CONTENTS

Blockchain enables the development of decentralized business models with enhanced security for critical data. In this chapter, we present a provenance tracking system based on blockchain technology, ensuring transparency and trust in digital records. This system considers multiple stakeholders within a single country but can be extended to a cross-border scenario, potentially supporting provenance data sharing at an international level. The system—B Provenance—handles processes related to digital asset tracking, such as verifying the origin of goods, ensuring authenticity, and maintaining tamper-proof records of ownership and transactions.

By leveraging blockchain's immutability, this system simplifies information exchange among multiple entities, ensuring secure and verifiable digital provenance. We analyze the benefits and implications of blockchain technology in provenance tracking and present an evaluation of the system's performance .

1.INTRODUCTION

Blockchain technology has reached the mainstream due to its role as the foundation of many cryptocurrencies, which have challenged traditional financial systems in recent years. Bitcoin, as the first and most well-known application of blockchain technology, is a digital currency that operates without a central authority [7]. The decentralization enabled by blockchain technology presents a direct alternative to centralized business models, including those used by governments and financial institutions. In a visionary scenario, blockchain technology could provide a decentralized collection of services competing with government functions such as asset provenance tracking, supply chain verification, and intellectual property management.

However, organizations can also view blockchain technology as an opportunity to enhance operational efficiency and ensure greater transparency. Many industries and governments are actively exploring and adopting blockchain technology to improve data integrity and security. A notable example is the Estonian government, which has been implementing blockchain-based services since 2012 [16].

Given the transformative potential of blockchain, this chapter introduces an immutable digital provenance system leveraging blockchain technology. A provenance tracking system based on blockchain can decentralize record-keeping, improving data availability and resilience against fraud or data manipulation. Since multiple stakeholders—including

businesses, regulatory bodies, and certification authorities—depend on provenance systems, a decentralized approach can enhance trust and security compared to conventional centralized solutions.

Despite blockchain's decentralized nature, it is essential to consider the role of regulatory authorities overseeing provenance verification. As discussed, a blockchain-based digital provenance system can balance decentralization while allowing trusted entities to maintain partial oversight. This approach ensures the authenticity of provenance data while leveraging blockchain's tamper-proof and transparent characteristics.

lthough blockchain technology is sometimes seen as having a limited range of applications, implementing an immutable digital provenance system could serve as a foundation for broader adoption in government and industry. In areas such as supply chain management, intellectual property protection, and historical record-keeping, a blockchain-based provenance solution could establish a unified framework for tracking authenticity, ownership, and compliance.

Furthermore, blockchain technology can contribute to improving government efficiency and transparency [16]. Since blockchain decentralizes data storage, it is crucial to assess the effort required for full-scale implementation, including system setup, hardware and software maintenance, and potential scalability challenges. Therefore, we propose an implementation of blockchain technology for immutable digital provenance using smart contracts and evaluate its impact on provenance tracking processes.

2.BACKGROUND

In this section we present background information about blockchain-based distributed ledgers and contracts, the adopted data provenance tracking model, and the usage control policy language used to encode the data accountability rules in our approach.

2.1 Distributed Ledgers Technology

The solution for data accountability and provenance tracking proposed in this paper relies on a public blockchain-based distributed ledger platform, namely the open-source Ethereum Virtual Machine (EVM) [17]. Public blockchain platforms such as the EVM can be seen as distributed databases of transactions that can be accessed and managed by many people who do not necessarily trust each other and do not share a common trusted third party.

The goal of the EVM is twofold: (1) to maintain a decentralized ledger of transactions performed using Ether (ETH), which is Ethereum's virtual currency; and (2) to support the decentralized execution of smart contracts, which are applications that can be assured to run exactly as programmed. Therefore, transactions in the EVM can involve transferring the virtual currency (ETH) from one account to another or executing contract functions.

In contrast to other public blockchain-based platforms such as Bitcoin [7], which only support mining of the digital currency and transaction management, the EVM also provides smart contract functionality. At the time of writing, Bitcoin and Ethereum are the first and second-largest cryptocurrencies by market capitalization, respectively [3].

Users submitting transactions to the EVM hold a private key (512 bits), which is used to generate a public key (160 bits). The generated public key is then used to generate a 256 bits EVM address that serves as public user identifiers. EVM addresses are also used to identify the deployed contracts, however, in contrast to users, contracts do not hold a private/public key pair. The transactions submitted to the EVM are organized in blocks that are transitively chained to each other using a cryptographic hash function, starting from a pre-computed genesis block. Once a block of transactions is added to the chain of blocks (a.k.a. blockchain) it cannot be modified or removed since the verification of the chain of hashes would not anymore be valid. The verification and addition of transactions to blockchains based on the proof-of-work paradigm, is done through the so called

”mining” process, which is the consensus algorithm establishing the node entitled to add a new block to the chain. In this process, miners have to solve a cryptographic challenge, and the winner (the first miner that solves it) is rewarded with a certain amount of coins (5 ETH in Ethereum) plus the transaction fees included in the block. The core idea of the challenge is to regulate the creation of new digital currency and also to reward miners for helping verifying and building the blockchain.

Regarding the contract functionality implemented by the EVM, it simply consists of the capability to allow the verified deployment and execution of stateful, Turing-complete programs. When a contract is deployed, it is assigned a unique address that is used in future transactions as a reference to invoke the contract functions. The invocation of functions that change the contract state must be done using paid transactions; however, EVM nodes can also invoke read-only functions without paying transaction costs, for example, to check the current state of the contract.

The execution fees for contracts are computed using a unit named gas (short for "gasoline"), and a dynamic market exchange value is defined in relation to ETH. When deploying or invoking contract functions, users can specify a gas limit for the transaction in order to restrict the amount of ETH they are willing to pay for the transaction. A gas limit is also set per block added to the chain, which is currently around 4 million gas per block, with a mining rate of approximately 10 seconds per block. For simple transfers of ETH—excluding more expensive contract transactions—the network is capable of processing around 15 transactions per second.

