Privacy-Aware Cloud Ecosystems and GDPR Compliance

Masoud Barati, Omer Rana, George Theodorakopoulos, Peter Burnap
School of Computer Science & Informatics
Cardiff University, UK
BaratiM@cardiff.ac.uk

***Abstract*—Understanding how cloud providers support the European General Data Protection Regulation (GDPR) remains an imporant challenge for new providers emerging on the market. GDPR influences access to, storage, processing and tranmission of data, requiring these operations to be exposed to a user to seek explicit consent. A privacy-aware cloud architecture is proposed that improves transparency and enables the audit trail of providers who accessed the user data to be recorded. The architecture not only supports GDPR compliance by imposing several data protection requirements on cloud providers, but also benefits from a blockchain network that securely stores the providers' operations on the user data. A blockchain-based tracking approach based on a shared privacy agreement implemented as a smart contract is described – providers who violate GDPR rules are automatically reported through a voting mechanism.**

***Index Terms*—data privacy, cloud architecture, blockchain, smart contracts, general data protection regulation**

I. Introduction

With the growth of available on-line services, often hosted over cloud infrastructure, there is a realization that such services can be hosted on an interlinked set of cloud providers [1]. Normally, the users of these services only interact with a Web server backend, rather than the larger, distributed ecosystem of providers that exist behind this server. Users often entrust their data without realizing that the providers may share their data with some back-end services such as cloud hosted analytic and advertisers. Though this was a simple problem in the past, it has recently become a main challenge with the expansion on Internet connected devices. In order to address this, the General Data Protection Regulation (GDPR) is implemented to impose some obligations on the providers' side in which non-expert users can make informed decisions about their privacy [2].

The key elements introduced in GDPR are a data subject, a controller or joint controller, and a processor. The data subject is directly or indirectly identified through an identifier, e.g., name, location, IP address, identification number, and so on. The controller is a person or organization specifies the aim of the processing of user personal data. The notion of joint controller is introduced where two or more controllers jointly determine 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 theses 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 and its processing steps. For instance, the integration of GDPR into the cloud ecosystems shares the new responsibility and accountability requirements to both the processes and controllers as providers to handle what is processed by them. Under such requirements, any operation of a cloud provider on personal data must be in accordance with user consent [2].

Given the GDPR requirements, several solutions have been provided to support the accountability and provenance tracking of the user data when it is delivered to a controller or submitted to a processor [4], [5]. Most solutions utilize blockchain-based technologies to improve transparency and trust between users and their data processing parties [4]–[9]. In addition to these solutions, the blockchain as a shared ledger has recently been integrated into many promising applications such as security services which cover privacy, authentication, provenance, and integrity domains [9], [10].

This paper proposes a blockchain-based approach to improve the provenance tracking of cloud user data under GDPR requirements. A systems architecture is proposed integrating a blockchain network into a service-oriented cloud environment, in order to enable the audit trail of providers on user data. The privacy management layer of the architecture not only supports trustable containers that securely log all providers' operations on personal data in a blockchain, but also provide a reactive mechanism in which the providers who violate GDPR rules can be detected through a consensus. Following that, a case study is presented to show how some GDPR rules can be deployed as smart contracts in the blockchain. The rules are related to the access, transfer, and profiling of user data. The rest of this paper is structured as follows. Section V-A presents a background about blockchain and smart contract. Section II proposes the new architecture for user privacy in cloud computing. Section III describes the interactions among the components giving rise to the realization of the architecture. Section IV represents the case study to illustrate the verification of the blockchain regarding some GDPR rules.

Identify applicable funding agency here. If none, delete this.

978-1-7281-2888-7/19/$31.00 ©2019 IEEE
DOI 10.1109/FiCloud.2019.00024

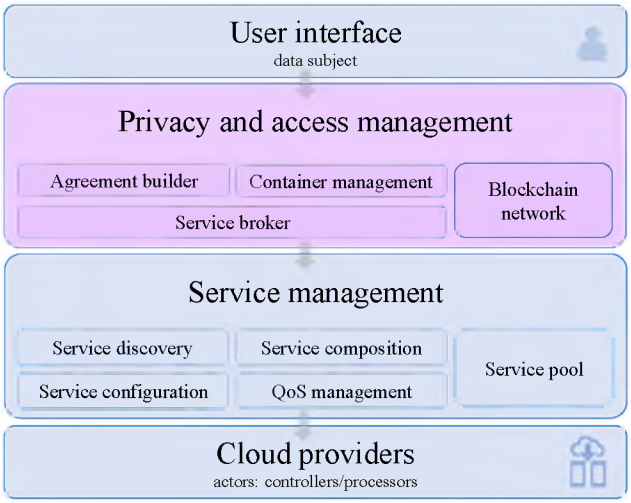

Fig. 1. An architecture for privacy-aware cloud ecosystems

Related work is reviewed in Section V, and finally, Section VI concludes the paper and gives some indications about future work.

