODE) in the operation of the system is not subject to controls or limitations.
DISTRIBUTED REGISTER	A data storage system intended to be conclusive, definitive and immutable and in which the archive itself is shared through a network of computers (NODES).

Characteristics of Smart Contracts

TRANSACTION	In this context, TRANSACTION means an action registered on the BLOCKCHAIN those results in a change of status on the BLOCKCHAIN itself. For example, a TRANSACTION is the transfer of CRYPTO-ASSET resulting in a reduction in the amount of CRYPTO-ASSET available to the owner of private key A and a corresponding increase in the amount of CRYPTO-ASSET available to the owner of private key B.
SMART CONTRACT	Computer program that, upon the occurrence of predetermined conditions, is automatically executed giving rise to predefined actions. A SMART CONTRACT may or may not represent the terms of a contract or be legally recognized. For the purposes of this document, SMART CONTRACTS are considered only in the context of DISTRIBUTED REGISTER SYSTEMS. However, it is acknowledged that SMART CONTRACTS are not limited to DISTRIBUTED REGISTER SYSTEMS and that the term may have a different meaning in other contexts.
VIRTUAL CURRENCY	A digital representation of value, which is not issued or guaranteed by a central bank or public authority and which is not necessarily linked to a legal tender currency, but which, is accepted by natural or legal persons as a means of exchange, as it can be transferred, stored and exchanged electronically.
WALLET	A device for storing private and public keys that enables DLT users to operate.

In addition, for ease of presentation, this document briefly refers to some regulatory sources. In the following table, for each normative source referred to in the text, the correct normative reference is indicated in full.

Regulatory reference in the text	**Regulatory reference**
Fintech Decree	Decree Law 25 of 17 March 2023 (note 40 of the document contains the number of the Decree), 'Urgent provisions on the issue and circulation of certain financial instruments in digital form and on the simplification of Fintech experimentation', converted with amendments by Law 52 of 10 May 2023
Simplification Decree	Decree Law 135 of 14 December 2018, 'Urgent provisions on support and simplification for businesses and the public administration', converted with amendments by Law 12 of 11 February 2019
DLT Pilot Regime	Regulation (EU) 2022/858 of the European Parliament and of the Council of 30 May 2022 on a pilot regime for market infrastructures based on distributed ledger technology and amending Regulations (EU) 600/2014 and (EU) 909/2014 and Directive 2014/65/EU
DORA Regulation	Regulation (EU) 2022/2554 of the European Parliament and of the Council of 14 December 2022 on digital operational resilience for the financial sector and amending Regulations (EC) 1060/2009, (EU) 648/2012, (EU) 600/2014, (EU) 909/2014 and (EU) 2016/1011
eIDAS Regulation	Regulation (EU) 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 and repealing Directive 1999/93/EC
MICA Regulation	Regulation (EU) 2023/1114 of the European Parliament and of the Council of 31 May 2023 on markets in crypto-assets and amending Regulations (EU) 1093/2010 and (EU) 1095/2010 and Directives 2013/36/EU and (EU) 2019/1937