This limit is needed to address non-deterministic issues in the creation or execution of a function, for example, non-termination or excessive usage of blockchain resources, since all transactions must be executed and verified by the miners. In addition to storing the state, contracts may also signal events, which are certified and stored in the blockchain but do not cost as much as storing a contract state variable. Contracts can be implemented on an online compiler, which also allows simulated execution to allow fast testing without the need to deploy in the real or test blockchain.

Many challenges need to be addressed in order to enable such a virtual platform, including synchronization issues, security, and soundness of the distributed protocol. Most of these challenges have been addressed, and it is out of the scope of this paper to discuss the details about the EVM design and implementation.

Our focus is on the use of the EVM as a distributed platform for tracking data provenance and evaluation of usage control policies in a trustworthy and verifiable way using the

contract concept. We restrict ourselves in this section to a high-level overview of the EVM concepts, and we introduce additional details about contracts as needed to understand our proposed solution in the following sections.

2.2 Data Provenance Tracking Model

In order to enable data provenance tracking with smart contracts we follow a structured data model including the specification of data types, data instantiations, and data instances. In our model primitive and composite data types can be specified, where a composite type includes a named list of contained data instantiations. For example, date and string can be specified as primitive data types while a user identity can be defined as a list of full name and date of birth instantiations of these data types. When data instances about a subject is exchanged with controllers the data model should be agreed beforehand using a data modeling language. For the purposes of this paper we adopt the data modeling approach used in the Model-based Security Toolkit (SecKit) [9], which also supports the specification of user identities. The data provenance tracking model is simply a list maintained by subjects where, for each controller, a list of references to the data provided is updated whenever data is transferred. When the controller forwards the subject's data to a processor, an entry for the processor with the list of data received is created in the data provenance tracking model. Since our goal is to store this model in a public blockchain, for privacy reasons we obfuscate the references to the data types, instantiations, and instances. In the following sections we describe in details how our data provenance tracking model is realized for the two possible models we considered.

2.3 Usage Control Policy Language

Data accountability smart contracts specify restrictions on the usage of the subject's data transferred to controllers .In our approach we specify these restrictions using the security policy language proposed by the Model-based Security Toolkit (SecKit) [9], already applied also to Internet of Things [8] and mobile devices [10] domains. In the SecKit policy language detective and preventive mechanisms are specified using ECA (Event-Condition-Action) rules.

Preventive mechanisms allow the specification of rules that do not allow an undesired behavior to occur, while reactive mechanisms simply take compensation actions when

something undesired is observed. In order to support these two types of mechanisms the Event part of the rule considers tentative and actual events, representing respectively activities that are about to take place but were not yet executed or activities already happened. In this paper we are concerned only with preventive mechanisms since our goal is to disallow undesired behaviors by controllers, for example, to prevent data from being misused or exchanged with 3rd parties.

In the Action part of preventive mechanisms enforcement actions can be specified to allow, deny, modify, or delay the execution of data usage activities by controllers. In the Condition part a logical expression using a combination of operators can be specified including event pattern matching, trust relationships, context-based, role-based, propositional, temporal, and cardinality operators. With this policy language it is possible to express, for example, that during a time window (e.g., 30 days) controllers should be allowed to perform an operation only once. Policies are evaluated using a discrete time step window with a configurable time granularity that is configured according to the policy requirements, for example, if the policy refers to restrictions on data usage considering hours, days, months, etc. For details about all operators supported in our policy language we refer the reader to [9].

The policy language proposed by the SecKit also supports the specification of modular configurable mechanism templates using variables. Templates can be dynamically instantiated and disposed using ECA rules following the same mechanism structure. This is particularly useful in order to enable re-use of mechanisms without repeating patterns of enforcement that should be applied for multiple subjects or situations. In the description of our approach we show examples of preventive mechanisms and templates encoded in smart contracts and describe how these mechanisms can be evaluated in the blockchain in a privacy-friendly way.

3.BLOCK CHAIN BASICS

3.1=>Decentralization:

Blockchain operates on a decentralized network, meaning no single entity controls the data.

This ensures that provenance records are not subject to tampering by any central authority.

3.2=>Immutability:

Once a transaction is recorded on the blockchain, it cannot be altered or deleted. This immutability ensures that digital provenance records remain trustworthy and resistant to forgery.

3.3=> Cryptographic Hashing:

Each record on the blockchain is secured using cryptographic hashing (e.g., SHA-256). Any changes to the data will result in a completely different hash, making tampering easily detectable.

3.4=>Transparency and Traceability:

Blockchain provides a transparent and auditable ledger where all transactions are timestamped and linked. This allows users to track the entire history of an asset, ensuring authenticity.

3.5=>Smart Contracts:

Smart contracts automate and enforce rules for data recording and verification without intermediaries. They help in ensuring that provenance records meet predefined conditions before being added to the blockchain.

3.6=> Tokenizatio & DigitalSignatures:

Assets can be tokenized on the blockchain, providing unique digital identities. Digital signatures ensure that only authorized entities can add provenance records.

3.7=>Consensus Mechanisms:

Blockchain relies on consensus protocols (e.g., Proof of Work, Proof of Stake) to validate and add new provenance records, preventing fraudulent transactions.

Would you like a deeper dive into a specific aspect of blockchain for digital provenance?

4. USECASE OVERVIEW

SOLUTION DESIGN CHOICES:

In this section we describe the three models for blockchain based data accountability and provenance tracking we identified. When designing such a solution, a few decisions should be taken with respect to the design of contracts and the blockchain architecture in order to proper address performance and authorization issues. The contract design issues are related to their lifecycle, the required state variables to store the contract information, and authorization policies specifying who should be allowed to read and update the contract variables.

The blockchain architecture decisions are concerned with the adoption of a public, semipublic, or private blockchain solution considering transaction management issues such as authorization and auditability properties. In the following, we first introduce these three models and discuss design and architecture issues that should be addressed in their implementation. The architecture issues we discuss are relevant only on the real-world deployment of the proposed models; in this paper, we are mostly concerned with design, implementation, and performance of these models using the EVM platform.

