Received: 16 January 2020 | Revised: 30 March 2020 | Accepted: 2 May 2020

DOI: 10.1002/cpe.5852

WILEY

SPECIAL ISSUE PAPER

Privacy-aware cloud ecosystems: Architecture and performance

Masoud Barati | **Omer Rana**

School of Computer Science and Informatics, Cardiff University, Cardiff, UK

Correspondence
Masoud Barati, School of Computer Science and Informatics, Cardiff University, Cardiff, UK.
Email: Baratim@cardiff.ac.uk

Funding information
The work reported in this paper has been funded by EPSRC under, Grant/Award Number: EP/R033439/1

Summary

With an increasing number of cloud providers offering services made use of by both individual users and other providers, there is a realization that service provision now involves an "ecosystem" of providers. Some providers may be directly visible to a user, while others may be contributors to composite services and not directly known to the user—as only the provider offering the composite service is visible. Such services may include: domain specific services (eg, simulation), advertising services, or profiling/analytics services. Understanding the impact on data privacy of a user for such a composite service remains a challenge, and providing transparency (and obtaining user consent for data use) remains a key requirement of the European General Data Protection Regulation (GDPR). An architecture that makes use of blockchains and smart contracts is proposed that addresses this requirement. An implementation of the architecture is used to demonstrate how access control can be managed and audited. The scalability and cost of undertaking access control, as the number of actors (both service providers and "voters") increases, is also described. The proposed approach can be used to support service aggregation across both private and public clouds.

KEYWORDS

blockchain, cloud architecture, data privacy, general data protection regulation, smart contracts

1 | INTRODUCTION

With increasing number of on-line services, often hosted over cloud infrastructure, there is a realization that such services can involve an interlinked set of cloud providers. To access a service, users primarily interact through a Web interface, and are (often) unaware of the larger collection of services that are made available behind the Web interface, and deployed across a distributed infrastructure. Users entrust their data without realizing that the providers may share their data with back-end services such as cloud hosted analytics and advertisers—the growth in Internet-connected devices adds further complexity to this challenge. In order to address this, the general data protection regulation (GDPR) is implemented to impose obligations on providers to ensure that consent is obtained from users before their data are made use of, thereby enabling nonexpert users to make informed decisions about their privacy.[1]

The key elements introduced in GDPR are a data subject, a controller or joint controller, and a processor.[2] The data subject is the data owner and the controller a person/organization specifying aims of processing a user's personal data. The notion of joint controller is introduced where two or more controllers jointly specify the purpose of data processing. Finally, the processor is responsible for processing personal data on behalf

of a controller or joint controller.[3] By defining these elements, GDPR gives the responsibility of any violation in data processing to the controller or joint controller, but also gives a shared responsibility to the processor when the user has no direct control on the data (or the analysis carried out on the data). The integration of GDPR into a cloud ecosystem helps make more explicit the responsibility and accountability requirements of both the processor(s) and controller(s). Under such requirements, any operation of a cloud provider on personal data must be in accordance with user consent.[1]

Given these GDPR requirements, several solutions have been proposed to support the accountability and provenance tracking of user data when it is delivered to a controller or submitted to a processor.[4] Some solutions utilize blockchain-based technologies to improve transparency and trust between users and actors.[4,5] In addition to these solutions, the blockchain as a shared ledger has been integrated into applications that cover privacy, authentication, provenance, and data integrity.[5] Blockchain technology has brought a provably secure, fully distributed, and consensus-driven solution to these applications. A review of recent blockchain-based techniques to enhance privacy and trust in cloud environment[6,7] shows that the impact of such techniques on cloud-based service deployment has not yet been studied.

This article proposes a blockchain-based approach to improve provenance tracking of cloud user data under GDPR requirements. The main contributions of this work can be summarized as follows: (i) a service-oriented architecture that makes use of a blockchain network, to enable an audit trail of service providers to be generated. The architecture supports trustable containers that securely log all provider operations on personal data; (ii) a blockchain-based logging mechanism to identify providers who violate GDPR rules; (iii) a case study to show how GDPR rules can be deployed as smart contracts in the blockchain—supporting access, transfer, and profiling operations on user data; (iv) performance evaluation showing the effect of increasing the number of parties who verify provider operations, and the amount of *gas* (a metric used to measure the computational complexity of carrying out the analysis) used for deploying smart contracts over a blockchain test network.

The rest of this article is structured as follows. Section 2 provides background material about blockchain and smart contracts. Section 3 describes the proposed architecture for supporting user privacy in a cloud computing environment, and Section 4 focuses on interactions between the software components in the architecture. Section 5 includes a case study to illustrate how verification can be carried out using a blockchain, with reference to GDPR rules. Section 6 provides experimental results describing the computational complexity associated with carrying out the verification process. Related research work is reviewed in Section 7 and conclusions are provided in Section 8.

2 | BLOCKCHAIN BACKGROUND

A blockchain is a public ledger comprising of a distributed, shared database (storing records in blocks) and a set of connected nodes. The blocks are structured as a chain, with each block containing a hash of its previous block. Each block also contains a time stamp and a nonce. The former shows the creation time of the block and the latter is an arbitrary number used just once in a cryptographic communication.[8] The nodes in a blockchain have a peer-to-peer relationship and can build a new block of valid transactions via a mining process. The nodes creating the blocks are called miners. Mining is a main concept of the blockchain through which a block is made and attached to a blockchain network. To this end, several techniques are currently available, namely, Proof of Work (PoW), Proof of Stake (PoS), Proof of Importance (PoI), Proof of Space (PoSpace), and Practical Byzantine Fault Tolerance (PBFT).[3]

A blockchain network can be public, federated, or private.[9] In a public blockchain, everyone can participate and access blocks without any permission (eg, Ethereum[10]). In the federated blockchain, the network is operated under the leadership of several organizations or groups, which limit the type of user who can take part in the verification of transactions (eg, Corda[11] and R3[12]). Finally in the private blockchain, only one organization has permission to create/verify blocks (eg, Monax[13] and Multichain[14]).