II. Privacy-aware cloud ecosystems architecture

A four layer architecture is proposed for supporting privacy-aware cloud ecosystems in Fig. 1. This architecture has two general workflows: (i) for service delivery; (ii) for 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.

A. User interface

This layer facilitates interactions between cloud users and the other layers of the architecture via an interface. It enables users to request their required services and to release their personal data that can be shared among a set of cloud providers. Through the interface, the users can also submit their preferences for verifying GDPR rules on the operations that will be executed by providers on their personal data.

B. Privacy and access management

This layer extends capability outlined in service-based cloud architectures [24], [25] and involves the following components: service broker, agreement builder, container manager, and blockchain network. This layer records the audit trail of providers through a blockchain network, recording the operations that are executed by providers on user data. Operations include: access, store, profile, or transfer of user data. If a provider's operation violates GDPR regulations, the layer can notify users about the violation.

Service broker: receives user requests and publishes the discovered services from the service management layer to the interface. Moreover, when a service is discovered for a user, the service broker provides the name, location and address of service provider to the agreement builder component.

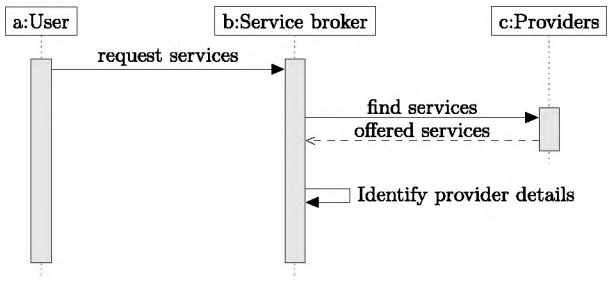

Fig. 2. A sequence diagram for the interactions to find composite services

Agreement builder: creates a shared agreement between a user and (a) providers. In this agreement, the user gives explicit consent for processing their data by providers. Given the operations executed on user data, this component builds a smart contract to record the information necessary for verifying operations under GDPR rules. The smart contract address is sent to providers to be deployed by their containers. For example, consider GDPR rule for operations that can only be accessible to users of 18 years or over. Given this operation and its related rule, the agreement builder can provide a field for inserting the age in the smart contract. The field is then filled by the container of provider executing profiling operation.

Container manager: the component launches a container on the provider to collect and submit data to a blockchain network. It deploys the smart contract supplied by the agreement builder for recording such data. The data may include: user and provider addresses, the operations processed on user data (e.g., access, transfer, profiling, and etc.), and other information for verifying operations under GDPR rules (e.g., age for profiling operation). Our assumption is that containers are trustable and they record every operations executed on the user data.

Blockchain network: a distributed ledger which makes immutable blocks for the data sent by the containers, generating an audit trail of operations performed by a provider. The blockchain can be accessed by other predefined parties—called as *voters*—to check whether a provider committed a breach in the GDPR rules or not.

The **Service management** layer is responsible for discovering cloud services, building composite services, and publishing the services or composite services to the upper layers. All the discovered services are stored in the service pool provided in the layer. Furthermore, they can be configured and installed via the service configuration component. The role of QoS management component is keeping all details about the quality aspects of cloud services, namely cost, availability, time constraint, and so forth.

III. Realization of the architecture

The realization of both workflows throughout the architecture requires some interactions among the components in

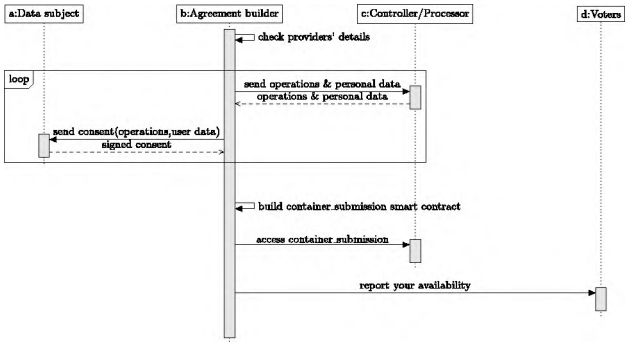

Fig. 3. A sequence diagram for building a shared agreement

the privacy and access management layer. To this end, such interactions are classified into four phases:

A. Phase 1: service discovery and composition

This phase consists of interactions through which the requested services of a user are discovered and suggested in the form of a set of atomic or composite services. The sequence diagram in Fig. 2 shows such interactions. The requested services are submitted to the service broker component and forwarded to the service management layer to find the designated providers able offer services. Finally, the service broker identifies the details of the designated providers involved in the offered services, e.g., their names, locations or addresses. These details can be accessed by the agreement builder for future references.