With respect to the contract lifecycle, we identified three possible models with different cardinality of contracts in relation to the number of data subjects and controllers. Data subject contract for specific controller (a): the subject creates a contract tailored for each controller that manages his/her data. The contract keeps track of the data shared with the controller, the policies regulating the use of the data, and registers the data usage events representing the activities performed by the controller using the subject data as input. Data subject contract for specific data (b): the subject creates a generic contract for each data instance that is shared for all controllers accessing the data. The contract contains the list of controllers that were given access to a particular data instance and the policies they should

respect. Controller contract for multiple data subjects (c): the controller creates a contract specifying how the data received from all subjects is treated. Data subjects then join the contract in case they accept the data usage policies of the controller, similarly to the Platform for Privacy Preferences Project (P3P) [15].

Cardinality of contracts and accountability granu-larity. The models differ with respect to the cardinality of contracts, the level of customization, and the data tracking granularity allowed to subjects. Model (a) has the highest number of contracts since there is a custom tailored subject contract for each controller accessing subject's data, which is more adequate when very sensitive data is exchanged (e.g., health information). In model (b) the subject creates one contract for each data type that is possibly accessed by many controllers. In model (c) there is just one contract for each controller, which is expected to be several orders of magnitude lower than the number of subjects, but also has the lower level of customization since subjects are left to choose between a limited number of contract options. Only model (a)
way.

And (b) consider the registration of fine-grained data provenance information including data usage events.

Trust model. A blockchain-based platform is essentially a public decentralized database without a single point of failure that can be trusted for correctness but not for privacy, can be accessed/managed by many people that do not necessarily trust each other and do not share a common trusted third party.

A public blockchain-based solution is needed in this scenario since the following conditions are met: (1) there is a need for a database tracking data provenance and usage, (2) many organizations need write access to this database, (3) the organizations that write to the database may not necessarily trust each other, and (4) the organizations do not have a shared trusted third party managing the database. Controllers and/or processors may intent to be legally bound to the EU law but may not trust an EU institution to manage this database since according to the GDPR they may also be organizations outside the EU. Since the EU regulation applies to subjects that are EU residents but maybe not EU citizens, they might not trust an EU institution for managing this database.

Public, consortium, or private blockchain. With respect to the blockchain architecture, the adoption of a public, semi-public, or private approach has an impact on the level of public auditability and censorship resistance properties of the solution [2].

A public blockchain approach such as the Bitcoin and Ethereum public blockchains allows anyone to read, send transactions, and participate in the consensus/mining process to determine which blocks are added to the chain and determine the current blockchain state. In a semi-public or consortium blockchain read access may be public or restricted and the consensus process is controlled by a set of organizations. In a private blockchain read access may be public or restricted and write permissions are decided by one central organization. The adoption of any of these approaches has an impact on the censorship resistance, privacy, anonymity, performance, and scalability.

Censorship resistance. Contract creation and joining is on the interest of controllers, since it allows them to receive and process data. However, subjects withdrawing their consent is not on their interest, because the data would be no longer available for processing.

Considering this fact, the adopted blockchain solution must ensure censorship resistance. In fact, if a private blockchain solution is adopted, the central authority could simply refuse to add transactions where subjects are withdrawing their consent in order to keep the rights to use the subjects data. Therefore, it is important to make sure transactions are fairly processed considering a minimum quality-of-service requirement, for example, it should be guaranteed that all subject transactions are added to the blockchain at most one day after the transaction submission. Data protection authorities can assume the role of verifying subject complains and censorship resistance in case a private blockchain solution is adopted.

Privacy and anonymity. From an anonymity point of view model (b) is not acceptable since subjects will need to use a unique address known by all controllers allowing for identity linkability between controllers. Models (a) and (c) allow subjects to use different pseudoidentifiers (EVM addresses) and by design prevent direct linkability of subscriptions among different controllers. In model (a) the only thing that can compromise the privacy of subjects is the policy structure since in our implementation we adopt a privacy-friendly way to encode data provenance and data usage events without revealing the details about the actual data instances exchanged with controllers (see Section 1.4).

Performance and Scalability.

From a scalability perspective, the controller-generic approach is the most efficient. However, it also limits the solution to public blockchains due to the high number of transactions required. A service provider like Facebook, which has an average of five new users per second, would generate at least five transactions per second solely for managing new users joining the data usage contract.

Given the transaction rates of current public blockchains—where Bitcoin has a theoretical limit of 7 transactions per second (tx/s) but typically achieves only around 3 tx/s, and Ethereum can handle approximately 15 tx/s—this throughput is clearly insufficient for large-scale service providers like Facebook.The only viable solution is an independent blockchain operated by the company itself, which can be implemented using a private Ethereum Virtual Machine (EVM) chain.

This private chain would periodically checkpoint transactions onto a public blockchain for verification.However, running the system on a private blockchain controlled by the service provider introduces the risk that users' opt-out requests from privacy policies could be ignored. To mitigate this, we propose using cryptographic acknowledgment of transactions by the service provider. This would allow users to prove that they submitted their request to the blockchain, ensuring accountability without altering the integrity of the system.Economics.

In case a public blockchain approach is taken the benefits regarding non censorship and public auditability are definitely compelling. However, in a public blockchain the transaction fees have to be paid for the processing contract invocations, which could make the approach (a) unfeasible due to the high number of transactions. Subjects and controllers would need to agree on an economic model to device who pays for the contract fees.

This is an additional argument to adopt a private blockchain solution where processing fees could be reduced significantly. We do not present in this paper a complete economic analysis of our solutions but from our evaluation results discussed in Sections 4 and 5 and considering the large number of subjects and controllers that process data, it is evident that these models could not be adopted only relying on the public EVM chain.

