Available online at www.sciencedirect.com

ScienceDirect

Computer Communications 00 (2024) 1–34

Computer Communi-cations

Blockchain-based Access Control System for Efficient and GDPR-compliant Personal Data Management

Cristòfol Daudén-Esmel, Jordi Castellà-Roca and Alexandre Viejo

Universitat Rovira i Virgili, Departament d'Enginyeria Informàtica i Matemàtiques
CYBERCAT-Center for Cybersecurity Research of Catalonia
Av. Països Catalans 26, E-43007 Tarragona, Catalonia (Spain)

{*cristofol.dauden,jordi.castella,alexandre.viejo*}*@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 and products, but also to fund themselves. However, processing personal data may result in the extraction of sensitive information, which, in turn, may lead to jeopardizing the users' privacy. To mitigate this significant risk, the European Parliament and Council of the European Union elaborated the General Data Protection Regulation (GDPR). This regulation forces Service Providers to obtain Data Subjects' explicit consent prior to collecting and processing their personal data. Nevertheless, the GDPR's legislative text does not define how Service Providers must transparently demonstrate that they already have these consents. Moreover, most individuals do not know the rights they have over their personal data, neither does this regulation provide them with efficient methods to be aware of what third parties are doing with such data. In order to address this situation, we propose a lightweight blockchain-based GDPR-compliant personal data management platform. The new solution provides public access to immutable evidences that reflect the reached agreements between Data Subjects and Service Providers. In this way, Service Providers can effectively demonstrate that they are fulfilling the regulation, and Data Subjects are able to control and manage their personal data according to their legitimate rights. We have implemented the new system, and we have performed a detailed study which includes: GDPR-compliance, provided functionality, security and privacy issues, and the cost in terms of gas and US dollars of the different operations to be run on the blockchain.

Keywords: General Data Protection Regulation (GDPR), Personal Data Management, Smart Contracts, Privacy

1. Introduction

The new digital technologies and its fuel, i.e., the personal data that they gather and consume, are the driving force of change and development which is transforming the current society by means of ground-breaking services and products. Well-known examples of those services are video-conference apps, massive information storage in the Cloud, location-based systems, and social networks among others. Regarding the products, cutting-edge devices such as smartwatches or smart home appliances equipped with Intelligent Personal Assistants (IPA) are becoming popular worldwide. Focusing on the aforementioned devices and in the personal data that they generate and take advantage of, smartwatches are capable of tracking biomedical data such as heart rate, sleep, activity or overall fitness level; in a similar way, smart home equipment deals with data related to security, lighting, and room temperature among others.

As it has been noted, those new technologies tackle huge amounts of information. In a final step, this data is gathered and processed by Service Providers which use it to improve and develop new services, but these entities also sell this data and make money by means of its processing, instead of directly charging users with the service

usage itself [1, 2]. The aforementioned information processing may result in extraction of sensitive data which may jeopardize the individuals' privacy. A significant example of this situation is the Cambridge Analytica scandal in which the personal data of thousands of Facebook users was misused in order to influence voters in the US Elections 2016[1]. This fact raised serious concerns about the technical, commercial, political, and ethical aspects of personal data collection and analysis by platform owners such as Facebook and other third parties.

In light of the above, the European Union drafted and approved the General Data Protection Regulation (GDPR) [3], which came into effect from May 2018, to mitigate the abuse of massive collection and processing of the individuals' personal data and, hence, keeping any data-hungry technology company at bay. The GDPR considers four key players: i) the Data Subject (DS), this is, the identified or identifiable natural person(s) from whom or about whom entities may collect personal information; ii) the Data Controller (DC), this is, the natural person or legal entity that determines the purposes for which and the means by which personal data is processed; iii) the Data Recipient (DR), this is, the natural person or legal entity to which the personal data is disclosed (it can be the same DC or another third party); and iv) the Data Processor (DP), this is, the natural person or legal entity that processes personal data on behalf of the DC. The main objective of the GDPR is to guarantee specific privacy rights to DSs, ensuring that their 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.

The GDPR has brought some relevant benefits such as boosting data security and protection awareness, or enabling consumers to control their preferences and to engage actively in the preservation of their rights. In particular, under this regulation, companies need methods to efficiently demonstrate that the information collected and processed fulfills the regulations, first, to show to the DSs that their personal data is lawfully processed, second, to prove the regulation compliance in front of a Supervisory Authority (SA). Nevertheless, works such as [4, 5] have shown that applying the GDPR in a practical scenario also raises significant issues that need to be tackled: on the one hand, the GDPR's requirements are highly abstract which, in turn, result in generic rules and principles that may be difficult to interpret and implement for both companies and consumers; on the other hand, traditional solutions follow a GDPR-compliance verification architecture in which all processes related to the GDPR are carried out in Service Providers' own servers, out of reach from the user control, which, in turn, brings relevant concerns regarding their *lack of transparency* [6]. In addition to that, the use of these independent provider-side verification architectures, each one managed by its hosting organization, makes it extremely difficult for DSs to ascertain where or to whom they have given their consent for processing their personal data. This issue may be acknowledged by DSs as a *lack of control* over their data.

