Assessment and Treatment of Privacy Issues in Blockchain Systems

Christoph Stach
University of Stuttgart, IPVS
Universitätsstraße 38
70569 Stuttgart, Germany
stachch@ipvs.uni-stuttgart.de

Clémentine Gritti
University of Canterbury
Jack Erskine 304
Christchurch 8041, New Zealand
clementine.gritti@canterbury.ac.nz

Dennis Przytarski
University of Stuttgart, IPVS
Universitätsstraße 38
70569 Stuttgart, Germany
przytads@ipvs.uni-stuttgart.de

Bernhard Mitschang
University of Stuttgart, IPVS
Universitätsstraße 38
70569 Stuttgart, Germany
mitsch@ipvs.uni-stuttgart.de

ABSTRACT

The ability to capture and quantify any aspect of daily life via sensors, enabled by the *Internet of Things* (*IoT*), data have become one of the most important resources of the 21st century. However, the high value of data also renders data an appealing target for criminals. Two key protection goals when dealing with data are therefore to maintain their permanent *availability* and to ensure their *integrity*. *Blockchain technology* provides a means of data protection that addresses both of these objectives. On that account, blockchains are becoming increasingly popular for the management of critical data. As blockchains are operated in a *decentralized* manner, they are not only protected against failures, but it is also ensured that neither party has sole control over the managed data. Furthermore, blockchains are *immutable* and *tamper-proof* data stores, whereby data integrity is guaranteed. While these properties are preferable from a data security perspective, they also pose a threat to privacy and confidentiality, as data cannot be concealed, rectified, or deleted once they are added to the blockchain.

In this paper, we therefore investigate which features of the blockchain pose an inherent privacy threat when dealing with personal or confidential data. To this end, we consider to what extent blockchains are in compliance with applicable data protection laws, namely the *European General Data Protection Regulation* (*GDPR*). Based on our identified key issues, we assess which concepts and technical measures can be leveraged to address these issues in order to create a *privacy-by-design blockchain system*.

CCS Concepts

•**Security and privacy** → **Distributed systems security; Privacy protections;** *Usability in security and privacy;*

Keywords

blockchain, decentralized, immutable, tamper-proof, GDPR, privacy assessment, data purging, data authentication, permission control, privacy filters, privacy control environment

1. INTRODUCTION

"Data is the new oil." is a commonly cited quote by Clive Humby used to emphasize the importance of data in modern times. Unlike oil, however, which was a key driver of the *Technological Revolution* only, data are revolutionizing society as a whole. Smart cars are able to drive autonomously [66], smart traffic enable more environment-friendly transportation [80], smart buildings enable green energy management [42], and smart healthcare facilitates the lives of both patients and physicians [4], just to name a few examples. However, all of this is only possible if the data of each participant is reliably made available to all other parties involved [71].

Due to the high value which data as a commodity have in our society, they become an attractive target for cyber-criminals. However, cyber-attacks can not only cause immense economic damage, but they also pose a threat to life and limb. For instance, cyber-criminals could tamper with location data of cars or traffic management data, causing accidents in the process [38], or they could render medical data unreadable, impeding the proper treatment of patients [93]. Therefore, modern data management systems require specialized security mechanisms, especially if human lives depend on the data they are dealing with. First and foremost, they must ensure that the data are *immutable* and *tamper-proof*. Since *blockchains* possess these two key properties, it is hardly surprising that they are commonly used as decentralized data stores in such instances [90].

Despite all of these undeniable benefits of blockchains regarding the protection of sensitive data, their usage is not uncontroversial if personal data are involved. Several legal requirements imposed by the *General Data Protection Regulation* (*GDPR*) [21] cannot be satisfied when using blockchains.

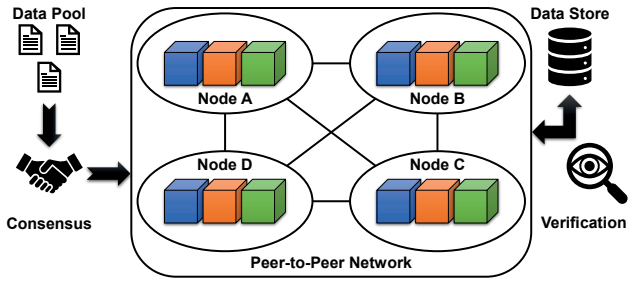

Figure 1: Simplified Architecture of a Blockchain System.

For example, immutability is an inherent violation of the right to be forgotten, while tamperproofing renders practicable anonymization of data subjects impossible [92].

That is why we investigate how a *privacy-by-design blockchain system* can be achieved without losing the immutability and tamperproofing required from a security point of view. To this end, we provide the following three contributions in this paper: 1. We elaborate on which articles of the GDPR blockchains are in conflict with when handling personal data. 2. We assess which research approaches can be used to resolve these conflicts and how they can be applied to blockchains. 3. We identify open research questions that need to be addressed in this context in order to provide an efficient privacy control in blockchains. Additionally, we outline how these research gaps can be overcome.

The remainder of this paper is structured as follows: In Section 2, we present the fundamental principles of blockchains that are responsible for the conflicts with the GDPR. Then, in Section 3, we elaborate on the articles of the GDPR with which blockchains are intrinsically in conflict. Section 4 presents related work and addresses how our work differs from these studies. We discuss technical approaches towards a GDPR-compliant blockchain in Section 5, before identifying open research questions in this regard in Section 6. Finally, Section 7 concludes this paper.

2. BLOCKCHAIN FOUNDATIONS

Whenever multiple parties operate on a common database and share their data with each other, centralized databases often pose a problem. On the one hand, the availability of the data depends entirely on this database — i.e., it represents for all participants a *single point of failure* for their operability. On the other hand, the central authority that operates the database has full control over the data and is capable of establishing the *single point of truth*, e.g., by manipulating the data or by withholding the data. To address such issues, *distributed ledger* technology has come to forth recently [96]. A distributed ledger represents a decentralized data storage, where each participant maintains the entire data stock. A consensus is reached among all participants as to which data or which transactions are authorized and added to the ledger [7]. The blockchain is a subtype of a distributed ledger.

In the following, we initially delve into the structure and architecture of blockchains in Section 2.1. Then, in Section 2.2, we present the distinctive security features that a blockchain entails. Since there are several approaches to reach a consensus, we outline the most important ones in Section 2.3. Furthermore, there are also different operating modes for blockchains, which are introduced in Section 2.4. Moreover, in Section 2.5, we identify the query capabilities of blockchain systems. Finally, we discuss four different application examples of blockchains in Section 2.6, which are representatives for fundamentally distinct types of use of blockchain systems.

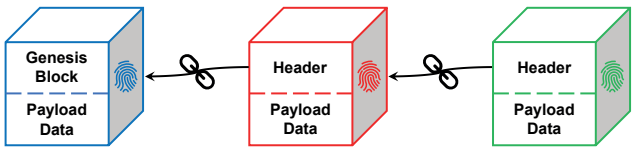

Figure 2: Internal Structure of a Blockchain.

2.1 Architecture of Blockchains

Although there are different types of blockchain systems, all of them apply distributed ledger technologies. That is, there is no central component that manages and controls the blockchain, but rather control is distributed among several emancipated nodes that mutually decide on and synchronize the content contained in the blockchain. In such a network of equal peers, it is crucial that there is a systematic mechanism by which all involved nodes can make a decision about which data to include in the blockchain [87].

Figure 1 shows how this is achieved. All data to be included in the blockchain are gathered in a data pool. A subset of these data is accepted for inclusion in accordance with certain boundary conditions. There are different approaches how to reach consensus on which subset to include. More details on such consensus technique can be found in Section 2.3. Each node of the peer-to-peer network holds an equal instance of this blockchain including all contents from the data pool on which the peers have agreed. Optionally, for performance reasons, the added data are additionally deposited in an external data store. To verify the integrity of this data store, it can be checked against any of the blockchain instances [52].

The actual blockchain, i.e., the storage structure in which the data are managed, consists, as the name suggests, of a chain of blocks. This chain structure is shown in Figure 2. Each of the blocks in the chain comprises two components: a header containing metadata and a body containing the payload data, i.e., the data picked from the data pool. The very first block of a blockchain is called *genesis block*. When additional data from the data pool have to be added, a *cryptographic hash* is calculated for them. This hash fulfills two purposes: First, this digital signature protects against unnoticed tampering of the contents — if the data would be manipulated, this would lead to a different cryptographic hash — and second, it serves as a unique fingerprint for the respective block containing these data. Each block except for the genesis block points to its predecessor by means of this fingerprint. Thereby, all blocks are inherently linked together [39].

From a technical point of view, both the fingerprints of the blocks and their coupling is realized via the metadata in the headers. To this end, a header contains five components. First of all, the *block version* indicates some key data regard-

ing the block generation and therefore, e.g., which validation rules have to be followed. Furthermore, the *hash of the preceding block* is included. This reference not only ensures the coupling of the blocks, but also guarantees that no tampering can occur in any of the predecessor blocks. Keep in mind that the header is a part of the block itself. That is, it is also protected via the cryptographic hashes. Thereby, the entire blockchain is rendered tamper-proof. Only the very last block of a blockchain can be removed unnoticed since there is no tight coupling to other blocks yet. However, beyond a certain depth — i.e., a block that has a certain number of subsequent blocks — it can be assumed that this coupling is sufficiently secure. Naturally, the *cryptographic hash* itself is also included in the header. For instance, this hash can be represented in the form of a *Merkle tree root* [48]. Furthermore, a *timestamp* is assigned to each block, representing the time at which it was created. This timestamp applies to all data contained in the block, since this is the time at which these data have been made valid in the context of the blockchain. The *target threshold* indicates the difficulty with which a consensus can be reached, i.e., how difficult it is to add a new block to the blockchain. Depending on the blockchain system and its consensus procedure, the header may include additional components [45]. The key facts about these header components are summarized in Table 1.

2.2 Security Features of Blockchains

From a security perspective, blockchains inherently possess three relevant properties, namely they are decentralized, immutable, and tamper-proof. Since identical instances of the blockchain are distributed and maintained by each participant in the peer-to-peer network, no central entity can gain full control over the blockchain and thus data sovereignty. Due to this redundant distribution of the blockchain on several nodes, an attacker would have to control the majority of the computational power or manipulate the majority of the nodes to constrain the availability of trustworthy data [73].

Table 1: Overview of the Components of a Block Header.

Header Component	Data Content
Block Version	This gives some insights about the block, e.g., regarding its supported protocols as well as which block validation rules have to be applied.
Previous Block Hash	This reference is used to tightly couple a block to its direct predecessor block.
Cryptographic Hash	A representation of the cryptographic hash over all contents of the respective block, e.g., a Merkle tree root.
Timestamp	This is the time at which the block was generated. This timestamp is also the time as of which the data contained in the block is valid.
Target Threshold	This parameter describes the difficulty to generate a new block.

The trustworthiness of the data is ensured by digitally signing each block with a cryptographic hash. Manipulations and thus a loss of data integrity would be immediately noticed by means of this signature. Furthermore, each individual block is protected by the tight coupling of the blocks to prevent an attacker from manipulating the signature in addition to the content of a block. These two interlocking mechanisms render the data tamper-proof beyond a certain block depth, i.e., in blocks that are already tightly coupled [55].

Lastly, immutability is a direct result of the two security features described above. On the one hand, the permanent availability of data is ensured due to the distributed data storage and the guarantee that no party can gain sole control over the blockchain. On the other hand, tampering can be detected immediately due to the guarantee that the data are protected via cryptographic hashes. Assuming that an attacker is unable to tamper with all nodes in the peer-to-peer network simultaneously, nodes that detect tampering in their own blockchain instance can replace it with a valid instance obtained by any peer in the network. Therefore, data in the blockchain can be considered immutable [33].

2.3 Consensus Procedures of Blockchains

A consensus protocol is used to agree on which data subset to include in the next block that extends the blockchain. To this end, there are different approaches. These approaches can be divided into two main classes: *absolute-finality consensus protocols* and *probabilistic-finality consensus protocols*. Absolute-finality consensus protocols (e.g., *Practical Byzantine Fault Tolerance* [11]) render a data record immediately valid and make it available to all parties as soon as it has been inserted into a block. Probabilistic-finality consensus protocols, meanwhile, only support eventual consistency. That is, a data record can be removed from the blockchain retrospectively under certain conditions. As discussed above, the last block of a blockchain can be removed without causing any problems, since it is not yet validated by other subsequent blocks. In probabilistic-finality consensus protocols, a data record in a blockchain is therefore not considered valid until it is in a block with a certain depth. Despite this limitation, however, probabilistic-finality consensus protocols are generally preferred in blockchains because absolute-finality consensus protocols require a single central leader that dictates which data records are valid for all parties [103].

