Data Protection Challenges in Distributed Ledger and Blockchain Technologies: A Combined Legal and Technical Analysis*

Danaja Fabčič Povše[1], Alfredo Favenza[2], Davide Frey[3], Zoltán Ádám Mann[4], Angel Palomares[5], Lorenzo Piatti[6], and Jessica Schroers[7]

[1] Vrije Universiteit Brussel, Belgium - Danaja.Fabcic.Povse@vub.be
[2] LINKS Foundation, Italy - alfredo.favenza@linksfoundation.com
[3] Univ Rennes, Inria, CNRS, IRISA, France - davide.frey@inria.fr
[4] University of Amsterdam, Netherlands - z.a.mann@uva.nl
[5] Atos, Spain - angel.palomares@atos.net
[6] InfoCert, Italy - lorenzo.piatti@infocert.it
[7] KU Leuven, Belgium - Jessica.Schroers@law.kuleuven.be

Abstract. Blockchain and the blockchain-based cryptocurrency Bitcoin revolutionized our ideas of decentralized data and transaction management. Based on and improving on the ideas of blockchain, a variety of distributed ledger technologies (DLTs) have been proposed in recent years. DLTs promise fully decentralized data and transaction management, with wide-ranging applications that go well beyond cryptocurrencies. However, DLTs are also associated with challenges, particularly with respect to compliance with applicable data protection regulations. This chapter presents a comprehensive analysis of the challenges associated with DLTs' compliance with the General Data Protection Regulation (GDPR) of the European Union (EU). It analyzes the impact of these challenges on different types of DLT approaches (public or private, permissioned or permissionless). It shows that three fundamental properties of DLTs—immutability, decentralization, and automation—make it very difficult to comply with the GDPR in the public permissionless setting. For other DLTs, GDPR compliance is less problematic, but some challenges remain. In particular, the uncertainty that remains about the exact interpretation of the GDPR in the context of DLTs means that the question of GDPR compliance cannot always be definitely answered.

Keywords: Distributed ledger; DLT; Blockchain; Data protection; Privacy; GDPR

1 Introduction

The importance and impact of distributed ledger technologies (DLTs) grew quickly in recent years. The success of Bitcoin [1] inspired many other cryp-

* This chapter was published in: N. El Madhoun, I. Dionysiou, E. Bertin (editors): *Building Cybersecurity Applications with Blockchain and Smart Contracts*, pp. 127-152, Springer, 2024. https://doi.org/10.1007/978-3-031-50733-5_6

tocurrencies, including Ethereum, Litecoin, and Ripple. Blockchain, the technology underlying Bitcoin and several other cryptocurrencies, also found many other applications beyond the field of cryptocurrencies, including accounting and auditing, regulatory reporting, supply chain management, and healthcare [2].

A blockchain is a ledger that supports appending new items, but does not allow changing or deleting items that were added to the ledger in the past. The distributed and decentralized nature of the blockchain implies that multiple parties need to agree on the content of the ledger, without needing to trust each other or a central authority. Since there are a variety of designs with these properties which significantly differ from the original blockchain, we use the more general term DLT (Distributed Ledger Technology).

Although DLTs were introduced for cryptocurrency, their flexibility has enabled applications in a variety of contexts. For example, notaries have been looking at DLTs as a solution to digitize their profession[1]. Some projects have tried to use blockchains for the management of medical data [3]. Others like MIT's Enigma started off with a vision that Blockchain would be a tool to "protect personal data" [4]. But applying DLT to personal data results in significant challenges, in terms of privacy [5] and compliance with data protection regulations.

This chapter provides a combined legal and technical analysis of data protection in DLTs. In terms of data-protection regulations, we focus our attention on the General Data Protection Regulation (GDPR) of the European Union (EU), and cover all relevant provisions of the GDPR. In terms of DLTs, we look at different setups according to who can access the ledger (public vs. private, permissioned vs. permissionless DLTs). In particular, we identify three core properties of DLTs that are relevant to data protection: immutability, decentralization, and automation. We analyze the challenges of GDPR compliance stemming from each of these properties.

We give here an example of the implication of each of these properties. First, as already mentioned, DLTs do not allow changing or deleting items that were added to the ledger. But the GDPR stipulates the "right to be forgotten": enabling data subjects to have their data removed from storing and processing. This clearly contradicts the immutability of DLTs [6]. Second, DLTs are based on a decentralized architecture with no central authority. Yet, the GDPR relies on the identification of a special actor: the data controller, which is responsible for ensuring compliance with the regulation. Unfortunately, it is currently unclear how a data controller should be identified in DLT-based applications [7]. Finally, several DLTs support smart contracts: software programs that can be stored on the DLT and executed in the context of the DLT in an automated way. The concept of smart contracts significantly extends the possibilities of DLTs and has contributed to their take-up [8]. But their usage in the presence of personal data raises important challenges with respect to several rights and

[1] https://joinup.ec.europa.eu/collection/blockchain-egov-services/solution/blockchain-based-notary-proof-concept

principles stated by the GDPR, for example, the right not to be subject to solely automated processing.

Our findings indicate that these three core properties (immutability, decentralization, and automation) make it very difficult to comply with the GDPR in the public permissionless setting. For other DLTs (public permissioned, private permissioned), our analysis reveals a multifaceted situation. In these cases, GDPR compliance is less problematic, but several challenges remain. For example, our analysis of the literature reveals that there is still uncertainty about the exact interpretation of the GDPR in the context of DLTs, so that the question of GDPR compliance cannot always be clearly answered.

The remainder of this chapter starts by providing a background on DLTs and the GDPR in Section 2. Then, in Section 3, it classifies the challenges associated with the combination of DLTs, personal data, and GDPR by considering the three dimensions we suggested here: immutability, decentralization, and automation. Finally, Section 4 concludes the chapter.

An analysis of how different technical and legal approaches can be used to mitigate the identified challenges is given in our companion chapter [9].

2 Background

This section introduces the background that is necessary to understand the subsequent analysis. We start with the relevant technologies underlying DLTs in Section 2.1, followed by the legal background on data protection in Section 2.2 Readers familiar with the technical or legal background can safely skip the corresponding subsection. We also note that Section 2 in Chapter "Blockchain-Integrated Identity Verification for E-Government Services" earlier in this book [10] already provided some background on Blockchain and smart contracts. Here we simply introduce some additional concepts and precisely define some terms that are relevant to our analysis.

2.1 Technological Background

Although the term Blockchain has now become an everyday word, its key property lies in the ability to implement a ledger, i.e. an append-only list of items. For this reason, a number of authors have proposed the term "Distributed Ledger" and the acronym DLT for "Distributed Ledger Technology". DLTs put together concepts and ideas from different areas of computer science. In the following, we define the main concepts and ideas employed in distributed ledger technology, and discuss how they are used in this context. Throughout the discussion, we will use the term node to refer to a device that contributes to the operation of the DLT (a miner or a full node in Bitcoin terminology) and the term clients to refers to devices that people use to connect to nodes or to intermediary services such as online Wallets.

We also observe that significant research is currently being devoted to alternatives to the Blockchain model [11–16]. In particular, multiple research groups

have shown that consensus, and thus a blockchain, is not necessary to implement a money-transfer abstraction, or even some types of smart contracts [15, 16]. This has led to significant work on consensus-less primitives [13, 14, 17]. In the following, nonetheless, we will concentrate on distributed ledgers that follow the classical blockchain model and establish an append-only and immutable total order among operations or blocks of operations.

2.1.1 Types of Ledgers A first way to classify distributed ledgers stems from the environment in which they are designed to operate. In this respect, we can distinguish two types of environments: permissionless and permissioned [18].