A smart contract is executable code that runs on the blockchain and translates a usual contract between two or more individuals into a program. A smart contract provides mediation between two parties so that it enforces them to follow the contract. Each contract can involve a set of transactions that may alter the state of the blockchain, for example, Ethereum.[10] Ethereum uses payments (referred to as "gas") for deploying a smart contract or executing transactions. Gas refers to a unit measuring the computational effort required to carry out operations in a smart contract (associated with operation or opcodes in the contract).[8,15] The gas is paid in Ether, being the cryptocurrency in Ethereum ecosystem and allows smart contracts to be executed.

3 | PRIVACY-AWARE CLOUD ECOSYSTEMS ARCHITECTURE

Figure 1 shows the architecture, which has two general work flows: (i) service delivery and (ii) improving user privacy. The former offers a set of services or composite services realizing user requirements. The latter proposes a blockchain-based technique for user data provenance tracking. The user interface enables submission of personal data to cloud-hosted services. Using the interface, a user can also submit preferences for verifying GDPR rules on the operations executed by providers.

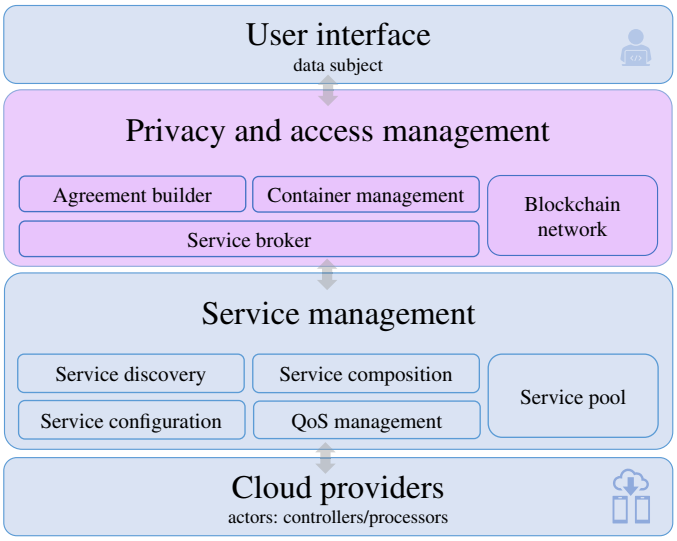


FIGURE 1 A new architecture for privacy-aware cloud ecosystems

3.1 | Privacy and access management

This layer includes four components: service broker, agreement builder, container management, and blockchain network—enabling an audit trail of provider operations to be stored in a blockchain network. Operations can be access, store, profiling, or transfer of user data. Any GDPR violations are also flagged to the user through this layer.

Service broker— identifies services that match user requirements—providing the name, location, and address of the service provider to the agreement builder component.

Agreement builder—acts as a broker to create a shared agreement between a user and provider(s). Given the operations to be executed on user data, this component builds a smart contract to record information required for verifying operations under GDPR rules. The smart contract address is sent to providers to be deployed on their containers.

Container management—launches and manages a container on the provider to get data from the provider and submit this to a blockchain network. It deploys the smart contract supplied by the agreement builder for recording such data. The data may involve user and provider addresses, the operations processed on user data (eg, access, transfer, profiling), and information for verifying operations under GDPR rules (eg, user age). Our assumption is that containers are trusted and they record every operations executed on the user data.

3.2 | Service management

This layer is responsible for discovering, building, and publishing cloud services (some of which may be an aggregation of multiple services). The QoS management component maintains details about cost, availability, and uptime associated with each service.

4 | REALIZATION OF THE ARCHITECTURE

Interaction between components is realized using four phases. Table 1 describes the symbols used.

Phase 1: service discovery and composition—this phase identifies requested services or the development of composite services. A service broker identifies providers involved in the offered services.

Phase 2: building a shared agreement—a protocol for the creation of a shared agreement (based on GDPR requirements) is illustrated in the sequence diagram of Figure 2. This phase is activated by the agreement builder using service details provided by the service broker component. The agreement builder confirms the identity of services requiring user data and sends a request to data controllers/processors (actors) about operations to be executed by them on user data. The agreement builder then waits for consent of the data subject. Given operations and associated GDPR rules, the agreement builder then builds a smart contract—referred to as *container_submission*. The smart contract consists of a template for storing data in the blockchain. This component also determines a set of voters for verifying operations.[1]The voters are third parties connected to the blockchain network and can give votes when executed operations do not comply with GDPR rules.

[1]The addresses of voters are accessible for the agreement builder.

Notation	Description
$\mathcal{A}$	A set of operations
α_i	An operation executed by actor i
$Deg(\alpha_i)$	Degree of compliance for α_i
V_i	A voter
v_i	A vote by V_i
θ_i	A threshold defined by V_i
G_α	A set of GDPR rules related to operation α
r_j	The successful detection rate for the violation j

TABLE 1 Summary of notation

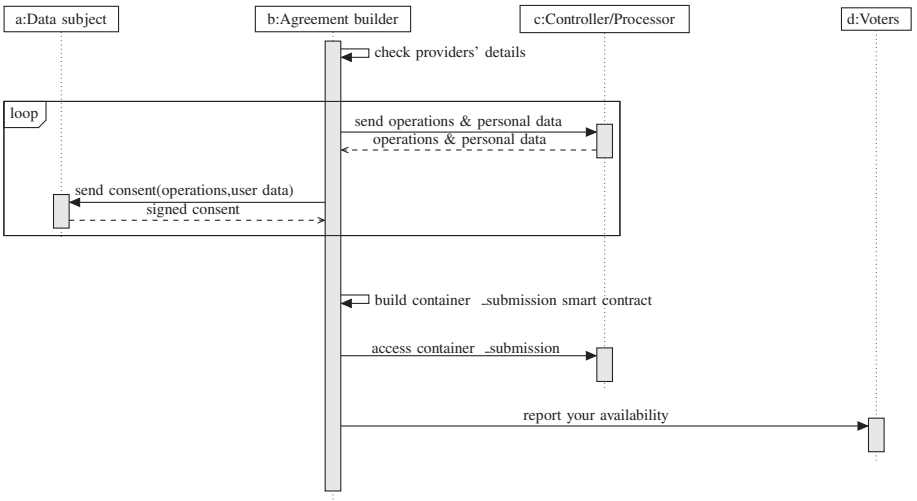


FIGURE 2 A sequence diagram for building a shared agreement

Phase 3: logging data processing—operations on user data, recorded using the trusted container, are stored in a blockchain (based on the container_submission smart contract). From Figure 3 the agreement builder requests personal data from the data subject, which is then forwarded to the actors for processing. A container_submission smart contract is used subsequently by a container to log the data, and which facilitates the verification process. On termination of data processing, a message indicating the finalization of the process is submitted to the agreement builder.