Two of the most relevant probabilistic-final consensus protocols in terms of their role in today's blockchain systems are *Proof-of-Work* and *Proof-of-Stake*. In Proof-of-Work approaches, a certain achievement has to be accomplished first in order to be allowed to generate a new block and add it to the blockchain. To this end, a so-called *miner* has to solve a puzzle. For instance, this can be a mathematical or cryptographic operation. The miner who solves this puzzle first for the selected subset of data, is allowed to generate the next block, i.e., add the respective data to the blockchain. However, solving the cryptographic challenge requires a tremendous amount of computational power. To compensate for this effort, the successful miner receives a reward in terms of coins of the respective cryptocurrency. Afterwards, this process starts all over again, i.e., miners pick new subsets from the data pool and try to solve another

puzzle [2]. To prevent wasting the use of computational power on solving a useless puzzle, there are also extensions to Proof-of-Work in which a meaningful task must be solved by the miners, e.g., training a deep learning model [5].

Nevertheless, Proof-of-Work is still highly profligate in terms of energy consumption. The Proof-of-Stake approach therefore simplifies this process by randomly selecting a participant who is entitled to generate the next block of the blockchain. Although the selection is basically random, it depends on the *stake* of a participant. In other words, the more coins of the respective cryptocurrency a participant owns, the more likely s/he will be entitled to generate the next block [6]. Yet, despite the significantly higher energy consumption, most of the large long-established blockchain systems such as *Ethereum* [100] rely on the Proof-of-Work approach.

2.4 Operating Modes of Blockchains

There are basically two different modes of how a blockchain can be operated. In a *public blockchain*, it is assumed that anyone can join the peer-to-peer network without having been authorized to do so beforehand and can also leave it at any time. This operating mode is therefore also referred to as *permissionless*. Consequently, it is not necessary (or provided) to verify the identity of any of these peers. The blockchain can only be successfully maintained if sufficient peers are motivated to provide their computing power and storage space needed to store a full instance of the blockchain locally. Typically, this is achieved by means of monetization, which raises the operating costs of the blockchain. However, this is necessary as the security features of the blockchain are only guaranteed if there are enough trusted peers, i.e., an adequate number of copies of the blocks. Since anyone can join the peer-to-peer network, anyone also gets full access to the data stored in the blockchain [41]. It is therefore obvious that public blockchains are not suitable for storing private or confidential data. Therefore, they are not further considered in the context of our work.

In contrast, private blockchains have a central regulatory authority that decides who is allowed to participate in the peer-to-peer network. This operating mode is therefore referred to as *permissioned* since authorization is necessary in order to gain access to the blockchain. In order to be able to grant such access rights, it is a fundamental prerequisite that all peers are uniquely identifiable. Although the number of nodes in private blockchains is thus smaller than in public blockchains, the security features can still be assumed since each peer can generally be trusted. It is also possible to exclude peers that show malicious behavior from further participation. Due to the smaller number of nodes and the low dynamics in terms of fluctuation, the peer-to-peer network can be operated much more efficiently. Furthermore, it is ensured that only trusted participants have access to the data. Although a central authority decides who is allowed to join the network, there is still not a single authority controlling all nodes [30].

Hybrid and consortium blockchains are somewhere in between. Here, a group of participants has joint control over the blockchain instead of a single central authority [39]. In the context of our work, however, they can be considered as subtypes of private blockchains.

2.5 Query Capabilities of Blockchains

Blockchains do not have a dedicated data model by default. In principle, any data objects can be stored in the blocks. The contents of a block can also be heterogeneous, i.e., each data object can have a different underlying data model. The only requirement posed by blockchains in this regard is that each object has a unique identifier. Therefore, the basic query capabilities of blockchains are similar to those of key-value stores. That is, for data access there is basically a get operator available, which receives the ID of a data object as a parameter and returns the corresponding data object [95]. To find the requested data object, all blocks must be traversed sequentially until the queried ID is found, i.e., the query costs increase with the data volume contained in the blockchain [1].

Similar to NoSQL data stores, there is no uniform query language. For instance, some blockchain systems also support a document-oriented access model — i.e., data objects are stored as JSON documents — which also facilitates more complex queries based on the attributes of the objects [97]. Yet, this requires the introduction of a stricter data schema which restricts the generality of the blockchain in the process. Furthermore, some blockchain systems also have additional features for query optimization, such as index structures for efficient data access [56] or an additional database that contains the most current state of all objects managed by the blockchain [58]. However, such approaches only represent island solutions and are not representative of blockchain technology in general.

Besides these types of access, which are comparable to key-value stores and document stores, blockchains offer a further, more sophisticated query mechanism in the form of *smart contracts*. The term smart contract was coined by Szabo in 1997 [88]. The smart contract is seen here as a version of a real-world contract implemented as an executable program. In simple terms, actions are subject to certain conditions. If these conditions are met, the contract is executed, i.e., the actions specified therein are carried out. Smart contracts have also made their way into the blockchain landscape. In this context, a smart contract describes which transactions are to be executed on certain data of the blockchain when a specified condition applies. The results of these transactions are automatically added to the data pool and thus eventually also become part of the blockchain eventually. The smart contracts themselves are also stored in the blockchain and are therefore publicly available and unchangeable for all parties, which renders them trustworthy [32].

In other words, a smart contract is a user-defined function specified in an imperative programming language. Such a function has a set of input parameters and a well-defined return value. Thus, a smart contract can also be leveraged as a parameterizable query for blockchain systems. A smart contract developer can implement any query logic provided that the supported programming language is Turing-complete. Thus, complex query functionalities which are widely used can be deployed as a smart contract and are then available to all users of the blockchain [14]. However, the definition of a smart contract is complex and comprehensive IT knowledge is required. Furthermore, smart contracts pose a huge security risk, as even the smallest errors contained in the contract's code jeopardizes the integrity of the blockchain

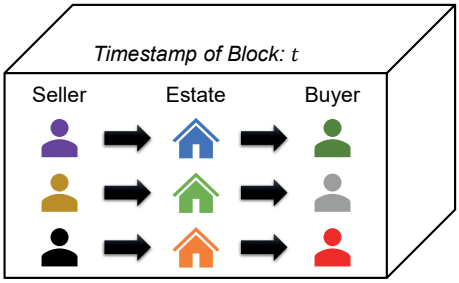

Figure 3: Structure of a Block for Transactions.

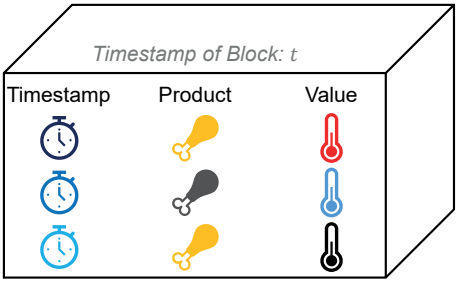

Figure 4: Structure of a Block for Time Series Data.

and all of its data [40]. Therefore, the utilization of smart contracts should rather be limited to a necessary minimum.

2.6 Application Examples of Blockchains

Originally, blockchains gained popularity in the context of digital currencies, such as the cryptocurrencies *Bitcoin* [57] and *Ripple* [74]. Today, however, they are being field-tested in many non-financial applications as well [91]. This has led to a plethora of novel application domains for blockchains [17]. Examples can be found in the form of *non-fungible tokens* (*NFT*) [60], *identity management systems* [77], and *access control systems* [18]. When looking at all these use cases, four categories of applications can be identified, which are differentiated based on the types of data that are managed in the blockchain. In the following, we take a closer look at these four types of applications.

Chain of Transactions.

The medium of exchange of a cryptocurrency, such as a *Bitcoin* or a *Satoshi*, are assumed to be *fungible*. That is, they are transferable objects, whereby every Bitcoin and every Satoshi are basically identical. For instance, the Bitcoin with ID 1 is as valuable as the Bitcoin with ID 2. However, just as with physical means of payment, it has to be technically ensured that only one entity can have a specific instance of a Bitcoin at any given time. In particular, this means that each instance is unique and cannot be duplicated, e.g., via exchange [54].

This property can also be applied to the exchange of non-fungible objects. These are objects where each instance has individual and possibly fundamentally different properties. In the context of NFTs, blockchain technologies are therefore used to manage virtual objects and to record the ownership of the respective object. More precisely, the blockchain itself records which transactions are executed with these tokens. This allows not only to unambiguously trace who the current owner of a particular token is, but also the complete history of the token with all its previous owners [64].

This is practical, however, for tangible assets as well. For instance, it must also be recorded in a land register who owns a real estate property and how this property came into their possession. In the case of such a land register, it is not only necessary that it is tamper-proof, but also that it can be consulted publicly. Both properties are guaranteed by a blockchain. In the course of the digitization of notary data, it is therefore reasonable to manage sales transactions of real estates in blockchain systems [101].

Figure 3 shows the data model used for this purpose. Essentially, the blockchain manages data triples consisting of a seller, the transferred goods, i.e., the real estate, and a buyer. These three instances can either be fully stored in the blockchain or merely exist as references to an external store. A transaction is not considered to have taken place until it appears in the blockchain. As a result, each of these transactions, i.e., each triple, additionally has an implicit timestamp, namely the creation time of a block, as this by definition marks the execution time of the transaction. The complete blockchain therefore represents the land register in which the histories of all real estates can be traced.

Time Series Data.

While the data model and the application of the blockchain in the chain of transactions largely correspond to the operating model used by cryptocurrencies — instances are transferred from A to B — blockchains can also be used in a completely different manner. Instead of using triples representing transactions, data objects can also be managed in the blockchain as a sequence of key-value pairs describing the attributes of the corresponding entities [9].

This can be applied productively in the area of supply chain management, for instance. Here, it is important to be able to model the life cycle of a product, i.e., its processing, from production to consumption. In this context, it is often necessary to comply with standards and regulations and to enable third parties, such as control authorities, to conduct a comprehensive audit [23]. One specific use case, e.g., is to check whether the cold chain of a product has been permanently maintained. To do this, IoT-enabled sensors regularly check the temperature and report it to the blockchain system [83].

Figure 4 shows the data model used for this purpose. In addition to the product to which the temperature reading belongs and, of course, the measured value itself, the time of the reading also has to be recorded in this case. In contrast to the previous application example, the timestamp that implicitly exists in the block header cannot be used for this purpose, since the time of the measurement can differ significantly from the creation time of the block. Furthermore, multiple measurements can be stored in a block, each recorded at an individual time. The assumption made in the previous use case (chain of transactions) that a fact, i.e., a

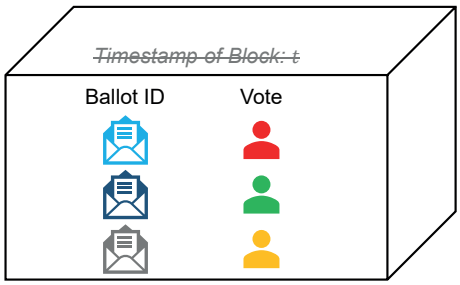

Figure 5: Structure of a Block for Singular Instances.

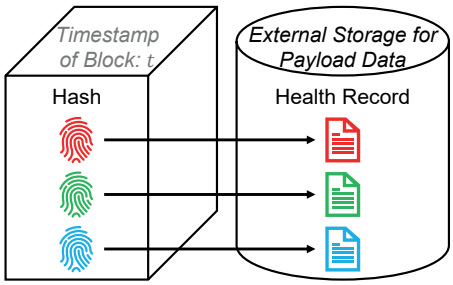

Figure 6: Structure of a Block for Validation Data.

transaction, is not valid until it is added to a block, does not apply here. That is, the timestamp in the block header only represents an upper threshold as to when the measurement has taken place at the latest.

Singular Standalone Instances.

In both use cases discussed so far, there was a time dependency within the data managed in the blockchain. That is, the set of all blocks represent a history for the particular objects. However, in certain use cases, there is no such dependency. This is always the case when singular standalone instances are managed in the blockchain.