As a result, nowadays, even in those scenarios in which the GDPR is actually being applied, DSs have no effective tools to know transparently and easily which data is being collected and processed and for which purposes. Instead of that, DSs are mostly limited to giving their consent beforehand, in a way that is based on an abstract clause. In order to bring the needed transparency to the data management, it is of paramount importance to provide mechanisms that grant all the involved parties to verify both the data custody and data processing. From the DSs' point of view, these methods would allow them to be aware of which data is being collected, who is storing/processing it, and for which purposes; and they would also empower them, handing over them an effective way to control and manage what happens with their personal data by means of granting/denying permissions to these Data Controllers/Processors. Regarding those entities, they would also be benefited greatly from the deployment of those tools. More specifically, Data Controllers/Processors would be able to retrieve a certified proof that would serve as evidence of GDPR-compliance in front on any Supervisory Authority.

Even though this issue has not been solved in a satisfactory way yet, it is worth mentioning that the scientific community has already paved the way to achieve it. In this regard, some authors have already introduced the use of Smart Contracts (SCs) implemented over the blockchain technology (BC) to design and create general-purpose data management and storage schemes which promise to offer features such as transparency, traceability, non-repudiation, integrity, immutability, and decentralization [7, 8, 9, 10, 11, 4, 12, 5].

1.1. Related work

It has been stressed above that there is a need of general-purpose data management systems that allow the different involved parties to prove the agreements made between them regarding the use and storage of the personal data. In

[1] 2018 Facebook and Cambridge Analytica Scandal. `https://www.cnet.com/news/facebook-cambridge-analytica-data-mining-and-trump-what-you-need-to-know/`

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

Table 1: Notations

order to fulfill that, in the last years, researchers have proposed different works based on the use of Smart Contracts and the blockchain technology, which offer desired features such as transparency, traceability, non-repudiation, integrity, immutability, and decentralization. Those proposals can be classified according to two main cases: i) when a Data Subject is willing to share her personal data with other parties and she will be in charge of its control and management; and ii) when a Service Provider collects the personal data of a DS for further processing in exchange for using a product that it provides, for instance, free of charge.

There are several proposals in the literature that fall into the first category, some relevant examples are [10], [13], [14], [11], [15] and [16]. These schemes follow a similar approach. In particular, they combine a decentralized storage system, the Ethereum blockchain, and Smart Contracts, in order to provide a fair access control to those Data Processors (DP) that want to collect and use the data shared by the DSs. An example can be found in [10], where authors combine technologies such as the interplanetary file system (as the decentralized storage system) and the Attribute Based Encryption (ABE) with the use of Smart Contracts running over the Ethereum blockchain to provide a data sharing secure system. While those schemes present significant differences in the way the access policies are created, used, and, in some cases, revoked, all of them require the Data Subjects to deal with the burden of managing the use of the protected data. This implies that DSs must employ their own infrastructure and control all the data's life cycle, an scenario which is not aligned with our envisaged use case. For this reason, in the following, we will focus on the second use case in which a Service Provider collects personal data from the DSs for further processing by DPs.

Turning the spotlight in this second category, the next main aspect to be taken into account is whether those proposals are GDPR-compliant or not. In this way, wide-purpose schemes such as [17], [9], [18], [19], or health-care related specific proposals such as [7], [8],[20], and [21] do not take into account the GDPR requirements in their procedures. These schemes, in general, share a similar approach based on providing a decentralized personal data management system that ensures that users own and control their data, using the blockchain as an automated access-control manager which, in some cases, is improved by means of Smart Contracts. For example, in [18] the authors use Smart Contracts to store the consumer access control polices that will be used in the access control process; also, in the health-care related system presented [8], Smart Contracts are used to link patients with their corresponding cryptographic material, hold a list of interactions among participants and, finally, establish a Patient-Provider Relationship Contract that identifies the records held by the care provider. Although these proposals address many issues, first, as already mentioned, they are not GDPR-compliant, which represents a significant shortcoming in the current context; last, but not less important, these works do not consider the different costs that may arise when being deployed on a large scale setting. In this way, the authors of those schemes only provide a high level description without any kind of implementation or evaluation of their proposed systems.

Focusing now on the GDPR-compliant proposals that can be found in the literature, the first scheme following this path is introduced in [4]. In this work, the authors present a scheme that uses Smart Contracts to embody the consent of the Data Subjects while their personal data is encrypted and directly held on the blockchain. Although the data is encrypted, an external entity may identify the person behind a private key and link all the data stored in the blockchain, hence this proposal does not provide anonymity but pseudonymity instead.

The authors in [12] propose a personal data and identity management system in which the consent given by the Data Subject is held by a controller by means of a Smart Contract, while her personal data is stored in an off-chain repository. The authors of this work only present a conceptual design. Technical details are not addressed.

Similarly, the solution presented in [22] uses Smart Contracts to maintain an Access Control List (ACL) that represents the rights to access a bundle of data; and it encrypts and stores the personal data into a decentralized file storage system. Again, the authors do not address relevant technical details such as how the ACL looks like or how the Smart Contracts are implemented.