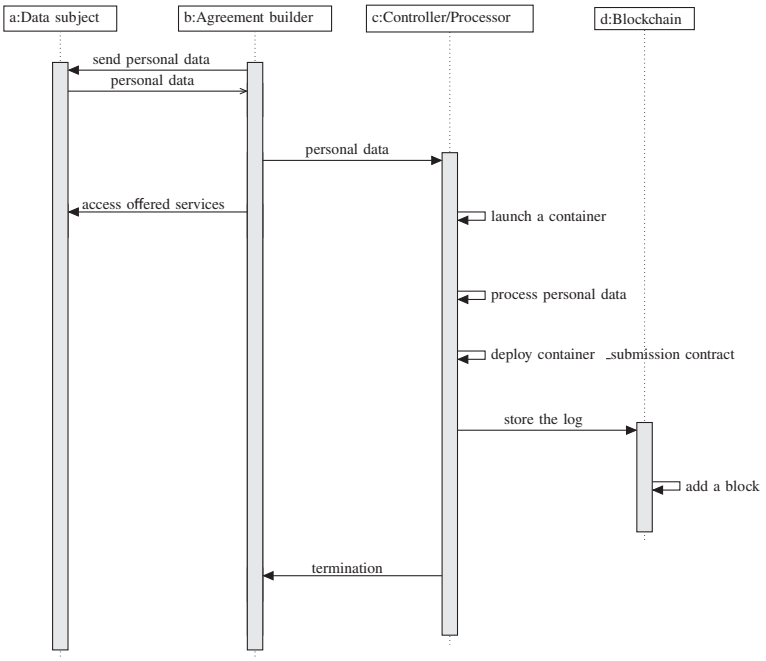
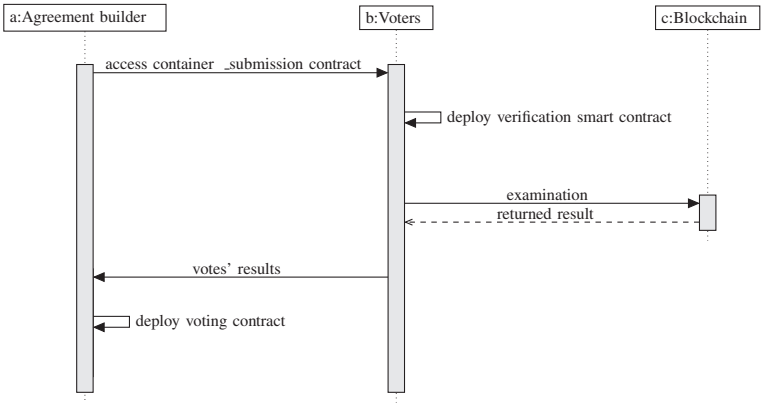


FIGURE 3 Data recording in blockchain

FIGURE 4 Verification using blockchain



Phase 4: verification of the operations—this phase verifies (using voters) violation of GDPR rules—as shown in Figure 4. Voters deploy a *verification* smart contract to check each operation for GDPR compliance (see Section 5). Once the agreement builder component receives a termination message, it sends the voters the address of container_submission contract to ensure that the executed operations conform to GDPR rules. The voters deploy the verification smart contract to access the data already stored by the container in the blockchain.

During agreement development between data subject and actors, the former should be notified in advance about operations that will be executed by actors on their personal data. Not all data subjects may be concerned about GDPR compliance—as verifying an operation is a costly process and a part of such cost should be paid by the data subject. We therefore introduce *degree of compliance* to enable a data subject to associate a value between [0, 1] to each operation that will be executed on their personal data:

Definition 1. Let $\mathcal{A}$ be a set of operations executed by actor(s) on personal data. A function $Deg : \mathcal{A} \rightarrow [0, 1]$ is defined to map the degree of compliance for verifying the operations into a real number between 0 and 1. For an operation $\alpha_i \in \mathcal{A}$ executed by actor i, the outputs $Deg(\alpha_i) = 1, 0 < Deg(\alpha_i) < 1$, and $Deg(\alpha_i) = 0$ show full-compliance, partial-compliance, and noncompliance, respectively.

Each voter can also define a threshold for the verification of each operation, that is, if the degree of compliance of data subject for an operation is greater than or equal to a voter's threshold, the operation is verified. The choice of a threshold is subjective and shows the interest of a voter for verifying GDPR compliance. Defining such a threshold is independent of the degree of compliance determined by data subjects. Setting it too high may limit the number of voters who engage. Setting it too low may not lead to an unuseful outcome. The degree of compliance can be considered as an input of the container_submission smart contract. The voting results can be reported according to the degree expressed by the data subject and the thresholds determined by voters. An actor is classified as a violator based on the following definition.

Definition 2. Let $\mathcal{V} = \{V_1, \ldots, V_l\}$ be a set of voters and v_j be a vote by V_j after verifying operation α_i by actor i such that

$$v_j = \begin{cases} 1, & \text{if } Deg(\alpha_i) \geq \theta_j \text{ and } \alpha_i \text{ violates } G_\alpha \\ 0, & \text{otherwise} \end{cases},$$

where G_α is a set of GDPR rules related to α_i, and θ_j is a threshold defined by V_j for verifying α_i. Moreover, let $m \leq l$ be the minimum number of acceptable votes for reporting a violation. The actor i is classified to be a violator if $\Sigma_{j=1}^{l} v_j \geq m$.