- The **permissionless** setting consists of an environment in which devices can join and leave the system without any authorization. These systems can thus comprise an unbounded number of participants whose identities are not known a priori. The well known Bitcoin [1] and Ethereum [19] blockchains belong to this category. Anyone can join the network and participate in the protocol at any time.
- The **permissioned** setting [20,21], on the other hand, requires participating nodes to have the authorization of a managing authority, for example a company or a bank. In this case, the set of participants is generally known a priori and a change like the addition of a new participant must go through the managing authority.

The permissioned setting offers a simpler and more controlled environment, while the permissionless (also known as unpermissioned) setting opens the door to what is commonly known as Sybil attack [22], in which a node can impersonate a large number of identities (either other nodes or made-up identities), thereby influencing the outcome of the protocol.

The distinction between permissioned and permissionless refers to the ability of network nodes to participate in the DLT protocol. On the other hand, users can also interface with the ledger by operating as clients that access the DLT's data without being part of the DLT protocol. From the perspective of clients, we can thus distinguish public and private DLTs.

- In a **public** DLT, any client can access the system without any authorization and without being part of a pre-defined group.
- In a **private** DLT, clients must be registered with some managing authority and can only access the system if they have the appropriate authorization.

If all users operate as nodes, then the two classifications coincide, but in the general case, they lead to four combinations: private permissionless, public permissionless, private permissioned, and public permissioned. The private permissionless combination does not make much sense, as it would mean that clients need authorization while nodes do not. So we obtain three meaningful types of DLTs:

- **Public permissionless DLT**: public ledger with no access restrictions. Anyone with an Internet connection can send transactions, become a block

validator, and participate in the execution of a consensus protocol. Examples include Bitcoin [1] and Ethereum [19].
- **Private permissioned DLT**: DLTs placing restrictions on who is allowed to participate in the network and in what transactions. They are mainly useful for business and industrial applications. In fully private permissioned DLTs, write permissions are kept centralized to one organization, while in consortium DLTs they may comprise a consortium of organizations. Examples of private permissioned DLTs are based on platforms like Hyperledger[2] and Corda[3].
- **Public permissioned DLT**: a new type of network that fills the gap between public permissionless and private consortium networks. A public permissioned blockchain network combines the permissioned nature of a private consortium with a decentralized governance model, trying to achieve the best properties of both models. This approach makes it possible to obtain features required for the implementation of use cases that do not fit any of the previously explained models. Alastria[4], EOS[5] and Ripple[6] are examples of such DLTs.

2.1.2 Inner workings of a ledger A distributed ledger essentially consists of two main components: a distributed data structure and a consensus protocol that allows nodes to agree on the content of the data structure. The most common type of data structure consists of a list of blocks, commonly known as a blockchain, but also other structures exist. For example, some ledgers operate by creating DAGs (directed acyclic graphs) of blocks or transactions. But in all cases, ledgers rely on a set of basic notions in the context of cryptography and distributed systems, which are described next.

2.1.3 Cryptographic background of DLTs On the cryptography side, all ledgers rely on cryptographic hash functions and public key cryptography. In the following, we define these terms.

Cryptographic Hash Functions. The term "hash function" generally refers to any function that takes an arbitrary size input and outputs a fixed size output. In the DLT context, the term refers to the more specific category of cryptographic hash functions. Any hash function must be deterministic (always return the same hash value for the same input), but a cryptographic hash function must also satisfy additional constraints:

1. It should be quick to compute.
2. It should be a one-way function: that is, it must be time-consuming and expensive to generate an input from its hash value.

[2] `https://www.hyperledger.org/`
[3] `https://www.corda.net/`
[4] `https://alastria.io/en/`
[5] `https://eos.io/eos-public-blockchain/`
[6] `https://ripple.com/`

3. It should present an avalanche effect: any small change in the input should produce big changes in its hash value.
4. It should be collision resistant: it should be unfeasible to find two different inputs with the same hash.

Property # 2 implies that the only way to go back to the original from a hash is to try all the possible variations to see if they produce a match, which is a time-consuming and very expensive task. For example, the Bitcoin network uses Secure Hash Algorithms (SHA), such as SHA-256 [23] to implement a crypto-puzzle. Property # 3 entails that if one single bit of input data is changed, the output changes significantly. For brevity, in the following, we will use the term hash function to refer to a cryptographic hash function with the above properties.

Public-Key Cryptography. Public-key cryptography belongs to the family of asymmetric cryptography systems. It is based on the use of two keys, in order to overcome the limitations of symmetric crypto-systems based on a single key. In a symmetric system, participants need a secured way (e.g., a physical meeting) to agree on the key to be used for the communication, which is not feasible for modern network-based communication. Public-key cryptography solves this issue by introducing two keys: the private key and the public key. The private key is only known by the owner and needs to be kept private, while the public key should be given to anyone in the network in order to be used by anyone to send encrypted messages to the owner of the public key. Messages encrypted with the public key can only be decrypted with the corresponding private key.

Zero Knowledge Proof (ZKP): ZKP is a cryptographic technique that can be used to hide a piece of information, while still making it possible to perform verification on this data. For example, a prover can use a ZKP to prove to a verifier that they own some secret data without leaking the actual content of this data. The verifier only knows the fact that the prover owns this data. For example, in the context of a cryptocurrency, a ZKP can be used to generate proof that the prover has enough funds for a transaction without revealing the exact amount it owns [24].

2.1.4 Distributed Systems background of DLTs From a systems' perspective, a DLT can be viewed as a distributed state machine [25, p. 313], an abstract object that creates the illusion of having a widely available unified computing system regardless of distribution or failures [26]. This has two direct implications. First, a ledger can support Turing-complete languages as done in the Ethereum cryptocurrency [19]. Second, it requires the ability to solve distributed consensus [26].

Distributed consensus models a set of participating devices that need to agree on an outcome. Each participant proposes a value, and at the end of the algorithm, each decides on a value. In the case of a ledger, the value can for example be the content of the next block that should be appended. A major result by Fischer, Lynch and Paterson [27] states that consensus cannot be solved in the presence of failures (even benign crash failures) in an asynchronous system

(i.e., where messages can experience unbounded delays). Intuitively, this results from the impossibility to distinguish a failed participant from an extremely slow one. In practical systems, this impossibility can be circumvented in several ways: e.g., assuming that communication is partially or eventually synchronous [28–30], or by employing randomized algorithms with probabilistic guarantees. This has led to the appearance of several systems that effectively exploit distributed consensus in the presence of crash [31] and byzantine (arbitrary) failures [32–36].

Consensus in permissionless settings. Classical distributed consensus algorithms [28–36] operate under the assumption that the distributed system consists of a predefined number of well identified participants. One of the novelties of blockchain protocols was to operate in an open (permissionless) environment, in which devices can join and leave the system without needing to register the fact that they are joining the system. This requires solutions to withstand identity-based attacks such as the Sybil attack, in which a node can impersonate a large number of identities.

As a result, classical consensus algorithms do not fit the permissionless setting defined by public ledgers. For this reason, Bitcoin and subsequent designs have based their operation on a novel family of consensus algorithms that we can name Proof-of-*. Proof-of-* follows a well-known means to achieve distributed consensus. It elects a leader, and then it allows this leader to take one (e.g., Bitcoin, Ethereum) or more (e.g., BitcoinNG) decisions. This model was already used in the closed context classical distributed systems [36–38], but the novelty of proof-of-* consists in enabling leader election, and thus consensus, in permissionless systems in the presence of Byzantine and Sybil attacks. The very nature of proof-of-* protocols consists in a Sybil resilience mechanism that makes it hard for attackers to generate arbitrary identities.