Truong et al. [5] introduces a GDPR-compliant personal data management platform that allows Data Subjects to impose data usage consent, ensuring that only designated parties can process their personal data, and logging all data activities in an immutable distributed ledger using Smart Contract and cryptography techniques. Mechanisms which are related to GDPR compliance are ported to a blockchain network from a traditional centralised server by means of Smart Contracts. The proposed system has been successfully implemented on a permissioned blockchain, however, its deployability on a public blockchain has not been tested yet. Moreover, this scheme tackles the collection and data processing consents in a coarse-grained fashion not allowing the Data Subject to make micro decisions regarding whether a certain party can process or not only a part of the whole personal data which is available to others. Last but not least, this scheme stores a hash of the collected personal data in the distributed ledger for integrity purposes which, in turn, may result in a significant processing overhead when gathering dynamic data due to the necessity of keeping this hash up to date.

Davari and Bertino, in [23], combine the blockchain technology and Smart Contracts with the XACML (eXtensible Access Control Markup Language) access control specification to provide attribute-based access control and data management, while allowing Service Providers to fulfill the GDPR requirements. The proposed system consists of four main components: i) an ontology to formalize the notion of the GDPR consents, data flows, and decisions; ii) a decentralized personal data management that ensures compliance by leveraging the blockchain; iii) a personal data repository which guarantees that only authorized Data Controllers and Data Processors can access the data; and iv) an XACML-based access control model containing consents, compliance, and a data management mechanism to enforce the legal privacy regulations specified by the GDPR. Even though this solution is promising, the authors do not provide any kind of implementation that proofs its deployability. Moreover, similarly to [5], this solution has issues when dealing with dynamic data, such as search logs or medical data, due the fact that they store the hash of the protected data in the Smart Contracts for integrity purposes.

The work presented by Barati and Rana in [24] shows how some GDPR rules can appear as opcodes in SCs. These opcodes allow to verify the operations that cloud providers carry out over users' personal data in a transparent and automatic way. The translation of GDPR rules into pseudo-code basically consists in writing questions related to data protection and privacy measures that should be supported by cloud services. This rules can then be assigned to each of the operations that can be carried out by providers (write, read, transfer, ...). The authors of this work explain how their scheme should be implemented and provide an extensive experimental section. In particular, they detail how four Smart Contracts are used to verify actor operations with respect to a set of GDPR rules, these are: the GDPR compliance contract (i.e., the policy that applies to the protected data); the Data Subject consent contract; the container contract that records all operations performed on personal data; and the Verification contract that allows to confirm whether the Data Subject consent has been obtained. Despite the significant positive qualities of this proposal, it also has two main issues to be considered: i) the use of four different SCs implies a significant computational cost that, in turn, may directly affect the efficiency of this proposal; and ii) a third party called the "Contract Activator" is required to generate all Smart Contracts, implying too many communications between all parties.

In [25], Merlec et al. propose a smart-contract-based dynamic consent management system backed by blockchain technology that targets personal data usage under the GDPR. More specifically, this scheme system allows individuals to control the collection and usage consent of their personal data throughout the data life-cycle. The proposed system cannot be considered fully decentralized due to the fact that a central entity behaving as regulator is required to allow entities to join the system and to approve their respective roles in the architecture. Moreover, the authors also expose other limitations such as the fact that an efficient and user-friendly key management scheme is required to guarantee the security and privacy features; and the fact that deleting the individuals' stored personal data is complex and challenging due to the immutable nature of the blockchain.

Finally, the authors in [26] use a private blockchain to provide a delegation-based personal data processing request notarization framework under the GDPR. This framework allows DSs to delegate the requests to process their personal data to the data controllers. These data processing requests and the corresponding processing results are stored in a private blockchain and notarized via a trusted institution of the linked blockchain network. A main issue of this solution is that it directly omits how DSs should give their consent to DCs in order to collect their data. In the same

way, it also omits how DPs should gather that data in order to process it.

1.2. Contribution and plan of this paper

In this article, we propose a blockchain-based GDPR-compliant personal data management system that, first, it provides mechanisms for Data Subjects to control and manage their personal data; and, second, it provides publicly accessible and immutable evidences which are useful for a SP to prove the agreements made between a Data Subject and her about the Data Subject's personal data. These evidences might be then used by a Supervisory Authority that performs an auditing procedure on the SP to verify that this entity is doing the data processing exactly as agreed with the affected Data Subjects. In the case that the SP misbehaves in any way, the Supervisory Authority will detect it and will be able to apply the corresponding punishment to those actions. In this way, the data processing guidelines which are stipulated between data subjects and Service Providers are effectively enforced.

As it has been stated in the review of the current literature, the considered research problem has been already tackled in several works. In comparison with them, our proposal brings a new conceptual design and system architecture for human-centric access control and personal data management that uses blockchain technology, Smart Contracts, and the XACML access control standard to allow Service Providers to fulfill the GDPR requirements. Our work differentiates between the data collection and data processing concepts by identifying the Data Controller and Data Processor actors and treating them in a related but separate way. We also reduce the overhead on Data Subjects. The target here is to reduce to the maximum the knowledge on blockchain technology that those actors may have and, also, reduce the operations that they should perform. We argue that both concepts, if not considered properly, may seriously discourage the usage of any personal data management system by the community.