5 | CASE STUDY

Consider a composite service combining: *Order creation*, *Payment*, and *Shipping*[16] services. A customer orders a product, ships it to a destination address, and organizes the payment process using an online portal (see Figure 5). A provider should access a customer's personal data to carry out the following data processing (operations): (i) *Order creation service provider:*should get the customer data: name, identification number, biometric information, age, and contact details; (ii) *Payment service provider*: needs to access customer name, identification number, and bank account details to handle the payment; (iii) *Shipping service provider*: requires the name and contact details of a customer. The provider remotely interacts with a subcontractor (a *Mail* service provider) to manage the product delivery. Given the roles defined in GDPR, both *Order creation* and *Payment* service providers are processors and directly access customer data. The *Shipping* service provider, however, can have both processor and controller roles. It acts as a processor when managing a part of data delivery and as a controller when transferring the data to the subcontractor. Finally, *Mail* service provider is assumed to be a data processor.

It is assumed that a shared GDPR-based agreement has been reached between the customer and actors (controllers/processors). The agreement is based on three GDPR rules: access, transfer, and profiling of customer data—where each actor must guarantee that (i) if customer data are

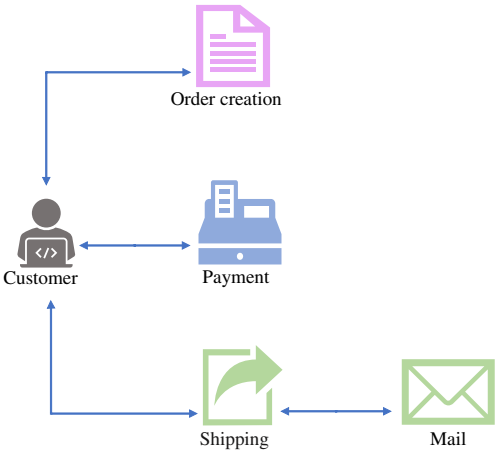


FIGURE 5 Composite service: e-commerce

sensitive, they provide an authentication control mechanism for preventing unauthorized access to customer data (Art. 32(1)(a) of GDPR); (ii) if customer data are transferred to a processor located outside of Europe, the receiver must belong to a country following BCR rules (Art. 44-47 of GDPR); (iii) automated profiling operations are not performed on data of customers whose ages are below 18 years (Art. 22 of GDPR). The first rule refers to a GDPR obligation for accessing customer data. In the case of sensitive data, services supplied by actors and delivered to customers need to support encryption. The second rule refers to a GDPR obligation for transferring customer data in which the data receiver must be hosted in a European member state or belong to a country certified by BCR clauses—essentially a code of conduct adopted by a community of multinational companies that want to move customer data internationally across various jurisdictions.[17] The third rule forbids automated profiling operations on customers whose ages are below 18 years.

Figure 6 provides an overview of smart contracts that must be deployed in accordance with the techniques proposed in phases 3 and 4. The parties deploying these smart contracts are actors involved in delivering the composite e-commerce service, voters, and agreement builder. These parties should be connected to the Ethereum network via a specific client interface.[6] These smart contracts can be shared among them—as shown in Figure 6.

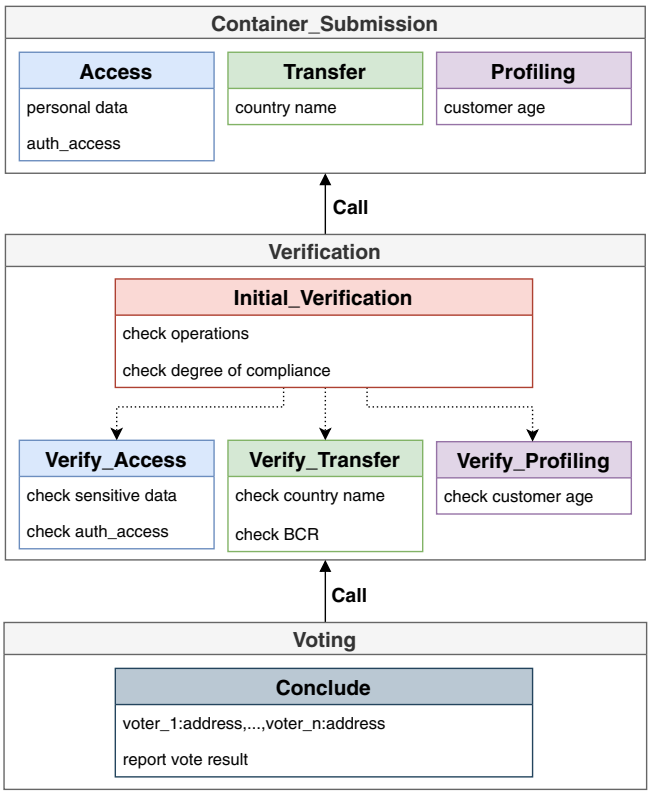


FIGURE 6 Smart contracts and operations

5.1 | Container_submission smart contract

This smart contract is deployed at containers and has three functions: *Access*, *Transfer*, and *Profiling*. Each function gets necessary information for verifying its related GDPR rule in the verification phase. The smart contract also enables customer to identify a degree of compliance for verifying operations under GDPR rules. *Access* uses the boolean `auth_access` variable, to identify whether the service supplied by the actor supports encryption of personal data or not; *Transfer* gets the country name of the provider receiving customer data; *Profiling* requests the age of the customer whose personal data are under an automated profiling operation (eg, obtaining some statistical results on customer data).

5.2 | Verification smart contract

This smart contract is deployed by predefined voters to verify GDPR rules on actors. Each voter detects GDPR violation and sends out a message. Four functions: *Initial_Verification*, *Verify_Access*, *Verify_Transfer*, and *Verify_Profiling* are used to implement this. Each function is implemented for the verification of an operation executed by actors on customer data.[2]

Initial_Verification—For each operation executed by an actor, the function compares the degree of compliance of customer for verifying the operation and the thresholds determined by voters. If the former is greater than or equal to the latter, then it (locally) calls the function for verifying the operation in the smart contract.