The best known proof-of-* protocol is the Proof-of-Work (PoW) invented by Malkhi [39] and first used in the context of DLTs by Bitcoin's creator, Satoshi Nakamoto [1]. Given the heavy utilization of energy and computational power consumed by PoW, other consensus mechanisms have emerged in the last few years to mitigate PoW inefficiencies, such as Proof-of-Stake (PoS), Delegated-Proof-of-Stake (DPoS), Proof-of-Burn (PoB), Proof-of-Authority (PoA), Proof-of-Elapsed Time (PoET), Proof-of-Capacity (PoC) [40].

Consensus in Permissioned Settings. Even if we were to ignore its high computational cost, PoW becomes much less appealing in permissioned settings, where attacks like the balance attack [41] become easier than in permissionless ones. But even other Proof-of-* solutions become less interesting. On the one hand, the main motivation for Proof-of-* solutions, the possibility of Sybil attacks, disappears in a permissioned setting where all participants are known. This has led the major permissioned solutions to adopt deterministic BFT consensus protocols from the domain of distributed computing. While these cannot currently operate in large-scale systems in permissionless settings, they present two major advantages with respect to Proof-of-*. First, they can achieve very high throughput. Second, they provide decision finality. In proof-of-*, a decision can be reverted with a probability that decreases with its depth in the chain.

BFT protocols make decisions final as soon as they are taken and do not even need a chain to achieve consensus [36, 42].

2.1.5 Automation and Smart Contracts Chapter "Blockchain-Integrated Identity Verification for E-Government Services" [10] already defined smart contracts. Here we simply observe that Smart contracts can bring several advantages with respect to traditional contracts, such as increased security (data managed through a smart contract are immutable and tamper-proof), increased speed (all transactions are automated and quicker than a manual system), increased accuracy (smart contracts automatically follow their built-in rules, greatly reducing the potential for error and simplifying verifiability by third parties), decreased cost (smart contracts streamline transactions and effectively remove the middleman, thereby lowering transaction cost)[7]. But they can also introduce new challenges as we discuss in the following.

A DAO (Decentralized Autonomous Organization) is a complex smart contract or a set of smart contracts. A DAO can be used to manage elaborate situations in which more parties (Data Subjects) are involved. DAOs usually implement decision-making systems to enable an online community to reach agreements. As a result of these agreements, the DAO operates automatically by executing the appropriate portion of code on the blockchain network [43].

2.2 Data protection under the GDPR

The European Union (EU) considers data protection to be a policy priority and a critical issue on the path of digitizing society toward a citizen-friendly single digital market [5]. The EU has adopted the General Data Protection Regulation (Regulation 679/2016, GDPR) which contains basic principles of data protection along with obligations, rights and duties for each of the data environment stakeholders.

2.2.1 The GDPR's scope of application The GDPR applies to the processing of personal data. The notion of 'personal data' covers "any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person" (art. 4 (1) GDPR).

Processing means "any operation or set of operations which is performed on personal data or on sets of personal data, whether or not by automated means" (art. 4 (2) GDPR), which includes every action done with personal data: the sole fact of starting data collection triggers the application of the Regulation.

[7] https://blockchain.ieee.org/topics/smart-contracts-and-energy-how-blockchain-smart-contracts-can-improve-the-energy-sector

In certain cases the GDPR is not applicable, for example when data processing is carried out by a natural person in the course of a purely personal or household activity (the household exemption), or when processing data that have been anonymized.

Anonymization of personal data means that the data is irreversibly de-identified [44]. This is different from pseudonymization or encryption, where the original data can be re-identified using appropriate additional information. Pseudonymized and encrypted data is still personal data, since it can be related to a natural person with additional information, such as a decryption key. Therefore, in case of pseudonymization or encryption, the GDPR is applicable.

It is not always clear whether a person can be considered identifiable. A decisive factor in this regard is the concept 'means likely reasonably to be used' [45]. There are two main approaches for interpreting this concept: the 'absolute' and the 'relative' approach [46]. The difference is whether the means must be available to anybody (absolute) or only to the controller (relative). For DLTs, the difference is important in the case of off-chain storing of personal data and public keys (see also the section on keeping personal data off-chain in our solutions chapter [9]):

- Following the absolute approach, the pointer on the blockchain to the off-chain data could be considered personal data since it relates to an identifiable person (identifiable via the additional information available off-chain). Since public keys can potentially be connected to information about a person, they would be considered personal data.
- Following the relative approach, it would only be personal data if the party processing the data on the blockchain is indeed able to obtain the off-chain information [46].

Currently, the direction of interpretation in practice seems to be towards the absolute approach, based upon the wording of the GDPR and the Article 29 Working Party opinions [46].

The European Court of Justice (CJEU) in the Breyer decision [45] considered that the possibility to combine a dynamic IP address with the additional data held by the internet service provider (ISP) could constitute a means likely reasonably to be used to identify the data subject. According to the case, it is important to consider whether identification is reasonable. Since legal channels exist to obtain information from the ISPs, IP addresses are considered personal data. However, it would not be considered personal data if the identification of the data subject was prohibited by law or practically impossible due to the required disproportionate effort in time, cost, and manpower, resulting in an insignificant risk of identification. Moreover, due to the way the question to the court was phrased, it is not entirely clear whether the court follows the relative approach [46].

2.2.2 Principles of data processing in the GDPR According to art. 5(1), data processing must be carried out in accordance with the following principles:

lawfulness, fairness and transparency; purpose limitation; data minimization; accuracy; integrity and confidentiality. Moreover, according to the art. 5(2) the data controller is responsible for, and must be able to demonstrate compliance (the accountability principle).

2.2.3 Role of the Data Controller Central to data protection law is the notion of the data controller, described in the GDPR as "the natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data". It can do so alone or jointly with other entities (in that case we speak of joint controllers) (art. 4(7)). The data controller carries the general obligation of compliance (art. 5(2) and 24 GDPR). It may entrust a data processor to process data on the controller's behalf, without having the power to determine its means or purposes. In practice, it may be difficult to distinguish between controllers and processors due to technological and societal developments as well as the lack of legal flexibility [47]. Furthermore, it is required to take appropriate measures to facilitate the exercise of data subjects' rights and to provide data subjects with clear information about their rights (art. 12(1) and 12(2) GDPR).

2.2.4 Data subject rights The GDPR gives the data subject the following rights in articles 13-22:

- Information rights (art. 13 and 14 GDPR): With regard to the transparency principle, the data subject has the right to receive information.
- Right of access (art. 15 GDPR): The data subject has the right to get confirmation from the controller whether his/her personal data are processed, and in such a case, access to the personal data and certain information, including the existence of automated decision-making/profiling, or about the existing safeguards if personal data is transferred to a third country. The right of access also specifies that the controller shall provide a copy of the processed personal data, as long as it does not adversely affect the rights and freedoms of others.
- Right to rectification (art. 16 GDPR): The data subject may obtain rectification of inaccurate personal data without undue delay. This right includes a notification obligation of the controller if the data has been transferred to other controllers, unless this proves impossible or involves disproportionate effort (art. 19 GDPR).
- Right to erasure (art. 17 GDPR): Applies under certain circumstances such as when the data is no longer necessary, the data subject withdraws consent or objects to the processing, the data have been unlawfully processed or must be erased for compliance with legislation, or if the data have been collected in relation to the offer of information society services. This right also includes a notification obligation of the controller if the data has been transferred to other controllers, unless this proves impossible or involves disproportionate effort (art. 19 GDPR).