The new proposal is an extension of a preliminary scheme presented in [27]. In particular, the new scheme has been partially re-designed to be deployed in a realistic setting. This includes: i) a modification in the process flow that makes the Data Subject the main responsible of her own personal data and the initiator of the whole proposed protocol; ii) the use of the well-known XACML framework to improve the robustness of the access control process; and iii) a refinement in the use of the Smart Contracts that allows to include all the purposes of a certain Data Processor in a single Smart Contract, thus, improving the general efficiency of the proposed system. Moreover, in the new proposal, we have performed a detailed study which includes: GDPR-compliance, provided functionality, security and privacy issues, and the cost in terms of gas and US dollars of the different operations to be run on the blockchain.

The rest of this paper is organized as follows: in Section 2, we provide a brief description about the technologies and concepts used in our contribution; Section 3 describes our proposal, presenting the requirements, the designed architecture, the Smart Contracts infrastructure, and its envisaged integration in a real-world setting; in Section 4, we provide an analysis and discussion of the described platform; and, finally, Section 5 summarizes the conclusions and future work.

2. Background

A brief description about the main technologies and concepts used in our contribution is next introduced.

2.1. The GDPR in a nutshell

The GDPR was created to give the citizens of the EU (i.e., Data Subjects) control over their personal data. This legislative text consists of 99 articles covering all of the technical and admin principles around how commercial and public organizations must handle clients and customers personal data [3]. We differentiate two main aspects of the GDPR; "personal data" and "processing of personal data":

- "personal data" pertains to "any information relating to an identified or identifiable natural person" [Art 4.1.]. For example, the name, online identifier, email, or even one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person;
- whereas "processing of personal data" refers to "any operation or set of operations which is performed on personal data or on sets of personal data" [Art 4.2.]. Therefore, a simple operation of storing a username or an IP address on a web server constitutes processing of personal data of a user.

So, the main goal of the GDPR legislation is to provide citizens with full control over their personal data. To fulfill this, this regulation differentiates between the roles: Data Subject, Data Controller and Data Processor; and it explicitly specifies associated rights and obligations under the EU data protection law. In this way, this regulation requires that personal data must be managed by a DC ensuring the rights of the DS, in such way that the DS can impose consents and withdraw those consents whenever she wants. The DS must also be able to know at any time *who*, *what*, *why*, *when* and *how* her data is being processed. For this data being able to be collected and processed, DCs and DPs need DS's valid legal consent. Furthermore, apart from being compliant with the GDPR, DCs have to be able to demonstrate it in front of a Supervisory Authority (SA) in an audit.

After a deep analysis of the legislative text, we have extracted some articles from which we obtain the requirements for our proposal. However, from the extracted ones, there are some articles we have not considered in this work because they are out of scope such as Art 5. (Principles relating to Processing of Personal Data); or because we will address them in a future work, such as Art. 16 (Right to Rectification), Art. 19 (Notification Obligation) or Art. 20 (Right to Data Portability). These articles are summarized in Table 2.

GDPR Article	Name	Description
5	Principles relating to Processing of Personal Data	Personal data controller must fulfill with the properties of lawfulness, fairness, transparency, purpose limitation, accuracy, storage limitation, integrity and confidentiality. It also must be able to demonstrate compliance with all of them.
6	Lawfulness of Processing	Data subject must give explicit consent for the processing of his personal data for one or more specific purposes and the processing must comply the legal obligation to which the controller is subject.
7	Conditions for Consent	Controller shall be able to demonstrate that the Data subject has given the explicit consent to process his personal data. This consent can be withdrawn at any time.
12	Transparent Information	Controller shall take appropriate measures to provide any information relating to processing to the Data Subject in a concise, transparent intelligible and easily accessible form.
13	Information to be provided	Controller shall provide the Data Subject with certain information where her personal data is collected.
15	Right of Access	Allows access by the Data Subject to any personal data held by a company.
16	Right to Rectification	Ensures that any data held by controllers can be rectified if inaccurate.
17	Right to Erasure	Ensures that any data held by controllers can be erased if the subject desires.
18	Right to restriction of processing	Ensures that any data gathered will only be used for the purposes for which consent was given.
19	Notification Obligation	Controller must notify, to each recipient to whom the personal data have been disclosed, regarding rectification or erasure of personal data or restriction of processing.
20	Right to Data Portability	Empowers Data Subjects to take their data with them if they leave the organisation.
21	Right to Object	Empowers Data Subjects to object to what how their data is being processed.
22	Automated Individual Decision Making	Prevents automated individual decision making, including profiling, of Data Subjects and selling this information to marketing firms.
25	Data Privacy by Design	Mandates that privacy and the trust it engenders must be built into the design using appropriate technical and organisational measures.

Table 2: GDPR Articles defining our requirements.

2.2. Blockchain technology and Bitcoin

The concept of blockchain architecture was first mentioned by S. Haber and W.S. Stornetta in 1991 [28]. In this work, they propose computationally practical procedures for digital time-stamping of documents (text, audio, picture, and video files). The idea behind this scheme is to make infeasible for a user either to back-date or to forward-date her document, even with the collusion of a time-stamping service. This represented a computationally practical solution for the order and management of digital documents, so that they could not be modified or manipulated.

