Blockchain and Compliance Verification

Group 3(Akshaya Mathur, Chen Zhou, Jinjin He, Wei Yu, and Xueyu Ni)

School of Computing, Newcastle University
Newcastle upon Tyne, United Kingdom

Abstract. In the information era, the amount of user information entering the network is increasing rapidly, and the protection of user data is becoming an important issue. The European Union has introduced the GDPR to govern what applications can do with user data. The GDPR restricts the access, storage, processing and transmission of personal data and requires explicit consent from user before performing these operations. In this report, we summarize previous works of blockchain-based models and architectures that aim to verify GDPR compliance for IoT and cloud services. Based on a summarized model, we discuss compliance verification of common regulations in the GDPR with a typical example. A blockchain-based architecture is then proposed, to show how the GDPR compliance verification based on the model, can be applied for IoT and cloud services.

Keywords: General Data Protection Regulation · Blockchain · Compliance Verification · User Privacy · Cloud Security · Smart Contracts · Internet of Things

1 Introduction

With the development of the Internet, increasing amount of data is being uploaded to the network and is accessed unrestricted by various software. Large amount of private information is transmitted and processed over the Internet without consent or encryption. It is becoming a substantial problem to protect data during usage, especially for sensitive data that needs higher level of security. To address the chaos and problems in this area, the General Data Protection Regulation (GDPR) is published by the European Union (EU), and becomes highly recognized. The GDPR is a strict regulation for privacy and security, and it imposes serious obligations onto organizations, so long as they target or collect data related to people in the EU.[8]

However, implementation is a gap between regulation and practice. To fill this gap, a model of the GDPR and a corresponding architecture are needed. The model should be able to express GDPR regulations, and the architecture should support verification of GDPR compliance in a secure and transparent way. Thus, blockchain is used in several papers to address related problems. Among the papers, [1] and [2] propose valuable models and architectures for GDPR compliance verification of IoT service and cloud service respectively.

Section 2 summarizes a model based on [1] and [2], by going through several common regulations in GDPR and describe their verification on a typical smartwatch application. Section 3 summarizes a blockchain-based architecture from [1] and [2] to support GDPR compliance verification for both IoT and cloud services. Section 4 concludes this article.

2 Data Protection Regulations and Compliance Verification

The GDPR data protection regulation requires system developers to show the purpose of using or processing performed on personal data in a transparent way. During the life cycle of a business process, a collection of design patterns is implemented to cover all activities undertaken on the user's personal data and the purpose of data processing by the actor. For example, the user steps, location information are used by the mobile application to track the user's health and provide location-based services (like weather). The mobile application hence needs to state the purpose of the process. The actor requires the data subject's (users) consent before using the personal data and processing it (according to the GDPR regulation).

Definition 1: Let PS. be a tuple of collection of business processes for system S represented as

$$P_S = (Act, P, A, D, D_h)$$

where,
Act is a set of actors,
P is a set of processes,
A is the set of activities such that $A = A_{op}$ (processing activities) $\bigcup A_{\bar{op}}$ (non-processing activities),
D is the data classes,
$D_h = Act \times Aop \times D \times P$ is a data handling relationship set mentioning which actor does what activity to what data for what process.
For the example in Fig.1
Act = {Walk record mobile application (MA), Walk record cloud service (CS)}
P = {walk record collection (COL), Walk record report (REP)}
A_op = {Read, write, transfer, profile}
$A\bar{op}$ = {display the step count, show the weather}
D = {user accelerate, user step count, user location}
D_h = {. . . , ¡MA, Read, user location, COL¿, . . . }

2.1 Privacy Policy

The purpose for collecting data subjects' personal data has to be clearly specified in the privacy policy. A privacy policy on D_h is denoted by $P_r(D_h)$ as the set of statements that the actor (act) executes a particular activity (α) on a data class (d) for a particular process (p), i.e., "act_i executes α on d for p". For the example in Fig.1, MA performs read activity on user location for COL.

When the privacy policy and data analysis is proposed to the user for consent, the vote (i.e., positive, or negative consent) of the data subject can be defined by a Boolean function as (*Definition 2*)

$$\Gamma_j : P_r(D_h) \longmapsto \{\top, \bot\}$$

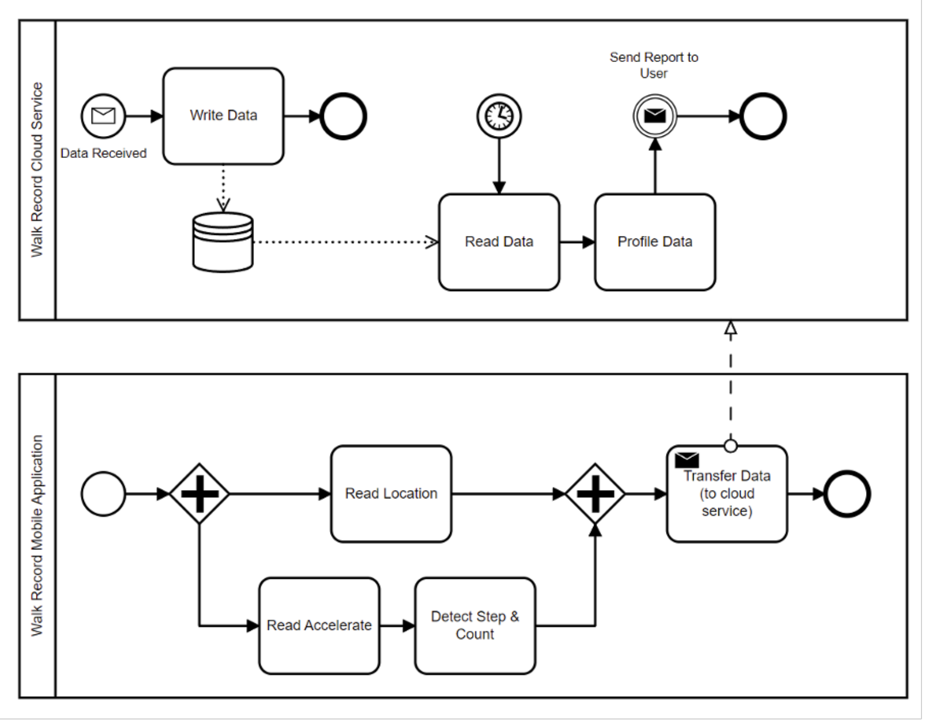


Fig. 1. Business processes of a fitness mobile application.

2.2 Data Subject (User) Consent

According to the GDPR rule (Recital 32 and Recital 43 of GDPR), the actors should obtain clear and free consent to process personal data of the data subject. The request must be clear, concise, and not disruptive to the use of service which must be provided in the privacy policy of the process collection. It should explicitly state the purpose for the collection of personal data. The consent can be given by written statement including electronic means (e.g., check box on a website) or oral statement [1].

A consent has to be given for a particular actor(act_i), activity (α), data class(d) and process (p), i.e.,

$$p_r(\gamma) \in P_r(D_h)$$

where $\gamma =< act_i, \alpha, d, p >\in D_h$.

For example in Fig.1, the user should have given the mobile app (actor) the consent to read his location data for the process of collecting the user's walk record.

To verify the process collection of the system and the data subject consent. *Definition 3:* Let $D'_h \subseteq D_h$ be a processed data relation set such that each relation determines what data was used by which processes and what processing activities were executed by the actor [1].

D'_h should satisfy $p_r(\gamma) \neq \emptyset$ and $\Gamma_j(p_r(\gamma)) = \top$ for each $\gamma \in D'_h$, otherwise there is a breach of the user consent rule. This rule means that a privacy policy is required and an user consent is required, for each data processing respectively.

Since D'_h is the data process happened, it can only be collected at run-time. Section 3 describes an architecture to record such info.

2.3 Data Minimization

The GDPR requires data controllers to limit the collection of personal data to the necessary data items, i.e., a set of operations must exist for any collected data to be processed (Art. 5(1)(c) of GDPR).[1]

To verify this regulation, we must first define the used data and the collected data, and then compare them.

Definition 4: Let P_S = (Act, P, A, D, D_h) be business processes within a system S. The used data set is denoted by D_u, where:

$$D_u = \{d \in D| < act, \alpha, d, p >\in D_h\}[1]$$

Definition 5: Let P_S = (Act, P, A, D, D_h) be business processes within a system S. $D_u \subseteq D$ be the set of used data. $D_r \subseteq D$ be the set of data coming from a user. When $D_r \subseteq D_u$, the data minimization rule of the GDPR is satisfied.

These definitions mean that, if the business process collects user data that will not be processed, it violates the data minimization requirement of the GDPR, holding more data than is necessary for its purpose.

For example in Fig.1, the smartwatch app should only collect and upload a user's walk record, but collects information that does not need to be processed, such as the user's nationality; otherwise it does not meet the data minimization regulation of the GDPR.

Also, this definition can be used in $D_r \subset D_u$ to express the situation where user data is generated and processed without the user's consent.[1] If a smartwatch app generates unique identity of user and use it without user consent, this data is not considered received from user, and thus $D_r \subset D_u$.

The models in both [1] and [2] share part of their data handling relation, which is "$Act \times Aop \times D$", and the data minimization verification proposed in [1] is based only on this part of the model by comparing D_r with D_u. Thus, by introducing D_r, the same verification can be applied to the model in paper2 to validate the GDPR's data minimization rules.

2.4 Data Protection

The term data protection can refer to the whole GDPR, but in this part, we focus on operation-level requirements of data protection, contrary to Data Subject Consent and Data Minimization, which restrict the procedure of data usage and the system-level data retrieval.

One of the requirements is data encryption. Data controllers, such as commercial companies developing apps or websites, are required by the GDPR to encrypt data before processing it. Data encryption reduces the risk of personal data breaches, and, for commercial companies, reduces the risk of penalties due to data breaches.

Although the GDPR does not prescribe a specific method of data encryption. Data encryption is a process of converting plain text into a hashed code using a key [4]. After encryption, decryption is only possible with a specific key. In this way, the risk of data leakage is greatly reduced. Encrypted data is not necessarily considered to be compromised even if the storage medium on which it is stored is lost.

Other requirements include user authentication, restriction to data storage time, support for data erasure, etc. We will focus on encryption here as an example, but the model we use should fit into general requirements.

Both [1] and [2] propose definitions to check GDPR compliance by the status of these data protection requirements, at the operation level. However, the definition in [2] is more generalized for any operation level requirements, so we will adopt the definition in [2].

Definition 6: Let α be an operation and $El = \{el_1, ..., el_n\}$ be a set of GDPR-concern elements of α such that $el_i \in El$ refers to an element concerned for α. The following Boolean-valued function is defined for the operation α to show its GDPR compliance status:

$$G_\alpha : \times_1^n V_i \longmapsto \{\top, \bot\}$$

where $V_i = dom(el_i)$ is the set of values associated with the domain of el_i . The operation α is GDPR compliant, if $G_\alpha(v_1, ..., v_n) = \top$, where $v_i \in V_i$ [2].

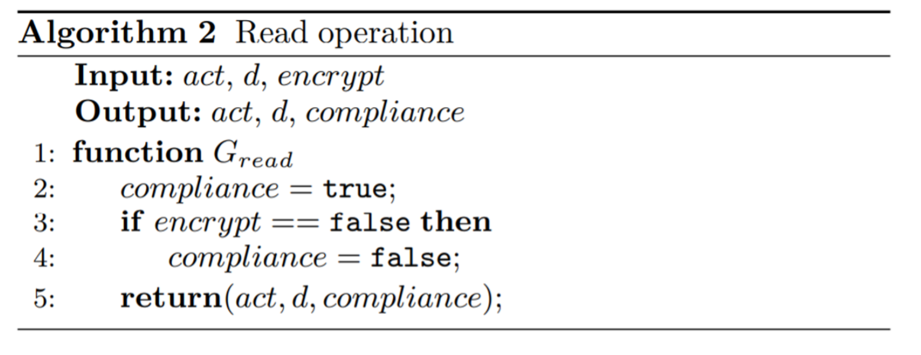
The G_α result is based on α and the GDPR-concerned elements, without act, d, or p in the model, as it extends the model to capture a different part of GDPR.

With this definition, the encryption requirement of a read operation in GDPR can be expressed as G_{read}(encrypt)=$\top$, only when encrypt is true, where encrypt means if encryption measure is taken or not. If there are other requirements in GDPR on a read operation, those concerns can be added to the input of G_{read}. Similarly, requirements on other operations can also be expressed with this definition. Algorithm 2 in [2] is an implementation of G_{read},. For a specific read operation by act on d, only when encrypt is true, the compliance in the result is true, regardless of act and d.

For the read operations in the smartwatch example, we can express their GDPR compliance with:

$$\forall < act, read, d, p > \in D_h, G_{read}(encrypt) = \top$$

Fig. 2. Algorithm 2 in [2]



Algorithm 2 Read operation

```
Input: $act$, $d$, $encrypt$
Output: $act$, $d$, $compliance$
function $G_{read}$
    $compliance$ = true;
    if $encrypt$ == false then
        $compliance$ = false;
    return($act, d, compliance$);
```

Since any $< act, \alpha, d, p >$ can have a GDPR compliance status determined by G_α , we can extend *Definition 1* and P_r.

Definition 1.1: ..., $D_h = Act \times Aop \times D \times P \times \{\top, \bot\}$, where $\{\top, \bot\}$ is the GDPR compliance status determined by G_α.

$P_r(D_h)$ will also provide the GDPR compliance status in the private policy.

2.5 Data Transfer

For data transfer inside the EU, the GDPR has no additional rules, as any receiver is under GDPR control. For data transfer to non-EU places, the GDPR lists detailed situations when a transfer is acceptable. For example, if the third country holds BCR certification, or if the receiver has additional safeguards.

[1] and [2] address different situations. [1] considers if the receiver outside the EU has sufficient protection (e.g. encryption) to user data. [2] on the contrary, checks if the receiver outside the EU is in countries holding BCR certification. We can express both points with the definition of GDPR-concerned elements.

Definition 7: Let $P_S = (Act, P, A, D, D_h)$ be business processes in system S and $A_{op} \subseteq$ A be the set of processing activities.

One situation [1] is that the encryption status of all operations of the actor *receiver* is true:

$$encrypt = \top, for \forall < receiver, \alpha, d, p > \in D_h$$

having *encrypt* as a GDPR-concerned element for its operation α.

Another situation [2] is that the *receiver* is in the EU or a BCR certified country: $loc \in EU$ or $loc \in BCR$, where *loc* is a GDPR-concerned element, the location of the receiver; *EU* is a set of EU countries; *BCR* is a set of BCR certified countries.

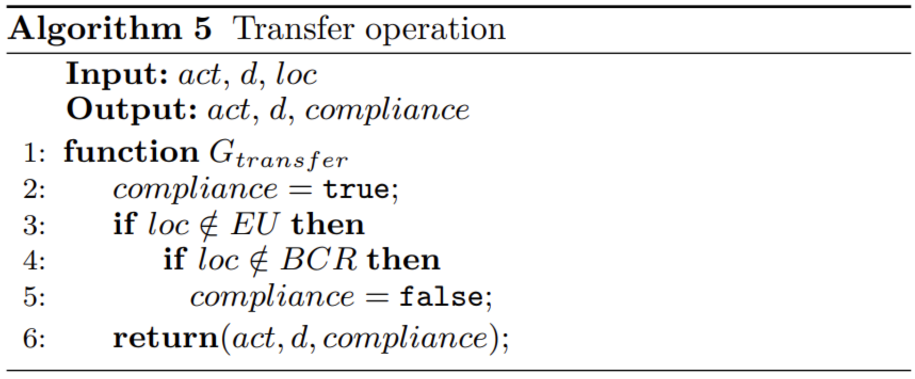
Thus, for a transfer from the actor *sender* to *receiver*, the GDPR compliance status of this transfer, based on Definition 6, is

$$G_{transfer}(encrypt, loc, encrypt_{receiver})$$

where *encrypt* is the encryption status of this transfer and the *loc* is the location of the *receiver* and $encrypt_{receiver}$ is the encryption status of all operations of the *receiver*.

If *loc* is in the EU or a country holding BCR certification, or if both *encrypt* and $encrypt_{receiver}$ are true, the transfer is GDPR compliant.

In practice, it is hard to summarize the encryption status of all operations of a receiver, especially if the receiver is external, and if we focus on *loc*, we will have Algorithm 5 in [2]:



Algorithm 5 Transfer operation

```
Input: $act$, $d$, $loc$
Output: $act$, $d$, $compliance$
function $G_{transfer}$
    $compliance = \texttt{true}$;
    if $loc \notin EU$ then
        if $loc \notin BCR$ then
            $compliance = \texttt{false}$;
    return($act, d, compliance$);
```

This algorithm is an implementation of $G_{transfer}$, where d is the transferred user data, and *loc* is the country of the receiver. The compliance in the result is true only when the receiver is in the EU or BCR certificated countries.

If we apply this algorithm to the smartwatch example and assume the cloud service is outside the EU, we can check the GDPR compliance of its transfer operation with $G_{transfer}$ (MA, user location and step data, location of the receiver). The compliance in the result is true only when the non-EU cloud service is in a BCR country. Comparing the model and verification proposed by the two papers, we can see that they cover different cases of the GDPR for the transfer operation. The model and verification in [1] can be extended by generalizing its ϵ_{act} function (Definition 7 in [1]) to include factors like "inside BCR country"; verification in [2] can express the same verification in [1] by adding safeguard measures like encryption to its GDPR-concerned elements in the $G_{transfer}$ function. We can see that the generalization will eventually make both verification methods equivalent to a G_α function with all necessary GDPR-concerned elements.

2.6 Data profiling

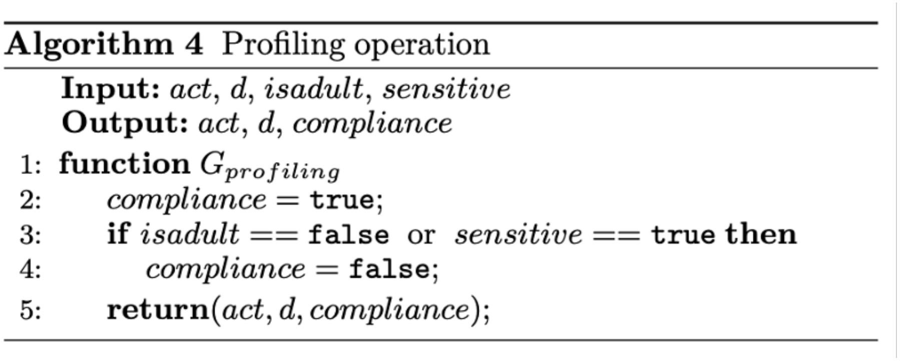
Data profiling includes "feature analysis", and any form of automatic processing for evaluating personal data related to natural persons, especially analysing, or predicting aspects related to the work performance, economic status, health, personal preferences or interests, reliability or behaviour, location, or movement of the data subject. It may have a legal effect related to the data subject or has a similarly significant impact.

Art 22 of the GDPR requires pre-confirmed of any automated profiling operations on customers under the age of 18 or whose per person data falls into a sensitive data category. Thus, [2] proposes 2 legal questions. The first legal question concerns profiling operations on personal data of underage customers. The second asks participants whether their services involve the profiling of sensitive data. Sensitive data defined by GDPR includes information such as religious or political beliefs, genetic data, biometric data and health-related data.

For example in Fig.1, a smart watch app may need to collect user location and step data for analyzing, to generate reports for the user, and it must notify user that their data needs to be profiled, and the user can decide whether the data can be analyzed.

With these 2 concerns, [2] implements $G_{profiling}$ to verify them
Let act $\in$ ACT and d $\subseteq$ D be, respectively, actor address and the personal data that will be processed. Let *sensitive* and *isadult* be the GDPR-concern elements of a profiling operation, that have Boolean values. The value of *sensitive* shows whether sensitive data will be profiled. The value of *isadult* indicates whether the actor performs profiling operation only on adult customers or not (e.g., its "true" value denotes the data profiling of adults). Algorithm 4 is an implementation of Gprofiling in [2]. A profiling operation violates GDPR rule (Art. 22) if sensitive is "true" or isadult is "false".

Fig. 3. Algorithm 4 in [2]



Algorithm 4 Profiling operation

```
Input: $act$, $d$, $isadult$, $sensitive$
Output: $act$, $d$, $compliance$
function $G_{profiling}$
    $compliance = $ true;
    if $isadult ==$ false or $sensitive ==$ true then
        $compliance = $ false;
    return($act, d, compliance$);
```

3 Blockchain-Based Architecture for Compliance Verification

Fig. 4 shows a blockchain-based architecture that supports GDPR compliance checking for cloud and edge applications. The architecture can record user data

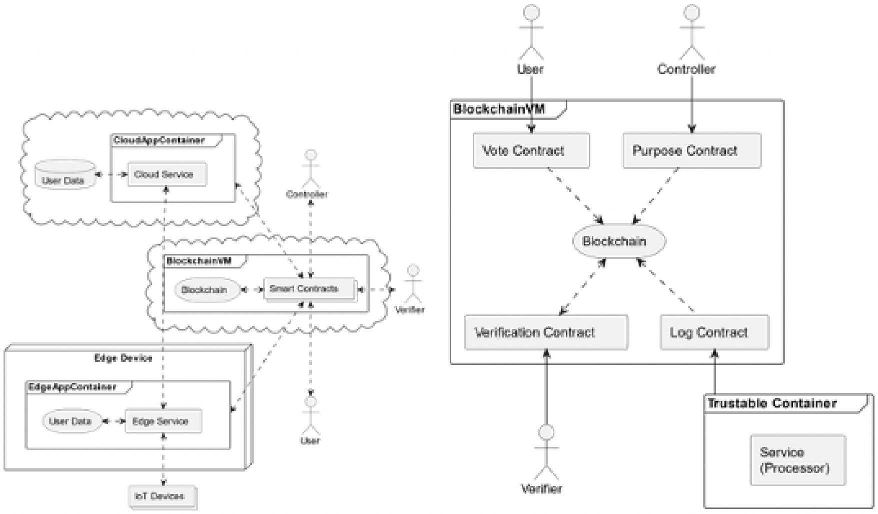


Fig. 4. Blockchain-Based Architecture

operations done by the actors to a blockchain. It can also verify if the operations recorded are compliant with GDPR regulations. The core component is a blockchain-based virtual machine hosting smart contracts. It interacts with the controller, processor, data subject and verifier. Other components are external but related, including trustable containers and IoT devices. An optional component can be added to manage the smart contract deployment, but for simplicity, we will only briefly mention it.

A blockchain-based virtual machine can hide the underlying blockchain infrastructure and provide an environment to deploy smart contracts. Because of this convenient technology, we can focus on the smart contract without looking into the details of blockchain.

The trustable container, introduced in [5], is a prerequisite of our architecture. In both edge and cloud environments, trustable containers can collect the operations of apps on user data and prevent or detect malicious interferes. This technology together with other container management measures in [6] and [7], allows us to collect non-tampered operation logs. IoT devices are related but not part of the architecture. They are one of the sources of user data and are used by edge services.

Inside the blockchain-based virtual machine, there will be smart contracts deployed:

1. Purpose contract, which checks GDPR-concerned elements of a purpose and records it to the blockchain.
2. Vote contract, which receives data subject's vote of each purpose and records it to the blockchain.

3. Log contract, which receives operation log from containers (the apps inside are processors) and records the log to the blockchain.
4. Verification contract, which reads records in the blockchain and verifies GDPR compliance, when activated by a verifier.

Optionally, a contract manager can be added to support more flexible smart contract deployment, like the contract factory and contract activator in [2]. For example, when a contract is updated, further activation of the contract will be on a new instance of the contract.

3.1 Architecture: Compliance Checking Phases

For checking the GDPR compliance we can divide the proposed architecture in three phases: ratification, submission and verification.

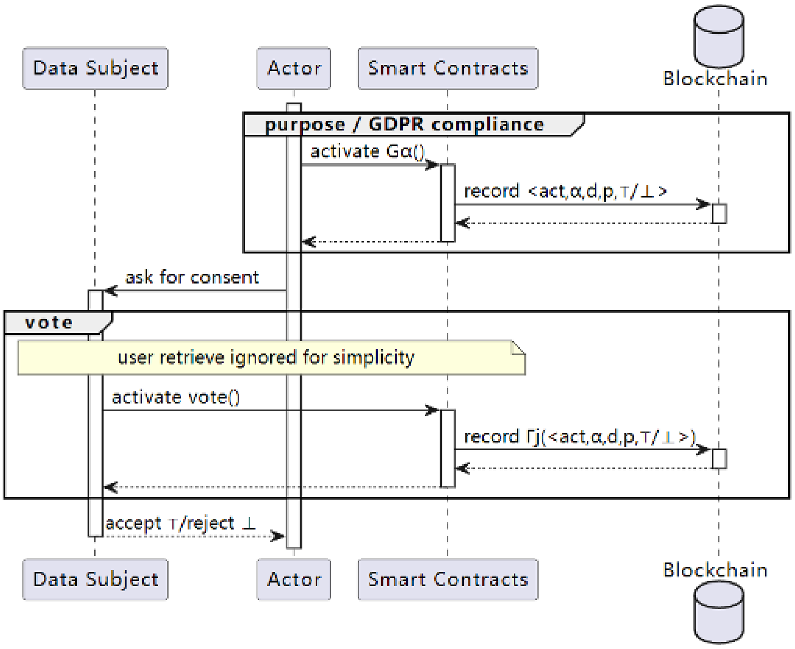


Fig. 5. Ratification Phase

Ratification Phase The main purpose of this phase is to fulfil the GDPR regulation of obtaining user consent, i.e. business process generation purposes and requesting data from the user. The steps are shown in Fig.5 There are two contracts here: the first records the data process relation, and the second records the consent of the data subject. They are implemented according to *definition 1.1* and definition 2 respectively.

First, the purpose contract is activated by controller or processor with the address of the participant (e.g., an ethereum account), the participant's activity on the user's data, the category of personal data, the name of the business process (purpose), and a series of GDPR-concerned elements.
These inputs corresponds to $D_h \subseteq \mathrm{ACT} \times \mathrm{Aop} \times \mathrm{D} \times \mathrm{P} \times \{\top, \bot\}$ in *Definition 1.1*, except that the GDPR compliance status is determined according to the values of GDPR-concerned elements provided by controller or actor (according to function G_α in *Definition 6*).
Then, the $< act, \alpha, d, p, \top/\bot > \in D_h$ is recorded to the blockchain.

After the data usage relation is recorded, the actor or contract can notify the user with privacy policy translated from the data usage relation, according to the function $P_r(D_h)$. We will ignore such contract/function (e.g. retrieve() in [2]) for simplicity.

Then, the user makes decision and activate the vote contract. The vote contract will record the user consent ($\mathit{\Gamma}_j(< act, \alpha, d, p, \top/\bot >)$) to the blockchain, as in *Definition 2*. In practice, the recorded data can either copy the data usage relation or refer to it. After the user vote is recorded, the result will be informed to the actor.

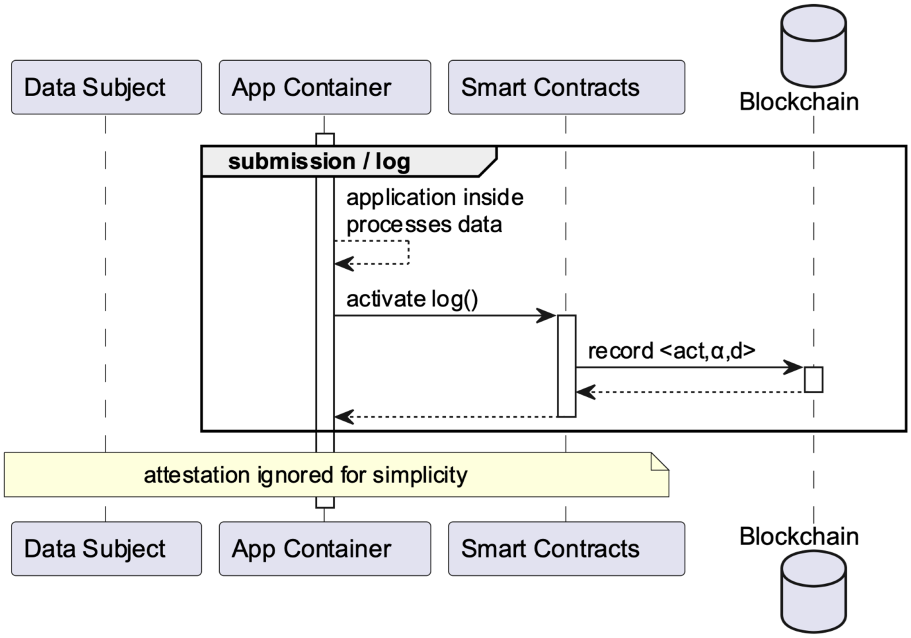


Fig. 6. Submission Phase

Submission Phase The submission phase is to record operations on user data during the execution of actors. The steps are shown in Fig.6

While the actors are processing data, the trustable containers in both edge and cloud environment, can capture the operations of the actors on user data, as $< act, \alpha, d >$. The containers then activate the log contract with $< act, \alpha, d >$, and the log contract will record the $< act, \alpha, d >$ to the blockchain.

An attestation contract can be added to notify user about data usage and transfer, but is ignored here for simplicity.

As an optimization, if two operations are of the same type, and are processed sequentially, and the data set processed by the second operation is a subset of the data set processed by the first operation, then the later operation is defined as data neutral, which indicates its log can be ignored, to avoid wasting resources. [2]

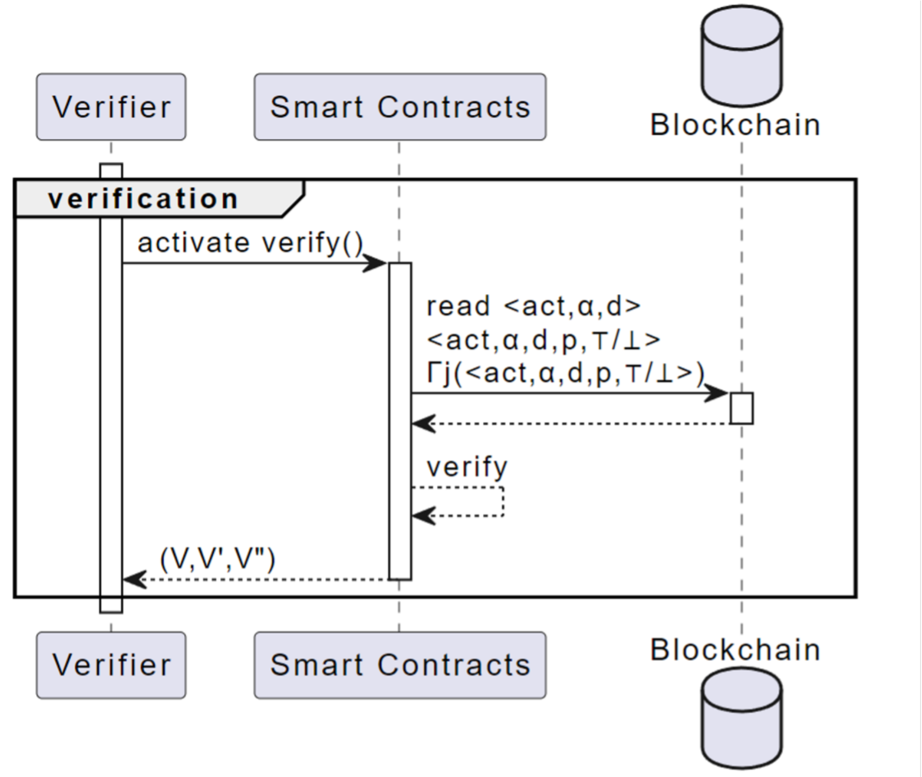


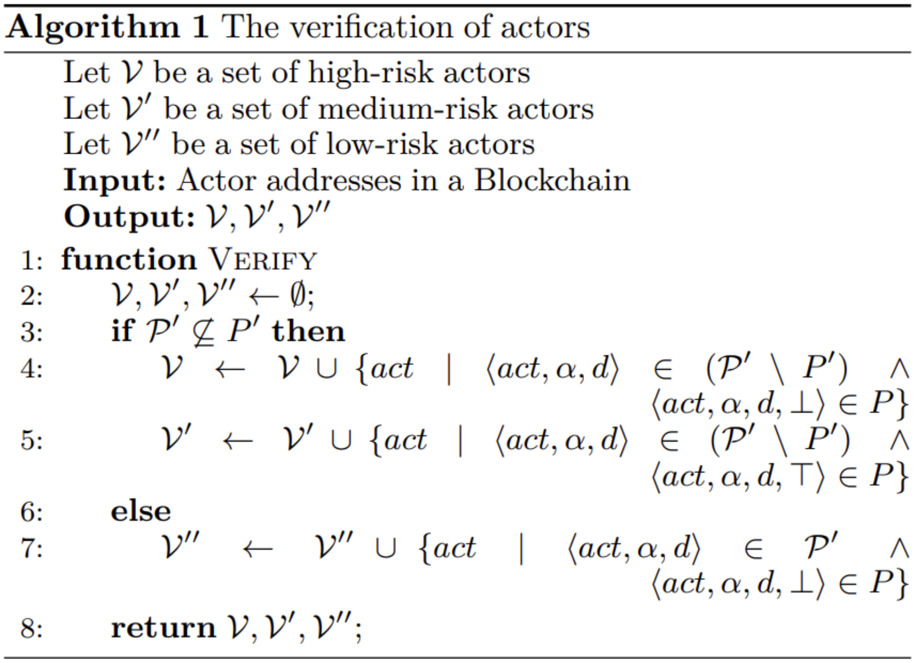
Fig. 7. Verification Phase

Verification Phase In this phase, the verifier will verify that the data usage of actors complies with the GDPR. The steps are shown in Fig.7

First, the verifier activates the verification contract. Then, the contract read all necessary data in the blockchain, that is recorded via the purpose, vote, and log contract.

The GDPR rules are then verified through the algorithm in Fig.8 proposed by [2]. The algorithm identifies actors with high-, medium-, low- risks, according to their operations: high risk means not GDPR compliant and without user consent; medium risk means GDPR compliant but without user consent; low risk means not GDPR compliant but with user consent.

In the end, the contract records the verification result to the blockchain and sends the result to the verifier as well.



Algorithm 1 The verification of actors

```
Let $\mathcal{V}$ be a set of high-risk actors
Let $\mathcal{V}'$ be a set of medium-risk actors
Let $\mathcal{V}''$ be a set of low-risk actors
**Input:** Actor addresses in a Blockchain
**Output:** $\mathcal{V}, \mathcal{V}', \mathcal{V}''$
function VERIFY
    $\mathcal{V}, \mathcal{V}', \mathcal{V}'' \leftarrow \emptyset$;
    if $\mathcal{P}' \not\subseteq P'$ then
        $\mathcal{V} \leftarrow \mathcal{V} \cup \{act \mid \langle act, \alpha, d \rangle \in (\mathcal{P}' \setminus P') \wedge \langle act, \alpha, d, \bot \rangle \in P\}$
        $\mathcal{V}' \leftarrow \mathcal{V}' \cup \{act \mid \langle act, \alpha, d \rangle \in (\mathcal{P}' \setminus P') \wedge \langle act, \alpha, d, \top \rangle \in P\}$
    else
        $\mathcal{V}'' \leftarrow \mathcal{V}'' \cup \{act \mid \langle act, \alpha, d \rangle \in \mathcal{P}' \wedge \langle act, \alpha, d, \bot \rangle \in P\}$
    return $\mathcal{V}, \mathcal{V}', \mathcal{V}''$;
```

Fig. 8. Verify Algorithm in [2]

3.2 Blockchain-based Model applied to Walk Record Mobile App

For our example walk record business in Fig.1, according to *Definition 1.1*, we have:
Act = {Walk record mobile application (MA), Walk record cloud service (CS)}
P = {walk record collection (COL), Walk record report (REP)}
A_{op} = {Read, write, transfer, profile}
D = {user accelerate, user step count, user location}

D_h={
$< MA, read, location, COL, \top >$,
$< MA, read, accelerate, COL, \top >$,
$< MA, transfer, location\ and\ step\ count, REP, \top >$,
$< CS, read, location\ and\ step\ count, REP, \top >$,
$< CS, write, location\ and\ step\ count, REP, \top >$,
$< CS, profile, location\ and\ step\ count, REP, \top >$
}
For simplicity, we assume that all GDPR-concerned elements related to the operations are satisfied, and all data usage can get positive user consent.

First, the app/service provider registers its data usage through the purpose contract, by providing corresponding $< act, \alpha, d, p >$ as well as GDPR-concerned elements related to these data usage. For example, the ¡MA, read, location, COL¿ requires data encryption, and it will be passed to the purpose contract along with its encryption status.

The purpose contract in the Blockchain-based virtual machine executes and decides GDPR compliance status for each data usage based on the corresponding Ga function. Then, the purpose contract records each data usage along with its GDPR compliance status to the blockchain. For example, $< MA, read,$location, COL,$\top >$ is recorded .

The recorded $< MA, read, location, COL, \top >$can be translated into plain English statements through Pr(), and be presented to the user of the app/service. This step can be done by another contract for querying and notification but is ignored here for simplicity.

The user then accepts $< MA, read, location, COL, \top >$ by activating the vote contract. The vote contract records the decision to the blockchain, as $< MA, read, location, COL, \top, \top >$ or $< ID, \top >$ where ID is a reference to $< MA, read, location, MA, \top >$.

After all required data usage relations are voted positive, the app/service can operate on user data - read location and accelerate from sensors, transfer data to its cloud service, store data, and do analysis. During execution, the operations are logged by the trustable containers holding the mobile app and cloud service. The containers activate the log contract with $< act, \alpha, d >$ for each operation, and the contract will record $< act, \alpha, d >$ to the blockchain.

The recorded operations may be sent to the user through another contract, but we ignore it here for simplicity.

In the end, a verifier may invoke the verification contract. The contract reads the data usage relation registered, the votes user submitted, and the logs the containers uploaded. It analyzes the recorded data with the verify algorithm in Fig.8 proposed by [2]. Then it records the result to blockchain and sends the result to the verifier as well.

3.3 Trust and Scalability

For IoT scenario, there are usually numerous devices exchanging and processing data (especially without human intervention in IoT devices), so they need to recognize and authenticate each other. Due to the size and other features of IoT, we need a decentralized authentication system as it would be impossible to create an efficient centralized authentication system. A decentralized method called bubble of trust is proposed in [9] to enhance the trust of the IoT users and smart nodes. It ensures a robust identification and authentication along with data integrity and availability of devices by building a virtual secure zone (bubbles) in IoT environment, where nodes trust each other. It relies on the security advantages provided by blockchain. A zone is a collection of devices that has a leader managing the zone. The leader is responsible for registering followers identities so that when any node in a zone requests access to the user data or requests the processing purpose of another node, it must already be registered by the leader. For an architecture verifying GDPR compliance for IoT actors, every zone can be equipped with a verifier that can use transaction log to identify GDPR violations or verify the data stored by the leader and followers.

For the blockchain-based verification architecture to achieve scalability, we can use multi-blockchain framework such as Cosmos [9] and Polkadot [10]. Cosmos is a decentralized network of independent parallel blockchain which supports distributed ledgers. Each network keeps the transactions done by a leader and followers on data in a certain zone. Cosmos framework provides inter-Blockchain protocol used for interaction between the networks for transferring personal data between different zones.

On the other hand Polkadot uses it's components called parachain and relay-chain to supply the communication and connection of various heterogeneous blockchain networks. For storing the transactions of nodes in different zones, Parachain can use blockchain. Coordination between parachain is handled by relay-chain. The leader of zones can act as a collators responsible for confirming the transactions created from registered valid nodes prior to sending them to validators, leading to the confirmation of new blocks [1].

4 Conclusion

This report summarises blockchain-based architecture for GDPR compliance verification for both IoT services [1] and cloud services in [2]. We first summarize a model that can describe common rules of GDPR, including user consent, data protection, data minimization, data transfer, and data profiling, explained with an example of the model applied to a typical smart watch app. In the second part, we summarize the blockchain-based architecture to support this model, and discuss the workflow and the smart contracts in this architecture, based on the same smartwatch app example. Finally, we discuss several technologies to support trust and scalability, including Bubble of trust and Cosmos and Polkadot.

The models proposed in [1] and [2] have a lot in common, and their more general parts can be merged into a single model. The core parts of their proposed architectures are similar, and can also be merged into one architecture, to support both IoT and cloud services, as long as trustable container technology is applied. With the foundation of [1] and [2], it is practical to implement blockchain-based system to support GDPR compliance verification.

References

1. M. Barati, O. Rana, I. Petri and G. Theodorakopoulous: GDPR compliance verification in Internet of things. IEEE Access,(vol. 8), pp.119697–119709 (2020)
2. M. Barati and O. Rana, Tracking GDPR compliance in cloud-based service delivery, IEEE Transactions on Services Computing, 2020, DOI: 10.1109/TSC.2020.2999559
3. GDPR Recitals: https://gdpr-info.eu/recitals/
4. GDPR Recitals: https://gdpr-info.eu/issues/encryption/
5. T. Al Said, O. F. Rana, and P. Burnap: VMInformant: An instrumented virtual machine to support trustworthy cloud computing. Int. J. High Perform. Comput. Netw., vol. 8, no. 3, pp. 222–234, (2015)
6. N. E. Ioini and C. Pahl: Trustworthy orchestration of container based edge computing using permissioned blockchain" in Proc. 5th Int. Conf. Internet Things, Syst., Manage. Secur., Oct. 2018, pp. 147–154.
7. E. Casalicchio and S. Iannucci, "The state-of-the-art in container technologies: Application, orchestration and security," Concurrency Comput., Pract. Exper., p. e5668, Jan. 2020, doi: 10.1002/cpe.5668.
8. What is GDPR, the EU's new data protection law?
   Available at: https://gdpr.eu/what-is-gdpr (Accessed: March 2022)
9. Cosmos Network: Internet of Blockchains. Accessed: Jun. 10, 2020. [Online]. Available: https://cosmos.network/
10. G. Wood, "Polkadot: Vision for a heterogeneous multi-chain frame-work," White Paper, 2017. [Online]. Available: https://polkadot.network/ PolkaDotPaper.pdf