B. Phase 2: building a shared agreement

This phase leads to the creation of a shared agreement between user and providers. The protocol is described in Fig. 3. This phase is activated by the agreement builder who accesses the details already provided through the service broker component, i.e. the names and addresses of service providers, checks these details to identify the parties requiring the user data and then sends a request to data controllers /processors (actors) in order to inform what operations will be executed by them on user data. In this step, actors should also supply the personal data that will be processed by them to the agreement builder. Once the details of operations and requested personal data are collected, the agreement builder sends them to the data subject and waits to get the consent of data subject for connecting to the actors. According to the GDPR requirement, data subject can give a consent with regards to the identity of the actors, the purpose of data processing, and the right to withdraw the agreements at any time. Since there can be offered more than one service or composite service for data subject, this process can be iterated until the final agreement is reached.

Given the operations and the GDPR rules legislated for them, the agreement builder then prepares a smart contract, called as *container_submission*. The smart contract consists of a template for storing data in the blockchain. This data is

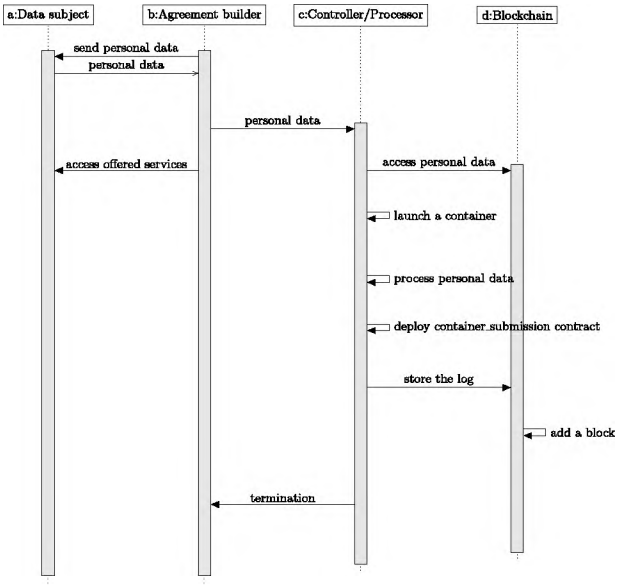

Fig. 4. A sequence diagram for logging data in the blockchain

used for the verification of operations under GDPR rules. For instance, if an actor transfers user data to another actor, the location of the data receiver should be stored by data sender in the blockchain. Following that, the agreement builder sends the smart contract address to the designated actors. The component also determines a set of voters for verifying operations.[1] The voters are third parties connected to the blockchain network and can give their votes when the executed operations do not comply GDPR rules.

C. Phase 3: logging the user data processing

This phase presents the interactions in which the operations of actors on user data are stored in a blockchain. Given the aforementioned assumption, each actor is equipped with a trusted container recording each operation on the personal data as a log. Each log includes the necessary data that must be provided by container with respect to the container_submission smart contract.

As demonstrated in Fig. 4, upon the creation of the shared agreement, the agreement builder asks the data subject to provide personal data required by actors. Once such data has been provided, it is forwarded to the actors. Then, the actor launches a container to store the logs after the execution of each operation on the user data. In fact, the container deploys the container_submission smart contract to send the data, which facilitate the verification of operation, to the blockchain. Finally, upon the termination of the operations of actors on user data, a message signifying the finalization of the process is submitted to the agreement builder.

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

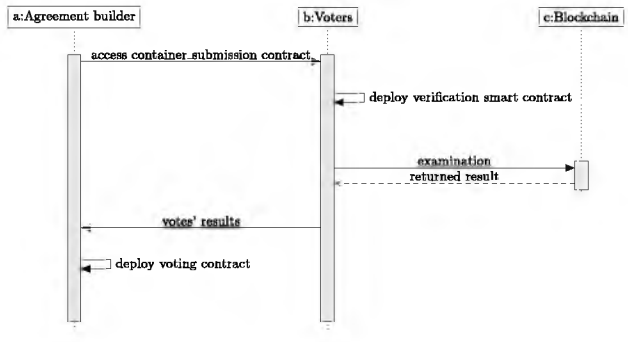

Fig. 5. A sequence diagram for verifying the blockchain

D. Phase 4: verification of the operations

This phase verifies whether actors violate GDPR rules or not. The sequence diagram depicted in Fig. 5 represents the series of interactions for the verification of the blockchain network. The voters in this diagram are the parties taking part for verifying the operations of actors on user data. Voters deploy a smart contract, called as *verification*, in order to check the operations. The smart contract for each type of operation processed on personal data, provides a function that verifies the operation in accordance with its related GDPR rule (for more details see Section IV).