Some years later, some developers working under the pseudonym *Satoshi Nakamoto* released a white paper establishing the model for a proof-of work based blockchain [29]. This blockchain consists of a distributed growing list of records, called blocks, that are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a Merkle tree [30]). This design allows to timestamp blocks without requiring them to be signed by a trusted party and introduces a difficulty parameter to stabilize the rate in which blocks are added to the chain. This system was implemented the following year by Nakamoto, becoming the core component of the cryptocurrency Bitcoin, where it serves as the public ledger for all transactions on the network.

The security of the blockchain relies on the hardness of the hash function used to link the blocks which, in the case of Bitcoin, is SHA-256. Once a block is added to the blockchain, it cannot be changed, as later blocks are chained after it and the work to change the block would include redoing all the blocks after it. The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. So, blockchain technology has the properties of immutability and persistence [31].

Payments in Bitcoin are made through structures called transactions, which can be seen as a payment made with a bank check. For a transaction to be valid, it must fulfill with some requirements, such as:

- *Non Double Spending*: inputs must be unspent outputs of a previous transaction.
- *Digital Signature*: each input must be digitally signed.
- *Authenticity*: the provided signature must match with the owner (receiver) of the inputs (transactions from where the money transferred in this new transaction come).
- *Correctness*: the sum of the values of all inputs must be higher or equal than the sum of all values of all outputs.

Digital signatures are used to verify that the user who issues a transaction is the one she claims she is. So, public cryptography is used in order to avoid that no one, except for the receiver of a transaction, could spend it in the future (use the money transferred to her as the input of another transaction). In the case of Bitcoin, the Elliptic Curve DSA (ECDSA) scheme is used to sign each transaction. The use of digital signatures adds the properties of authenticity, integrity and non-repudiation [32].

2.3. Smart contracts and Ethereum

Later on, blockchain technology separated from the currency and its potential for other financial, inter-organisational transactions was explored. Blockchain 2.0 was born, referring to applications beyond currency, in which Ethereum has been one of the most significant ones. The Ethereum blockchain system was proposed in 2014 by Vitalik Buterin [33]. This system introduces computer programs into the blocks, currently known as Smart Contracts.

A Smart Contract is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. It has the property of automatically verifying the contract and execute the agreed terms, so it allows the performance of credible transactions without third parties, providing a security that is superior to traditional contract law and reducing other transaction costs associated with contracting. Smart contract transactions are traceable, transparent and irreversible [34].

In general, the use of Smart Contracts over the blockchain technology provides the following properties:

- *Immutability*: impossibility of changing the state of an object after it is created.
- *Persistence*: once an object is created, it cannot be removed or forgot.
- *Authenticity*: ability to prove that a user or application is genuinely who that person or what that application claims to be.

- *Integrity*: maintenance of, and the assurance of the accuracy and consistency of data over its entire life-cycle.
- *Non-Repudiation*: assurance that someone cannot deny the validity of something.
- *Traceability*: capability of keeping track of a given set of information to a given degree.
- *Transparency*: ability to easily access and work with data no matter where they are located or what application created them.
- *Irreversibility*: once a transaction is added into the blockchain, it cannot be turned back to the previous state.

2.4. The XACML standard

XACML stands for "eXtensible Access Control Markup Language". The standard defines a declarative fine-grained, attribute-based access control policy language, an architecture, and a processing model describing how to evaluate access requests according to certain rules defined in policies [35].

It is primarily an attribute-based access control system (ABAC), also known as a policy-based access control (PBAC) system, where these attributes and/or polices associated with a certain user, action or resource are used to decide if certain entity gets access to a resource in a particular way.

The XACML model differentiates the stages of access control identifying them as different points (see Table 3), and supports and encourages their separation as individual identities. One of the goals of this system is to promote common terminology and interoperability between access control implementations by multiple vendors.

Abbreviation	Term	Description
PAP	Policy Administration Point	Manages access authorization policies.
PDP	Policy Decision Point	Evaluates an access request against the authorization policies before issuing an access decision.
PEP	Policy Enforcement Point	Intercepts user's access request to a resource, makes a decision request to the PDP to obtain the access decision (i.e., access to the resource is approved or rejected), and acts on the received decision.
PIP	Policy Information Point	Source of attribute values (i.e., a resource, subject, environment)
PRP	Policy Retrieval Point	Where the XACML access authorization policies are stored, typically a database or the filesystem.

Table 3: XACML terminology

The current workflow of this model is described in Figure 1:

1. A user sends a request which is intercepted by the Policy Enforcement Point (PEP).
2. The PEP transforms the user request into a XACML authorization request.
3. The PEP forwards the authorization request to the Policy Decision Point (PDP).
4. The PDP evaluates the authorization request against the policies in use. The policies are acquired via the Policy Retrieval Point (PRP) which is, in turn, managed by the Policy Administration Point (PAP). If indicated by the policies, the PDP also retrieves any required attribute value from underlying Policy Information Points (PIP).
5. The PDP reaches a decision (Permit / Deny / NotApplicable / Indeterminate), and returns it to the PEP.

3. Our proposal

In this section, we describe in detail our proposed data management platform, which is built on top of a blockchain-based access control system, and allows the fulfillment of the GDPR to the Service Providers using it.