Off-blockchain communication. All proposed models
consider the need for off-blockchain communication requirements including the transfer of the subject’s data to the controllers. The purpose of the blockchain is to register and provide auditability of the communications when needed. The EVM proposes Swarm and Whisper respectively for off blockchain decentralized data storage/distribution and messaging. The integration of these approaches in our solution is out of the scope of this paper.Guarantees.

The goal of our approach is not to prevent controller and providers from unlawfully holding or redistributing data, but to enable verifiability of data custody and a mechanism for subjects to issue notifications about the misuse of their data. From a controller and processor point of view the main benefit is a certified proof that can be

presented to supervisory authorities showing the data was obtained in a GDPR compliant way.

Secondly, it provides a regulatory framework mechanism to enable GDPR compliance checking by Supervisory Authorities in case a noncompliance activity (i.e., an unlawful data access) is signaled by subjects. For example, when a subject receives an unsolicited e-mail advertisement from an organization containing his/her personal data for which provenance cannot be determined, it could be an indication of a non-compliant activity.

The signaling of non-compliant activities could trigger investigation and auditing of the organization by the Supervisory Authorities and possibly lead to a sanction in case the organization cannot show by informing the respective transactions in the blockchain allowing the data to be received and used for the specific purpose.

Subjects could also by default require all interactions with them to be backed by Links to the respective transactions in the blockchain, even in non-digital communications, for instance, paper letters could contain a machine readable code encoding the transaction number allowing the specific data to be collected. Subjects could then read this code using their mobile phone and check the compliance of the obtained personal data.

5.IMPLEMENTATION

5.1 SUBJECT CONTRACT MODEL:

Figure 5.1.1 presents the high-level architecture of the more fine-grained data accountability and provenance tracking model proposed in this use case . In this architecture, three main entities are depicted following the GDPR terminology:

The Data Subject, the Data Controller, and the Data Processor.

When the subject subscribes with a controller, which is typically the role of a service provider, it creates a policy based Data Usage Contract specifying constraints on the usage and redistribution of any data obtained explicitly or implicitly by the controller.

Explicit data is any data provided directly through interactions with the subject such as the e-mail addresses or birth date. Implicit data is any data acquired automatically, for example, sensor data from IoT devices in the environment surrounding the subject, data acquired by apps installed in mobile devices, or even server log files registering details of the network interactions between subject and controller services (e.g., IP addresses).

The contract in this model acts as a data provenance tracker, policy evaluation entity, and event logger that allow the subject to easily check all data transfers and usage transactions providing assurance that only transactions conforming to the contract policies are authorized and registered in the blockchain.

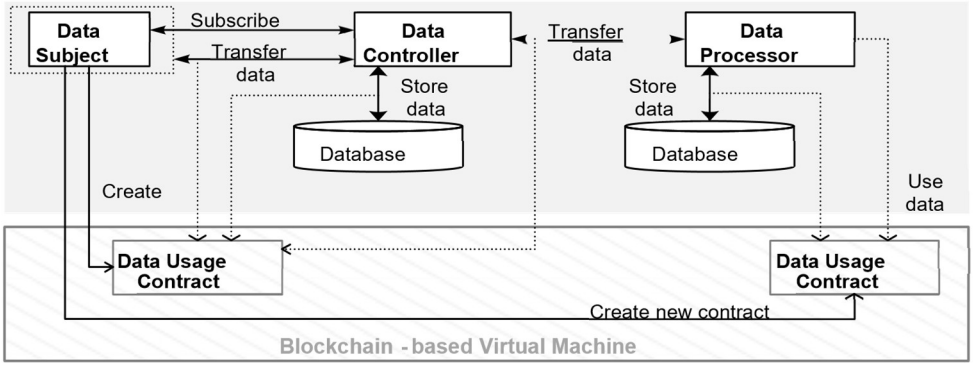

Figure 5.1.1: Architecture

Figure 5.1.2 shows the sequence diagram of the interactions between the entities and contracts. The subject subscribes with the data controller, creates the custom contract for this controller regulating the use of his/her data, and transfers the data to the controller.

For each new established contract the subject uses a new blockchain address to prevent linkability of the contracts established with each controller, requiring the subject to maintain a list of all addresses used and the respective nonce established with each controller or processor. After creating the contract, the subject transfers the data to the controller.

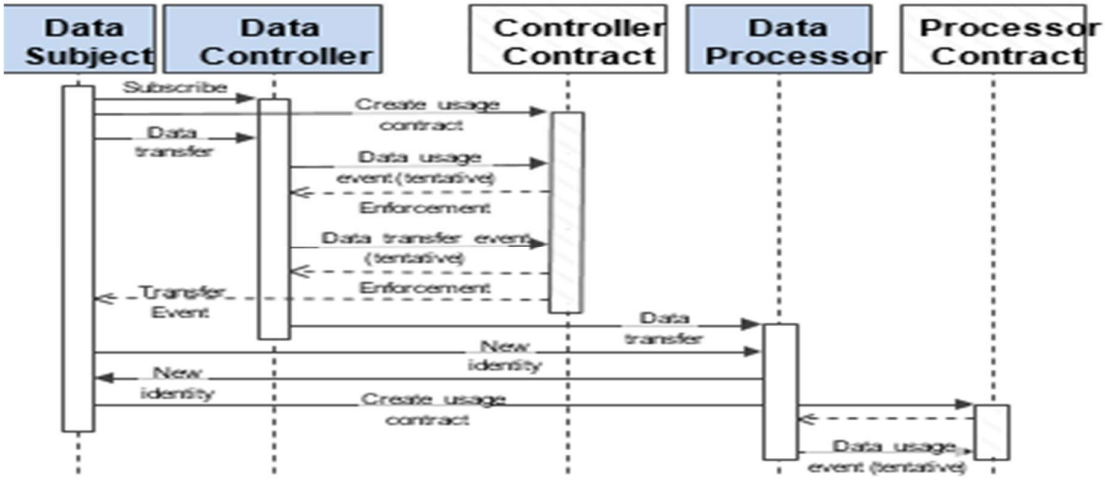

Figure 5.1.2: Sequence diagram

The newly created contract stores a list of the data transferred to the controller including data instantiation and instance values. This information is not store in plain since the contract is deployed in a public blockchain where anyone is able to access the contract state variables and transactions, which would imply public access to all the subject's data. Figure 5.1.3 shows the schema we adopt to obfuscate the provenance tracking information stored in the contract.