A good example for this is an electronic election. Each vote represents thereby an independent instance, which has neither a connection to other votes, nor a history [24]. In order to conduct electronic elections in a trustworthy manner, the results must be audible and verifiable for everyone [12]. This can be ensured by a blockchain-based management of ballots.

As shown in Figure 5, each vote can be stored in the blockchain in a largely anonymous manner[1] [89]. For this purpose, in addition to an ID of the ballot to prove that it is a valid vote, the person for whom the vote is for, is also stored in the blockchain. Such an entry does not have a timestamp, since it is ensured for all valid votes, i.e., all votes that end up in the blockchain, that they were received in due time. An individual timestamp for each ballot is therefore not required. Although an implicit timestamp is given by the block, it is completely meaningless in this use case.

Validation Data for External Storages.

In all preceding use cases, we assumed that the payload data are stored directly in the blockchain. This is referred to as *on-chain data management*. As a result, full transparency is achieved, as anyone with access to the blockchain can verify the integrity of the data. Furthermore, availability is also guaranteed, as all contents of the blockchain are immutable, i.e., they cannot be deleted. However, it is not always possible to store the full payload data in the blockchain, for instance due to their size. This is referred to as *off-chain data management*. In this case, the actual data objects are outsourced to an external data store. In the blockchain itself, only verification data are stored, which can be used to check whether the payload data on the external storage have been manipulated. Yet, this represents massive limitations in terms of availability and integrity, since payload data on the external storage can be fully deleted and manipulations can be detected, but not prevented or undone [31]. Nevertheless, this is often the only option when big data need to be managed or comprehensive data analytics are required, since on the one hand the storage space in the blockchain is expensive, and on the other hand the data processing capabilities of a blockchain are inherently rather limited [99].

One such use case are electronic health records [68]. While managing the entire record in the blockchain would be reasonable from a theoretical point of view, as this would provide the maximum protection against loss or manipulation, this is generally not feasible in practice. Large entries in the patient record, such as high-resolution CAT scans (*computer-assisted tomography*), would inflate the volume of data to be stored excessively. Rather, the blockchain serves only as a security measure that can be used to verify the integrity of the actual health records [46].

Figure 6 shows the data model for such a use case. Only fingerprints for each health record are stored in the blockchain, e.g., as cryptographic hashes. When the record is accessed, its hash value can be calculated and verified against the hash stored in the blockchain. If they match, it is ensured that the record has not been tampered with. The payload data can be stored in any data store and data format. However, since a health record must be constantly updated — for instance, as new diagnostic findings or examination results are added — a new fingerprint must be stored in the blockchain for each legitimate modification. That is, only the most recent fingerprint of each record is actually relevant, as there is typically no history for the record itself. Yet, the most recent fingerprints for all records can be spread across the entire blockchain. The timestamp of a block therefore only matters in the sense that the blockchain has to be scanned sequentially, starting with the block with the most recent timestamp, until a fingerprint for the respective record is found. For efficiency reasons, this scan can be facilitated by access structures, such as a *world state*, which points directly to the most recent fingerprint for each record. *FalconDB* [63] is an example of this usage of blockchain technology.

[1] However, it has to be noted that certain conclusions about the respective voter are still possible due to the required information regarding the ballot. Therefore, a public blockchain is still not an option.

In the context of our work, however, this type of blockchain application, in which the blockchain merely serves as a means of verification for external data storage, is of no further interest. Since the privacy-critical data is held in the external storage, conventional protection measures can be applied directly to the data. The fingerprints stored in the blockchain are not relevant from a privacy perspective, as they do not enable to draw any confidential or compromising inferences [61].

3. BLOCKCHAINS AND THE GDPR

The GDPR is intended to give *data subjects*[2] full control over their personal data in an increasingly digitalized world — they must be empowered to control who has access to their data. So, it is not surprising that blockchains, which are primarily designed to make data permanently and immutably accessible to all interested parties, are in conflict with such regulations if personal data are involved.

In her study, Finck [22] therefore examines whether blockchains can be squared with the GDPR. Here, a fundamental problem becomes apparent, namely that the GDPR presumes that there is a **data controller** who is responsible for compliance with the data protection rights of data subjects (Article 24). However, due to the decentralized nature of blockchains, such a central control authority does not exist. As a result, the study concludes that it is difficult to achieve GDPR-compliance, especially for public blockchains. Therefore, we focus on *permissioned blockchains* (e.g., private blockchains), where there are organizational and technological regulatory means.

Going over the articles of the GDPR in consecutive order, the first articles that seem to be relevant for blockchains are Articles 5 and 7. They specify the legal framework within which processing of personal data is allowed. If the data are processed directly in the blockchain, smart contracts can specify exactly for which purpose the data are processed as well as in which way they are processed. Thereby, a kind of **purpose limitation** (Article 5(1)(b)) is achieved. As all data stored in the blockchain are available to all participants of the peer-to-peer network, this nevertheless raises a problem with regard to **data minimization** (Article 5(1)(c)). Furthermore, since the data in the blockchain are immutable, neither the **accuracy** of the data can be improved retroactively (Article 5(1)(d)) nor any **storage limitation** (Article 5(1)(e)) can be enforced as blockchain are an *append-only* data structure. Moreover, the **consent** of the data subject (Article 7) is only reliably respected within the scope of a smart contract. If the data are processed outside the blockchain, the agreements reached in the smart contracts no longer apply.

A data processor has the duty to **inform the data subjects** about the collection and processing of their personal data (Article 12 – 15). In order to enable a data subject to do this, however, a query interface is needed that can be used to retrieve all aspects regarding the collected data. The query capabilities supported by blockchains, namely a low-level key-value query interface and the option to define smart contracts, are by no means sufficient for this purpose. With the former, no targeted queries about a certain data subject are possible and with the latter, a lot of contracts would have to be implemented to cover all use cases, which poses a considerable security threat — and an overhead as well.

[2]In accordance with the GDPR, a data subject is an identified or identifiable natural person whose personal data are processed (Article 4(1)).

Table 2: Summary of the GDPR Articles with which Blockchains Inherently Conflict due to Technical Reasons.

GDPR Article	Conflicting Blockchain Property
Article 5(1)(b)	By default, blockchains do not impose a *purpose limitation*. However, a kind of purpose limitation can be achieved via well-defined smart contracts.
Article 5(1)(c)	All data on the blockchain are accessible to all nodes. Therefore, there is no *data minimization*.
Article 5(1)(d)	Data on the blockchain are immutable, i.e., their *accuracy* cannot be improved retroactively.
Article 5(1)(e)	Since blockchains are append-only data structures, *storage limitation* cannot be achieved.
Article 7	In general, blockchains do not require the *consent* of a data subject to process his or her data. However, this can be realized via smart contracts.
Article 12 – 15	In order to fulfill the duty to *inform the data subjects*, comprehensive query capabilities must be provided. Yet, the query capabilities of blockchains are rather limited.
Article 16	The *right to rectification* cannot be enforced in blockchains as the data are stored immutable and tamper-proof.
Article 17	The *right to erasure* cannot be enforced in blockchains as this would destroy the internal blockchain structure.
Article 18	When using smart contracts, the *right to restriction of processing* can only be enforced if a majority of blockchain nodes agree to the requested changes.
Article 22	If data processing is handled by autonomously acting smart contracts, the *automated individual decision-making* is violated.
Article 24	In order to enforce the rights of data subjects, an all-embracing *data controller* is required. Yet, such a central authority fundamentally contradicts the decentralized nature of a blockchain.
Article 25	Only if all the ten technical issues listed above are addressed, a blockchain can support *data protection by design*.

Furthermore, since each node operates independently, there is no holistic view covering all peers and their activities, e.g., with regard to local data processing.

If the data stored in the blockchain are incorrect, a data subject also has no means to have them corrected as required by the **right to rectification** (Article 16) due to immutability and tamperproofing. It is also not possible for a single data subject to exercise its **right to erasure** (Article 17) — due to the linkage between the blocks only the last block can be erased without destroying the structure of the blockchain. Furthermore, even the last block can only be deleted completely or not at all. Since a block contains an arbitrary subset of the data from the data pool, the deletion of a block therefore always affects the data of several data subjects.

As smart contracts execute transactions automatically and without human intervention, data subjects also have issues exercising their **right to restriction of processing** (Article 18). Only when a smart contract has been modified on the majority of the nodes of the peer-to-peer network according to the requested restrictions, the modifications will take effect. In any case, the **automated individual decision-making** (Article 22) bears another conflict potential, since smart contracts can be used for such decision-making. Therefore, the usage of smart contracts in the context of personal data has to be regarded as problematic in general.

Other regulations such as the **territorial scope** (Article 3) and the **lawfulness of processing** (Article 6) are primarily organizational issues. In our work, however, we focus on technical aspects of the blockchain that inherently conflict with the GDPR.

In summary, it can be observed that the immutability and tamperproofing of blockchains in particular cause problems with regard to the correction and deletion of data. Furthermore, the decentralized management of the data poses a challenge in terms of restricting access to the available data. This also results in an issue regarding a central controller that ensures compliance with data protection regulations. These issues must be overcome in order to support **data protection by design** (Article 25) for blockchains.

Table 2 outlines the key conflicts that we identify in blockchains with regard to the GDPR. Here, however, we focus only on the first and foremost technical aspects.

4. RELATED WORK

Due to an increasing number of novel use cases for blockchains, there is a large body of research regarding blockchains and privacy in addition to the aforementioned study by Finck [22]. Haque et al. [29] conduct a comprehensive literature review on different aspects of how to improve the GDPR-compliance of blockchains. They conclude that there is basically a lot of prior work on the topic of GDPR-compliant blockchains. Yet, besides some well researched application areas, such as the healthcare sector, there are many unexplored areas where there are still open research questions regarding GDPR-compliance issues with blockchains.

This is due to the fact that studies such as those by Campanile et al. [10] or Miyachi and Mackey [50] deal with a very specific use case for blockchains in the area of smart cars or smart healthcare, respectively. They are developing a privacy-aware blockchain solution for exactly these use cases. However, these solutions require a dedicated infrastructure and cannot be transferred to other application areas and use cases due to their high degree of specialization.

While these studies focus on technical solutions to make blockchains GDPR-compliant for specific use cases, studies like the one by Shuaib et al. [76] provide administrative guidelines on how blockchains can be used to store sensitive data, such as electronic health data. In a similar direction, the work by Molina et al. [51] presents high-level design guidelines for administrators to set up a GDPR-compliant infrastructure with blockchains.

Furthermore, blockchains are also assessed from a purely legal perspective. Poelman and Iqbal [65] come to the disillusioning conclusion that GDPR-compliant blockchains are basically impossible. However, they water this statement down by adding that it might be possible in a permissioned private blockchain with appropriate extensions. Yet, this requires that certain limitations have to be accepted regarding the key characteristics of the blockchain, namely decentralization, immutability, and tamperproofing. In contrast, Manteghi [47] concludes that current privacy laws also need to be adjusted to enable a "peaceful" coexistence with blockchains. One way or the other, there is a need for action.

Related work thus can be divided into four categories: literature reviews, privacy-aware blockchain solutions tailored to specific use cases, administrative guidelines, and legal assessments. Our work differs significantly as we investigate technical measures that can be added to any blockchain to achieve compliance with the GDPR. To this end, we discuss techniques that are well-known from other application areas and describe how they can be applied to blockchains in order to comply with data protection requirements.

5. APPLICABLE SOLUTIONS

As discussed in Section 3, a major problem of blockchains with regard to the GDPR is their inability to rectify or erase personal data. To this end, we explain in Section 5.1 how hierarchical data encryption can be used to achieve data purging in blockchains. Yet, better than correcting data retrospectively is to ensure that data quality is as high as possible beforehand. Therefore, we explain in Section 5.2 how attribute-based authentication can be used to prevent data from dubious sources from being included in the blockchain in the first place. Moreover, these techniques also enable purpose-based permission control, as we show in Section 5.3. These permissions are able to minimize the disclosed information about the data subject by applying privacy filters to the data. In Section 5.4, we discuss how these filters can be used to realize the principles relating to processing of personal data in blockchains. To this end, we detail three variants of privacy filters, namely pattern-based privacy filters, time series privacy filters, and statistical privacy filters. Finally, the distributed nature of blockchains poses a problem with respect to the GDPR, as there is no distinct data controller. For this reason, we conclude in Section 5.5 with a reflection on how all these techniques can be incorporated into a central privacy control architecture for blockchains.