First, the set of requirements considered in the design of the new proposal are introduced; then, an overview of the proposed system is presented; after that, the use of Smart Contracts to represent the consents is discussed; next, both the architecture and the system's workflow are deeply explained; finally, the integration of the new proposal in a real-world setting is discussed.

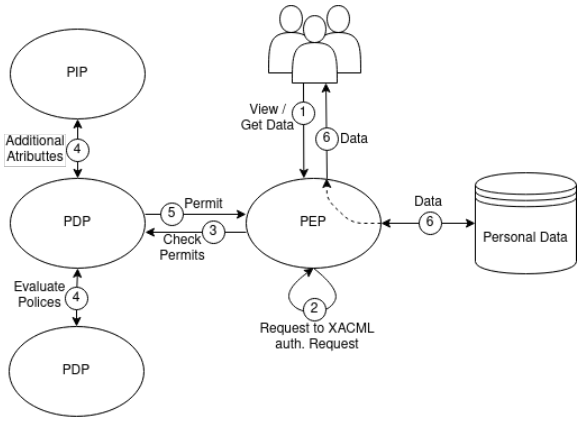


Figure 1: XACML architecture and general workflow

3.1. Requirements

According to the regulations, DSs should have access to all consents given to DCs and DPs at any time. This can be easily achieved by making them freely accessible to everyone. However, making them publicly accessible would imply that any non-authorized party may find a way to tamper them with. In order to prevent this undesirable situation, the proposed system must ensure that: i) all actions linked to an agreement (i.e., consent) can only be performed by authorized actors; ii) these actions must be immutable, i.e., once those actions have been validated they can not be modified or deleted; and iii) the result of validating these actions, i.e., the agreement/consent achieved by the different parties, must be persistent and it may be modified only by subsequent validated actions.

In summary, firstly, the proposed system has to provide public access to the immutable evidences that show the agreements/consents achieved between DSs and DCs regarding DS's personal data. Those agreements/consents can only be updated by means of further validated actions which, in turn, are represented by additional immutable evidences. Secondly, the proposed system should try to mitigate identity and/or attribute disclosure risks by means of preventing adversaries from linking different agreements/consents to a certain DS. For example, if a DS grants her consent to her hometown web page, a technical web, and some webs related to her hobbies, an adversary should not be capable of re-identifying the DS, or retrieving any significant piece of information from her.

In order to achieve that, the new proposal and the evidences that it generates must fulfill the following requirements:

3.1.1. GDPR requirements

According to the rights and duties we have addressed in section 2.1, here we extract the main requirements our proposal must fulfill in order to provide GDPR compliance to Data Controllers, Data Processors and Data Subjects:

R1.1. DC needs DS's explicit consent in order to collect her personal data (Art. 6, 12).

R1.2. Data-collecting consent is specific for a certain period of time (Art. 17) and for a specific set of personal data, so it must contain the following information (Art. 13): i) identity of the DC; ii) consent lifetime; iii) set of personal data; and iv) identities of all Data Recipients.

R1.3. DP needs DS's consent in order to process her personal data. DP also needs the DC's consent due to the fact that the processing is done under her responsibility (Art. 6, 12).

R1.4. Data-processing consent is specific for a certain period of time (Art. 17), purpose and for a specific set of personal data, so it must specify the following information (Art. 13): i) identity of the DC; ii) identity of the DP; iii) consent lifetime; iv) set of personal data; v) identities of all Data Recipients; and vi) the processing purpose.

R1.5. User consent must be available for the DS in order to control and manage her data, in such a way that she can:

R1.5.1. Modify which personal data can be collected (Art. 18).

R1.5.2. Request their data not to be longer collected (Art. 7, 18).

R1.5.3. Request their data to be erased (Art. 17).

R1.5.4. Revoke the processing consent for a specific processor at any time. (Art. 7, 21, 22).

R1.5.5. Revoke the processing consent for a specific purpose at any time. (Art. 7, 21, 22).

R1.6. User consent must be available for DCs and DPs in order to have evidence about the lawfulness of the processing in the case of an audit (Art. 7).

R1.7. DCs determine the purposes and means of the processing of personal data and, hence, they must be able to revoke the consent to process DS's personal data given to any DP.

3.1.2. Functional requirements

The proposed system must also fulfill a set of functional requirements in order to overcome the current literature:

R2.1. It has to be fast and efficient, becoming as lightweight as possible.

R2.2. It must provide a single resilient point of access to all consents/agreements given by a DS to process her personal data. This requirement addresses the current unfeasibility for DSs to keep track of all places in which they have given these consents.

R2.3. It must work in a distributed way, hence, consent management must not be centralized at the Service Providers' IT systems. This requirement addresses trust concerns on the traditional model in which any consent reached between a Service Provider and a DS is stored in the Service Provider's servers.

R2.4. It must allow DSs, DCs and DPs to interact with and modify the agreements reached between them by means of certain actions.

R2.5. It must keep track of all the actions related to a certain, allowing then to reconstruct that consent life-cycle.

3.1.3. Security and privacy requirements

Finally, for the system to be robust enough, the agreement-evidences that it provides and the actions that the different actors may do over them must fulfill a set of security requirements which are next summarized:

R3.1. Actors cannot perform an action over the agreements for which they have no permissions according to the GDPR.