For each controller the subject selects the data to be shared, which is encoded according to a data model, and establishes a secret random nonce that is only shared with the data controller in an encrypted format using the contract initialization parameters.

The subject then stores in the contract only the hashes (generated using the SHA3-256 function) of the data instantiations and data instances values (e.g., e-mail="…") concatenated with the nonce, which is only known at this point by the subject and the controller.

The objective is to provide indistinguishability and to prevent crawling of the blockchain searching for the same data value in order to re-identify the subject contracts, for example, if a controller would know also the same data value (e.g., e-mail) it could simply search the blockchain transactions for this same exact hash.

Figure 5.1.3 : Data provenance encoding in contract

Whenever the controller is about to perform a data usage activity it should check using a read-only local transaction to the contract interface if this activity is allowed according to the policies. In case the activity is allowed by the contract, the controller can record it using a blockchain transaction signaling the actual data usage event, which is recorded in the blockchain for a later accountability check.

Data usage activities include access to the subject's data, storage of the data in a local database managed by the controller, execution of any purpose specific activity, generation of derived or consolidated data, and transfer/redistribution of data to processor. In order to protect the privacy of subjects, events and policies are also anonymized using a hash function and the nonce already established with the data controller.

The behavior of the policy is encoded in the contract and uses a privacy-friendly policy evaluation algorithm that also relies on hash functions.

Figure 5.1.4 describes how the policy behavior is encoded in a contract for a sample policy that allows a billing message to be sent to the subject e-mail at most once every 30 days and denies any other usage activities.

The first step is the parsing of the policy structure and identification of the event patterns and operators creating a tree of the policy structure. For each event pattern specified in the policy, the activity name and attributes are obfuscated using the secret random nonce and a hash function, in the same way already done for the data provenance information.

The event matching is then encoded in the notifyEvent function of the contract, which updates the states of the operator tree and verifies if the final evaluation of the policy is true or false. We do not store the events in the contract, we simply update the states in case the event is observed, and for cardinality and temporal operators a time step window with the previously observed events is updated. The encoding of each operator is custom considering the specified policy, for example, the within operator just requires a boolean flag and the last update time for evaluation purposes.

Reactive policies that are triggered based on time instead of an event such as "raise a notification if my data is not deleted in 30 days" require the subject to verify periodically the blockchain state in order to observe violations and generate a time step event in order to trigger the execution.

In case the user does not generate the time step event the violation is detected whenever the contract is notified about any relevant event. In our current implementation we only support equality matching of obfuscated events and attributes, in order to support other comparisons such as ”allow something if value is bigger or lower then a threshold” would require more sophisticated schemes such as homomorphic encryption.

Furthermore, if policies depend on any external information (e.g., context or trust Relationships),this information has to be pushed in the block chain.
The activity of transferring data from a controller to a processor should be also authorized by the contract policies.
In case the transfer is authorized by the contract, the data is transferred and an event is generated containing the address of the processor known by the controller in an encrypted format only readable by the subject. The subject is able then to establish a new identity
with the processor, which is even unaware of the controller contract address established with the subject.
Finally, the subject creates a data usage contract for the processor following the approach used with the controller; the processor contract consists of a new shared secret nonce, the data provenance information, and the encoded policy behavior that should be enforced by the controller.
Subjects may withdraw their consent by deleting the contract from the blockchain, which simply makes the contract inactive from that moment in time on, while the complete contract history stays forever recorded. Furthermore, users may also release additional data or impose more policies/constraints on the data released to the controllers/processors.

Figure 5.1.4: Generation of contract from policy

This can be achieved by adding additional data provenance information to the parent contract and by creating child contracts encoding more desired policy behaviors.
We have implemented a sample contract using the policy example in Figure 4 and evaluated the performance for transactions creating and executing contract functions in a private blockchain implemented using the EVM.

In our implementation we evaluated two choices regarding the storage of data usage events in the contract state variables. In the first version 10 hashes of data types and instances were stored as state variables in the contract with a total of 736 bytes (2 times byte32 for each data), while in the second version the data hashes were signaled as events associated to the contract (each event signaling 2 times byte32).

Just creating the contract without any data associated consumed 0.82 million gas, with the events the consumption increased to 0.89 million gas, and with the state variables the consumption was as high as 1.25 million gas. These results show that it is far better to use events in case of simply logging activities instead of using contract state variables. We also evaluated the cost of notifying a data usage event to the contract, which generates a contract
event with the respective enforcement result (i.e. allow, deny, modify or delay). The event notification cost was only of 0.023 million gas.

Since contracts have a limited amount of storage and processing capabilities it is unfeasible to allow flexible configuration of policies in a single contract with a more complex interfaces. Contracts should embed the policy logic and in case a new policy or data transfer takes place a new child contract should be created. Data controllers are allowed to add new child contracts to the existing one where two approaches can be followed depending on the number of users and constraints.
All users that joined a parent contract are automatically added to the child contract unless they add a restriction. All users that joined a parent contract must explicitly re-join the child contract to be included in the new service conditions. Nevertheless users are able to watch the contracts and be notified about these events in order to be able to react. Data controllers may choose to link new child contracts with additional features that users will be allowed to receive from the service by associating data requirements to the provided features.

With respect to user's privacy, this model guarantees user pseudoymity and unlinkability among different controllers and processors and unobservability by third party entities of the user policies and data events in the public blockchain that are necessary to evaluate the respective policies. Linkability can be achieved in case processors and controllers collude by exchanging their addresses and nonces used in each contract if users provide unique identifiers (e.g. e-mail) allowing entities that received this identifier to know the associated addresses of a user.