5.1 Data Purging by Encryption

In order to permanently delete data, there are two methods that are considered to be reliable: Either the data carrier on which the data is stored is physically destroyed or the sectors containing the data in question are overwritten several times. Both approaches guarantee that the data cannot be restored. However, there are use cases in which neither of the two methods can be applied, as organizational reasons speak against the destruction of the data carrier — e.g., because there are also data on it that must not be deleted or because the costs for frequent deletions would skyrocket — or because it is not possible for technical reasons to access an explicit data sector via the available interfaces — e.g., when dealing with databases.

In such cases, another method has proven to be extremely reliable: *data purging by encryption*. Here, all data in the data store are encrypted. This is done completely automatically in the background and is entirely transparent to the user. The keys are kept outside the data store containing the payload data. To delete data, it is sufficient to destroy the associated keys, since subsequently the data cannot be decrypted and are therefore rendered unreadable. Since the keys are much smaller than the payload data, they can be held in special data stores that ensure secure erasure, e.g., by providing interfaces via which read and write operations can be performed at sector level. Such an encryption-based erasure procedure also fulfills the purging requirements of privacy laws [75].

Although it is not possible to delete data in blockchains due to the cryptographic hashes and the links between the blocks, a data purging by encryption approach can grant the right to rectification as well as the right to erasure, without thereby rendering the immutability and tamperproofing as such obsolete. If personal data are stored in the blockchain in encrypted form, the blockchain can still guarantee immutability via the hashes. However, data can only be processed as long as the key required for decryption exists. If this key is stored externally in a trusted environment and the data subjects are given full control over their keys, they can make their data unreadable at any time. Also, not all data on the blockchain have to be encrypted, but only those that are considered personal data according to Article 4 of the GDPR.

For instance, the blockchain system *Hyperledger Fabric*[3] uses *CouchDB*[4] to represent the world state, i.e., the consolidated view of all nodes. Stach and Mitschang [84] have shown that secure deletion is feasible efficiently on such document-oriented databases via an encryption-based approach. Opposed to a regular database, blockchains are append-only, i.e., such an approach also results in less overhead since update operations involving multiple decryption and encryption operations are omitted. Thus, data purging by encryption can be considered an applicable technical solution for a blockchain, e.g., to implement the right to erasure.

[3]see `https://www.hyperledger.org/use/fabric` (accessed on August 22, 2022)

[4]see `http://couchdb.apache.org/` (accessed on August 22, 2022)

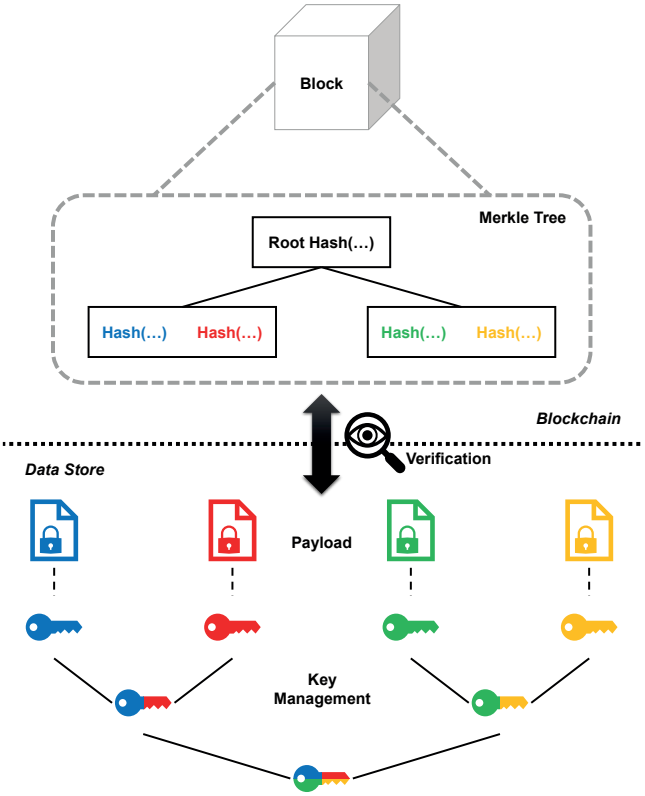

Figure 7: Data Purging by Encryption in a Blockchain.

Yet, with regard to the management of the keys, an effective strategy has to be adopted due to the large amount of data that can accumulate in a blockchain and thus the potentially large number of keys. Here, a structural property of many blockchains can be exploited. In blockchains, so-called *Merkle trees* are often used for data verification. This is a hash tree in which the leaves contain the hashes of the payload data, and the inner nodes contain a hash of its child nodes (see Figure 7 upper part). That is, a hierarchical structure is established, where each node is responsible for the consistency of all data contained in the subtree rooted at that node [39].

Waizenegger et al. [98] have introduced a tree-like data structure for managing keys. The keys with which payload data are encrypted are located at the leaf level. Each key is based on its parent node. In this way, all keys in a subtree become invalid if the key in its root node is deleted (see Figure 7 lower part). These two tree-structures can be mapped to each other, so that the required keys can be deleted very easily as soon as the node in whose subtree the data to be purged is located has been identified in the Merkle tree. This interrelation is outlined in Figure 7.

5.2 Attribute-Based Data Authentication

Data purging by encryption can also be used to correct data in a blockchain by deleting the incorrect data and then adding the corrected data to the data pool of the blockchain. However, this is a costly process. It is therefore much better to ensure the highest possible data quality in advance. One

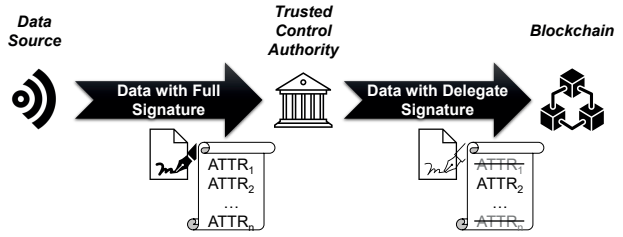

Figure 8: Illustration of a Privacy-Aware Data Authentication (cf. Gritti et al. [27]).

way to achieve this is to accept data from reliable sources, only. That is, an authentication of data sources is required.

Attribute-based authentication methods are suitable for this purpose as they can distinguish between different sources at a fine-grained level. These methods are based on a digital signature that contains certain attributes of the signer — i.e., the data source. As a result, the signature can not only be used to verify that the data has been transmitted genuinely, but it can also be used to determine unambiguously which properties the sender has. These properties are then checked against a policy. Only if they meet this policy the data are considered authentic [13].

Thereby, it is possible, e.g., to verify that the device that captured data about a data subject has the necessary software and hardware to capture this kind of data with a sufficient degree of accuracy. The attribute-based approach enables an arbitrary fine-grained distinction of entities, since the number of attributes used is not restricted. Nevertheless, it is an effective way to specify a policy to determine which requirements a source has to meet in order to provide a certain kind of data. It is only necessary to specify a threshold for the relevant attributes that a source must at least satisfy. All other attributes can be ignored when verifying the signature.

However, the virtually unlimited number of attributes that a signature can contain also harbors an inherent threat with regard to the privacy as some of the attributes might reveal too much information about the sender. This would represent a significant drawback if, in order to protect the privacy of one data subject, another data subject (in this case the sender) is exposed. Gritti et al. [26] therefore introduce a privacy-preserving attribute-based authentication. For this purpose, they use *delegated authentication*. That is, a trusted control authority acts as an intermediary between the source and the designated sink, i.e., in our case the blockchain.

This approach is illustrated in Figure 8. A data source, e.g., a sensor, signs the data with its full signature and sends it to the control authority. In the depicted example, this full signature consists of the attributes 1 to n of the source ($ATTR_1$, $ATTR_2$, ..., $ATTR_n$), e.g., its serial number, model, and location. The trusted control authority verifies these attributes. If the source is appropriate to provide the respective payload data, the control authority filters out all attributes from the signature that are not required by the sink for authentication and applies the resulting *delegate signature* to the payload data. Here, the serial number and the location of the data source are filtered out since they reveal too much information about the data source. The model of the sensor ($ATTR_2$) is sufficient to legitimate the submitted payload data. That is, the blockchain is able to verify the authenticity and origin of the data based on the delegate signature, but no privacy-critical information about the source is revealed.

5.3 Purpose-Based Permission Control

As discussed in Section 2, public blockchains are not suitable for storing sensitive information because anyone can join the network and thus gain unrestricted access to all data in the blockchain. This is not the case with private blockchains, since the number of parties with access to the data is severely restricted in such blockchains. Furthermore, smart contracts can be used to further regulate the processing of data by making it dependent on certain conditions.

Smart contracts, however, have to be hard-coded in *chaincode*. Therefore, they are comparable to the transformation operators defined in a data warehouse. There, the data are also automatically pre-processed according to predefined rules and optimized for certain use cases that are fully known in advance [35]. Yet, this implies that these use cases have to be identified and specified in advance. In dynamic environments, like today's smart environments, such a concept is too rigid. Data consumers require more flexibility, as new use cases are constantly emerging. The goal should therefore be to keep the data as generic and unprocessed as possible and to leave the processing entirely to the data consumers [34]. Therefore, it must also be possible to process the data outside of smart contracts.

However, this also entails that there have to be well-defined permissions as to which parties are allowed to access which data on the blockchain. In permissioned blockchains such as Hyperledger Fabric, this is regulated by means of *access control lists*. With these policies, it is not only possible to define who can participate in the network in general, but also which resources they are allowed to access. In addition, it is possible to restrict who is allowed to make updates — in terms of adding new data — to the blockchain. These access control lists rely on *role-based access control*. Yet, this rather traditional form of access control is often not dynamic and flexible enough, which is why Khan et al. [37] introduce *DistU*. DistU monitors the data of a blockchain permanently and grants or revokes permissions depending on how a data object is used.

Nevertheless, a data consumer still has either full access to a data object or none at all. That is, the permission model itself also needs to be extended in order to enable effective data minimization. Stach et al. [82] present such a *fine-grained permission model* for distributed Internet of Things applications that can be adapted to blockchains. Figure 9 shows this adapted permission model. A *permission rule* describes which *accessor* — i.e., which data consumer — may access which *resource* — i.e., which data. In the case of a permissioned blockchain, the data consumer can be identified using its access credentials. Since personal data may only be processed for a given *purpose*, such a purpose can be attached to a permission rule. Using an attribute-based authentication method as described in Section 5.2, the purpose can be specified by means of identifying attributes of

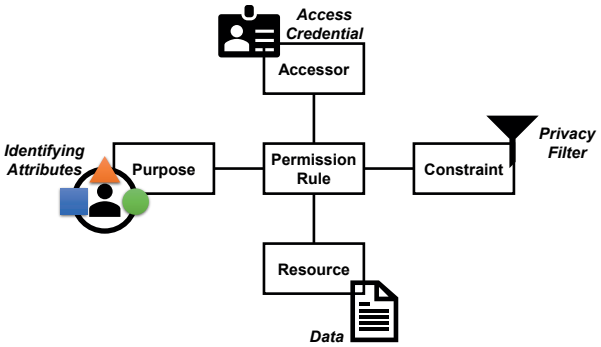

Figure 9: A Permission Model for Blockchain Applications.

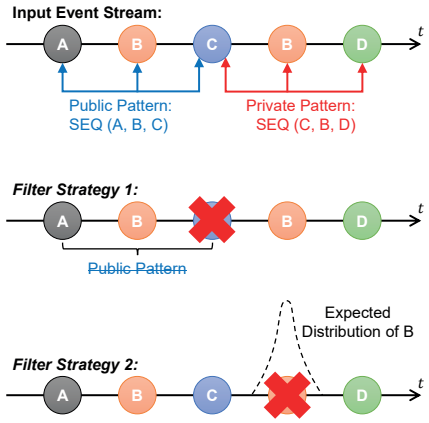

Figure 10: Examples of Pattern-Based Privacy Filters.

the data consumer as well as the processing environment. Finally, *constraints* can be imposed on the processing. They are described in terms of privacy filters (see Section 5.4) that have to be applied to the data prior to processing.