Once the agreement builder component receives a termination message, it sends the voters the address of container_submission contract to check the blockchain to ensure that the executed operations conform to GDPR rules. Then, the voters deploy the verification smart contract. The contract allows voters to access the data already stored by container in the blockchain. By retrieving such data, voters can activate the functions supplied by verification smart contract to check whether the stored data comply with GDPR rules or not. After the verification, they give their votes to the agreement builder component. Finally, the component by deploying a smart contract, called here voting, collects the votes of voters. Given the smart contract, if most voters vote for a violation in GDPR rules, the actor committed the violation is reported. Figure 8 in Appendix presents a sample of voting smart contract.

1) Degree of compliance for verifying operations: During building an agreement between data subject and actors, the former should be notified in advance about the operations that will be executed by actors on their personal data. In some cases ,a data subject may not be concerned about some operations or the personal data that must be processed by them. Hence, the verification of such operations under GDPR rules is not likely to be important for data subject in these cases. Normally, verifying an operation is a costly process and a part of such cost should be paid by the data subject. Hence, some operations may be unimportant for a data subject. We introduce the notion of *degree of compliance*, a metric in which data subject gives a value between 0 and 1 to each operation that will be executed on their personal data. This recognises that not all user data should be treated in the same way, and some information may be considered more *sensitive* by a user than other.

There can be a classification for the degree of compliance of an operation (or a provider): *fully-compliance*, *partially-compliance*, and *non-compliance*. More precisely, when a full-compliance degree is given to an operation, it must inevitably be verified under GDPR rules. However, the verification has a lower level of importance for data subject when it comes to the partially-compliance degree. Finally, a non-compliance degree can be given when the verification is never a concern from the data subject's point of view.The following definition formally presents such a classification.

Definition 1. Let $\mathcal{A}$ be a set of operations executed by actors 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 fully-compliance, partially-compliance, and non-compliance degrees, respectively.

Given the degree of compliance, each voter can also define a threshold for the verification of each operation. If the degree of compliance for an operation is greater than or equal to a voter's threshold, the operation is verified. 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 degrees expressed by the data subject and the thresholds determined by voters. A violation can finally be detected with respect to the calculation of voting results. In order to formally define the conditions under which an actor is classified to be violator, the following definition is provided.

Definition 2. Let $\mathcal{V} = \{V_1, \cdots, V_l\}$ be a set of voters and v_j be a vote reported by V_j after verifying operation α_i executed 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 minimum number of acceptable votes for reporting a violation. The actor i is classified to be violator if $\Sigma_{j=1}^{l} v_j \geq m$.

IV. A CASE STUDY FOR VERIFYING SOME GDPR RULES

Consider an e-commerce sales service integrated from three different services: *Order creation*, *Payment*, and *Shipping* [26]. A customer is connected to the cloud to make a product order, ship it to a destination address, and organize the payment process in an online portal. To use this composite service, providers should access customer personal data. The purposes of data processing (operations) for each service provider is as follows:

- Order creation service provider: should get the customer data: name, identification number, biometric information,

age, and contact details. Its service performs a mathematical function on some data of customers ordering an specific product to obtain a number of statistical results which are published in public.

- Payment service provider: needs to access the customer data comprising of name, identification number, and bank account details, in order to handle the payment.
- Shipping service provider: requires the name and contact details of customer. The provider remotely interacts with a sub contractor (a *Mail* service provider) to manage the product delivery. In this end, the customer data should be forwarded to the sub contractor as well.

Given the roles defined in GDPR, both *Order creation* and *Payment* service providers are supposed to be processors and directly manage or process the customer data. The *Shipping* service provider, however, can have both processor and controller roles. It plays as a processor when manages a part of data delivery and plays as a controller when transfers 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 proposed for the operations: access, transfer, and profiling of customer data. The details of such rules are:

1) the actors guarantee that if customer data is sensitive, they provide an authentication control mechanism for preventing unauthorized access to customer data.
2) the actors guarantee that if customer data is transferred to a processor located outside of Europe, the receiver must belong to a country following BCR rules.
3) the actors guarantee that automated profiling operations such as statistics are not performed on data of customers whose ages are under 18.

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 an encryption or authentication technique like a secure login. Sensitive data consists of information such as religious or philosophical beliefs, genetic data, biometric data, and data concerning health. The second rule refers to a GDPR obligation for transferring customer data in which the data receiver should belong to an European member states or belong to a country certified by BCR clauses. The BCR is internal rules (e.g. code of conduct) which are adopted by a community of multinational companies that want to move customer data internationally across various jurisdictions [37]. The third rule forbids automated profiling operations on customers whose ages are under 18.