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

R3.3. No actor can falsely claim that she has not performed a certain action.

R3.4. The proposed system must provide immutable and permanent evidences of all actions carried out over the consents/agreements.

R3.5. No personal data of the DS must be stored in the system. It cannot provide any way of identifying the DS either.

R3.6. Only authorized actors must have access to DS's collected personal data.

3.2. Overview of the proposed architecture

In order to address the different problems and issues presented in previous sections, the proposed solution ports the current GDPR-compliance verification process to a blockchain network, decentralizing it, and making the consents/agreements with the different Service Providers easily available for Data Subjects.

In the introduction, we have described a scenario in which a Service Provider collects personal data from a Data Subject for further processing in exchange for using a product that it provides free of charge. Performing the aforementioned port to the GDPR regulation first requires us to relate the GDPR actors (i.e., DS, DC, DR and DP) with the entities of the depicted scenario. In this way, there is a first direct relation between the GDPR's DS and the Data Subject that uses the Service Provider product. Next, focusing on the Service Provider and the three GDPR actors left, there is a wide range of different options in that case: for example, the DC, DP and DR may be fully independent third

parties and be hired by the Service Provider to perform their assigned duties; the Service Provider may simultaneously be those three actors; or it could even be something in between, this is, a Service Provider acting as the DC and the DR, and a fully independent third party acting as the DP.

Once the different main actors of the proposal have been stated, we next differentiate the two types of consents/agreements that DSs and DCs may reach:

- *Data-collection Consent*: Consent between a DS and a DC that encloses all the required information and conditions that the DC must fulfill in order to collect DS's personal data in exchange for using a service.
- *Data-processing Consent*: Consent between a DP and a DC that encloses all the required information and conditions that the DP must fulfill in order to access the data collected by the DC and process it.

The workflow of the presented framework relies on three main steps: *Consent Generation* and *Data Management* steps are briefly represented in Figure 2 (i.e., blue and green sections respectively). The third step, *Consent Management*, consists of the interactions of the different actors with the generated Smart Contracts in order to manage the given consent.

In the Consent Generation procedure, first, a DS requests access to certain service; then, the DC asks the DS for permission to collect her personal data in exchange for accessing the requested service. If both agree with the conditions, a new *Data-collection Consent* is generated, the DS gets access to the offered service, and the DC starts collecting DS's personal data. Next, if a DP request the DC to process some of the collected data, the DC will create a new *Data-processing Consent* for that DP, specifying which data can be processed and for which purpose. Note that, if a *Data-Processing Consent* between a DC and a DP already exists, it can be extended with new collected data and new purposes at any time.

In the Data Management step, a DP that owns a valid *Data-Processing Consent* uses the proposed XACML extension to request the DR to supply the stored DS's personal data. Similarly, DSs can also use that method to get access to their collected data.

Finally, in the Consent Management step, all actors can interact with the different consents that have been created according to their respective rights over them. For example, a DS can revoke any previously established consent for processing or collecting her data at any time. In the same way, a Supervisory Authority can check these consents at any time in order to audit that a DC is fulfilling the legislation.

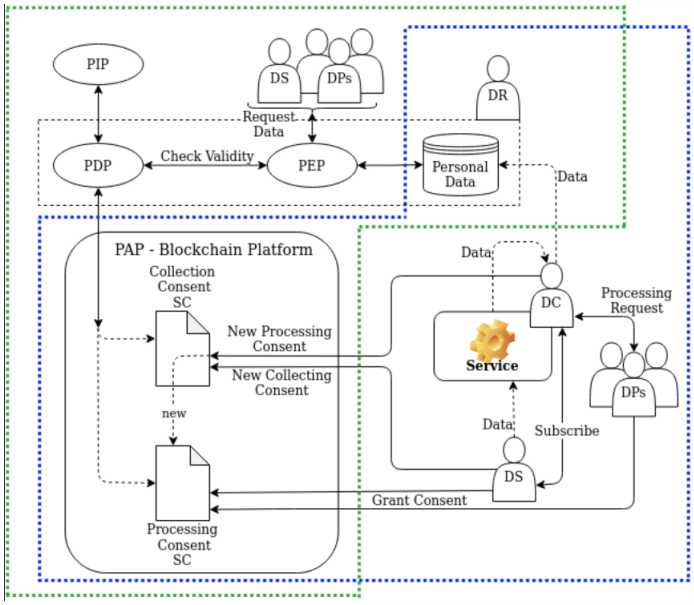


Figure 2: System architecture

3.3. Smart contracts

In order to fulfill the requirements indicated in 3.1, the new proposal implements the *Data-collection* and *Data-processing* consents introduced above by means of Smart Contracts. In particular, the following two Smart Contracts are defined:

- *Collection Consent Smart Contract*: It embodies the authorization that a DS gives to a DC to collect her personal data.
- *Processing Consent Smart Contract*: It contains the authorization that a DC and the DS give to a DP to process a subset of DS's personal data for a specific purpose.