In addition to the higher flexibility, as these permission rules do not have to be hard-coded as chaincode, and the possibility of assigning very fine-grained permissions, this approach has another advantage over smart contracts. They are much easier to define as no coding skills are required. That is, they can also be comprehended and specified by IT laymen according to their privacy requirements.

5.4 Privacy Filters

By means of the permission rules, data access can be restricted quite well, but in order to be able to ensure data minimization effectively, it must also be possible to reduce the information contained in the data. Smart contracts could realize this, as they can transform the data and thus, e.g., filter out certain features during processing. However, the implementation of such a function is far too complex, so that data subjects are not capable of specifying such a smart contract to reduce the information content.

A more user-friendly solution is to provide out-of-the-box *privacy filters* that are able to blur certain privacy-relevant aspects in the data before releasing them to an accessor. However, it is important that the data are not rendered invalid in this process. Therefore, a collection of privacy filters adapted to specific data types and use cases is needed [49]. In addition to generic filters that can be applied to any type of data (e.g., withholding some data or adding noise to the data), also specialized filters are required for cloaking of location data [3] or distortion of time series data [15]. By applying the appropriate filter, it is possible to filter out certain aspects that are less relevant for processing but contain a lot of privacy-relevant information. In this way, information minimization can be achieved for any use case.

Besides such filters that operate on the data of a single user or even single data points, it is also feasible to use privacy filters tailored to large multi-user data stores like a blockchain. For instance, a privacy filter based on differential privacy enables statistical analyses without identifying individual users [104]. There are also filter operators that are designed to filter out large amounts of data without impairing the usability of the underlying data too much [62].

Looking at the three privacy-critical application examples that we identified in Section 2.6 as particularly relevant in the context of blockchains, it becomes evident that different aspects need to be concealed in each case. Therefore, in the following, we present three privacy filtering approaches that are specifically geared to these application examples, namely pattern-based privacy filters, time series privacy filters, and statistical privacy filters. We also describe how these filters can be deployed in a privacy platform so that they can be applied in a user-friendly way.

Pattern-Based Privacy Filters.

When transaction chains are stored in the blockchain, it can be particularly privacy-critical if certain patterns can be recognized in these data. In the case of land registers, for instance, a data subject might be interested in ensuring that periodic purchase transactions in which that data subject is involved are not identifiable. At the same time, all other aspects of the land register must not be affected when filtering out such privacy-critical patterns. In particular, there may be patterns that are essential to be identifiable as they are mandatory for the operation of the land register.

Each individual purchase transaction, i.e., each data triple, can be considered as an event from a technical point of view. Since each triple has a unique timestamp via the header of the block, the complete contents of a blockchain in which such transactions are stored can be regarded as an event stream, which can be traversed sequentially starting from the genesis block. Therefore, *private patterns*, i.e., patterns that have to be concealed, and *public patterns*, i.e., patterns that are required for the legitimate maintenance of a service, can be defined as sequences of events [85].

Palanisamy et al. [59] present techniques for concealing certain patterns by chronologically reordering such event streams. While this may sound simple at first, this turns out to be a hard task in practice. We provide examples for some problems in Figure 10. To filter out the private pattern $C \rightarrow B \rightarrow D$ in the given input event stream, one approach might be to remove the event C (Strategy 1). Since dropping this event would unnecessarily reduce the overall data qual-

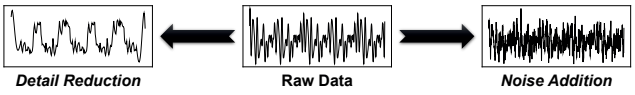

Figure 11: Examples of Time Series Privacy Filters.

ity, an earlier or later timestamp could be assigned to the event C by the privacy filter instead. In this filter strategy, however, the required public pattern ($A \rightarrow B \rightarrow C$) would no longer be recognizable. Whereas, if the timestamp of the second B would be manipulated, as depicted in Strategy 2, the problem seems to be solved sufficiently at first. However, additional constraints may exist, such as that event B occurs every t time units. Therefore, an attacker would detect this manipulation immediately and could deduce what pattern has been concealed.

A pattern-based privacy filter must therefore use heuristics to devise appropriate filter strategies and evaluate all possible manipulations using a quality metric. Such a quality metric assesses how successful the respective strategy conceals the privacy patterns and how destructive it is regarding the public patterns. On the one hand, it must be taken into account how many public patterns are removed by the manipulations. On the other hand, it has to be considered how many public patterns were falsely generated in the stream by the manipulations. A penalty weight can be assigned to each of these parameters, e.g., if false positives — i.e., the detection of public patterns that are generated by the privacy filter — are very harmful in the intended use case [86]. The best strategy can then be applied to the blockchain data, i.e., the timestamps of the respective events can be manipulated accordingly.

Time Series Privacy Filters.

If time series data are stored in the blockchain, other privacy techniques are required. Contrary to the transactions which do not necessarily have any direct correlation, a time series is a continuous series of measurements. Manipulation of the timestamps of the individual measured values is therefore not an option for this kind of data. Here, the progression is especially relevant. However, manipulating the timestamp would completely disrupt it.

Basically, there are two opposing approaches that can be applied by a privacy filter for time series data instead. These two approaches are illustrated in Figure 11. On the one hand, the level of detail can be reduced. For instance, certain measurements that seem particularly privacy-relevant can be removed and replaced by syntactic values using *data interpolation* [43]. Alternatively, if all details should be removed, this can be realized using *data smoothing*, e.g., by means of a *Fourier transform* [72]. On the other hand, artificial noise can be added to the data. Moon et al. [53] present a method that adds *Gaussian noise* to a time series which cannot be removed decisively by noise filtering. As shown in Figure 11, both approaches ensure that the general progression is still recognizable — which is a prerequisite for time series data — but details of individual measurements are concealed [78].

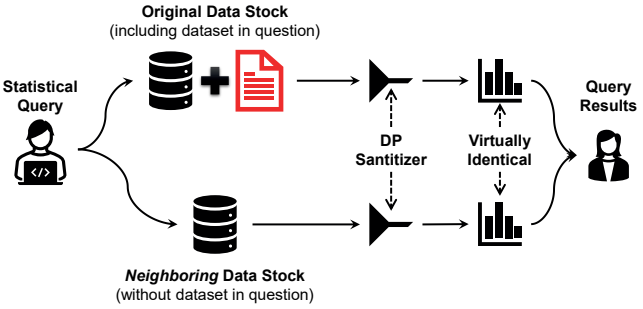

Figure 12: Examples of Statistical Privacy Filters.

Statistical Privacy Filters.

If singular standalone instances are stored in the blockchain, no assertions can be made with regard to the underlying data schema. Therefore, an individual privacy approach must be applied to each data object. If there is no approach available for the respective type, the only option is to conceal the entire affected object. In case the blockchain uses a document-oriented model internally, at least individual attributes of the object can be concealed in a more fine-grained manner.

Yet, in our example of an electronic election, another option is available. Here, it can be assumed that primarily statistical information is retrieved, whereas information on individual objects is hardly ever needed. The latter can therefore either be completely prevented or highly restrictive privacy filters, such as the concealing of entire objects, can be applied. Thus, it only has to be ensured that the statistical queries do not expose any individuals. This would be the case, for instance, if certain properties only apply to a single data object. Then, this data object could be uniquely identified out of the mass of all data objects due to these properties.

To this end, Dwork [19] introduces *differential privacy*. In Figure 12 its underlying concept is shown. If a statistical query is performed on a data stock, it can be checked to what extent a specific data subject is exposed by performing the same query on a *neighboring data stock*, i.e., an identical data stock without any data on the data subject in question. If the results of the two queries differ only by a very small ϵ, it is obvious that the privacy of the data subject is not compromised by the query. If the given ϵ is exceeded, then the data subject would be exposed by the query. In this case, noise has to be added to the query result (indicated by the DP Sanitizer). Dwork et al. [20] use *Laplace noise* for this purpose. This procedure is repeated until the privacy of the data subject is no longer compromised by the query.

Since contrary to the singular standalone instances the data schema is known for the query results, differential privacy can also be applied to our blockchain application example. In addition to the very simple ϵ-*differential privacy* variant described above, further variants, such as (ϵ, δ)-*differential privacy*, can also be implemented as part of statistical privacy filters. Here, an additional parameter δ has to be specified, which describes how likely the ϵ is to be exceeded by a query. This makes the filter less restrictive if rather improbable privacy threats are accepted by the data subjects.

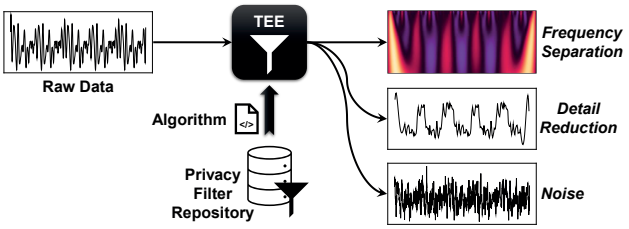

Figure 13: Application of Privacy Filters in a TEE.

Privacy Platform.

Stach et al. [79] present an architecture in which a set of different privacy filters is gathered in a repository and a suitable filter for the respective type of data and use case is selected. A utility metric is used to determine which of the applicable filters provides the best privacy protection but at the same time has the least impact on the quality of the data for the particular use case. The code of the selected filter algorithm is then loaded into a data processor and applied to the data before releasing them to the data consumer.

Yet, this requires a trusted environment in which the filter operator is executed. A blockchain, however, is a trustless system, i.e., the individual parties cannot trust the other participants in the blockchain. Only by reaching a consensus among all participants for any operation, trust in the overall system is established. That is, for the application of privacy filters, a *Trusted Execution Environment* (*TEE*) is required in the blockchain system. A TEE is an isolated execution environment on which only approved applications can be executed. Cryptographic primitives ensure the integrity of the code executed in this environment and other processes have no influence on the execution as well as the outcomes [36]. In a TEE, it can therefore be ensured that the privacy filters cannot be manipulated and are executed correctly.

Figure 13 shows how the privacy filters are applied. Depending on the requested type of data, applicable algorithms are selected from the privacy filter repository. For instance, for time series data, frequency separation can be used to abstract the data progression so that only changes in frequency are visible, the resolution of the data can be lowered to reduce details, or noise can be added to the data. Depending on the use case, the most suitable privacy filter is selected and applied to the raw data in the TEE. This way, both sides (data subject and data consumer) can trust that the privacy filter is applied correctly. For the data subject this means that the desired privacy level is maintained and for the data processor that the promised data quality is delivered.

5.5 Trusted Privacy Control Environment

The technical solutions shown in this section for ensuring data protection principles in a blockchain system, such as purpose limitation, data minimization, right to rectification, or right to erasure, however, also require an extension of the conceptual infrastructure of a blockchain so that they can be applied reliably. This is necessary whenever personal data are managed and processed by the blockchain, otherwise, as discussed in Section 3, a blockchain cannot be operated in a GDPR-compliant manner. Such sensitive data cannot be

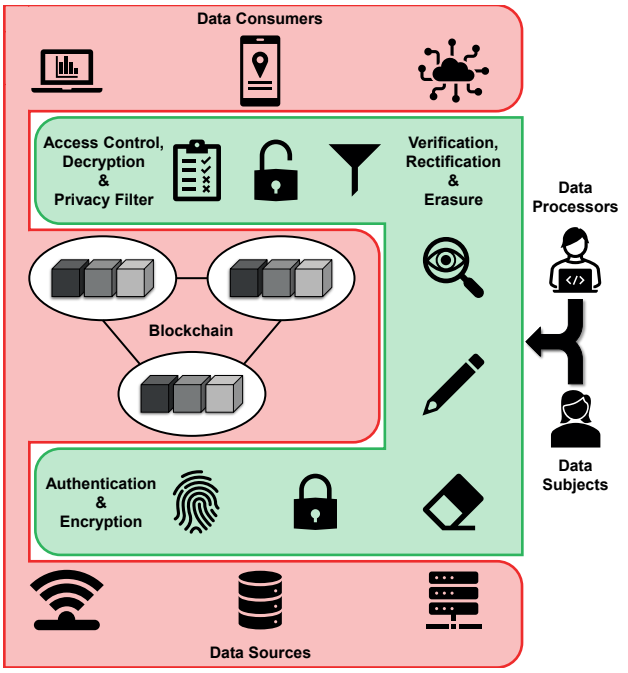

Figure 14: Embedding of a Blockchain in a Trusted Privacy Control Environment (trusted components are depicted in green while trustless components are depicted in red).

kept confidential in public blockchains for obvious reasons. Thus, if such data is involved, a closed set of participants, i.e., a private blockchain, can be assumed. Nevertheless, each node is considered trustless in its own right (as well as the data sources and data consumers) — otherwise, one would not need a blockchain.

Our approach is therefore to embed the trustless distributed components in a *trusted environment* that can be controlled by a central authority. In this way, our approach also meets the demand for a data controller. Stach et al. [81] present such a control environment for distributed Internet of Things applications. Figure 14 shows how we adapted this approach to a blockchain environment.

The blockchain is completely isolated from both, data sources and data consumers. If a source wants to add data to the blockchain (or rather its data pool), this must be done via an interface controlled by the trusted environment. Here, the attribute-based data authentication comes into play (see Section 5.2), which can verify whether the source has the necessary properties to be able to provide trustable data. If the verification is successful, personal data are encrypted before they are forwarded to the blockchain to enable data purging by encryption (see Section 5.1). The blockchain then processes the data autonomously and unaffected by the control environment. That is, its crucial key properties (decentralized, immutable, and tamper-proof) are not impaired by any means.

If data consumers want to gain access to the data stored on the blockchain, this is also done via a restricted interface. In this interface, the purpose-based permission rules (see

Section 5.3) are checked to determine whether the consumer has the required access rights. If this is the case, personal data is decrypted, and privacy filters are applied to them according to the relevant permission rule (see Section 5.4). Yet, these privacy filters tamper with the data. Therefore, an additional verification interface is required for data processors to prove the authenticity of the underlying raw data (guaranteed by the blockchain itself) as well as the correct execution of the applied privacy filters (guaranteed by the TEE). This user interface also needs to enable data subjects to enforce rectification and erasure via the data purging techniques and express new or changed privacy requirements in terms of permission rules.

While all of this is feasible from a technical perspective, from an organizational perspective, however, it has to be resolved who has the responsibility for operating the trusted control environment. This operator has complete control over the data, as s/he controls which data are added to the blockchain and which data from the blockchain are made available to whom. Therefore, both, data subjects and data processors have to trust this controller implicitly. From our point of view, only the data protection officer of the organization which operates the private blockchain is eligible, as s/he is unbiased and trustworthy.

All technical solutions discussed in this paper, as well as their role in terms of a privacy-aware blockchain, are outlined in Table 3.

Table 3: Summary of the Discussed Technical Solutions and their Contribution towards Data Protection.

Technical Solution	Contribution towards Data Protection
Data Purging by Encryption	Due to the full encryption of all data in the blockchain, it is possible to delete data by deleting their respective decryption key. From a technical perspective, the data are still available, but they are no longer readable. This addresses all privacy issues that are related to the *revision or deletion of data*, e.g., Article 5(1)(d), Article 5(1)(e), Article 16, and Article 17.
Attribute-Based Data Authentication	By authenticating data sources and determining certain characteristics, inappropriate data sources can be easily identified. Their data can thus be excluded from the blockchain, as the expected data quality from such sources is low. This addresses all privacy issues that are related to the *quality and correctness of data*, e.g., Article 5(1)(d), Article 16, and Article 17.
Purpose-Based Permission Control	Fine-grained access control enables data subjects to specify who gets access to their data and for what purpose, without requiring a smart contract. This addresses all privacy issues that are related to the *processing of data*, e.g., Article 5(1)(b), Article 7, Article 18, and Article 22.
Privacy Filters	By applying privacy filters, data quality can be adjusted to reduce the amount of disclosed sensitive information. This addresses all privacy issues that are related to the *information value of data*, e.g., Article 5(1)(c), Article 5(1)(d), and Article 18.
Trusted Privacy Control Environment	By embedding these four techniques in a central control environment and isolating the blockchain from data sinks and data sources, a privacy-aware operation of the blockchain can be realized. This addresses all privacy issues that are related to the *management of data*, e.g., Article 12 – 15, Article 22, and Article 24, and enables *data protection by design* (Article 25).

6. FUTURE RESEARCH DIRECTIONS

As shown in the previous section, the five discussed technical solutions represent a major step towards a privacy-aware blockchain in the sense of the GDPR. By means of data purging by encryption, compliance with Article 16 and Article 17 — i.e., the right to rectification and the right to erasure — is achieved. Since rectification is realized by deleting the incorrect data and resubmitting the corrected data (which requires considerable effort due to the consensus protocol), attribute-based data authentication helps to ensure that the data sources are appropriate before they can add data to the blockchain in order to maintain high data accuracy (Article 5(1)(d)). Via the purpose-based permission control, data subjects are empowered to exercise their right to restriction of processing (Article 18), as they can specify in fine-grained manner which data are processed for which purpose (Article 5(1)(b)). The associated privacy filters achieve data minimization (Article 5(1)(c)), since only the information required for processing is passed on to a data consumer. The trusted privacy control environment, in which all of these concepts can be embedded, provides an additional *virtual* storage limitation (Article 5(1)(e)), since on the one hand the incoming data and on the other hand the visibility of the available data can be restricted. Furthermore, with this environment, data protection officers are enabled to take on the role of data controllers and thus represent a central point of contact for data subjects (Article 24). This environment also limits the power of smart contracts, as they can no longer be used for automated individual decision-making (Article 22), as their results initially remain completely isolated in the blockchain until they are approved by the data controller.

Even though we have achieved a lot in terms of privacy-aware blockchains with the concepts presented in Section 5, to us there are still two major research gaps that need to be addressed, namely *verifiable control of the datasets* held by the nodes and *comprehensive query capabilities* for blockchains. From our point of view, these two components are the key research gap towards a comprehensive privacy-

by-design blockchain (Article 25). We flesh out these two future research directions in the following:

Verifiable Control of the Datasets.

Our proposed trusted privacy control environment (see Section 5.5) assumes that there is a central authority which is responsible for enforcing the rights of data subjects. To this end, however, it is imperative that the nodes governing the instances of the blockchain, i.e., the payload data, are in general not malicious. This can be assumed, especially for private blockchains. Yet, there are no guarantees. While a few malicious nodes may not impair data security, a third party could gain control over data sovereignty by taking over the majority of nodes. This third party could corrupt the data in all of its nodes. Thus, the majority of nodes agrees on this alternate version of the blockchain and thereby rendering the corrupted data to the single point of truth — i.e., untrue information about data subjects is spread.

To prevent this, it is necessary for the central authority to regularly verify that all nodes are using an authentic instance of the blockchain. Complete verification of all data is out of the question for reasons of efficiency. Nevertheless, it is necessary that all nodes provide an unforgeable proof. In the context of cloud storage, proofs of retrievability are used for such a purpose. This enables file owners to check whether a cloud server is storing their files correctly [25]. As part of future work, it is necessary to investigate whether such an approach can be transferred to the nodes of a blockchain system, or how it needs to be adapted for this purpose.

Comprehensive Query Capabilities.

Comprehensive query capabilities are required in blockchain systems in order to fully comply with the information obligations towards data subjects (Article 12 – 15). However, as discussed in Section 2.5, blockchain systems have only rudimentary query capabilities. For instance, in order to identify all data concerning a specific data subject, all blocks must be processed sequentially, and all contained data objects have to be read one by one to check whether they concern the data subject in question. Yet, this represents a huge overhead.

There are various research approaches that deal with this topic. For instance, there is an SQL-like query language for Ethereum [8]. SQL-like queries for smart contracts are also studied [28]. With the help of such query capabilities for blockchain systems, simple analyses on blockchain data are also possible [44]. Yet, the special characteristics of blockchains are insufficiently taken into account, which means that current use cases (see Section 2.6) are not efficiently supported by such query layers. Xu et al. [102] address the structure of a blockchain in their work by enabling range queries over the block and extending the query language accordingly. Other approaches extended query capabilities for very specific use cases or data types, such as spatio-temporal data [70].

What is missing, however, is a holistic approach for generic multipurpose blockchains. From our point of view, three components are needed for this, which are seamlessly intertwined:

Query Language. Contrary to today's systems, blockchains should support a descriptive query language. This enables data subjects to formulate their information requests accordingly. Yet, extensions are necessary as there are additional metadata available in blockchains. For instance, all data have an implicit inherent timestamp. Furthermore, a history of the data objects exists, which describes how an object has changed over time. These special properties of blockchain data must also be reflected by a query language. Such aspects are not sufficiently addressed in current approaches [69].

Index Structures. Concepts are needed that enable efficient data access, e.g., when searching for all data on a specific data subject. Przytarski [67] proposes a triple-based data model for this purpose, i.e. subject – predicate – object. Conventional triple stores provide six different indexes for data access — one for each permutation of the triples. With regard to the types of queries mostly used in blockchains, it is necessary to study whether all six indexes are also required here. In particular, since blockchains are append-only stores, whereby the data volume is constantly growing, it would be beneficial if the number of indexes could be kept small. Too few indexes, however, would result in high query costs.

Federation Concept. As illustrated in Section 2.6, often only a small part of the payload data (or sometimes none at all) is actually stored in the blockchain and there are external data stores that contain the majority of the payload data. In order to be able to gather all data that are available about a particular data subject, the data from the blockchain must be merged with the data from the data stores. This requires a federation concept that routes incoming queries to both the blockchain and the data stores and consolidates the query results. There are federation layers for different blockchain systems [94] and federation layers over multiple databases [16]. Yet, there is no fusion of these two approaches.

From our perspective, these are the two most significant challenges researchers have to overcome, in order to enable a privacy-friendly blockchain system.

7. CONCLUSION

Whenever data have to be shared securely between several parties, the use of a blockchain is a suitable option. Blockchains ensure that the data are immutable, tamper-proof, and available to all participants in a transparent manner. Yet, it is due to these characteristics that they conflict with data privacy laws such as the GDPR.

To this end, we assessed in this paper, whether privacy-aware blockchains are feasible. In this regard, we provided the following three contributions: 1. First, we identified with which articles of the GDPR there is a conflict. 2. Then, we presented five technical solutions that address these conflicts (namely data purging by encryption, attribute-based data authentication, purpose-based permission control, privacy filters, and a trusted privacy control environment) and described how they can be applied to a blockchain. 3. Finally, we discussed why mechanisms to verify the data stored on the nodes of a blockchain as well as comprehensive query capabilities are crucial, yet open research questions. They need to be solved in order to facilitate a privacy-by-design blockchain fully compliant with the GDPR.

Although blockchains have inherent privacy issues, they can be reconciled with data protection laws. Technical and organizational adjustments are required, however, and there is still a lot of research necessary to make these data protection measures in blockchain systems efficient and effective.

Acknowledgments

Some work presented in this paper was performed in the project 'NUCLIDE' as part of the Software Campus program, which is funded by the German Federal Ministry of Education and Research (BMBF) under grant number 01IS17051.

8. REFERENCES

[1] A. Abuhashim and C. C. Tan. Smart Contract Designs on Blockchain Applications. In *Proc. of the 2020 IEEE Symposium on Computers and Communications*, ISCC, pages 1–4, 2020.

[2] T. Ali Syed, A. Alzahrani, S. Jan, M. S. Siddiqui, A. Nadeem, and T. Alghamdi. A Comparative Analysis of Blockchain Architecture and its Applications: Problems and Recommendations. *IEEE Access*, 7:176838–176869, 2019.

[3] Z. A. Almusaylim and N. Jhanjhi. Comprehensive Review: Privacy Protection of User in Location-Aware Services of Mobile Cloud Computing. *Wireless Personal Communications*, 111:541–564, 2020.

[4] A. M. Antoniadi, M. Galvin, M. Heverin, O. Hardiman, and C. Mooney. Prediction of Quality of Life in People with ALS: On the Road towards Explainable Clinical Decision Support. *ACM SIGAPP Applied Computing Review*, 21(2):5–17, 2021.

[5] A. Baldominos and Y. Saez. Coin.AI: A Proof-of-Useful-Work Scheme for Blockchain-Based Distributed Deep Learning. *Entropy*, 21(8):723, 2019.

[6] I. Bentov, A. Gabizon, and A. Mizrahi. Cryptocurrencies Without Proof of Work. In *Proc. of the 20th International Conference on Financial Cryptography and Data Security (Workshops)*, BITCOIN, pages 142–157, 2016.