For verifying the actors, Figure 6 represents an architectural 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 the containers of actors involving in e-commerce composite service, voters, and agreement builder. These parties should be connected to

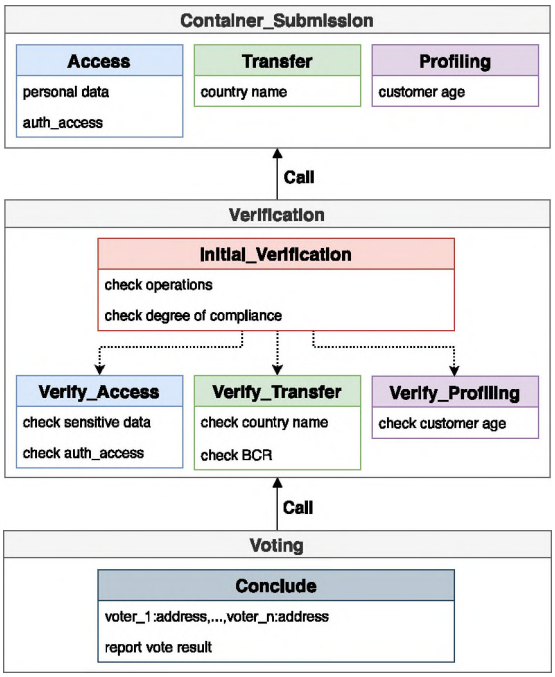

Fig. 6. Architectural overview of smart contracts

the Ethereum network and have an access to an Ethereum client [11]. Then, the smart contracts can be shared among them.[2] The details about the implementation of smart contracts depicted in Fig. 6 are provided as follows.

A. *Container_submission smart contract*

This smart contract is deployed by containers and has three functions, called here *Access*, *Transfer*, and *Profiling*. Each function gets necessary information that are necessary for verifying its related GDPR rule in the verification phase. The smart contract also enables customer to give degrees of compliance for verifying operations under GDPR rules.

Access—It involves a variable, called here `auth_access`, through which the container can assign a boolean value to identify whether the service supplied by actor supports the encryption of personal data or not. Moreover, the container through this function can log the personal data accessed by each actor.

Transfer—It gets the country name of the provider receiving customer data.

Profiling—It gets the age of customer whose personal data is under an automated profiling operation (e.g., obtaining some statistical results on customer data).

Figure 7 in Appendix illustrates a part of the implementation of the smart contract in Solidity [29].

[2]Smart contracts have public addresses on Ethereum blockchain and they are accessible by Ethereum clients.

B. Verification smart contract

This smart contract is deployed by predefined voters to verify the GDPR rules on actors. In the case of detecting a violation in the rules, the contract enables the voters to give a vote. In the implementation of the smart contract, there exist four functions, called here *Initial_Verification*, *Verify_Access*, *Verify_Transfer*, and *Verify_Profiling*. Each function is implemented for the verification of an operation executed by actors on customer data.[3]

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 which is considered for verifying the operation in the smart contract.

Verify_Access—In the implementation of this function, the assumption is that the verification smart contract involves a list containing all sensitive data identified by GDPR standard. For the aim of verification, the function, first, retrieves the personal data logged by container. Following that, through the sensitive data list, the function detects whether the personal data is sensitive or not. In the case of sensitive data, the status of safeguard or authentication in the service used by customer should then be verified. In this end, the logs recorded by container is checked here to observe the authentication status. Providing that authentication variable (`auth_access`) has a `false` value, the actor accessing the customer data is detected as a violator.

Verify_Transfer—This function first obtains the country name of a data receiver through checking the blocks created by container. If the receiver is outside of Europe, the function then checks the list of countries certified by BCR.[4] In the case that the data receiver country is not found in the list, the sender of customer data is classified to be violator.

Verify_Profiling—The function retrieves the customer age already recorded in the blockchain via deploying container_submission smart contract. If the age is less than 18, the voters vote for a violation committed by the actor executing the operation.

C. 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. If at least m voters, which is the minimum number of acceptable votes, vote for the detection of a violation, the violator is reported. Figure 8 in Appendix shows the implementation of the smart contract in Solidity.

[3]The smart contract calls the container_submission contract to retrieve the information already sent by container to the blockchain.

[4]The smart contract is supposed to have a list containing countries that follow BCR.