- Right to restriction of processing (art. 18 GDPR): Under certain circumstances and for certain periods of time, the data subject can require the controller to restrict the processing of his or her personal data. If the right is employed, then again a notification obligation of the controller exists if the data has been transferred to other controllers, unless this proves impossible or involves disproportionate effort (art. 19 GDPR).
- Right to data portability (art. 20 GDPR): The right to data portability enables the data subject to transfer data between different controllers. It requires the controller to provide the data in a structured, commonly used and machine-readable format and that the data subject has the right to transmit those data to another controller without hindrance from the original controller, and where technically feasible and requested by the data subject, the personal data should be directly transmitted from one controller to another.
- Right to object (art. 21 GDPR): The data subject has the right to object against the processing of his or her personal data on grounds relating to his or her particular situation. This right can only be invoked by the data subject if the processing is based upon public interest or the legitimate interest of the controller, or processed for direct marketing purposes.
- Right not to be subject to a decision based solely on automated processing (art. 22 GDPR): The data subject has the right that decisions that produce legal effects or similarly significantly affect him or her, should not be solely based on automated processing, including profiling. There are certain exceptions to this right, e.g. if it is necessary for entering into, or performance of, a contract between the data subject and a data controller; if it is authorized by Union or Member State law to which the controller is subject and which also lays down suitable measures to safeguard the data subject's rights and freedoms and legitimate interests; or if it is based on the data subject's explicit consent.

2.2.5 Transfer of personal data to other jurisdictions Many organizations that store or process personal data are located outside the EEA (EU, Norway, Iceland and Liechtenstein). Such data transfers are only compliant if the level of protection guaranteed by the GDPR is not undermined (art. 44 GDPR) [48]. From a legal perspective, the GDPR defines four options for establishing that a non-EEA country satisfies the required standards of data protection: (1) adequacy decisions (art. 45 GDPR), (2) appropriate safeguards (art. 46 GDPR), (3) binding corporate rules (art. 47 GDPR), and (4) exception-explicit consent (art. 49 GDPR).

- Adequacy decision: only allowing transfers to countries which the Commission considers adequate in their level of protection (art. 45 GDPR, examples are UK, Israel, and South Korea).
- Appropriate safeguards: if appropriate safeguards are ensured (art. 46 GDPR, meaning inter alia adopting measures preventing access by national security agencies).

- Binding corporate rules: if the transfer is within the same group of undertakings, subject to binding corporate rules and having been confirmed by a supervisory authority (art. 47 GDPR).
- Exception-explicit consent: if the data subject has given explicit consent after being informed about the possible risks. A transfer under this exemption is only possible if not prohibited by EU or national law (art. 49 GDPR).

A large number of countries do not fall into any of these categories. For example, the US, which, in the case of Bitcoin, comprises 16% of all the identified nodes[8], is not considered by the European Commission to have an adequate level of protection. Until 2020, US-based data controllers relied on the Privacy Shield Agreement, through which they were able to comply with some essential data protection requirements. However, the agreement was invalidated by the Court of Justice in its Schrems II decision [49]. The same decision also invalidated the Commision-approved standard contractual clauses. Although a new regime is being discussed on the political level, controllers are now left with the option of transferring data based on binding corporate rules within their own group of undertaking or appropriate safeguards under art. 47 of the GDPR. Both of these appear difficult to apply in a blockchain context.

3 Data protection challenges of DLTs

This section gives an overview of the main challenges in data protection in connection with DLTs. DLTs have some intrinsic features that, on the one hand, make them suitable for specific purposes and use cases, but on the other hand, may give rise to some potentially severe data protection issues. The key properties of DLTs that make data protection challenging are **immutability**, **decentralization**, and **automation**. We start by reviewing the challenges stemming from these properties, and then provide some practical examples of the challenges. Table 1 provides a visual summary of the issues discussed in this section.

3.1 Challenges resulting from the immutability of DLTs

Immutability constitutes one of the pillars of DLT solutions. In the context of DLTs, immutability refers to the fact that information in old-enough blocks cannot be changed [50]. This constitutes a challenge from a data protection point of view, in particular for data subject rights and the implementation of data protection principles. Since the data stored in a block normally cannot be modified afterward, means that it is not possible to comply with certain data subject rights, in particular the **right to rectification** (art. 16 GDPR) and the **right to erasure** (art. 17 GDPR), since the data cannot be changed or erased at request. Furthermore, the immutability arises from the regular verification that the hashes of the data on the chain are still correct, which means that the data on the chain is processed. This gives rise to some uncertainty regarding

[8] `https://bitnodes.io/`

the **right to restriction of processing** (art. 18 GDPR) and the **right to object** (art. 21 GDPR). With respect to the former, it has not yet been clearly decided whether verifying the content and hashes of a blockchain would mean an infringement of the restriction of processing or whether this could be considered functional to storage and therefore it would fall under the storage exemption. The **right to object** (art. 21 GDPR) does not have such a storage exemption, but the fact that processing is functional to the operation of the DLT could constitute grounds to invoke the legitimate interest of the controller to continue using the DLT. However, this would have to be evaluated on a case-by-case basis with a careful balancing analysis.

The immutability of DLTs also threatens data protection principles, which may overlap with data subject rights. The inability to rectify incorrect data makes it impossible to comply with the **accuracy principle** (art. 5 (d) GDPR). Similarly, the inability to remove or anonymize data clashes with the **principle of storage limitation** (art. 5 (e) GDPR): data can legally be stored at length only for archival purposes in the public interest, scientific or historical research purposes, which is not the case in many DLT use cases. The **principle of purpose limitation** (art. 5 (b) GDPR) requires that personal data be collected only for specified, explicit and legitimate purposes and not further processed for a different purpose, except if it can be argued that the processing is compatible with the original purpose. The challenge is establishing whether processing on a DLT is really needed for the purpose at hand. This can only be dealt with on a case-by-case basis. Similarly, the **principle of data minimization** (art. 5 (c) GDPR) requires that data should be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed. Also, this principle could be breached if the data is still further processed in the DLT: since the data cannot be deleted, it is known from the start that in the end more data would be processed than necessary. Immutability also poses problems if **consent** is used as legal ground for processing, as the data subject has the right to withdraw consent at any time.

Moreover, the GDPR requires the implementation of **data protection by design and by default** (art. 25 GDPR). In a DLT, this means that the controller, when deciding on the technology to be used for processing, but also during the processing, should implement appropriate technical and organizational measures to implement data protection principles and to integrate the necessary safeguards into the processing to meet the requirements of the GDPR and protect the data subject rights. Furthermore, the controller must make sure that by default, only personal data which are necessary for each specific purpose of the processing are processed. However, if the controller implements a DLT with the safeguards that seem sufficient today, but later the safeguards become insufficient, then the immutability of the DLT may prohibit the controller from moving to another solution.

3.2 Challenges resulting from the decentralization of DLTs

The decentralized nature of DLTs leads to two categories of challenges. First, the potentially large number of users and devices that can constitute a blockchain makes it particularly hard to identify the roles they should take in the application of the GDPR. Second, the geographical spread of these nodes, together with the replication strategies adopted by DLTs, causes data to be stored and processed across diverse jurisdictions, making compliance with the GDPR particularly difficult.

Several principles and rights laid out in the GDPR, starting with, but not limited to, the **principle of accountability** of art. 5(2) and **general responsibility** under art. 24 rely on the identification of a party as the data controller. However, identifying roles (data controller, data subject, and data processor) constitutes a fundamental challenge in a DLT that handles personal data, since data subject rights are at stake if a controller cannot be identified [51]. The decentralized nature of distributed ledgers makes it impossible to identify these roles without ending up with absurd interpretations. For example, [7] discusses how the decentralized and peer-to-peer nature of permissionless ledgers only allows two solutions for the identification of the controller: either each node is a controller, or there is no controller. Both of these solutions would result in the inapplicability of GDPR to the blockchain.