Verify_Access—the verification smart contract has a list of all sensitive data identified by the GDPR standard. For verification, personal data logged by a container is compared with a sensitive data list. The log recorded by a container is checked to identify the authentication status. Providing that authentication variable (`auth_access`) is `false` value, the actor accessing the customer data is identified to be a violator.

Verify_Transfer—checks the location of the data receiver, and if outside of Europe the function then checks the list of countries certified by BCR country. If neither of these match, a GDPR violation is flagged.

Verify_Profiling—checks customer age via deploying container_submission smart contract. If the age is less than 18 years, then a violation is flagged.

The approach proposed in Reference 3 can be used for translating the aforementioned GDPR rules into opcodes for use in smart contracts. The approach focuses on most frequently used operations, for example, access, data transfer, and so on. These operations are used to support data processing (on personal user data) by service providers and their execution can be directly monitored and verified.

5.3 | Voting smart contract

This smart contract is deployed by the agreement builder and collects the votes returned by voters in order to check whether a violation is committed by actors or not. The function of this contract—called here *Conclude*—gets the addresses of voters participating in the verification process. Given Definition 2, if at least m voters report a violation, the actor is reported.

6 | EXPERIMENTAL RESULTS

An initial prototype was built by Ganache[18] and Ropsten,[19] and smart contracts in Figure 6 were programmed in Solidity.[20] The Ganache local test network supplied default gas and Ether to alter blockchain states under the function calls. Ropsten is a public test network, supporting miners, and with a gas limit of 4 712 388 for deploying a contract. Remix Ethereum was used as the framework to compile and run deployed contracts. The smart contracts *container_submission*and *verification* were deployed on Ganache and Ropsten, respectively. The amount of gas used for the contracts was 773 721 for *container_submission* and was 1 814 952 for *verification*.[3]The gas consumption for the deployment of voting smart contract depends on the number of voters involved in the contract. The variation in the amount of gas consumed by changing the number of voters, and the variation in gas used by changing the number of operations of actors are taken into account. Moreover, the impact of the deployments of voting contract with different number of voters on the (average) time taken for the mining process is evaluated. When detecting violations in GDPR rules identified in the case study, experiments are carried out to show the rate of violation detection under different threshold levels determined by voters. Furthermore, the relationship between the violation detection rate and the number of acceptable votes for reporting a violation is investigated with regards to different diversity of voters' thresholds.

[2]The smart contract calls the container_submission contract to retrieve the information already sent by container to the blockchain.
[3]The gas price was 1 Gwei in our experiments.

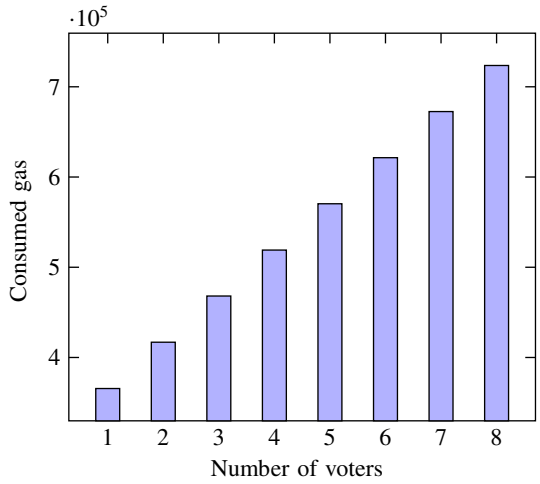


FIGURE 7 Number of voters vs gas consumption

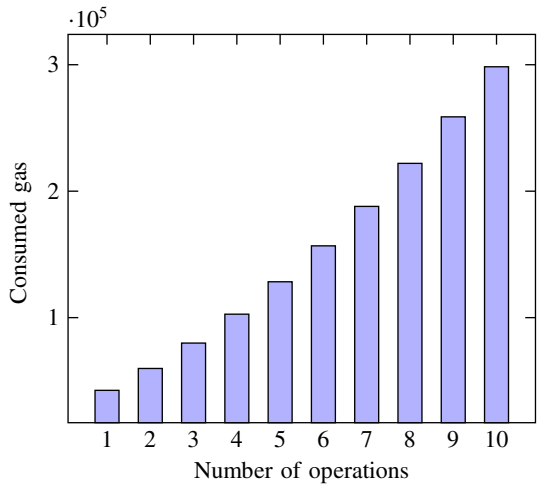


FIGURE 8 Number of operations vs gas consumption

Number of voters vs gas consumption: Figure 7 shows the relationship between the number of voters and the used gas spending on the deployment of voting contract. In this experiment deployed in Ganache test network, the number of voters varies from one to eight. As seen from the figure, when the number of voters increases, the amount of consumed gas increases constantly from 365 597 (one voter) to 723 608 (eight voters).

Number of operations executed by an actor vs gas consumption: We consider one actor and one voter, with number of operations (ie, *Access*, *Transfer*, *Profiling*) executed by the actor varying from 1 to 10. Transaction costs for the verification of `Initial_Verification` function (with different parameters) were repeated and measured five times to calculate the average used gas. In each activation, the operations, degrees of compliance, and voters' thresholds were selected randomly. Given these assumptions, the relationship between the number of operations and the transaction costs used for verifying them is shown in Figure 8. It can be seen from the figure that as the number of operations increases, the used gas varies from 42 503 (one operation) to 298 384 (10 operations).

Number of voters vs mining time: This experiment was performed on the Ropsten test network, to measure the time taken from the deployment to mining of a contract—repeated five times to calculate the average. Number of voters in the contract was altered from one to eight. Figure 9 identifies the time taken (in seconds) for the voting contract to be successfully mined since its deployment time. Similar to Reference 21, our results indicate that the time depends on the interest of miners in the voting contract and does not depend on the number of voters or contract parameters. It results that miners can normally take an arbitrary time for the process of mining.