V. Related work

The challenge of the user trust and privacy for sharing data in the cloud environment has recently motivated the cloud researchers to find a solution based on smart contract and blockchain-based techniques. In this end, a blockchain-based data sharing framework has been presented to provide privacy for recording medical data within the cloud environments [32]. The framework was based on a lightweight and permissioned blockchain giving the access right to only verified users. The potential of using blockchain-based techniques in order to protect the healthcare data located in the cloud ecosystem was scrupulously studied in [30]. The authors also described some practical challenges to highlight the importance of privacy for recording the data in the blockchain network. In [11], a blockchain-based approach has been proposed for storing cloud attestation. The authors implemented a smart contract in order to record the migration of the user data between cloud providers. The deployment of the smart contract enabled the cloud users to inform about the location of their data through the submission of a query to the contract. In [34], the same authors extended the smart contract to provide the cloud users more control on the migration of their data so that the data can only be shared among the providers existing in the users' white lists. In the line of [34], the trust of cloud customers was improved by supplying some consumer-based data movement policies in a high integrity way, which was realized through a blockchain-based technique [31]. The authors in [33] 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. However, none of these approaches applied the GDPR-based requirements in their smart contracts to clearly give some standard regulations to the actors processing the user data.

The integration of blockchain-based approaches into several security services, including authentication, privacy, data provenance, and integrity, has been reviewed in [9]. Given some recent approaches in the data provenance domains, 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 the log of provenance data in a blockchain network [35]. However, the verification of the data operations is locally performed outside the network. In [4], a blockchain-based approach for supporting data accountability and provenance tracking, which meets the GDPR requirements, has been proposed. The approach presented two different models for deploying a smart contract in the blockchain. The fist model deployed the data subject consent rules in a blockchain at which each actor (controller/processor) should follow the rules. The second model deployed the actor policies as a smart contract in a blockchain that allows the users as subscribers join or leave the contract. However, the verification of the blockchain to check whether the actors violated the consent rules is manually

done by the user in both models. Moreover, a combination model to enable the negotiation between a data subject and an actor for reaching a shared agreement was not studied in [4]. A personal health data sharing system has been proposed in [36], which enabled users to securely share their health data and help data consumers to get necessary data in a transparent manner and in compliance with GDPR. The system used blockchain technology supplemented by cloud storage to share the health data. Likewise, 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 both GDPR and blockchain for improving the privacy of users' health data, it still lacks a methodology through which the verification of the stored data in the blockchain network is provided.

A. Blockchain background

A blockchain is a public ledger comprising of a distributed, shared database and a set of connected nodes. The database stores the records in blocks. The blocks can be accessible by users, whereas they cannot be altered or deleted. The blocks are structured as a chain and each of them keeps the hash address of its previous block. Each block 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 [16]. The nodes in a blockchain have a distributed and peer-to-peer form and can build a new block of valid transactions through a mining process. The nodes creating the blocks are called miners. Mining is a key concept of the blockchain through which a block is created and attached to the blockchain network. To this end, several techniques are currently available, namely Proof of Work (PoW), Proof of Stake (PoS), Proof of Space(PoSpace), Proof of Importance (PoI), and Practical Byzantine Fault Tolerance (PBFT).

A blockchain network can be categorized into a public, federated, or private blockchain [17]. In the public blockchain, everyone can participate and access blocks without any permission (e.g., Bitcoin and Ethereum). In the federated blockchain, the network is operated under the leadership of several organizations or groups, which do not permit any user with access to the Internet to take part in the verification of transactions (e.g., Corda [18] and R3 [19]). Finally in the private blockchain, only one organization has the permission of creating or verifying the blocks (e.g., MONAX [20] and Multichain [21]). A *smart contract* constitutes programming code which runs on the blockchain and translates a 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, each of which may change the state of the blockchain.

VI. Conclusion

A cloud-based architecture was designed to enhance user privacy. The architecture used a blockchain-based technique in which the GDPR requirements were supported. The technique provided a reactive mechanism so that the providers violating GDPR rules were detected by a number of voters. Iin contrast to the data provenance tracking approach proposed in [4], where the blockchain network was manually checked by users, the presence of voters in our approach led to an automatic way for verifying the network. In this paper, the notion of degree of compliance enabled the data subject to give a weight as their level of importance for verifying GDPR rules on providers' operations. This aspect recognises that not all data has equal privacy constraints, and reducing these constraints can support the discovery of providers at a lower cost to the user. Voters therefore performed the verification process when their thresholds were less than or equal to the degree of compliance of the data subject. To illustrate a real-world scenario, the paper presented a case study leading to the implementation of smart contracts for checking GDPR rules over the blockchain network. The rules were related to the access, profiling, and transfer of user data in the cloud environment. Finally, the experimental results showed that a high violation detection rate is achievable when voters select lower levels of threshold.