A more reasonable stance with respect to this challenge is put forward by Moerel, who observes that DLTs constitute a form of general-purpose technology, like the Internet itself [52]. Therefore, the GDPR should not be directly applied to DLTs, but to the applications built on top of DLTs. For example, a company or user that uses a smart contract to process personal data would become the data controller, the nodes running the blockchain, and thus executing the contract would be at most data processors following the definition under art. 4(8). In this respect, ledger nodes can be compared to nodes in the cloud that store data on behalf of some client of the cloud platform [53].

The other challenge related to distribution results from the geographical spread of blockchain nodes, which may be located anywhere in the world. The GDPR requires that **transfers to third countries** can only take place insofar **an adequate level of protection** of personal data can be guaranteed (art. 45) or other guarantees are put in place. Since most blockchain platforms adopt a total replication strategy, i.e., they replicate each block on all the nodes that make up the system, this raises not only scalability issues, but it also implies that data will probably be stored outside the EU, possibly in countries that do not provide an adequate level of data protection guarantees. This problem presents significant challenges, particularly in the context of permissionless blockchains. A public permissionless blockchain may involve nodes from any country, and, at least in currently deployed systems, no one can have control on where nodes, let alone clients, are located. This makes permissionless blockchains the most challenging scenario from the point of view of the tension between GDPR and decentralization.

Paradoxically, while determining a controller on blockchains can be challenging, sometimes blockchains can instead lead to easier designation of a controller in some instances [54]. This is due to the persistence and record-keeping functions of the blockchains – shipping companies such as Maersk use blockchains to enhance transparency in their supply chains.

Lastly, true decentralization involves the possibility of global use of blockchain technology: this means that a user — data subject in the GDPR semantics — can access services at any time and without any control. It becomes, therefore, difficult to show privacy notice, in compliance with Article 12, or—if necessary—to collect consent to processing, in compliance with Articles 6(1)(a) and 7.

3.3 Challenges resulting from the automation in DLTs

Automation—as described in Section 2.1.5—is relevant to the GDPR for three main reasons. First, because the regulation sets a specific perimeter for automation in data processing. Second, because automation can be a tool for unlawful processing by third party or external code invocation. Third, automation may create some friction with certain GDPR principles, as explained below.

Regarding the first issue—specific perimeter—the European regulator focuses on three points:

- *Automatic decision-making.* According to article 22, The data subject has the right not to be subject to a decision based solely on automated processing, which produces legal effects concerning him or her or similarly significantly affects him or her. In a DLT environment, there are two types of applications that can potentially threaten this right: smart contracts and Decentralized Autonomous Organizations (DAOs). Using smart contracts and/or DAO solutions obliges the data controller—assuming that a data controller can be identified—to implement suitable measures to safeguard the data subject's rights and freedoms and legitimate interests, at least the right to obtain human intervention on the part of the data controller, to express the data subject's point of view and to contest the decision. Also, Data Controller shall then communicate in an accurate way how the smart contract itself works, according to Recital 63 "Every data subject should therefore have the right to know and obtain communication in particular with regard to the purposes for which the personal data are processed, where possible the period for which the personal data are processed, the recipients of the personal data, the logic involved in any automatic personal data processing and, at least when based on profiling, the consequences of such processing".
- *Profiling.* The definition of profiling is also based on the concept of automation, as—per recital 71—it is "any form of automated processing of personal data consisting of the use of personal data to evaluate certain personal aspects relating to a natural person". Under this provision, the data controller has the burden to grant a correct and specific legal basis for the processing.
- *Portability.* Automation is also relevant under article 20 (1) (b), which grants to the data subject the right to portability when "the processing is carried

out by automated means". According to this provision, Data Controller shall grant the data portability each time the processing is made by tools — in our case smart contracts/DAOs — that enable automated processing.

Regarding the second issue—unlawful processing—automation can be misused in a Blockchain environment. Open access and open use of DLTs may make GDPR compliance difficult: third and unknown parties, which may not be allowed for a specific data processing, may use the related data. For example, in a scenario where a specific set of data is used within automation processes without authorization criteria, the vast number of subjects who can—by exploiting the automatism and the functioning of smart contracts—access personal data could make it difficult to reach a GDPR-compliant approach. A specific example could be the misuse of a database with the records of a decentralized gaming platform: the operation of the platform would have a very high degree of automation and access to data could be indiscriminate in the absence of precise rules in this regard.

Regarding the third issue—friction with certain GDPR principles—automation in DLTs may affect further general rules and duties as follows:

- *Principle of lawfulness, fairness and transparency:* automation by smart contracts may violate this principle, as the data subject could be faced with a computer object that is not intelligible and therefore violates the principle of transparency. The same could apply also to lawfulness and fairness, given that the tool used may not reflect all the information necessary for the data subject to be aware of how their data is being processed, resulting in a non-compliance of article 5(1)(a).
- *Principle of confidentiality:* smart contracts create a mechanism whereby access to data—in the event of incorrect, or malicious, implementations—may be indiscriminate, violating the principle in question.
- *Controller shall maintain a record of processing activities under its responsibility (art. 30 GDPR):* in a highly automated context, it is difficult to analyze and identify each treatment. Smart contracts enable massive treatments, often independent of the will of the individual. Without proper tools, it may be difficult to keep track of individual operations, despite the transparency of the Blockchain.
- *Controller shall notify personal data breach (art. 33):* identifying and reporting data breaches could be difficult in a context where data access is available to software that automates processing, e.g. smart contracts. If there are no precise authorization rules, it may be hard to identify when and how an access has caused a data breach. Moreover, in its broadest sense, the data breach includes the impossibility to access the data, and a process exploiting DLT automation may lead to a situation in which a software malfunction prevents the use of some personal data.

As an example of the above three issues, we can imagine a scenario where a seller (acting as a data controller) of digital goods or services may use a smart contract that automatically processes data about a buyer (data subject). The

seller uses the buyer's data to create, complete and deploy the smart contract to manage the sale and the execution of the contract between the parties. These data, once entered into the blockchain, will be used automatically, by both the seller and the buyer, for the purpose set out in the contract, but also by other actors, who can call the same contract for related purposes.

Another example could arise in connection with DAOs. A DAO (Decentralized Autonomous Organization) is a complex smart contract or a set of smart contracts. A DAO can be used to manage elaborate situations in which more parties (data subjects) are involved. A DAO is used to autonomously manage different rights and functions of the "Organization", such as voting and sharing. DAOs, like smart contracts, may pose a threat to personal data under two circumstances. On the one hand, they can access personal data on databases, making themselves a tool for data processing. On the other hand, they receive information on how users of the DAO behave and interact, creating a potential profile of the data subjects. Such an organization may—under the abovementioned circumstances—process a huge amount of personal data, exacerbating the problems of smart contracts. Each participant gets one or more tokens which can be used to express preferences on a certain topic: for example, to automate crowdfunding or complex company actions and decisions. The movement of tokens and related decisions can be tracked on the ledger, potentially exposing personal data regarding choices or other actions and relationships of the data subjects.

A third example of DLT automation arises from cross-chain smart contracts [55]. Cross-chain smart contracts are decentralized applications composed of multiple smart contracts deployed across different DLTs that interoperate to create a single application. Decentralized services built on a single chain are often inefficient for enterprise applications that generate millions of transactions per second. Cross-chain approaches can increase computational efficiency, but also aggravate issues regarding the protection of data in a cross-chain application with intensive use of automated smart contracts. These issues fall within the area of interoperability, which is an emerging trend in DLT research, and require solutions to preserve data privacy across different blockchain networks when interactions between multiple smart contracts happen.