Violation detection rates: the effect of changing voter thresholds on the rate of violation detection in the GDPR rules (from Section 5) is measured—where a violation in a rule is based on the conditions proposed in Definition 2. We consider nine actors (controllers/processors) and one voter (with one operation per actor). Moreover, the data subject assigns a degree of compliance for verifying operations executed by actors. Voter thresholds are set to 5, 7, and 9[4] From Figure 10, the *x*-axis indicates the number of actors who committed a violation, and the *y*-axis the rate of violation detection based on the voter's thresholds. For each threshold, the *verification* smart contract was activated 10 times to calculate the average

[4]Integer numbers are used as solidity does not support real numbers.

FIGURE 9 Number of voters vs mining time

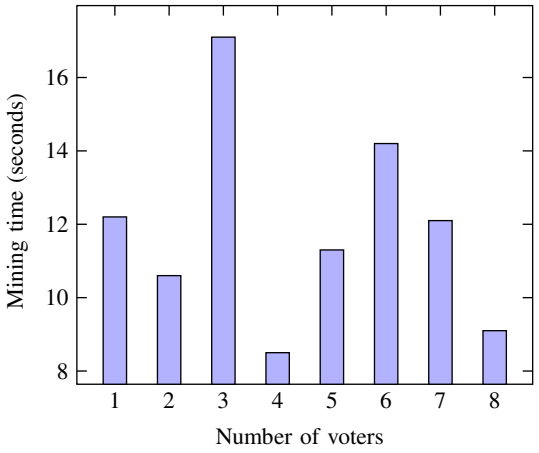
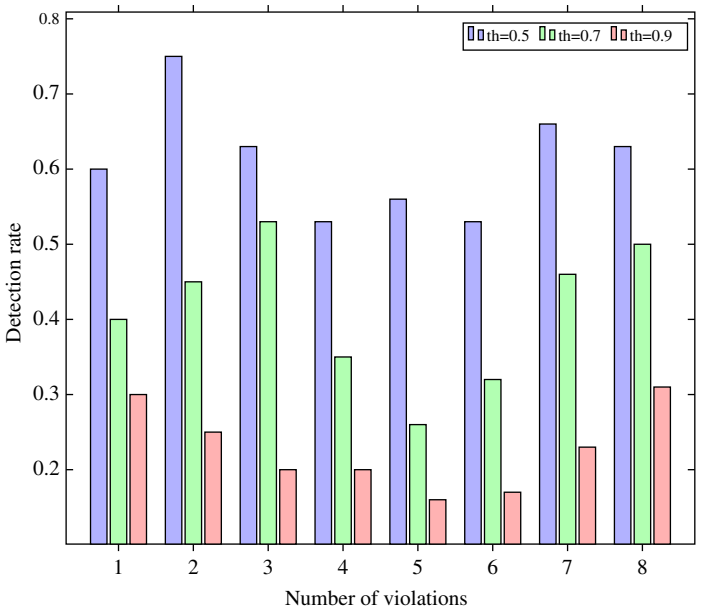


FIGURE 10 Violation detection rate vs no. of violations



rate of violation detection. In each activation, a random number between 0 and 10 was generated to indicate the degree of compliance with GDPR rules on the operation executed by each actor. Given such assumptions, the average rate of violation detection is calculated as:

$$\frac{\sum_{i=1}^{n} r_i}{n},$$

where n is the number of violations (varies from 1 to 8) and r_i is the average number of successful detection for the violation i. When a voter increases its threshold for verifying operations, the average rate of violation detection decreases. The fluctuations are due to the random generation of compliance degrees. We observe that changing the number of violations does not impact on the detection rate, as GDPR compliance of operations is automatically verified and any violation is flagged when the threshold of voter is less than or equal to the degree of compliance required by a data subject.

Violation detection rate vs number of votes: this experiment evaluates the impact of changing the minimum number of acceptable votes for reporting a violation on the average rate of violation detection. The evaluation is done under different thresholds. Considering eight actors, one is randomly selected as a violator breaching a GDPR rule. Moreover, there are eight voters (with thresholds between 0 and 10), and the compliance degree of data subject for verifying the rules is randomly chosen between 0 and 10. In the experiment, the diversity of thresholds for voters is assumed to be 4 and 8, that is, voters have 4 and 8 different thresholds for verifying operations, respectively. Regarding the eight voters involved in the evaluation, every two voters have the same threshold. However, for the diversity of eight, each voter has a different threshold with the others.

As illustrated in Figure 11, the x-axis shows the minimum number of acceptable votes to issue a violation report. The y-axis indicates the average rate of violation detection, calculated by the formula provided in the previous experiment with n=1. The smart contracts of the case study were

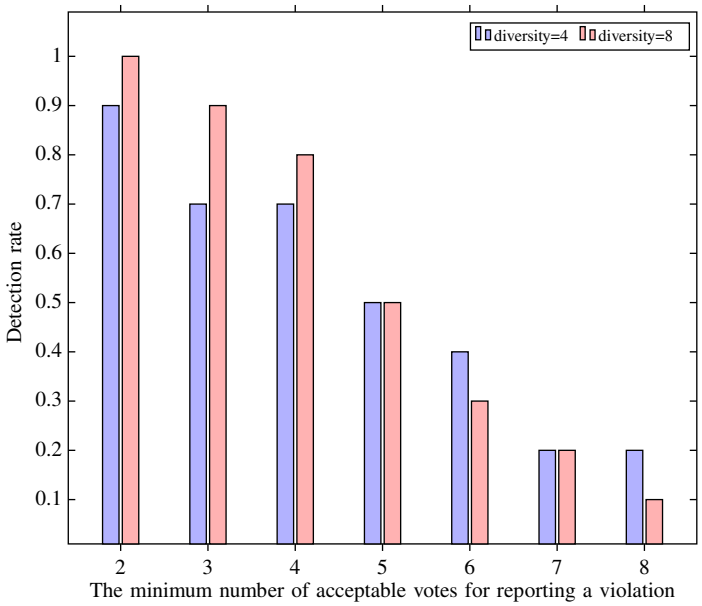


FIGURE 11 Rate of violation detection vs accepted number of votes with varying threshold

deployed and executed 10 times to calculate the average rate of violation detection. From Figure 11, by increasing the number of acceptable votes for reporting a breach, the average rate of violation detection declines gradually. It is also observed that a higher detection rate (or a more precise violation detection) can be achieved when the number of thresholds used increases and the number of acceptable votes decreases.