Our solution also considers that, in particular cases, upon request of a third party such as a the Data Protection Supervisor Authority, both subject or controller/processor may be requested to prove the possession of the data and to prove the data usage activities performed did not violate the subject's policies. In this case, the release of the encrypted

nonce by subject and/or processors/controllers is enough to enable verification of the policies and the data provenance information.

5.2 CONTROLLER CONTRACT MODEL :

In this section we describe the more coarse-grained data accountability and provenance tracking model proposed in this paper. In this model the controller or processor creates a contract specifying the shared constraints on the usage and redistribution of any explicit or implicit data obtained from all subjects that subscribe with the controller.

The contract in this model acts simply as a repository of the configurable policy templates that will be instantiated for all subjects and do not store anything except the list of subjects. Transactions are only requests by subjects to join or leave the contract.

Figure 5.2.1 shows an example of policy enforcement and configuration templates that can be used in a data usage contract in this model. The policy enforcement template specifies an enforcement pattern using an entity variables, which in this example is one variable for the data subject that allows a billing message to be sent no more than once per month.

This policy is configured for all users that subscribe to the service in the defined configuration template, and a special data assignment function is used to determine in the policy evaluation if the e-mail is associated to the data subject encoded in the variable.

```
Default Enforcement: Deny
PolicyTemplate0
 Variables: Entity(s)
      Event: sendMessage(purpose= billing ,
isDataSubject(e-mail, s))
 Condition: not(within(30 days, sendMessage(purpose= billing ,
isDataSubject(e-mail, s))))
    Action: Allow
ConfigurationTemplate0
   Variables: Entity(s)
 Assignments: s = xpath(\\trigger\user)
       Event: subscribeUser()
   Condition: true
      Action: configure(PolicyTemplate0, s)
```

Figure 5.2.1: Enforcement and configuration templates

Since this model is more focused on controllers processing large amounts of data we also allow the data controller to log events representing bulk executions of the contract, for example, using an event template send message where the e-mail data attribute is left unspecified. Bulk events make sense when regular activities are completed nearly at the same specific moment in time such as on a daily time window.

Bulk events may also be parametrized, for example, in case an activity happens for a subset of subscribers matching a particular condition using also a policy template. Bulk attributes may be specified using the same ECA template of policies using a tentative event structure,

for example, all data of data subjects from a particular country will be shared. This example policy is specified in 5.2.2.

```
PolicyTemplate1
 Variables: Entity(s)
     Event: shareData(purpose= marketing , country= Italy ,
isDataSubject(country, s))
 Condition: true
    Action: Allow
```

Figure 5.2.2 : Data sharing policy for subjects of a particular country

We have implemented a sample contract where the policy is configured as a string, which could be an XML or JSON encoding of the policy specified using the SecKit policy language. As expected, this contract requires a much lower amount of gas in contrast to the model detailed in the previous section.

The creation of the contract requires only 0.34 million gas and the joining and leaving of data subjects only requires around 40 thousand gas. The contract does not require any other operation since it does not log any data usage events or enforcement actions.

5.3 RELATED WORK

The nature of blockchain is particularly suitable for tracking assets exchanged by different entities. In their most famous implementations, actually blockchains do not exchange any asset (i.e. coin or token), but the ownership of a certain amount of them is just the result of
evaluating the transaction list containing the quantity of assets exchanged between two entities.
Therefore, when applied to digital or physical assets, it is fundamental that the latter can be unequivocally represented and identified, and that they cannot be (trivially) cloned.

One of the first application of blockchains to the tracking of physical goods was for the provenance of diamonds [12]. Here the strong point is that diamonds can be uniquely fingerprinted through their physical properties, making them distinguishable one from another. This fingerprinting allows the diamond discovery processing and registration in the blockchain to easily match the concept of mining in a standard proof-of-work based blockchain system.

When the assets to be tracked are digital, the first problem is related to the possibility of easily cloning and transferring them, so that the main goal is at least to prove their first origin. The tracking of digital data provenance using blockchains was first proposed in a discussion paper presented at the ID2020 Design Workshop [11].

Although at high level, the paper describes a proof of concept for verification and attestation of provenance of individuals' data using blockchains.

The latter do not directly store the data, which are referenced with hashes and associated with tokens, so that it is possible to prove their (immutable) origin and track all the data exchanges. Although the proposed approach is novel, it does not cover the definition of advanced policies or contracts regulating the usage of the exchanged data in the way proposed by us in this paper.

An example of users' data protection and privacy policy enforcement on a blockchain is provided in [18]. The authors propose to control access permissions to private data collected by a service (e.g., location from a mobile phone) through a Bitcoin blockchain.

Every time a users subscribes to a service a new transaction specifies the access permissions and another contains the hash of the data, which are stored in a off-chain database. Policies encoded in a protocol executed by the blockchain grant or deny access to the data referenced in the chain. Although this solution is in part similar to ours, their proposed policies only specify simple allow/deny enforcement (i.e. white/blacklisting) without the possibility to express more complex policy conditions.

Moreover, scalability is not taken into consideration: issuing minimum two transactions for every subscription of a user application to an online service would easily saturate a Bitcoin network even considering few service providers.Inspired by the example above, is the application of data tracking to health care proposed in [6]. Here the blockchain is again a medium to control the access to data stored off chain.

The paper provides only a high level description, but already identifies the main limitations, mainly in terms of scalability and data privacy, of Bitcoin-based blockchains for this kind of applications.Applied to the same field but with a more comprehensive approach is MedRec [1]. The authors propose to give patients control over their Electronic Health Records (EHRs) through the use of Ethereum blockchain and smart contracts, which manage authentication, confidentiality, accountability and sharing of the data.

The latter are referenced in the chain using their hashes but are stored externally. Miners are rewarded with anonymized aggregated data useful for research. Although the proposed approach addresses many issues, considerations about costs and scalability on a large scale deployment are not mentioned.