[7] D. Berdik, S. Otoum, N. Schmidt, D. Porter, and Y. Jararweh. A Survey on Blockchain for Information Systems Management and Security. *Information Processing & Management*, 58(1):102397, 2021.

[8] S. Bragagnolo, H. Rocha, M. Denker, and S. Ducasse. Ethereum Query Language. In *Proc. of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*, WETSEB, pages 1–8, 2018.

[9] C. Cachin. Architecture of the Hyperledger Blockchain Fabric. IBM Research, 2016.

[10] L. Campanile, M. Iacono, F. Marulli, and M. Mastroianni. Designing a GDPR compliant blockchain-based IoV distributed information tracking system. *Information Processing & Management*, 58(3):102511, 2021.

[11] M. Castro and B. Liskov. Practical Byzantine Fault Tolerance. In *Proc. of the Third Symposium on Operating Systems Design and Implementation*, OSDI, pages 173–186, 1999.

[12] D. L. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[13] Y. Chen, J. Li, C. Liu, J. Han, Y. Zhang, and P. Yi. Efficient Attribute Based Server-Aided Verification Signature. *IEEE Transactions on Services Computing (Early Access)*, pages 1–9, 2021.

[14] M. S. Chishti, F. Sufyan, and A. Banerjee. Decentralized On-Chain Data Access via Smart Contracts in Ethereum Blockchain. *IEEE Transactions on Network and Service Management*, 19(1):174–187, 2022.

[15] M.-J. Choi, H.-S. Kim, and Y.-S. Moon. Publishing Sensitive Time-Series Data under Preservation of Privacy and Distance Orders. *International Journal of Innovative Computing, Information and Control*, 8(5(B)):3619–3638, 2012.

[16] T. M. De Farias, C. Dessimoz, A. A. Benitez, C. Yang, J. Long, and A.-C. Sima. Federating and querying heterogeneous and distributed Web APIs and triple stores. In *Proc. of the 30th Conference on Intelligent Systems for Molecular Biology*, ISMB, pages Q–001:1–Q–001:2, 2022.

[17] D. Di Francesco Maesa and P. Mori. Blockchain 3.0 applications survey. *Journal of Parallel and Distributed Computing*, 138:99–114, 2020.

[18] D. Di Francesco Maesa, P. Mori, and L. Ricci. Blockchain Based Access Control. In *Proc. of the 17th IFIP International Conference on Distributed Applications and Interoperable Systems*, DAIS, pages 206–220, 2017.

[19] C. Dwork. Differential Privacy. In *Proc. of the 33rd International Colloquium on Automata, Languages and Programming*, ICALP, pages 1–12, 2006.

[20] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In *Proc. of the Third Theory of Cryptography Conference*, TCC, pages 265–284, 2006.

[21] European Parliament and Council of the European Union. Regulation on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/EC (Data Protection Directive). Legislative acts L119, Official Journal of the European Union, 2016.

[22] M. Finck. Blockchain and the General Data Protection Regulation: Can distributed ledgers be squared with European data protection law? European Parliamentary Research Service PE 634.445, 2019.

[23] Z. Gao, L. Xu, L. Chen, X. Zhao, Y. Lu, and W. Shi. CoC: A Unified Distributed Ledger Based Supply Chain Management System. *Journal of Computer Science and Technology*, 33(2):237–248, 2018.

[24] K. Gjøsteen, C. Gritti, and K. N. Moran. Ballot Logistics: Tracking Paper-based Ballots Using Cryptography. In *Proc. of the Fifth International Joint Conference on Electronic Voting*, E-Vote-ID, pages 259–274, 2020.

[25] C. Gritti and H. Li. Efficient Publicly Verifiable Proofs of Data Replication and Retrievability

Applicable for Cloud Storage. *Advances in Science, Technology and Engineering Systems Journal*, 7(1):107–124, 2022.

[26] C. Gritti, M. Önen, and R. Molva. CHARIOT: Cloud-Assisted Access Control for the Internet of Things. In *Proc. of the 2018 16th Annual Conference on Privacy, Security and Trust*, PST, pages 1–6, 2018.

[27] C. Gritti, M. Önen, and R. Molva. Privacy-Preserving Delegable Authentication in the Internet of Things. In *Proc. of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC, pages 861–869, 2019.

[28] J. Han, H. Kim, H. Eom, J. Coignard, K. Wu, and Y. Son. Enabling SQL-Query Processing for Ethereum-Based Blockchain Systems. In *Proc. of the 9th International Conference on Web Intelligence, Mining and Semantics*, WIMS, pages 9:1–9:7, 2019.

[29] A. B. Haque, A. K. M. N. Islam, S. Hyrynsalmi, B. Naqvi, and K. Smolander. GDPR Compliant Blockchains – A Systematic Literature Review. *IEEE Access*, 9:50593–50606, 2021.

[30] C. V. Helliar, L. Crawford, L. Rocca, C. Teodori, and M. Veneziani. Permissionless and permissioned blockchain diffusion. *International Journal of Information Management*, 54:102136, 2020.

[31] T. Hepp, M. Sharinghousen, P. Ehret, A. Schoenhals, and B. Gipp. On-chain vs. off-chain storage for supply- and blockchain integration. *it – Information Technology*, 60(5–6):283–291, 2018.

[32] T. M. Hewa, Y. Hu, M. Liyanage, S. S. Kanhare, and M. Ylianttila. Survey on Blockchain-Based Smart Contracts: Technical Aspects and Future Research. *IEEE Access*, 9:87643–87662, 2021.

[33] F. Hofmann, S. Wurster, E. Ron, and M. Böhmecke-Schwafert. The immutability concept of blockchains and benefits of early standardization. In *Proc. of the 2017 ITU Kaleidoscope: Challenges for a Data-Driven Society*, ITU K, pages 1–8, 2017.

[34] B. Inmon. *Data Lake Architecture: Designing the Data Lake and avoiding the garbage dump*. Technics Publications, Basking Ridge, New Jersey, USA, 2016.

[35] W. H. Inmon, D. Strauss, and G. Neushloss. *DW 2.0: The Architecture for the Next Generation of Data Warehousing*. Morgan Kaufmann Publishers Inc., Burlington, Massachusetts, USA, 2008.

[36] P. Jauernig, A.-R. Sadeghi, and E. Stapf. Trusted Execution Environments: Properties, Applications, and Challenges. *IEEE Security & Privacy*, 18(2):56–60, 2020.

[37] M. Y. Khan, M. F. Zuhairi, T. Ali, T. Alghamdi, and J. A. Marmolejo-Saucedo. An extended access control model for permissioned blockchain frameworks. *Wireless Networks*, 26(7):4943–4954, 2020.

[38] P. Kohli, S. Sharma, and P. Matta. Security Challenges, Applications and Vehicular Authentication Methods in VANET for Smart Traffic Management. In *Proc. of the 2021 2nd International Conference on Intelligent Engineering and Management*, ICIEM, pages 327–332, 2021.

[39] S. Krishnan, V. E. Balas, E. Golden Julie, Y. H. Robinson, S. Balaji, and R. Kumar, editors. *Handbook of Research on Blockchain Technology*. Academic Press, London, San Diego, Cambridge, and Oxford, 2020.

[40] S. S. Kushwaha, S. Joshi, D. Singh, M. Kaur, and H.-N. Lee. Systematic Review of Security Vulnerabilities in Ethereum Blockchain Smart Contract. *IEEE Access*, 10:6605–6621, 2022.

[41] R. Lai and D. Lee Kuo Chuen. Blockchain – From Public to Private. In D. Lee Kuo Chuen and R. Deng, editors, *Handbook of Blockchain, Digital Finance, and Inclusion, Volume 2*, chapter 7, pages 145–177. Academic Press, 2018.

[42] S. Lazarova-Molnar, H. t. Logason, P. G. Andersen, and M. B. Kjærgaard. Mobile Crowdsourcing of Occupant Feedback in Smart Buildings. *ACM SIGAPP Applied Computing Review*, 17(1):5–14, 2017.

[43] M. Lepot, J.-B. Aubin, and F. H. Clemens. Interpolation in Time Series: An Introductive Overview of Existing Methods, Their Performance Criteria and Uncertainty Assessment. *Water*, 9(10):796, 2017.

[44] Y. Li, K. Zheng, Y. Yan, Q. Liu, and X. Zhou. EtherQL: A Query Layer for Blockchain System. In *Proc. of the 22nd International Conference on Database Systems for Advanced Applications*, DASFAA, pages 556–567, 2017.

[45] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla. ProvChain: A Blockchain-Based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability. In *Proc. of the 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, CCGRID, pages 468–477, 2017.

[46] A. A. Mamun, S. Azam, and C. Gritti. Blockchain-Based Electronic Health Records Management: A Comprehensive Review and Future Research Direction. *IEEE Access*, 10:5768–5789, 2022.

[47] M. Manteghi. Blockchain and the European Union's General Data Protection Regulation: From Conflict to "Peaceful" Coexistence? SSRN, 2021.

[48] R. C. Merkle. A Digital Signature Based on a Conventional Encryption Function. In *Proc. of the Conference on the Theory and Applications of Cryptographic Techniques*, CRYPTO, pages 369–378, 1988.

[49] K. Mindermann, F. Riedel, A. Abdulkhaleq, C. Stach, and S. Wagner. Exploratory Study of the Privacy Extension for System Theoretic Process Analysis (STPA-Priv) to elicit Privacy Risks in eHealth. In *Proc. of the 2017 IEEE 25th International Requirements Engineering Conference (Workshops)*, REW/ESPRE, pages 90–96, 2017.

[50] K. Miyachi and T. K. Mackey. hOCBS: A privacy-preserving blockchain framework for healthcare data leveraging an on-chain and off-chain system design. *Information Processing & Management*, 58(3):102535, 2021.

[51] F. Molina, G. Betarte, and C. Luna. Design principles for constructing GDPR-compliant blockchain solutions. In *Proc. of the 2021 IEEE/ACM 4th*

International Workshop on Emerging Trends in Software Engineering for Blockchain, WETSEB, pages 1–8, 2021.

[52] A. A. Monrat, O. Schelén, and K. Andersson. A Survey of Blockchain From the Perspectives of Applications, Challenges, and Opportunities. *IEEE Access*, 7:117134–117151, 2019.

[53] Y.-S. Moon, H.-S. Kim, S.-P. Kim, and E. Bertino. Publishing Time-Series Data under Preservation of Privacy and Distance Orders. In *Proc. of the 21th International Conference on Database and Expert Systems Applications*, DEXA, pages 17–31, 2010.

[54] K. B. Muthe, K. Sharma, and K. E. N. Sri. A Blockchain Based Decentralized Computing And NFT Infrastructure For Game Networks. In *Proc. of the 2020 Second International Conference on Blockchain Computing and Applications*, BCCA, pages 73–77, 2020.

[55] M. Muzammal, Q. Qu, and B. Nasrulin. Renovating blockchain with distributed databases: An open source system. *Future Generation Computer Systems*, 90:105–117, 2019.

[56] T. Nakaike, Q. Zhang, Y. Ueda, T. Inagaki, and M. Ohara. Hyperledger Fabric Performance Characterization and Optimization Using GoLevelDB Benchmark. In *Proc. of the 2020 IEEE International Conference on Blockchain and Cryptocurrency*, ICBC, pages 1–9, 2020.

[57] S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. Bitcoin Project, 2008.

[58] M. S. Ozdayi, M. Kantarcioglu, and B. Malin. Leveraging blockchain for immutable logging and querying across multiple sites. *BMC Medical Genomics*, 13(7):82, 2020.

[59] S. M. Palanisamy, F. Dürr, M. A. Tariq, and K. Rothermel. Preserving Privacy and Quality of Service in Complex Event Processing through Event Reordering. In *Proc. of the 12th ACM International Conference on Distributed and Event-Based Systems*, DEBS, pages 40–51, 2018.

[60] A. Park, J. Kietzmann, L. Pitt, and A. Dabirian. The Evolution of Nonfungible Tokens: Complexity and Novelty of NFT Use-Cases. *IT Professional*, 24(1):9–14, 2022.

[61] Y. R. Park, E. Lee, W. Na, S. Park, Y. Lee, and J.-H. Lee. Is Blockchain Technology Suitable for Managing Personal Health Records? Mixed-Methods Study to Test Feasibility. *Journal of Medical Internet Research*, 21(2):e12533, 2019.