3.4 Practical Examples

We now consider three practical examples that further highlight the challenges resulting from the use of DLTs for the storage or management of personal data. We start by considering the case of public-keys (Section 3.4.1) which pose challenges even if they constitute an integral part of the DLT ecosystem. Then we discuss managed wallets (Section 3.4.2), and finally (Section 3.4.3), we consider one of the current killer applications of DLTs: Self Sovereign Identity systems.

3.4.1 Public keys Public keys are often used in the context of DLTs, and their relevance to data protection represents an interesting dilemma. Since it is

Table 1. Summary of challenges

	Public permissioned	Public permissionless	Private permissioned
IMMUTABILITY			
1st challenge: Irreversibility of DLT ⇒ challenges for data subject rights	Possibility to change content, ledger not immutable, depends on consensus mechanism & number of nodes. 0 +	Very challenging to comply with data protection rules. + +	Possibility to change content, ledger not immutable, depends on consensus mechanism & number of nodes. 0 +
DECENTRALIZATION			
2nd challenge: Identification of Controllers and Processors	Nodes are identified and authorized to create the ledger, data protection rules are enforceable. 0	Nodes not identified nor authorized to create the ledger, data protection rules are NOT enforceable. + +	Nodes are identified and authorized to create the ledger, data protection rules are enforceable. 0
3rd challenge: Transfer of data outside the EU	Restrictions on location can be implemented, data protection rules are enforceable. 0	No clear solutions for restricting node location, data protection rules are NOT enforceable. + +	Restrictions on location can be implemented, data protection rules are enforceable. 0
4th challenge: Consent management in a decentralized environment	Reading non-authorized ⇒ difficult to design correct consent management procedure. +	Reading non-authorized ⇒ difficult to design correct consent management procedure. +	As reading is authorized, consent and privacy notice can be managed. 0
AUTOMATION			
5th challenge: Automation of decision made with personal data,	Challenging to implement a correct data protection approach. +	Challenging to implement a correct data protection approach. +	Solvable with the correct data protection approach (consent or other legal basis). 0

0 **Not a problem**: due to the technological characteristics of the given type of DLT, the challenge does not pose a problem.

\+ **Issue**: the challenge does pose an issue, but it can be easily solved with a legal or a technical solution, without distorting the DLT approach.

\+ \+ **Big issue**: the challenge does pose an issue, which cannot be easily solved, neither with a legal nor with a technical work-around.

often possible to connect public keys—with additional information— to natural persons, thus allowing the identification of a natural person, public keys can generally not be considered anonymous, but rather pseudonymous [7, pp. 13-16] [56, p. 40] [57, p. 95].

Whether a public key is personal data or not may depend on information that is not directly visible. A public key consists of a number and is often represented by a string corresponding to the hexadecimal encoding of the number. In general, it is impossible to determine, by looking at a public key, whether it refers to a natural person. Even when it does, the additional data that makes this association concrete may not be available to any third parties. In some cases, there may be a company that serves as an interface between a person and the blockchain system, and that company records a link between this person's public key and his/her identity or other personal information. In other cases, this additional data may only exist on devices that are entirely under the control of the person to whom the public key refers. This is for example the case for a user that connects to a blockchain platform from his/her own computer using a client program or using a full-fledged blockchain node.

The French data protection authority, CNIL, considers public keys as personal data that are essential for the proper functioning of the blockchain [58]. But in general, there is no clear agreement in the literature as to the status of public keys as personal data. Nonetheless, since they may represent personal data, the challenges discussed in Sections 3.1-3.3 may apply to public keys.

Let us consider immutability, data subjects whose public key is processed in the DLT will not be able to use their data subject rights, such as the right to rectification and the right to erasure. The public key will stay in the DLT, and the principle of storage limitation or data minimization cannot be complied with. As it is not possible to comply with data protection principles and data subject rights, the use of DLTs with public keys would mean that the principle of data protection by design and by default would not be complied with. Considering the decentralized nature of DLTs, it would also not be possible to identify who is responsible (i.e., the controller) for the processing of the public keys, and the public key might be transferred to third countries. Similarly, in most cases it is difficult if not impossible to ensure the art. 13 right of information with respect to processing of public keys. It is, in fact, difficult to provide the information about the processing to the data subjects, if only their public key is known.

Finally, a further issue, related to public keys, and to any other identifier, is that they can enable linkability [59] between different records on the blockchain. For example, the ability to determine the amount in a person's Bitcoin wallet, relies on the fact that this person uses the same public key for all their transactions.

3.4.2 Wallets and addresses Blockchain is a public ledger where transaction information is stored. Each transaction happens between addresses. Wallets are a way to process and manage addresses. As long as the identity of the natural person behind an address or a wallet is not disclosed, there is no trace of

personal data. However, ensuring that an address or a wallet cannot be linked to a natural person is challenging. Attackers may correlate different transactions involving the same address or may combine on-chain transaction history information with publicly available or easily obtainable off-chain information to break the anonymity of addresses or wallets [60]. Users wallet address are public and, although blockchain network are pseudonymous, they can be tracked, potentially revealing transaction patterns and user behavior [61] and posing a significant challenge for blockchain wallet users who desire anonymity. Another potential issue is related to private key management, where the security of the wallet depends on the safeguarding of the private key associated with the user's public address [62]. An inadequate storage and management of private keys can expose the wallet to theft, unauthorized access, or loss of funds. This is particularly true when wallets rely on third-party services to provide additional features such as data storage, exchange services, or multi-signature functionality [63]. Integrating with these services can introduce new security vulnerabilities and data protection risks if the third-party providers do not adhere to strict security standards.

3.4.3 Self-Sovereign Identity Self Sovereign Identity (SSI) has emerged in recent years as a novel paradigm that seeks to improve over centralized or federated identity management systems. Essentially, SSI seeks to put users back at the center of the management of their digital identities by making them not only user-centric, but also completely decentralized.

Typically, SSIs are build on the concept of anonymous credentials. An anonymous credential (AC) consists of a cryptographic primitive that makes it possible for a user to prove some aspects of their identity without disclosing any information other than what they want to prove. In other words, any participant in the SSI can issue a credential to any other or verify someone else's credential. Essentially, an SSI system consists of three roles: users, issuer, and verifier. A user can be any person or device who wishes to access a service. An issuer is any entity that issues credentials to users. A verifier is whoever verifies the validity of credentials. As suggested above, a single person or entity may assume any number of roles.

A recent series of blogposts [64,65] by Bilgesu Sümer examines the similarities and differences between the GDPR and the concept of SSI. It is not within the scope of this chapter to report all the points discussed by these blog posts; nonetheless, a few of them are worth mentioning.

In particular, the posts highlight that while the two share common concepts and ideas, they also exhibit important differences. For example, one seeks to protect the the right to identity, while the other the right to privacy. Moreover, some terms assume different meanings depending on whether one is referring to SSI or the GDPR. In particular, transparency and access, refer to public access to transparent data for SSIs while it refers to a more restrictive notion in the GDPR.

According to Sümer, the main hurdle in reconciling SSI and GDPR remains associated with the persistence and immutability features of the underlying blockchain [65]. Sümer observes that the principle of data minimization requires that data be kept for as little as necessary for the required purpose, and argues that the verification of a credential typically requires one second. However, to the best of our knowledge, existing SSI's do not store credentials on the blockchain. Rather, they only use a blockchain only to store information that has been chosen to be publicly available (e.g. a public DID referring to a publicly available service provider).