Our future work will focus on proposing a preventative mechanism in the blockchain-based technique exploited in our architecture. The mechanism enforces providers to log their data in the blockchain via the deployment of a smart contract considering GDPR rules. The smart contract does not allow providers to process user data based on an operation violating GDPR rules. Such a preventative mechanism can then be compared with the reactive one proposed in the paper in terms of performance, scalability, and trust. Furthermore, the translation of available GDPR rules, particularly legislated for cloud environments, into smart contracts is another potential research avenue for future consideration.

References

[1] K. Bila, O. Khali, A. Erbad, and S. U. Kha, Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers, *Computer Networks*, vol. 130, pp. 94–120, 2018.

[2] B. Russo, L. Valle, G. Bonzagni, D. Locatello, M. Pancaldi, and D. Tosi, Cloud Computing and the New EU General Data Protection Regulation, *IEEE Cloud Computing*, vol. 5, no. 6, pp. 58–68, 2018.

[3] M. Virvou and E. Mougiakou, Based on GDPR privacy in UML: case of e-learning program, *in Proc. of the 8th International Conference on Information, Intelligence, Systems & Applications*, Larnaca, Cyprus, 2017.

[4] R. Neisse, G. Steri, and I. Nai-Fovino, A Blockchain-based Approach for Data Accountability and Provenance Tracking, *in Proc. of the 12th International Conference on Availability, Reliability and Security*, Reggio Calabria, Italy, 2017.

[5] I. Belic, Data Protection Challenges of public permissionles blockchains in relation to the GDP, *Master Thesis*,Tilburg University, 2018.

[6] C. Wirth and M. Kolain, A Blockcchain-enabled GDPR-compliant Approach for Handling Personal Data. In: W. Prinz and P. Hoschka (eds.) *Proc. of the 1st ERCIM Blockchain Workshop*, Reports of the European Society for Socially Embedded Technologies, vol. 2, no. 6, 2018.

[7] M. Finck, Blockchains and Data Protection in the European Union, *Max Planck Institute for Innovation & Competition Research Paper No. 18-01, SSRN Electronic Journal*, 2017.

[8] D. Schmelz, G. Fischer, P. Niemeier, L. Zhu, and T. Grechenig, Towards Using Public Blockchain in Information-Centric Networks: Challenges Imposed by the European Union's General Data Protection Regulation, *in Proc. of the 1st IEEE International Conference on Hot Information-Centric Networking*, Shenzhen, China, 2018, pp. 223–228.

[9] T. Salman, M. Zolanvar, A. Erbad, R. Jain, and M. Samaka, Security Services Using Blockchains: A State of the Art Survey, *IEEE Communications Surveys and Tutorials*, 2018.

[10] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, MedRec: Using Blockchain for Medical Data Access and Permission Management, *in Proc. of the 2nd International Conference on Open and Big Data*, Vienna, Austria, 2016, pp. 25–30.

[11] S. Kirkman and R. Newman, Using Smart Contracts and Blockchains to Support Consumer Trust Across Distributed Clouds, *in Proc. of the 13th International Conference on Grid, Cloud, and Cluster Computing*, Las Vegas, NV, 2017, pp. 10–16.

[12] E. Gaetani, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, Blockchain-based Database to Ensure Data Integrity in Cloud Computing Environments, *in Proc. of the First Italian Conference on Cybersecurity*, Venice, Italy, 2017, pp. 146–155.

[13] Y. Zhang, S. Wu, B. Jin, and J. Du, A Blockchain-based Process Provenance for Cloud Forensics, *in Proc. of the 3rd IEEE International Conference on Computer and Communications*, Chengdu, China, 2017, pp. 2470–2473.

[14] F. Casino, T.K. Dasaklis, and C. Patsakis, A systematic literature review of blockchain-based applications: Current status, classification and open issues, *Telematics and Informatics*, vol. 36, 2019, pp. 55–81.

[15] D. Ulybyshev, M. Villarreal-Vasquez, B. Bhargava, G. Mani, S. Seaberg, P. Conoval, R. Pike, and J. Kobes, (WIP) Blockhub: Blockchain-Based Software Development System for Untrusted Environments, *in Proc. of the 11th International Conference on Cloud Computing*, San Francisco, CA, 2018, pp. 582–585.

[16] G. Wood, Ethereum: A secure decentralised generalised transaction ledger, *Ethereum Project Yellow Paper*, 2014.

[17] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends, *in Proc. of the IEEE 6th International Congress on Big Data*, Honolulu, USA, 2017, pp. 557–564.

[18] "Corda." https://www.corda.net/, 2019. [Online].