7 | RELATED WORK

Blockchain and smart contracts have recently motivated cloud security researchers to improve user trust and privacy for sharing data in cloud ecosystem. The potential of using blockchain-based techniques to protect healthcare data located in cloud was described in Reference 22, highlighting practical challenges for recording medical data in a blockchain network. Moreover, the authors in Reference 23 presented a patient centric healthcare data management system with the aid of blockchain. The system ensured that the private health-care data in cloud is only monitored by patient. A blockchain-based approach was proposed for storing cloud attestation in Reference 6. The authors implemented a smart contract for recording the migration of user data between cloud providers. The deployment of the smart contract enabled cloud users to identify the location of their data through the submission of a query to the contract—extended in Reference 24 to provide cloud users more control on the migration of their data to providers in user-defined white lists. Consumer-based data movement policies realized through a blockchain-based technique was reported in Reference 25. The authors in Reference 21 proposed an automatic way for tracking and enforcing data sharing agreements between a user and cloud providers with the aid of smart contract and blockchain technology. In this approach, the providers who violated the shared agreements were detected through a set of voters or arbiters listed in a voting contract. In Reference 26, a secure smart home architecture based on cloud and blockchain technology was proposed. The authors used an encryption and hashing algorithm to obtain confidentiality and trust. The integration of blockchain-based approaches into several security services, including authentication, privacy, data provenance, and integrity are reviewed in Reference 5. A conceptual model—called ProvChain—was designed to collect cloud data provenance and provide the assurance of data operations in a cloud storage application through logging provenance data in a blockchain network.[27] The integration of blockchain and attribute-based *signcryption* was proposed in Reference 28 to support secure data sharing in a cloud ecosystem. Although the aforementioned approaches take advantages of blockchain and smart contracts to enhance cloud user privacy and trust, none of them applied GDPR rules in their methods to clearly give some standard regulations to the actors processing user data.

Work that combines blockchain and GDPR in a cloud environment includes a blockchain-based approach for supporting data accountability and provenance tracking, which meets GDPR requirements proposed in Reference 4. The approach presented two different models for deploying a smart contract: (i) data subject consent rules recorded in a blockchain under which each actor (controller/processor) should follow the rules and (ii) actor policies supported as a smart contract that allows users as subscribers join or leave the contract. Verification to check for compliance of consent rules was undertaken manually in both cases. A personal health data sharing system has been proposed in Reference 29, which enables users to securely share their health data and for data consumers to get necessary data in a transparent manner and in compliance with GDPR. The system used blockchain technology supplemented by cloud storage shares the health data. A data quality inspection module relied on machine learning approaches was introduced in the system to monitor the quality of personal health data. Although the system benefits from GDPR and

blockchain for improving the privacy of users' health data, it still lacks a methodology whereby the verification of stored data in the blockchain network is supported. The authors in Reference 30 designed a conceptual and a high-level architecture for an identity management system that provides control on personal data usage with the aid of GDPR. The architecture also utilized blockchain technology to supply transparency, trust, and security. However, the validation and deployment of architecture in real-world applications was not discussed. A blockchain-based personally identifiable information management system, called BcPIIMS, was proposed in Reference 31 so that storing personal data in the system complies with GDPR rules. The verification was limited to the rule: *right to be forgotten*. The authors in Reference 32 took advantage of blockchain and GDPR to develop a digital *onboarding* framework that defines security policies for users' identity attributes stored on multiple centralized repositories. However, the verification of GDPR rules legislated for data transfer and profiling over the framework was not studied. Generally, though all these approaches make use of blockchain and GDPR to improve user privacy, they lack an automatic, transparent, and consensus mechanism under which GDPR violations are flagged. In Reference 33, a privacy-aware cloud architecture was proposed to improve transparency and enable the audit trail of providers who accessed user data through blockchain. The core of architecture took the advantages of GDPR and smart contracts to verify GDPR compliance. However, the smart contracts proposed for the verification of providers were not evaluated. In Reference 3, the audit trail of IoT devices under GDPR rules was presented through which several GDPR rules were translated as opcodes in smart contracts to automatically protect IoT user data. The verification of devices, however, was not undertaken by a consensus mechanism and the degree of compliance of users for verifying operations was not studied.

8 | CONCLUSION

A cloud-based architecture is proposed to enhance user privacy—using a blockchain-based technique in which GDPR requirements are supported. The technique provides a reactive mechanism so that the providers violating GDPR rules are detected—the likelihood of detection increasing as the voters increase in number (but at a high gas cost). In contrast to data provenance tracking approaches, where the blockchain network is manually checked by users, the presence of voters in our approach provides an automatic verification approach. The degree of compliance enables data subjects to trade-off GDPR rule checking vs the cost of performing such checks. Based on this concept, voters perform the verification process when their thresholds were less than or equal to the degree of compliance of the data subject. A case study is used to illustrate the implementation of smart contracts for checking GDPR rules, for accessing, profiling, and transferring of user data in the cloud environment. Experimental results show that violation detection rate increases when voters select lower levels of threshold. Future work will focus on a preventative approach, whereby a smart contract does not allow providers to process user data if an operation is likely to violate GDPR rules. Such a preventative mechanism can then be compared with the reactive one proposed in this article in terms of performance, scalability, and trust. Translation of legal GDPR clauses into rules that can be programmatically verified remains another challenge.

ACKNOWLEDGMENT

The work reported in this article has been funded by EPSRC under grant EP/R033439/1, Project title: "PACE: Privacy-aware Cloud Ecosystems."

ORCID

Masoud Barati https://orcid.org/0000-0001-7829-2240

REFERENCES

1. Russo B, Valle L, Bonzagni G, Locatello D, Pancaldi M, Tosi D. Cloud computing and the new EU general data protection regulation. *IEEE Cloud Comput*. 2018;5(6):58-68.
2. Virvou M, Mougiakou E. Based on GDPR privacy in UML: case of e-learning program. Paper presented at: Proceedings of the 8th International Conference on Information, Intelligence, Systems & Applications; 2017; Larnaca, Cyprus.
3. Barati M, Petri I, Rana, OF. Developing GDPR compliant user data policies for Internet of things. Paper presented at: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing; 2019:133-141; Auckland, NZ.
4. Neisse R, Steri G, Nai-Fovino I. A blockchain-based approach for data accountability and provenance tracking. Paper presented at: Proceedings of the 12th International Conference on Availability, Reliability and Security; 2017; Reggio Calabria, Italy.
5. Salman T, Zolanvar M, Erbad A, Jain R, Samaka M. Security services using blockchains: a state of the art survey. *IEEE Commun Surv Tutor*. 2018.
6. Kirkman S, Newman R. Using smart contracts and blockchains to support consumer trust across distributed clouds. Paper presented at: Proceedings of the 13th International Conference on Grid, Cloud, and Cluster Computing; 2017:10-16; Las Vegas, NV.
7. Ulybyshev D, Villarreal-Vasquez M, Bhargava B, Mani G, Seaberg S, Conoval P, Pike R, Kobes J. (WIP) Blockhub: blockchain-based software development system for untrusted environments. Paper presented at: Proceedings of the 11th International Conference on Cloud Computing; 2018:582-585; San Francisco, CA.
8. Wood G. Ethereum: a secure decentralised generalised transaction ledger. *Ethereum Project Yellow Pap*. 2014.
9. Zheng Z, Xie S, Dai H, Chen X, Wang H. An overview of blockchain technology: architecture, consensus, and future trends. Paper presented: Proceedings of the IEEE 6th International Congress on Big Data; 2017:557-564; Honolulu.
10. Ethereum. https://www.ethereum.org/. Accessed March 2020.
11. Corda. https://www.corda.net/. March 2020].

12. R3. https://www.r3.com/. March 2020.
13. Monax. https://monax.io/. March 2020.
14. Multichain. https://www.multichain.com/. March 2020.
15. ETH Gas Station. https://ethgasstation.info/. March 2020.
16. Afzal A, Shafiq B, Shamai S, Elahraf A, Vaidya J, Adaml NR. ASSEMBLE: attribute, structure and semantics based service mapping approach for collaborative business process development. *IEEE Trans Serv Comput*. 2017;10(2):1-14.
17. Corrales M, Jurcys P, Kousiouris G. Smart contracts and smart disclosure: coding a GDPR compliance framework. *SSRN Electron J*. 2018. https://doi.org/10.2139/ssrn.3121658.
18. Ganache. https://github.com/trufflesuite/ganache. March 2020.
19. Ropsten testnet pow chain. https://github.com/ethereum/ropsten. March 2020.
20. Solidity. https://solidity.readthedocs.io/en/v0.5.3/. March 2020.
21. Desai H, Liu K, Kantarcioglu M, Kagal L. Enforceable data sharing agreements using smart contracts; 2018. arXiv:1804.10645v1[cs.CY].
22. Esposito C, De Santis A, Tortora G, Chang H, Choo KR. Blockchain: a panacea for healthcare cloud-based data security and privacy? *IEEE Cloud Comput*. 2018;5(1):31-37.
23. Omar AA, Bhuiyan MZA, Basuc A, Kiyomoto S, Rahman MS. Privacy-friendly platform for healthcare data in cloud based on blockchain environment. *Future Generat Comput Syst*. 2019;95:511-521.
24. Kirkman S, Newman R. A cloud data movement policy architecture based on smart contracts and the Ethereum blockchain. Paper presented at: Proceedings of the IEEE International Conference on Cloud Engineering; 2018:371-377; Orlando, FL.
25. Kirkman S. A data movement policy framework for improving trust in the cloud using smart contracts and blockchains. Paper presented at: Proceedings of the IEEE International Conference on Cloud Engineering; 2018:270-273; Orlando, FL.
26. Singh S, Ra I-H, Meng W, Kaur M, Cho GH. SH-BlockCC: A secure and efficient Internet of things smart home architecture based on cloud computing and blockchain technology. *Int J Distrib Sens Netw*. 2019;15(4). https://doi.org/10.1177/1550147719844159.
27. Liang X, Shetty S, Tosh D, Kamhoua C, Kwiat K, Njilla L. ProvChain: a blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. Paper presented at: Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing; 2017:468-477; Madrid, Spain.
28. Eltayieb N, Elhabob R, Hassan A, Li F. A blockchain-based attribute-based signcryption scheme to secure data sharing in the cloud. *J Syst Arch*. 2020;102 101653. https://doi.org/10.1016/j.sysarc.2019.101653.
29. Zheng X, Mukkamala RR., Vatrapu R, Ordieres-Mere J. Blockchain-based personal health data sharing system using cloud storage. Paper presented at: Proceedings of the 20th International Conference on e-Health Networking, Applications and Services (Healthcom); 2018; Ostrava, Czech Republic.
30. Faber B, Michelet G, Weidmann N, Mukkamala RR, Vatrapu R. BPDIMS: a blockchain-based personal data and identity management system. Paper presented at: Proceedings of the 52nd Hawaii International Conference on System Sciences; 2019:6855-6864; Hawaii.
31. Al-Zaben N, Onik MMH, Yang J, Lee N-Y, Kim C-S. General data protection regulation complied blockchain architecture for personally identifiable information management. Paper presented at: Proceedings of the International Conference on Computing, Electronics & Comms. Engineering; 2018:77-82; Southend, UK.
32. Soltani R, Nguyen UT, An A. A new approach to client onboarding using self-sovereign identity and distributed ledger. Paper presented at: Proceedings of the IEEE Conferences on Internet of Things, Green Computing and Communications; 2018:1129-1136; Halifax, Canada.
33. Barati M, Rana O, Theodorakopoulos G, Burnap P. Privacy-aware cloud ecosystems and GDPR compliance. Paper presented at: Proceedings of the 7th International Conference on Future Internet of Things and Cloud; 2019; Istanbul, Turkey.

How to cite this article: Barati M, Rana O. Privacy-aware cloud ecosystems: Architecture and performance. *Concurrency Computat Pract Exper*. 2020;e5852. https://doi.org/10.1002/cpe.5852