[62] R. Patgiri, S. Nayak, and N. B. Muppalaneni. Is Bloom Filter a Bad Choice for Security and Privacy? In *Proc. of the 2021 International Conference on Information Networking*, ICOIN, pages 648–653, 2021.

[63] Y. Peng, M. Du, F. Li, R. Cheng, and D. Song. FalconDB: Blockchain-Based Collaborative Database. In *Proc. of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD, pages 637–652, 2020.

[64] C. Pinto-Gutiérrez, S. Gaitán, D. Jaramillo, and S. Velasquez. The NFT Hype: What Draws Attention to Non-Fungible Tokens? *Mathematics*, 10(3):335, 2022.

[65] M. Poelman and S. Iqbal. Investigating the Compliance of the GDPR: Processing Personal Data On A Blockchain. In *Proc. of the 2021 IEEE 5th International Conference on Cryptography, Security and Privacy*, CSP, pages 38–44, 2021.

[66] A. Poibrenski, M. Klusch, I. Vozniak, and C. Müller. Multimodal Multi-Pedestrian Path Prediction for Autonomous Cars. *ACM SIGAPP Applied Computing Review*, 20(4):5–17, 2021.

[67] D. Przytarski. Using Triples as the Data Model for Blockchain Systems. In *Proc. of the 18th International Semantic Web Conference (Workshops)*, BlockSW, pages 1–2, 2019.

[68] D. Przytarski, C. Stach, C. Gritti, and B. Mitschang. A Blueprint for a Trustworthy Health Data Platform Encompassing IoT and Blockchain Technologies. In *Proc. of the ISCA 29th International Conference on Software Engineering and Data Engineering*, SEDE, pages 56–65, 2020.

[69] D. Przytarski, C. Stach, C. Gritti, and B. Mitschang. Query Processing in Blockchain Systems: Current State and Future Challenges. *Future Internet*, 14(1):1, 2022.

[70] Q. Qu, I. Nurgaliev, M. Muzammal, C. S. Jensen, and J. Fan. On spatio-temporal blockchain query processing. *Future Generation Computer Systems*, 98:208–218, 2019.

[71] M. Romero, W. Guédria, H. Panetto, and B. Barafort. Towards a Characterisation of Smart Systems: A Systematic Literature Review. *Computers in Industry*, 120:103224, 2020.

[72] T. Sakamoto, M. Yokozawa, H. Toritani, M. Shibayama, N. Ishitsuka, and H. Ohno. A crop phenology detection method using time-series MODIS data. *Remote Sensing of Environment*, 96(3):366–374, 2005.

[73] S. Sayeed and H. Marco-Gisbert. Assessing Blockchain Consensus and Security Mechanisms against the 51% Attack. *Applied Sciences*, 9(9):1788, 2019.

[74] D. Schwartz, N. Youngs, and A. Britto. The Ripple Protocol Consensus Algorithm. Ripple, 2014.

[75] N. Scope, A. Rasin, J. Wagner, B. Lenard, and K. Heart. Purging Data from Backups by Encryption. In *Proc. of the 32nd International Conference on Database and Expert Systems Applications*, DEXA, pages 245–258, 2021.

[76] M. Shuaib, S. Alam, M. Shabbir Alam, and M. Shahnawaz Nasir. Compliance with HIPAA and GDPR in blockchain-based electronic health record. *Materials Today: Proceedings*, pages 1–6, 2021.

[77] W. L. Sim, H. N. Chua, and M. Tahir. Blockchain for Identity Management: The Implications to Personal Data Protection. In *Proc. of the 2019 IEEE Conference on Application, Information and Network Security*, AINS, pages 30–35, 2019.

[78] C. Stach. VAULT: A Privacy Approach towards High-Utility Time Series Data. In *Proc. of the Thirteenth International Conference on Emerging*

Security Information, Systems and Technologies, SECURWARE, pages 41–46, 2019.

[79] C. Stach, J. Bräcker, R. Eichler, C. Giebler, and C. Gritti. How to Provide High-Utility Time Series Data in a Privacy-Aware Manner: A VAULT to Manage Time Series Data. *International Journal on Advances in Security*, 13(3 & 4):88–108, 2020.

[80] C. Stach and A. Brodt. vHike – A Dynamic Ride-sharing Service for Smartphones. In *Proc. of the 2011 IEEE 12th International Conference on Mobile Data Management*, MDM, pages 333–336, 2011.

[81] C. Stach, F. Dürr, K. Mindermann, S. M. Palanisamy, and S. Wagner. How a Pattern-based Privacy System Contributes to Improve Context Recognition. In *Proc. of the 2020 IEEE International Conference on Pervasive Computing and Communications (Workshops)*, CoMoRea, pages 238–243, 2018.

[82] C. Stach, C. Gritti, and B. Mitschang. Bringing Privacy Control Back to Citizens: DISPEL — A Distributed Privacy Management Platform for the Internet of Things. In *Proc. of the 35th ACM/SIGAPP Symposium on Applied Computing*, SAC, pages 1272–1279, 2020.

[83] C. Stach, C. Gritti, D. Przytarski, and B. Mitschang. Trustworthy, Secure, and Privacy-aware Food Monitoring Enabled by Blockchains and the IoT. In *Proc. of the 2020 IEEE International Conference on Pervasive Computing and Communications Workshops*, PerCom, pages 50:1–50:4, 2020.

[84] C. Stach and B. Mitschang. CURATOR—A Secure Shared Object Store: Design, Implementation, and Evaluation of a Manageable, Secure, and Performant Data Exchange Mechanism for Smart Devices. In *Proc. of the 33rd Annual ACM Symposium on Applied Computing*, SAC, pages 533–540, 2018.

[85] C. Stach and F. Steimle. Recommender-based Privacy Requirements Elicitation – EPICUREAN: An Approach to Simplify Privacy Settings in IoT Applications with Respect to the GDPR. In *Proc. of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC, pages 1500–1507, 2019.

[86] C. Stach, F. Steimle, C. Gritti, and B. Mitschang. PSSST! The Privacy System for Smart Service Platforms: An Enabler for Confidable Smart Environments. In *Proc. of the 4th International Conference on Internet of Things, Big Data and Security*, IoTBDS, pages 57–68, 2019.

[87] A. Sunyaev. Distributed Ledger Technology. In *Internet Computing: Principles of Distributed Systems and Emerging Internet-Based Technologies*, pages 265–299. Springer, 2020.

[88] N. Szabo. Formalizing and Securing Relationships on Public Networks. *First Monday*, 2(9), 1997.

[89] Y. Takabatake and Y. Okabe. An Anonymous Distributed Electronic Voting System Using Zerocoin. In *Proc. of the 2021 International Conference on Information Networking*, ICOIN, pages 163–168, 2021.

[90] N. Tariq, A. Qamar, M. Asim, and F. A. Khan. Blockchain and Smart Healthcare Security: A Survey. *Procedia Computer Science*, 175:615–620, 2020.

[91] P. Tasatanattakool and C. Techapanupreeda. Blockchain: Challenges and applications. In *Proc. of the 2018 International Conference on Information Networking*, ICOIN, pages 473–475, 2018.

[92] U. Tatar, Y. Gokce, and B. Nussbaum. Law versus technology: Blockchain, GDPR, and tough tradeoffs. *Computer Law & Security Review*, 38:105454, 2020.

[93] N. Thamer and R. Alubady. A Survey of Ransomware Attacks for Healthcare Systems: Risks, Challenges, Solutions and Opportunity of Research. In *Proc. of the 2021 1st Babylon International Conference on Information Technology and Science*, BICITS, pages 210–216, 2021.

[94] D. Trihinas. Datachain: A Query Framework for Blockchains. In *Proc. of the 11th International Conference on Management of Digital EcoSystems*, MEDES, pages 134–141, 2019.

[95] L. Tseng, X. Yao, S. Otoum, M. Aloqaily, and Y. Jararweh. Blockchain-based database in an iot environment: challenges, opportunities, and analysis. *Cluster Computing*, 23(3):2151–2165, 2020.

[96] S. Underwood. Blockchain beyond Bitcoin. *Communications of the ACM*, 59(11):15–17, 2016.

[97] K. S. S. Wai, E. C. Htoon, and N. N. M. Thein. Storage Structure of Student Record based on Hyperledger Fabric Blockchain. In *Proc. of the 2019 International Conference on Advanced Information Technologies*, ICAIT, pages 108–113, 2019.

[98] T. Waizenegger, F. Wagner, and C. Mega. SDOS: Using Trusted Platform Modules for Secure Cryptographic Deletion in the Swift Object Store. In *Proc. of the 20th International Conference on Extending Database Technology*, EDBT, pages 550–553, 2017.

[99] K. Wang, Y. Yan, S. Guo, X. Wei, and S. Shao. On-Chain and Off-Chain Collaborative Management System Based on Consortium Blockchain. In *Proc. of the 7th International Conference on Artificial Intelligence and Security*, ICAIS, pages 172–187, 2021.

[100] G. Wood. Ethereum: A Secure Decentralised Generalised Transaction Ledger. Ethereum Yellow Paper Berlin Version 888949c, 2021.

[101] H. P. Wouda and R. Opdenakker. Blockchain technology in commercial real estate transactions. *Journal of Property Investment & Finance*, 37(6):570–579, 2019.

[102] C. Xu, C. Zhang, and J. Xu. vChain: Enabling Verifiable Boolean Range Queries over Blockchain Databases. In *Proc. of the 2019 International Conference on Management of Data*, SIGMOD, pages 141–158, 2019.

[103] S. Zhang and J.-H. Lee. Analysis of the main consensus protocols of blockchain. *ICT Express*, 6(2):93–97, 2020.

[104] M. T. Zia, M. A. Khan, and H. El-Sayed. Application of Differential Privacy Approach in Healthcare Data – A Case Study. In *Proc. of the 2020 14th International Conference on Innovations in Information Technology*, IIT, pages 35–39, 2020.

ABOUT THE AUTHORS:

Christoph Stach is a postdoctoral researcher at the Applications of Parallel and Distributed Systems department at the University of Stuttgart in Germany. He received his PhD in Computer Science from the University of Stuttgart in 2017 for his research in information security and data privacy in mobile applications. Following his successful doctorate, he was appointed academic councilor at the Institute for Parallel and Distributed Systems at the University of Stuttgart. From June 2020 to September 2021, he held the deputy professorship in Data Engineering at the University of Stuttgart. His current research focuses on concepts and tools that enable trustworthy and demand-oriented data provisioning. To this end, his research also addresses issues related to data acquisition, data management, data security, and data protection.

Clementine Gritti is a lecturer at the Computer Science and Software Engineering department within the University of Canterbury. Her current research interests are the design and evaluation of public-key cryptographic protocols for security and privacy in various environments, such as cloud computing, the Internet of Things and blockchains. She previously worked on several research projects dealing with information security and privacy for electronic health and electronic voting. Prior to joining the University of Canterbury in 2020, she worked at the Norwegian University of Science and Technology in Norway and at the graduate school and research center in digital sciences Eurecom in France. She obtained her PhD in Computer Science in 2016 from the University of Wollongong in Australia.

Dennis Przytarski is a PhD student at the Institute for Parallel and Distributed Systems at the University of Stuttgart in Germany. He received his M.Sc. in Software Engineering from the University of Stuttgart in 2016. In his PhD project, he investigates the research questions of how blockchains can be leveraged as trusted data stores for IoT data and how they can be integrated into prevailing Big Data infrastructures. To this end, he examines novel storage concepts, efficient access structures, and advanced query capabilities in the context of blockchain technologies. In 2020, he was admitted to the Software Campus, a research program of the Federal Ministry of Education and Research (BMBF), due to his hands-on research work.

Bernhard Mitschang is professor for Database and Information Systems and head of the department ‘Applications of Parallel and Distributed Systems’ that is part of the Institute of Parallel and Distributed Systems at the Universität Stuttgart, Germany. Both research and teaching spectra of his department cover on one hand data-intensive applications ranging from business applications to engineering systems and on the other hand fundamental data management techniques, data analytics as well as scalable data processing architectures. Since 2013, he is CEO of the Graduate School of Excellence on advanced Manufacturing Engineering and head of the Technology Partnership Lab at the university.