6.BENEFITS

Blockchain technology enhances trust, security, and transparency in digital provenance. Here are the key benefits:

6.1. Immutability and Tamper-Proof Records

- **Why It Matters:** Once data is recorded on the blockchain, it cannot be altered or deleted.
- **Benefit:** Prevents fraud, counterfeiting, and unauthorized modifications of provenance records.

6.2. Enhanced Transparency and Trust

- **Why It Matters:** Every transaction is stored on a public or permissioned ledger, accessible for verification.
- **Benefit:** Increases accountability in supply chains, ensuring ethical sourcing and regulatory compliance.

6.3. Decentralization and Security

- **Why It Matters:** No single entity controls the data, reducing the risk of manipulation.
- **Benefit:** Eliminates reliance on intermediaries, reducing corruption and data breaches.

6.4. Efficient and Automated Verification

- **Why It Matters:** Blockchain enables instant verification of ownership, authenticity, and history.
- **Benefit:** Reduces paperwork, speeds up transactions, and prevents fake documentation.

6.5. Cost Reduction

- **Why It Matters:** Traditional provenance systems rely on third-party verification services, increasing costs.

- **Benefit:** Blockchain eliminates middlemen, reducing expenses in industries like luxury goods, real estate, and digital content.

6.6. Smart Contracts for Automated Compliance

- **Why It Matters:** Smart contracts can enforce rules automatically (e.g., royalty payments, supply chain compliance).

- **Benefit:** Reduces human error, speeds up processes, and ensures compensation.

6.7. Improved Data Privacy and Ownership

- **Why It Matters:** Blockchain allows users to control their data using cryptographic keys.

- **Benefit:** Protects sensitive information, such as personal identity and medical records, from unauthorized access.

6.8. Global Accessibility and Interoperability

- **Why It Matters:** Blockchain-based provenance systems work across borders without intermediaries.

- **Benefit:** Enables cross-border trade, digital identity verification, and global tracking of assets.

6.9. Fraud Prevention and Anti-Counterfeiting

- **Why It Matters:** Counterfeit products and fraudulent records cause billions in losses.

- **Benefit:** Unique digital signatures and NFT-based ownership prevent forgery in art, luxury goods, and supply chains.

6.10. Regulatory Compliance and Auditing

- **Why It Matters:** Companies need to prove compliance with industry regulations.

- **Benefit:** Blockchain provides auditable, time-stamped records that regulators can verify instantly.

7.CHALLENGES

While digital provenance is essential for tracking authenticity, ownership, and history of assets, traditional systems face several challenges. Blockchain provides solutions to overcome these limitations effectively.

7.1. Data Tampering & Fraud Challenge:

- Traditional databases can be altered or manipulated, leading to fake ownership claims, counterfeit goods, and fraudulent transactions.
- In industries like luxury goods, supply chains, and real estate, forged documents cause major financial losses.

Blockchain Solution:

1. Immutability – Once recorded, data on a blockchain cannot be changed or deleted, ensuring tamper-proof provenance.
2. Cryptographic Hashing – Each transaction is secured with a unique hash, making unauthorized modifications detectable.

7.2. Centralized Control & Single Point of Failure Challenge:

- Centralized provenance systems depend on trusted authorities, making them vulnerable to hacking, corruption, and loss of records.
- Example: A government or company managing land registries could alter or erase property ownership records.

Blockchain Solution:

1.Decentralized Network – Data is stored across multiple nodes, eliminating single points of failure.

2.Resilience – Even if some nodes fail, the blockchain remains functional and secure.

7.3. High Costs & Inefficiencies in Verification Challenge:

- Manual audits and verification in provenance tracking require third-party intermediaries, increasing costs and slowing down processes.
- Example: Authenticating luxury handbags or verifying supply chain compliance requires expensive and time-consuming third-party inspections.

Blockchain Solution:

1. Smart Contracts – Automate verification processes, reducing the need for middlemen.
2. Instant Audits – Businesses and regulators can verify transactions immediately on the blockchain.

7.4. Privacy Concerns & Data Ownership Challenge:

- Users often have limited control over their personal data in centralized provenance systems.
- Example: A musician might lose control over how their digital content is distributed and monetized.

Blockchain Solution:

1. Self-Sovereign Identity – Users own their data and grant access only when necessary.
2. Encrypted Transactions – Sensitive information is stored securely while still being verifiable.

7.5. Scalability Issues & Transaction Speed Challenge:

- Public blockchains like Bitcoin and Ethereum have limited transaction speeds (e.g., Bitcoin ≈7 transactions per second).
- Large-scale provenance applications (e.g., tracking millions of supply chain items) require high throughput.

Blockchain Solution:

1. Layer 2 Scaling Solutions – Technologies like sidechains, rollups, and sharding improve scalability.
2. Private & Hybrid Blockchains – Companies can use private blockchains for speed while periodically anchoring data to public chains for security.

7.6. Regulatory & Compliance Challenges Challenge:

- Governments and industries have varying regulations on data ownership and digital transactions.
- Example: Some regions may not fully recognize blockchain-based land registries as legal proof of ownership.

Blockchain Solution:

1. Auditable Compliance Records – Blockchain automatically stores time-stamped, verifiable logs for regulators.
2. Hybrid Blockchain Models – Combining private blockchains for internal processes with public blockchains for regulatory transparency.

8.CONCLUSION & FUTURE WORK

In this use case we analyze the feasibility of using a blockchain based contract approach to support data accountability and provenance tracking in light of the new requirements of the GDPR.

We discuss several solution design choices and introduce three models, including two concrete implementations, with an extensive analysis with respect to data accountability and provenance tracking granularity, privacy, anonymity, performance, and scalability. We show that for more sensitive data with less frequent exchanges, such as medical data, a more fine-grained solution where subjects create contracts with each controller and processors is more adequate.