Yet, even if the blockchain stores only public DIDs, this still poses problems with respect to the protection of personal data. Consider a user, Bob, who opens a business that requires the verification of a person's age. Bob calls the business AnonACME, an anonymous-looking name because he does not want to be publicly associated it with it. The company's DID that goes on the blockchain therefore only refers to AnonACME and not to Bob. However, Bob still needs to register the business with the public company registry, which has a publicly available database that lists Bob as the owner of AnonACME. This makes AnonACME sensitive personal data, as the public registry makes it easy to associate it with Bob.

If Bob runs his business for all his life or is otherwise proud of his business, everything looks fine. But let's imagine that Bob's runs into trouble with his business and wishes to delete it from the blockchain. All the blockchain can do is revoke its business by adding a revocation entry, but information about the previous existence of AnonACME cannot be erased. It is true that this information is also present in the public company database, but the government has legal grounds for maintaining this information, while it is not clear that such legal grounds exist for the SSI's blockchain. The persistence of the data in the blockchain can therefore be a problem even for public DIDs contradicting the principle of minimization and the right to erasure.

Although not strictly related to SSIs, another blogpost [66] analyses the specific case of the eIDAS Regulation (a proposal by the European Union for a European Digital Identity). The authors observe that the use of persistent identifiers—implied by art. 12(4)(d) of eIDAS2[9]—directly clashes with rights and principles of GDPR, such as data minimization or the right to erasure.

4 Conclusions and outlook

This chapter provided a thorough analysis of the challenges associated with the use of DLTs for the management of personal data. Our analysis concerned both legal and technical aspects, to discuss how different technical approaches to DLTs may or may not comply with the GDPR.

We identified three fundamental properties of DLTs that make compliance with—different provisions of—the GDPR challenging: immutability, decentralization, and automation. These properties create a significant problem for public

[9] `https://digital-strategy.ec.europa.eu/en/policies/eidas-regulation`

permissionless DLTs. For other types of DLTs, achieving compliance with the GDPR is less difficult, but still not trivial as compliance may depend on many factors, such as the type of data stored in the DLT, the type of processing that is performed by the DLT, links to data outside the DLT etc. We also observed that several questions about the interpretation of the GDPR in the context of DLTs are still not completely clear, for example determining the controller for a DLT application, or identifying whether public keys and hashes should be considered personal data.

These challenges make it particularly difficult to develop and operate a DLT-based application in a GDPR-compliant way while handling personal data. In our companion chapter [9], we analyze how different technical and legal approaches can be used to mitigate some of these challenges, and outline directions for future research.

References

1. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. https://git.dhimmel.com/bitcoin-whitepaper/ (2008)
2. Berdik, D., Otoum, S., Schmidt, N., Porter, D., Jararweh, Y.: A survey on blockchain for information systems management and security. Information Processing & Management **58**(1), 102397 (2021)
3. Panetta, R., Cristofaro, L.: A closer look at the EU-funded My Health My Data project. Digital Health Legal pp. 10–11 (Nov 2017). https://doi.org/10.5281/zenodo.1048999, https://doi.org/10.5281/zenodo.1048999
4. Zyskind, G., Nathan, O., Pentland, A.S.: Decentralizing privacy: Using blockchain to protect personal data. In: 2015 IEEE Security and Privacy Workshops. pp. 180–184 (2015). https://doi.org/10.1109/SPW.2015.27
5. Timan, T., Mann, Z.: Data protection in the era of artificial intelligence: trends, existing solutions and recommendations for privacy-preserving technologies. In: The Elements of Big Data Value: Foundations of the Research and Innovation Ecosystem, pp. 153–175. Springer (2021)
6. Bayle, A., Koscina, M., Manset, D., Perez-Kempner, O.: When blockchain meets the right to be forgotten: technology versus law in the healthcare industry. In: 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI). pp. 788–792. IEEE (2018)
7. Finck, M.: Blockchain and data protection in the European Union. Max Planck Institute for Innovation & Competition Research Paper No. 18-01 (2017)
8. Hewa, T., Ylianttila, M., Liyanage, M.: Survey on blockchain based smart contracts: Applications, opportunities and challenges. Journal of Network and Computer Applications **177**, art. 102857 (2021)
9. Fabčič Povše, D., Favenza, A., Frey, D., Mann, Z.Á., Palomares, A., Piatti, L., Schroers, J.: Solutions to data protection challenges in distributed ledger and blockchain technologies: A combined legal and technical approach. In: El Madhoun, N., Dionysiou, I., Bertin, E. (eds.) Building Cybersecurity Applications with Blockchain Technology and Smart Contracts. Springer (2024)
10. Talukder, S., Alam, M., Hossain, I., Puppala, S.: Blockchain-integrated identity verification for e-government services. In: El Madhoun, N., Dionysiou, I., Bertin, E. (eds.) Building Cybersecurity Applications with Blockchain Technology and Smart Contracts. Springer (2024)

11. Auvolat, A., Frey, D., Raynal, M., Taïani, F.: Money transfer made simple: a specification, a generic algorithm, and its proof. Bulletin of EATCS **132** (2020)
12. Guerraoui, R., Kuznetsov, P., Monti, M., Pavlovič, M., Seredinschi, D.A.: The consensus number of a cryptocurrency. Distributed Computing **35**, 1–15 (2022)
13. Albouy, T., Frey, D., Raynal, M., Taïani, F.: Byzantine-tolerant reliable broadcast in the presence of silent churn. In: SSS'21. pp. 21–33 (2021)
14. Albouy, T., Frey, D., Raynal, M., Taïani, F.: A modular approach to construct signature-free BRB algorithms under a message adversary. In: OPODIS'22. vol. 253, pp. 26:1–26:23 (2022)
15. Alpos, O., Cachin, C., Marson, G.A., Zanolini, L.: On the synchronization power of token smart contracts. In: IEEE ICDCS'21. pp. 640–651 (2021)
16. Frey, D., Gestin, M., Raynal, M.: The synchronization power (consensus number) of access-control objects: The case of allowlist and denylist (2023)
17. Guerraoui, R., Kuznetsov, P., Monti, M., Pavlovič, M., Seredinschi, D.A.: Scalable byzantine reliable broadcast. In: DISC'19. vol. 146, pp. 22:1–22:16 (2019)
18. Swanson, T.: Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems. `https://www.the-blockchain.com/wp-content/uploads/2016/04/Permissioned-distributed-ledgers.pdf` (2015)
19. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper (2014)
20. Monrat, A.A., Schelén, O., Andersson, K.: Performance evaluation of permissioned blockchain platforms. In: IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE). IEEE (2020)
21. Dabbagh, M., Choo, K.K.R., Beheshti, A., Tahir, M., Safa, N.S.: A survey of empirical performance evaluation of permissioned blockchain platforms: Challenges and opportunities. Computers & Security **100**, art. 102078 (2021)
22. Douceur, J.R.: The Sybil attack. In: Peer-to-Peer Systems: First International Workshop. pp. 251–260. Springer (2002)
23. Sobti, R., Geetha, G.: Cryptographic hash functions: a review. International Journal of Computer Science Issues **9**(2), 461–479 (2012)
24. Morais, E., Koens, T., Van Wijk, C., Koren, A.: A survey on zero knowledge range proofs and applications. SN Applied Sciences **1**, art. 946 (2019)
25. Raynal, M.: Fault-tolerant message-passing distributed systems: an algorithmic approach. Springer (2018)
26. Lamport, L.: Time, clocks, and the ordering of events in a distributed system. Communications of the ACM **21**(7), 558–565 (1978)
27. Fischer, M.J., Lynch, N.A., Paterson, M.S.: Impossibility of distributed consensus with one faulty process. Journal of the ACM **32**(2), 374–382 (1985)
28. Dwork, C., Lynch, N., Stockmeyer, L.: Consensus in the presence of partial synchrony. Journal of the ACM **35**(2), 288–323 (1988)
29. Dutta, P., Guerraoui, R., Lamport, L.: How fast can eventual synchrony lead to consensus? In: International Conference on Dependable Systems and Networks (DSN'05). pp. 22–27. IEEE (2005)
30. Lamport, L.: The part-time parliament. ACM Transactions on Computer Systems **16**(2), 133–169 (1998)
31. Hunt, P., Konar, M., Junqueira, F.P., Reed, B.: ZooKeeper: wait-free coordination for internet-scale systems. In: USENIX Annual Technical Conference (2010)
32. Lamport, L., Shostak, R., Pease, M.: The Byzantine generals problem. ACM Transactions on Programming Languages and Systems **4**(3), 382–401 (1982)
33. Veronese, G.S., Correia, M., Bessani, A.N., Lung, L.C., Verissimo, P.: Efficient Byzantine fault-tolerance. IEEE Transactions on Computers **62**(1), 16–30 (2013)

34. Aublin, P.L., Guerraoui, R., Knežević, N., Quéma, V., Vukolić, M.: The next 700 bft protocols. ACM Trans. Comput. Syst. **32**(4) (jan 2015). https://doi.org/10.1145/2658994, https://doi.org/10.1145/2658994
35. Kotla, R., Alvisi, L., Dahlin, M., Clement, A., Wong, E.: Zyzzyva: Speculative byzantine fault tolerance. ACM Trans. Comput. Syst. **27**(4) (jan 2010). https://doi.org/10.1145/1658357.1658358, https://doi.org/10.1145/1658357.1658358
36. Castro, M., Liskov, B.: Practical byzantine fault tolerance. In: Proceedings of the Third Symposium on Operating Systems Design and Implementation. p. 173–186. OSDI '99, USENIX Association, USA (1999)
37. Lamport, L.: The part-time parliament. ACM TCS'98 **16**, 133–169 (1998)
38. Lamport, L.: Paxos made simple, fast, and byzantine. In: Bui, A., Fouchal, H. (eds.) OPODIS. Studia Informatica Universalis, vol. 3, pp. 7–9. Suger, Saint-Denis, rue Catulienne, France (2002), http://dblp.uni-trier.de/db/conf/opodis/opodis02.html#Lamport02
39. Franklin, M.K., Malkhi, D.: Auditable metering with lightweight security. In: International Conference on Financial Cryptography. pp. 151–160 (1997)
40. Zhang, S., Lee, J.H.: Analysis of the main consensus protocols of blockchain. ICT Express **6**(2), 93–97 (2020)
41. Natoli, C., Gramoli, V.: The balance attack or why forkable blockchains are ill-suited for consortium. In: 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). pp. 579–590. IEEE (2017)
42. Singh, A., Kumar, G., Saha, R., Conti, M., Alazab, M., Thomas, R.: A survey and taxonomy of consensus protocols for blockchains. Journal of Systems Architecture **127**, Art. 102503 (2022)
43. Faqir-Rhazoui, Y., Arroyo, J., Hassan, S.: A comparative analysis of the platforms for decentralized autonomous organizations in the Ethereum blockchain. Journal of Internet Services and Applications **12**(1), 1–20 (2021)
44. Article 29 Data Protection Working Party: Opinion 05/2014 on anonymisation techniques. https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf (2014)
45. CJEU: Patrick Breyer vs. Bundesrepublik Deutschland. ECLI:EU:C:2016:779 / C-582/14, https://curia.europa.eu/juris/liste.jsf?num=C-582/14 (2016)
46. Laan, V., Rutjes, A.: Privacy-issues bij blockchain: hoe voorkom of minimaliseer je die? Computerrecht **6**, 253–263 (2017)
47. Van Alsenoy, B.: Regulating data protection: the allocation of responsibility and risk among actors involved in personal data processing. Ph.D. thesis, KU Leuven (2016)
48. Schoenen, S., Mann, Z.Á., Metzger, A.: Using risk patterns to identify violations of data protection policies in cloud systems. In: Service-Oriented Computing – ICSOC 2017 Workshops. pp. 296–307. Springer (2018)
49. CJEU: Data Protection Commissioner vs Facebook Ireland Limited, Maximillian Schrems. ECLI:EU:C:2020:559 / C-311/18, https://curia.europa.eu/juris/liste.jsf?language=de&num=C-311/18 (2020)
50. Politou, E., Casino, F., Alepis, E., Patsakis, C.: Blockchain mutability: Challenges and proposed solutions. IEEE Transactions on Emerging Topics in Computing **9**(4), 1972–1986 (2021)
51. Berberich, M., Steiner, M.: Blockchain technology and the GDPR – how to reconcile privacy and distributed ledgers. European Data Protection Law Review **2**(3), 422–426 (2016)

52. Moerel, L.: Blockchain & data protection... and why they are not on a collision course. European Review of Private Law **26**(6), 825–851 (2018)
53. Palm, A., Mann, Z.Á., Metzger, A.: Modeling data protection vulnerabilities of cloud systems using risk patterns. In: Proceedings of the 10th System Analysis and Modeling Conference (SAM). pp. 1–19. Springer (2018)
54. IBM Security: Blockchain and GDPR: How blockchain could address five areas associated with GDPR compliance. https://iapp.org/resources/article/blockchain-and-gdpr/ (2018)
55. Zhang, L., Hang, L., Jin, W., Kim, D.: Interoperable multi-blockchain platform based on integrated REST APIs for reliable tourism management. Electronics **10**(23), Art. 2990 (2021)
56. Bacon, J., Michels, J.D., Millard, C., Singh, J.: Blockchain demystified. Queen Mary University of London, School of Law Legal Studies Research Paper no. 268 (2017)
57. Pesch, P., Böhme, R.: Datenschutz trotz öffentlicher Blockchain? Datenschutz und Datensicherheit **41**(2), 93–98 (2017)
58. Commission Nationale Informatique & Libertés: Blockchain – solutions for a responsible use of the blockchain in the context of personal data. https://www.cnil.fr/sites/default/files/atoms/files/blockchain_en.pdf (2018)
59. Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM **24**(2), 84–90 (feb 1981). https://doi.org/10.1145/358549.358563, https://doi.org/10.1145/358549.358563
60. Andola, N., Yadav, V.K., Venkatesan, S., Verma, S., et al.: Anonymity on blockchain based e-cash protocols—a survey. Computer Science Review **40**, art. 100394 (2021)
61. Biryukov, Tikhomirov, G.: Privacy and linkability of mining pool payments. IEEE Conference on Communications and Network Security (CNS) pp. 118–123 (2019)
62. Tschorsch, S.: Bitcoin and beyond: A technical survey on decentralized digital currencies. IEEE Communications Surveys & Tutorials **18**(3), 2084–2123 (2016)
63. Huaqun, G., Xingjie, Y.: A survey on blockchain technology and its security. Blockchain: Research and Applications **3**(2) (2022)
64. Sümer, B.: Can Self-Sovereign Identity (SSI) fit within the GDPR?: a conceptual data protection analysis (part I). https://www.law.kuleuven.be/citip/blog/can-self-sovereign-identity-ssi-fit-within-the-gdpr-part-i/ (2022)
65. Sümer, B.: Can Self-Sovereign Identity (SSI) fit within the GDPR?: a conceptual data protection analysis (part II). https://www.law.kuleuven.be/citip/blog/can-self-sovereign-identity-ssi-fit-within-the-gdpr-part-ii/ (2022)
66. Sümer, B., Schroers, J.: The new digital identity regulation proposal and the EU data protection regime. https://www.law.kuleuven.be/citip/blog/the-new-digital-identity-regulation-proposal/ (2021)