[19] "R3." https://www.r3.com/, 2019. [Online].

[20] "Monax." https://monax.io/, 2019. [Online].

[21] "Multichain." https://www.multichain.com/, 2019. [Online].

[22] "ETH Gas Station." https://ethgasstation.info/, 2019. [Online].

[23] "Ethereum." https://www.ethereum.org/, 2019. [Online].

[24] F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf, NIST Cloud Computing Reference Architecture, *NIST Special Publication*, vol. 500, p. 292, 2011.

[25] L. Li, Z. Jin, G. Li, L. Zheng, and Q. Wei, Modeling and analyzing the reliability and cost of service composition in the IoT: a probabilistic approach, *in Proc. of the IEEE 19th International Conference on Web Services*, Honolulu, HI, 2012, pp. 584–591.

[26] A. Afzal, B. Shafiq, S. Shamai, A. Elahraf, J. Vaidya, and N. R. Adaml, ASSEMBLE: Attribute, Structure and Semantics based Service Mapping Approach for Collaborative Business Process Development, *IEEE TRANSACTIONS ON SERVICES COMPUTING*, vol. 10, no. 2, 2017, pp. 1–14.

[27] "Ganache." https://github.com/trufflesuite/ganache, 2019. [Online].

[28] "Ropsten testnet pow chain." https://github.com/ethereum/ropsten, 2019. [Online].

[29] "Solidity." https://solidity.readthedocs.io/en/v0.5.3/, 2019. [Online].

[30] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K-K R. Choo, Blockchain: A Panacea for Healthcare Cloud-Based Data Security and Privacy?, *IEEE Cloud Computing*, vol. 5, no. 1, 2018, pp. 31–37.

[31] S. Kirkman, A Data Movement Policy Framework for Improving Trust in the Cloud Using Smart Contracts and Blockchains, *in Proc. of the IEEE International Conference on Cloud Engineering*, Orlando, FL, 2018, pp. 270–273.

[32] Q. Xia, E. B. Sifah, A. Smahi, S. Amofa, and X. Zhang, BBDS: Blockchain-Based Data Sharing for Electronic Medical Records in Cloud Environments, *Information*, vol. 8, no. 2, p. 44, 2017.

[33] H. Desai, K. Liu, M. Kantarcioglu, and L. Kagal, Enforceable Data Sharing Agreements Using Smart Contracts, *arXiv:1804.10645v1[cs.CY]*, 2018.

[34] S. Kirkman and R. Newman, A Cloud Data Movement Policy Architecture Based on Smart Contracts and the Ethereum Blockchain, *in Proc. of the IEEE International Conference on Cloud Engineering*, Orlando, FL, 2018, pp. 371–377.

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

[36] X. Zheng, R. R. Mukkamala, R. Vatrapu, and J. Ordieres-Mere, Blockchain-based Personal Health Data Sharing System Using Cloud Storage, *in Proc. of the 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*, Ostrava, Czech Republic, 2018.

[37] M. Corrales, P. Jurcys and G. Kousiouris, Smart Contracts and Smart Disclosure: Coding a GDPR Compliance Framework, *SSRN Electronic Journal*, 2018.

APPENDIX

This appendix provides the implementation of the smart contracts presented in Section IV.

```
contract Container_Submission{
  struct Status{
        bool  auth_access;
        bytes32  country;
        uint  age;
  }
  mapping (address => Status) statusTerms;
  ...
  function Access(address Provider) public returns(bool){
    var StatusTerm=statusTerms[Provider];
    Personal[Provider]=data;
    StatusTerm.auth_access=_auth;
    return (StatusTerm.auth_access);
  }
  function Profiling(address Provider) public returns(uint){
    var StatusTerm=statusTerms[Provider];
    StatusTerm.age=_age;
    return (StatusTerm.age);
  }
  function Transfer(address Provider) public returns(bytes32){
    var StatusTerm=statusTerms[Provider];
    StatusTerm.country=_country;
    return (StatusTerm.country);
  }
  function Logs(address Provider,bytes32 OP, uint degCompliance){
    ProOP[Provider].push(OP);
    deg[Provider].push(degCompliance);
  }
}
```

Fig. 7. Container_Submission smart contract

```
contract voting{
   event report(string title, address[] violator);
   function Conclude(address voter_1, address voter_2,...,address voter_n){
      Verification v_1=Verification(voter_1);
      Verification v_2=Verification(voter_2);
      ...
      Verification v_n=Verification(voter_n);
       if(v_1.vote()+v_2.vote()+...+v_n.vote()>= m){
         emit report("GDPR is not compliance for",v_i.violators());
       }
   }
}
```

Fig. 8. Voting smart contract