On the other hand, for more dynamic data with more frequent exchanges and strict scalability and performance requirements, controllers or processors should manage a contract that registers all subjects accepting all or part of the data usage conditions. A possible solution for scalability issues we are currently investigating is the use of sharding, where the blockchain is divided in separate chains that are responsible for contracts of a subset of all controllers and processors.

These separate private chains then synchronize with the public chain on regular intervals, for example every 1000 blocks, in order to allow for public verifiability [16].

In case the separated chains are managed privately, data protection supervisory authorities can then join all chains just as observers in order to prevent censorship and guarantee that transactions of data subjects are not indiscriminately refused.

As future work we also plan to investigate the possibility of using business blockchain approaches such as the Hyperledger solution, which allows for a custom consensus algorithm and also has a more ambitious scalability and performance goal with thousands of transactions per second [13, 14].

For the use case of immutable digital provenance, a private or permissioned blockchain is often the most suitable choice due to the need for control, privacy, and security. However, if maximum transparency is required, a public blockchain may be appropriate.

The specific parameters of the blockchain—such as immutability, transparency, scalability, security, consensus mechanism, interoperability, and the use of smart contracts—should be carefully considered to ensure that the solution meets the unique needs of the organization and its stakeholders.

We also plan to work on a model-based translation mechanism to automatically generate contracts from policies specified in our policy language.

9.SDG's ADDRESSED

Blockchain technology enhances digital provenance by ensuring transparency, security, and accountability, contributing to multiple United Nations Sustainable Development Goals (SDGs). Here's how:

9.1. SDG 12 – Responsible Consumption and Production

Why It's Addressed:

- Blockchain enables traceability in supply chains, ensuring ethical sourcing and reducing waste.
- Consumers can verify whether products (e.g., fashion, food, electronics) are sustainably and ethically produced.

Example: Companies use blockchain to track carbon footprint, fair trade compliance, and sustainable sourcing of raw materials.

9.2. SDG 9 – Industry, Innovation, and Infrastructure

Why It's Addressed:

- Blockchain improves industrial transparency, streamlining logistics and reducing fraud in provenance tracking.
- Strengthens technological infrastructure by providing tamper-proof, decentralized records for industries like manufacturing, logistics, and finance.

Example: Blockchain-powered digital twins help track real-time conditions of physical goods in global trade.

10.REFERENCES

[1] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman. 2016. MedRec: Using Blockchain for Medical Data Access and Permission Management.
In 2016 2nd International Conference on Open and Big Data (OBD). 25–30.

[2] V. Buterin. 2015. On Public and Private Blockchains. https://blog. ethereum.org/2015/08/07/on-public-and-private-blockchains/. (2015). Online; accessed March 15th 2025.

[3] CoinMarketCap. 2017. CryptoCurrency Market Capitalizations. https: //coinmarketcap.com/. (2017). Online; accessed March 14th 2025.

[4] European Parliament and the Council of the European Union. 1995.
Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data.

Official Journal of the European Union (1995).

[5] 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data. Official Journal of the European Union (4 May 2016).

[6] L. A. Linn and M. B. Koo. 2016. Use of Blockchain in Health IT and Health-related Research. https://www.healthit.gov/sites/default/files/ 11-74-ablockchainforhealthcare.pdf. (2016). Online; accessed March 15th 2025.

[7] S. Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf. (2008). Online; accessed March 12th 2025.

[8] R. Neisse, G. Steri, and G. Baldini. 2014. Enforcement of security policy rules for the Internet of Things. 2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (Wimob) 00, undefined (2014), 165–172.

[9] R. Neisse, G. Steri, I. Nai Fovino, and G. Baldini. 2015. SecKit: A Model-based Security Toolkit for the Internet of Things.computers and Security 54 (2015), 60 – 76. Secure Information Reuse and Integration & Availability, Reliability and Security 2014.

[10] R. Neisse, G. Steri, D. Geneiatakis, and I. Nai Fovino. 2016. A privacy enforcing framework for Android applications.Computers & Security 62 (2016), 257 – 277.

[11] G. Samman and K. Dow. 2016. Immutable Me A Discussion Paper Exploring Data Provenance to Enable New Value Chains. https://github.com/WebOfTrustInfo/ID2020DesignWorkshop/ blob/master/topicsand-advance-readings/immutable-me.pdf. (2016). Online; accessed March 12th 2025.

[12] G. Volpicelli. 2017. How the blockchain is helping stop the spread of conflict diamonds. http://www.wired.co.uk/article/ blockchain-conflict-diamondseverledger. (2017). Online; accessed March 15th 2025.

[13] M. Vukolic. 2016.´ The Quest for Scalable Blockchain Fabric: Proof-ofWork vs. BFT Replication. Springer International Publishing, Cham, 112–125.

[14] M. Vukolic.´ 2017. Rethinking Permissioned Blockchains. In ACM Workshop on Blockchain, Cryptocurrencies and Contracts (BCC'17). Available at: http://vukolic.com/ rethinking-permissioned-blockchainsBCC2017.pdf.

[15] W3C. 2007. Platform for Privacy Preferences (P3P) Project. https: //www.w3.org/P3P/. (2007). Online; accessed March 17th 2025.

[16] Ethereum Wiki. 2017. Sharding FAQ - On sharding blockchains. https: //github.com/ethereum/wiki/wiki/Sharding-FAQ. (2025).

[17] G. Wood. 2014. Ethereum: A Secure Decentralised Generalised Transaction Ledger - EIP-150 Revision. http://http://gavwood.com/paper. pdf. (2014). Online; accessed March 16th 2025.

[18] G. Zyskind, O. Nathan, and A. '. Pentland. 2015. Decentralizing Privacy: Using Blockchain to Protect Personal Data. In 2015 IEEE Security And Privacy Workshops. 180–184.

11. APPENDIX A

Drive Link:

https://drive.google.com/drive/folders/1sO9vl2ggrlX4L_lXGBaujC8_yF60px1v?usp=sharing

QR Code: