Lightweight Blockchain-based Platform for GDPR-Compliant Personal Data Management

Cristòfol Daudén-Esmel, Jordi Castellà-Roca, Alexandre Viejo and Josep Domingo-Ferrer
Universitat Rovira i Virgili
Departament d'Enginyeria Informàtica i Matemàtiques
UNESCO Chair in Data Privacy, CYBERCAT-Center for Cybersecurity Research of Catalonia
Av. Països Catalans 26, E-43007 Tarragona, Catalonia
{cristofol.dauden,jordi.castella,alexandre.viejo,josep.domingo}@urv.cat

***Abstract*—New digital technologies generate large amounts of information. This data is processed by Service Providers in order to improve and develop new services or products, but also to fund themselves. However, processing these personal data can result in the extraction of sensitive information. In consequence, it can lead to users' privacy risk. To mitigate this risk, the EU elaborated the GDPR. It forces Service Providers to have Data Subjects' explicit consent for collecting and processing their personal data. The problem is that legislative text does not define how to transparently demonstrate that they already have this consent. Also, most users do not know the rights they have over their personal data, neither this regulation provides techniques for them to be aware about what happens with it. In this paper, we propose a lightweight blockchain-based GDPR-compliant personal data management platform. It provides public access to immutable evidences that show the agreements between the Data Subjects and Service Providers. The Service Providers can demonstrate that they are fulfilling the regulation, and Data Subjects are aware about what happens with their personal data and can manage it according to their rights.**

***Index Terms*—General Data Protection Regulation (GDPR), personal data management, smart contracts**

I. Introduction

The rapid advancement and development of new digital technologies has changed the dynamics of our daily lives by providing us with new services and products. Among the services, we can stress the social connectivity, information storage and location (GSP and mapping). Regarding the products, Smartwatches and Smart Home Devices with an Intelligent Personal Assistant (IPA) are two examples that are becoming more popular worldwide. Smartwatches track a set of biomedical data which can be useful in medical treatments and health control. Users can improve the energy consumption or remotely control their houses (lights, room temperature, security, etc.) through their IPAs. These services and devices generate huge amounts of information, which is processed by the Service Providers (SPs) in order to improve and develop new services. We cannot however ignore that also represents an important source of revenue for the SPs. As a result, their products can be cheaper or even free [1], [2].

The processing of the aforementioned information may result in extraction of sensitive information which can jeopardise the users' privacy. Thus, the EU took a decisive step with the General Data Protection Regulation (GDPR) [3], that came into effect from May 2018, in order to protect users' rights. The GDPR wants to mitigate the abuse of massive collection and processing of users' personal data (PD). The regulation guarantees specific privacy rights to Data Subjects (physical or legal entities to which the personal data belongs) ensuring that personal data "can only be gathered legally, under strict conditions, for a legitimate purpose"; as well as bringing full control back to the data owners.

Under GDPR, companies are required to prove compliance in case of suspicion of a violation or when a Data Subject (DS) lodges a complaint with the Supervisory Authority (SA). However, the legislative text does not specify how to transparently demonstrate that the information collected and its processing fulfills with the regulation. In the same way, DSs need tools to know and control what happens with their Personal Data. The problem here is that individuals have no tools to know transparently and easily which data is being collected and processed and for which purposes. As a result, DSs are mostly limited to giving their consent beforehand, in a way that is based on an abstract clause. In this regard, due the current centralised GDPR-compliance verification architecture, critical concerns on the lack of transparency have been imposed [4]. Consequently, we need a framework to enable the agreement verification between the users (DS), data controllers (DC) and data processors (DP) in relation to the data custody and processing. At the same time, the users should be capable to know and control which data is being collected, who is processing it and for which purposes. From the DS point of view, the main benefit is a way to manage his personal data, which does not depend on the DC, i.e. the tool can be used to manage all agreements with SPs. In addition, from a DC and DP point of view, the main benefit is a proof that can be presented to SAs showing that data was obtained and processed in a GDPR compliant way. So, the proof should have the following properties: i) public access; ii) verifiable; iii) authentic; iv) immutable and; v) non-repudiable. According to these properties, some authors have proposed the use of Smart Contracts (SCs) implemented over the blockchain technology (BC) as a general-purpose data management and storage [5]–[12]. This is a promising measure in GDPR-compliant personal data management thanks to its properties.

In this article, we propose a lightweight blockchain-based

GDPR-compliant personal data management system, which provides public access immutable evidences showing the agreements between a Data Subject and a Service Provider about DS's personal data.

The rest of the paper is organised as follows. First, we provide a concise description of related work (sec. 2). In section 3 we introduce and describe our proposed design. Sec. 4, provides the analysis and discussion about the platform. Last section (sec. 5) summarizes the conclusions and future work.

II. Related Work

The aforementioned properties are similar to the ones required in other systems concerning secure data sharing, data accountability, health records management, etc. In this line, Linn et al. [5] propose the use of a public BC to register the accesses to the EHRs. The health BC carries out the access control function, while the medical records itself are stored off-BC in secure data custodians. In a similar way, Azaria et al. [6] present MedRec, a decentralized record management system to give patients control over their Electronic Medical Records (EMRs) through the use of Ethereum BC and SCs. However, these works only provide a high level description without any kind of implementation or evaluation of their systems.

Neisse et al. [7] introduce two BC-based approaches for data accountability and provenance tracking. They rely on the use of publicly auditable SCs that increase the transparency with respect to the access and usage of personal data, which is stored locally. The first approach allows the flexible configuration of polices in a single contract, but it is unfeasible due to the limited amount of storage and processing capabilities of SCs. On the other hand, the second approach solves the previous issues, but has a lower level of customization, since subjects are left to choose between a limited number of contract options.

Other works in the same research line, focus on the case where DS wants to share his personal data in an integer and secure way. Wang et al. [8] designed a framework that combines the decentralized storage system interplanetary file system (IPFS), the Ethereum BC, and Attribute Based Encryption (ABE) technology. The main limitation of this system is data owners are responsible for all required tasks and the scheme does not implement the functions of user's attribute revocation and access policy update. Similarly, Chowdhury et al. present in [9] a data sharing framework that guarantees the authenticity of the shared data in real-time and provides transactional privacy in a BC network. In this case, the BC guarantees the data authenticity and integrity while personal data is stored using Personal Data Storage (PDS) technology.

Table I
Notations Table

Notation	Description
BC	Blockchain Technology
DC	Data Controller
DP	Data Processor
DR	Data Recipient
DS	Data Subject
GDPR	General Data Protection Regulation
SA	Supervisory Authority
SC	Smart Contract
SP	Service Provider

After the GDPR coming into effect in 2018, different works addressed the processing of personal data according to the new regulation using BC technology. In [10], Wirth and Kolain introduce an architectural blueprint that embodies the legal concept of the DS's consent in a BC-enabled and GDPR-compliant manner. The fact of representing the consent of the DS in a smart contract ecosystem, makes the processing of personal data a question of control rather than trust. Nevertheless, their proposal stores DS's personal data on the BC, which despite being encrypted, it becomes permanent, in such a way that some entity in the future can decrypt this data leading to privacy issues and not fulfilling the Right to Erasure principle. Faber et al. [11] designed a GDPR-compliant personal data and identity management system based on the BC technology (BPDIMS - BC-based Personal Data and Identity Management System). Their proposal consists of holding the consent given by the DS to a DC on a SC, while using an off-chain repository to store this user's personal data. However, they only present the conceptual approach, (evaluation) technical details on platform were missed out; neither do they propose any design for the GUI. Finally, in [12] Truong et al. developed a GDPR-compliant personal data management platform leveraging the BC and SC technologies. Their platform allows data owners to impose data usage consent, ensures only designated parties can process personal data, and logs all data activities in an immutable distributed ledger using SC and cryptography techniques. The BC issues an access token showing that a SP has been granted to access a particular dataset. This token will be validated by a resource server (who holds that dataset) before returning the requested data. So, all personal data is stored off-chain for better scalability and higher efficiency. They do not also differentiate between the data collection consent and the data processing consent: it could be the case on which data subject allows the processing of his personal data for certain purposes but only consents the processing of part of his personal data for others. Moreover, they store a Hash of personal data in the distributed ledger, which must be updated when the data is modified. This implies a significant processing overhead when the data is dynamic, for example the query logs.

When compared to the existing research, our work proposes a new conceptual design and system architecture for human-centric personal data and identity management, by using BC and SCs technology that is in compliance with the GDPR. Our work differentiates between the data collection and data processing concepts by identifying the Controller and Processor actors and treating them in a related but separate way. We also try to reduce the overhead on data owners, as if they need to have a wide knowledge on BC technology or they have to be constantly operating over our platform, it will be hard to be accepted and used by the community.

III. Proposal Description

In this section, we describe our proposal for a blockchain-based GDPR-compliant personal data management platform. The main objective is to provide a personal data usage control system for complying with the regulation manifold legal requirements.

A. Requirements

The proposed system must provide public access immutable evidences showing the agreements between a Data Subject and a Service Provider about DS's personal data. The ystem must also allow the DSs to interact with these agreements, in such a way they can manage their personal data. To accomplish that, our proposal and the evidences it emits must fulfill some GDPR, functional and security requirements.

1) GDPR Requirements: These requirements are obtained from the GDPR, analyzing the rights and duties that DSs and SPs have over personal data.

R1.1. Data Controller and Data Processor needs the Data Subjects' explicit consent for collecting/processing their personal data. This consent is specific for a certain period of time and for a specific set of personal data (Art. 6, 12, 17).

R1.2. Data collection and processing consent must contain the following information (Art. 13): i) Identity of the Data Controller; ii) Identity of the Data Subject; iii) Data collection or processing consent lifetime; iv) Identities of all Data Recipients; and v) for each Data Processor that requests processing the collected personal data, its identity, and the processing purpose.

R1.3. User consent must be publicly available for the Data Subject to manage what happens with his data and for the controller and processors to have evidence about the lawfulness of processing, in the case of an audit (Art. 7).

R1.4. Data Subjects must be able to

- R1.4.1. modify which personal data can be collected (Art. 18);
- R1.4.2. request their data not to be longer collected (Art. 18);
- R1.4.3. request their data to be erased (Art. 17), and;
- R1.4.4. revoke the processing consent for a specific processor and/or for a specific purpose at any time. (Art 21, 22).

2) Functional Requirements: Our system must also fulfill some functional requirements in order to compete and overcome existing proposal on the working topic.

R2.1. Reduce the interactions that actors have with the system, becoming as lightweight as possible.

R2.2. System must work in a distributed way, hence consent management is no longer centralised on SPs IT systems.

R2.3. Once an agreement has been set, it can be modified but not removed. So, it can be used in the future for an audit, when the consent has already expired.

3) Security and Privacy Requirements: Finally, for the system to be robust enough we need that the agreement-evidences that it provides fulfill some security requirements, as well as the actions that the different actors can do over these evidences.

R3.1. No actors must be able to carry out an action over the agreements for which they have not permission.

R3.2. Actors must be able to identify themselves, in such a way that they can prove who they claim to be.

R3.3. Information concerning a consent agreement cannot be modified since it was emitted.

R3.4. No actors can falsely claim that they have not performed an action on the system.

R3.5. System must provide immutable evidences of all events carried out.

R3.6. No personal data of the DS must be stored on the system. Neither can it provide any way of identifying the DS.

B. System Architecture Overview

Mechanisms related to the GDPR, focusing on the consent given by Data Subjects to a Controller/Processors to collect/process their personal data and its management, are ported to a BC network from the traditional centralised GDPR-compliance verification architecture.

As it can be seen in Figure 1, when a subject subscribes with a controller, wants to have access to certain service provided by the SP itself, the DC creates a new Consent Smart Contract (step 1 and 2). This contract specifies which DS's personal data can be collected while using the service and for how long this data can be kept. If the DS agrees with DC's conditions, the SC is validated and the subject has access to that service (step 3). This personal data will be collected by the DC and stored off-chain by the Data Recipient (DR) (it can be the DC itself or any other third party) (step 4). So, no personal information is stored in the contracts, only which data categories can be collected.

When the Consent SC is valid, a processor (any third party or the controller itself) can request to process DS's collected data (step 5). In this case, the DC creates a new Purpose Smart Contract (step 6). If the DS agrees on the processing purpose, the SC is validated, in such a way that the DP can request to the DR DS's personal data to process it (step 7 and 8). Then, the DR checks if the Purpose SC is valid and, gives access to that DS's personal data for which the DP has permits to process (step 9 and 10). Finally, the Supervisory Authority can check that the DC and DPs have the authorization to collect/process DS's personal data in the case of an audit (step 11).

C. Smart Contracts

As we have already said, the proposed system consists of two smart contracts: i) the Consent contract, which embodies the authorisation given to a DC from a DS to collect his personal data for a period of time and ii) the Purpose contract, which contains the permission given to a DP to process a subset of DS's personal data for a specific objective and for a limited period of time.

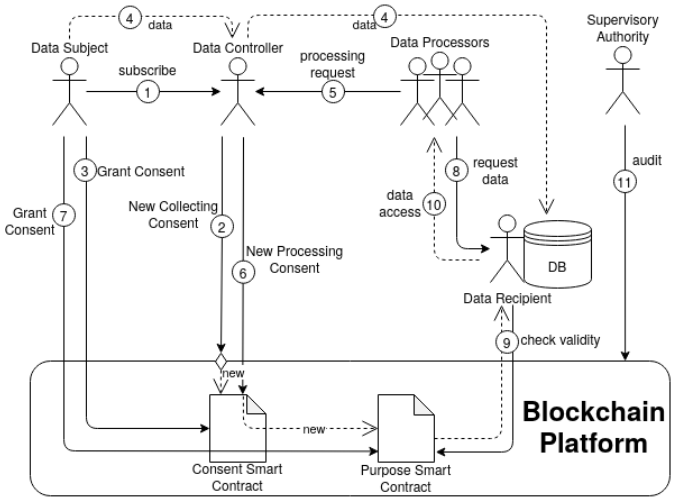


Figure 1. System Architecture

1) Consent Smart Contract: The Consent smart contract consists of a single class that holds:

- Identities: public keys of the Data Subject, Controller and Recipients (*dataSubject*, *controller* and *recipients* arguments);
- Data: this argument contains the data categories that can be collected, not the data itself (*Data*);
- Consent lifetime: period during which data can be collected and stored (*beginningDate* and *expirationDate*);
- Default Purposes: list of purposes for which the DSs allow the processing of their persona data (*defaultPurposes*);
- Purposes: link to all Purpose SCs that depend on this consent (*purposes*);
- Valid flag: flag that must have its value set to one for the Consent SC to be valid (*valid*).
- Erasure flag: flag that, if its value is set to one, the DC must erasure all DSs' personal data he has collected (*erasure*).

When a new Consent SC is instantiated, DC needs to indicate the *dataSubject, controller, recipients, defaultPurposes* arguments, and an integer indicating the duration of the contract (in days). By default, the *valid* and *erasure* flags are set to zero. The SC also provides some methods to interact with Consent instances:

- *grantConsent()* and *revokeConsent()* methods can only be used by the DS. They modify the *valid* flag by setting a one or a zero on it.
- *verify()* method checks if the Consent SC is valid. In order to be valid, the *valid* flag value must be one and lifetime has not expired.
- newPurpose() method is used to create a new Purpose SC instance. It can only be used by the controller and needs the Consent SC to be valid.
- modifyData() can only be used by the DS and updates the *Data* argument. If the new data argument is more restrictive than the previous one (allowing the collection of less personal data than before), those Purpose SCs that can process data which cannot be collected also will have this argument updated, by using the *modifyData()* method of the respective Purpose SC.
- eraseData() can also only be used by the DS. It sets the *erasure* flag to one, forcing the DC to erase all personal data collected from that DS.
- *revokeConsentPurpose()*: this method revokes the processing consent of all Purpose SC, created from this Consent SC, that have as purpose argument the one specified when calling this method. It also removes the specified purpose from the *defaultPurposes* list, if it is in there. This method can only be used by the DS.

2) Purpose Smart Contract: The Purpose smart contract also consists of a single class that holds the following information:

- Identities: public keys of the DS and DC plus the address of the Data Processor (*dataSubject*, *controller* and *processor* arguments);
- Data: which data can be processed, not the data itself (*Data*);
- Processing purpose: specific purpose for which data can be processed (*purpose*);
- Contract lifetime: specific period during which data can be processed (*beginningDate* and *expirationDate*);
- Validity array: four positions array (*valid*): one for the DS, another for the DC and the last for the DP. It must have all positions set to one for the Purpose SC to be valid.

When a new Purpose SC is instantiated, DC needs to indicate the *dataSubject, controller, processor* arguments, an integer indicating the duration of the contract (in days) and the processing purpose (*purpose*) for which data can be processed. Purpose SC also implements the *grantConsent()* and *revokeConsent()* methods that can be used by Data Subject, Controller and Processor, each one modifying its assigned position in the *valid* array. SC also implements the *modifyData()* method, which can only be used by the DS. It modifies which data can the DP use for a specific purpose. Finally, it implements the verify() method that returns if the Purpose SC is valid or it is not. In order to be valid, all positions of the *valid* array must be one and lifetime has not expired.

D. Workflow

Figure 2 shows the main interactions between the entities and SCs. When a Data Subject subscribes with a Data Controler, the DC creates a new Consent SC specifying which data will be collected and for how long it will be stored. It is also specified who (Data Recipients) will hold this data and a set of default purpose for which it can be processed without an additional consent of the DS. If DS agrees with the conditions, grants his consent to the SC (*grantConsent()*), and gets access to the requested service. From this moment, the DC can start collecting DS's personal data which will be stored by a Data Recipient. For each new Consent SC, the DS

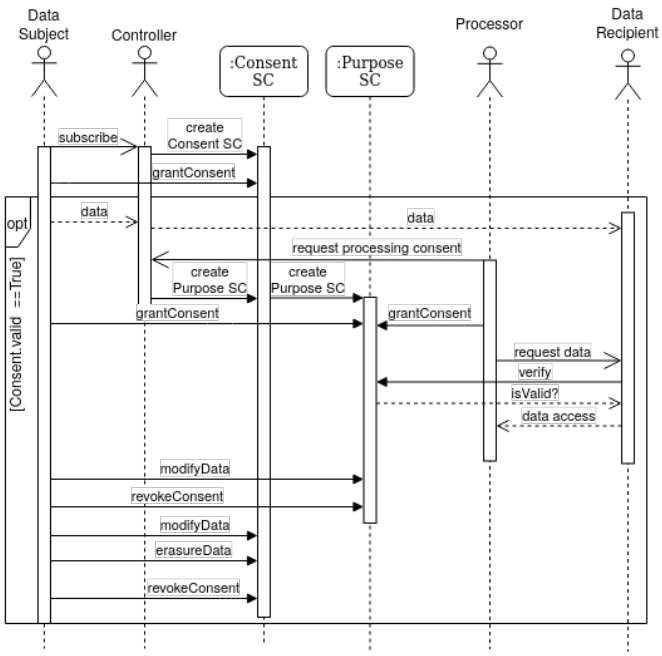


Figure 2. Sequence Diagram

uses a new address (public and private key pair) to identify himself and sign all transactions. We need to do it in this way in order to prevent a linkage attack using the SCs established with other DCs.

Once the Consent SC is valid, a Data Processor can request DC to process DS's collected personal data. If DC agrees, creates a new Purpose SC specifying which data will be processed and for which purpose, who will process this data and the lifetime of this permit. Then, if DS agrees to the processing purpose, he will grant his consent to the Purpose SC (*grantConsent()*). This SC also needs the DP to grants his consent to be valid, in order to ensure that he was the one who requested to process this data. If a DP request to process DS's personal data for a purpose included in the Consent SC *defaultPurposes* list, the Purpose SC does not require the DS to grant his consent to be valid. When the Purpose SC is valid, the DP can request the DS's personal data to the Data Recipient. DR checks if the contract is valid (*verify()*) and, in an affirmative case, gives access to the requested data.

Concerning DS's rights over his personal data, he can modify at any time which of his personal data can be collected or processed (*modifyData()*). Even, he can request his personal data to be erased (*erasureData()*). Finally, if DS's wants his data not to be longer collected or processed, he can revoke the consent previously given.

IV. Discussion

This section provides analysis and discussion of the described platform, including GDPR-compliance, functionality, security, and privacy.

A. GDPR-Compliance

The proposed platform allows SPs to comply with GDPR main requirements. This is due to SCs design and the allowed interactions.

- "Art. 6 Lawfulness of processing": SCs abstract the consent given by the DSs to collect their personal data and to process it for one or more specific purposes (R1.1.).
- "Art. 7 Conditions for consent": SCs showing DCs consent are public, so the controller is able to demonstrate that the DSs have consented to processing their personal data (R1.3.).
- "Art 12 Right to be informed": DSs must grant their explicit consent (using the *grantConsent* methods) for DCs and DPs be able to collect/process his personal data (R1.1.).
- "Art. 13 Required Information": The Smart Contracts contains all information that must be provided where personal data is collected from the Data Subject (R1.2.).
- "Art. 17 Right to erasure ('right to be forgotten')": Consent SCs have a specific lifetime, when this expires DC can no longer collect DSs personal data and all collected data must be removed. DSs can also use the Consent SC *erasureData()* method at any time (R1.4.3.).
- "Art. 18 Right to restriction of processing": DSs can limit which of their personal data can be collected by using the *modifyData()* methods (R1.4.1. and R1.4.2.).
- "Art. 21 Right to Object": DSs can also revoke at any time their consent for collecting and processing his personal data by using the *revokeConsent()* method (R1.4.4.).

B. System Functionality and Security

We have designed the smart contracts infrastructure minimizing the interactions that a Data Subject has to do over it (R2.1.). In other works, the Data Subject is required to perform to much actions over the system, which can difficult the acceptance of the proposals. For example, in [7] the DS is required to create the Data Usage Contracts to interact with the DC and DPs. On [12] proposal, when a DP requests to processing DS personal data: the DS receives the request (1), signs it and redirects the request to the DC (2), receives the response form the DC (3) and has to resend it to the DP (4). In order to reduce the overhead on the DS, we took the following design decisions:

- Data Controller creates all SCs.
- SCs only requires one transaction from the DS (*grantConsent()*) to be valid.
- When a DP requests to process DS's personal data for a purpose included in the Consent SC *defaultPurposes* list, the DS does not need to validate that SC, as it has previously agreed on that processing purpose. Nevertheless, DSs can revoke the consent for processing their personal data for a certain purpose by using the *revokeConsentPurpose()* method.

So, DSs are only required to interact once with the Consent SC -granting his consent- and once with that Purpose SCs that request to process their personal data, for that purposes not included in the *defaultPurposes* list.

Implementing the smart contract infrastructure over the blockchain makes our platform distributed (R2.2.), in contrast with current centralised GDPR-compliance verification architectures. It also provides the system with some functional and security requirements by inheritance. Operations over the system consists of transactions, which are digital signed using public key cryptography. So, while the signature scheme used to sign the transactions is secure, it provides authenticity (R3.2.) non-repudiation (R3.4.) and integrity (R3.3) properties. When these transactions are added into the blockchain, they become permanent (R2.3.), immutable (R3.5) and publicly available (R1.3.). Finally, no personal information of DSs is stored in the SCs, furthermore the IDs of the different actors are public keys, providing users anonymity. In order to enforce DSs resilience to linkage attacks, they use a different key pair for each DC (R3.6.). Smart contracts also control internally that the actor who signs a transaction, invoking any of the methods implemented on it, has the right of using it. So, no actor is able to carry out an action for which he has not permission (R3.1.).

We have implemented and deployed the proposed system in a local blockchain in order to validate the feasibility of our design[1]. The validation has consist on the emulation of the different actors and to perform the actions enumerated in Figure 1. As result, all smart-contracts methods have been tested.

V. Conclusion

In this work, we propose a system that provides public access immutable evidences showing the agreements between a Data Subject and Service Providers. We have implemented our proposal by using SCs deployed over the blockchain, due to all properties it provides.

These immutable evidences that our system provides, allow DCs and DPs to fulfill the GDPR main requirements by demonstrating that they have the authorization to collect/process DS's personal data in the case of an audit. This model also allows users to be aware at any time about which data is being collected and who is processing it and for which purposes. Users can also interact with the SCs at any time. They can limit which data can be collected and processed, the processing purposes or even which processors are allowed to work with their personal data.

As we have mentioned in the discussion section, we have implemented the proposed system validating the Smart Contracts functionality.

A. Future Work

We plan to add a graphical user interface (GUI) to facilitate that users can manage and control their personal data. The final framework will also internally manage user's wallets, in such a way that it will automatically generate and store a new address (key pair) when the user signs up into a new service. Finally, we will study using our framework on existing BCs or design and implement a new one. The cost of interacting with the SCs (the Gas that each transaction demands), the efficiency and scalability are the main properties the we will consider in order to take the decision.

[1] github.com/toful/BC_GDPR-Compliant_PDManagement_System

Acknowledgments and disclaimer

We acknowledge support from the European Commission (project H2020-871042 "SoBigData++"), the Government of Catalonia (Grant 2017 SGR 705), and the Spanish Government (project RTI2018-095094-B-C21 "Consent"). We are with the UNESCO Chair in Data Privacy, but the views in this paper are our own and not necessarily shared by UNESCO.

References

[1] Asunción Esteve. The business of personal data: Google, Facebook, and privacy issues in the EU and the USA. *International Data Privacy Law*, 7(1):36–47, 03 2017.

[2] Kimberly Houser and W. Voss. Gdpr: The end of google and facebook or a new paradigm in data privacy? *SSRN Electronic Journal*, 07 2018.

[3] Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation) (text with eea relevance). *Official Journal of the European Union L 119*, 59:1–88, 5 2016.

[4] Shakila Bu-Pasha, Anette Katariina Alen-Savikko, Jenna-Sofia Mäkinen, Robert Guinness, and Päivi Hannele Korpisaari. Eu law perspectives on location data privacy in smartphones and informed consent for transparency. *European Data Protection Law Review*, 2(3/2016):312–323, September 2016.

[5] Laure A Linn and Martha B Koo. Blockchain for health data and its potential use in health it and health care related research. In *ONC/NIST Use of Blockchain for Healthcare and Research Workshop. Gaithersburg, Maryland, United States: ONC/NIST*, pages 1–10, 2016.

[6] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman. Medrec: Using blockchain for medical data access and permission management. In *2016 2nd International Conference on Open and Big Data (OBD)*, pages 25–30, Aug 2016.

[7] Ricardo Neisse, Gary Steri, and Igor Nai Fovino. A blockchain-based approach for data accountability and provenance tracking. In *ARES '17: Proceedings of the 12th International Conference on Availability, Reliability and Security*, pages 1–10, August 2017.

[8] S. Wang, Y. Zhang, and Y. Zhang. A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. *IEEE Access*, 6:38437–38450, 2018.

[9] Mohammad Chowdhury, Alan Colman, Ashad Kabir, Jun Han, and Paul Sarda. Blockchain as a notarization service for data sharing with personal data store. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 1330–1335, August 2018.

[10] Christian Wirth and Michael Kolain. Privacy by blockchain design: A blockchain-enabled gdpr-compliant approach for handling personal data. In *Reports of the European Society for Socially Embedded Technologies (EUSSET)*, 2018.

[11] Benedict Faber, Georg Michelet, Niklas Weidmann, Raghava Rao Mukkamala, and Ravi Vatrapu. Bpdims:a blockchain-based personal data and identity management system. In *Conference: Hawaii International Conference on System Sciences*, January 2019.

[12] N. B. Truong, K. Sun, G. M. Lee, and Y. Guo. Gdpr-compliant personal data management: A blockchain-based solution. *IEEE Transactions on Information Forensics and Security*, 15:1746–1761, 2020.