3.3.1. Collection Consent Smart Contract

This Smart Contract consists of a single class that holds:

- Identities: public keys of the Data Subject, Controller and Recipients (*dataSubject*, *dataController* and *dataRecipients* arguments). These public keys are used as pseudonyms by the different actors. In this way, a DS may use a different key pair for each DC in order to improve her anonymity.
- Data categories: the data categories that can be collected are stored in the *Data* argument. Note that this argument does not contain the data itself.
- Consent lifetime: period of time during which the data can be collected and stored (*beginningDate* and *expirationDate* arguments).
- Default purposes: list of processing purposes for which a DS allows the processing of her persona data (*defaultPurposes* argument).
- Blacklist: list with all the DPs that the Data Subject has revoked her consent to process her personal data (*processorsBlacklist* argument).
- Processing Consents: link to all the *Processing Consent Smart Contracts* that depend on this consent (*processingConsentContracts* argument);
- Valid flag: flag that indicates the validity of the Smart Contract (*valid* argument).
- Erasure flag: flag that, when it is set to true, compels the DC to erase all the personal data collected from the DS (*erasure* argument).

When a new *Collection Consent Smart Contract* is instantiated, the DS must provide the following arguments: i) *dataSubject*; ii) *dataController*; iii) *dataRecipients*; iv) *defaultPurposes*; v) *beginningDate*; and vi) *expirationDate*. Automatically, *valid* argument is set to one while *erasure* argument is set to zero.

Once the *Collection Consent Smart Contract* instance is created, the different actors may interact with it by means of the following methods:

- *grantConsent()* and *revokeConsent()* methods are used by the DS to modify the *valid* flag, enabling or disabling the SC.
- *verify()* method checks whether the Smart Contract is valid or not. In order to be valid, the *valid* flag value must be true and the linked lifetime period must have not expired.
- *newPurpose()* method is used by the DC to create a consent for a DP to allow it to process the personal data of the bounded DS. More specifically, two different situations apply: i) if in the Smart Contract already exists a *Processing Consent Smart Contract* instance linked to this DP, this method adds a new processing purpose to that contract; and ii) otherwise, a new *Processing Consent Smart Contract* related to the DS, the DC, and the new DP is created and its address is stored in the *processingConsentContracts* argument. Note that this method requires the *Collection Consent Smart Contract* to be valid (i.e., the *verify()* method must return "true").

- *modifyData()* method is used by the DS to update 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 *Processing Consent Smart Contracts* that state the process of data which cannot be collected anymore will also have this argument updated. This is achieved by using the *modifyData()* method of the corresponding *Processing Consent Smart Contract*.

- *eraseData()* method is used by the DS to set the *erasure* flag to "true". This compels the DC to erase all personal data collected from that DS.

- *revokeConsentPurpose()* method is used by the DS to revoke the processing consent for all processors that has requested to process DS's personal data for a specified purpose; and remove that purpose from the *defaultPurposes* list (if included). This method requires as an input argument the purpose for which its processing consent will be revoked.

- *revokeConsentProcessor()* method is used by the DS to revoke all processing consents given to a Data Processor; and add the specified Data Processor to the *processorsBlacklist* list (if she is not included).

3.3.2. Processing Consent Smart Contract

The DC responsible of a *Collection Consent Smart Contract* creates a *Processing Consent Smart Contract* for each processor that requests to process DS's personal data. This Smart Contract holds different policies for each processing purpose, specifying which subset of data can be processed and for which period of time.

This Smart Contract consists of a single class and holds the following information:

- Identities: the identities of the data subject and controller plus the address of the processor (*dataSubject*, *dataController* and *dataProcessor* arguments).

- Data: which data can be processed (*Data* argument) and for which specific period of time (*beginningDate* and *expirationDate* arguments).

- Purpose: the specific purpose for which the data will be processed (*purpose* argument).

- Valid Flag: flag that indicates if the Smart Contract is valid (*valid* argument).

This contract encapsulates the *Data, beginningDate, expirationDate, purpose*, and *valid* arguments in blocks. There are as many blocks as processing purposes the bounded DP has requested.

There are two main cases when a DP requests to process DS's personal data for a certain purpose: i) it is the first time that the DP request to process the data, so the DC needs to instantiate a new *Processing Consent Smart Contract*; or ii) there is an existing *Processing Consent Smart Contract* for that specific DP, so the DC just has to add a new processing purpose to the that Smart Contract.

When a new *Processing Consent Smart Contract* is instantiated, the DC indicates the *dataSubject, dataController, dataProcessor* arguments, and all the information related to the processing purpose: the processing purpose (*purpose* argument) for which the data will be processed, the personal data to which this purpose is applicable, and the duration of the consent (*beginningDate* and *expirationDate* arguments). On the other hand, when a DP wants to add a new processing purpose, the DC only adds an additional block of information.

Regarding how the different actors may interact with this Smart Contract, similarly to the *Collection Consent Smart Contract*, the *Processing Consent Smart Contract* implements the *grantConsent()*, *revokeConsent()*, *modifyData()* and *verify()* methods. Those methods been already explained above.

3.4. Architecture and procedures

We next explain in detail the architecture of the proposed system and the different procedures that are used to generate and manage the consents and the data.

3.4.1. Collection Consent Smart Contract generation and data collection

Figure 3 depicts the workflow followed by the new system when a new consent for data collection must be generated and the collection of the corresponding personal data is started. In particular, the following steps apply:

1. A DS request access to certain service provided by a DC.
2. The DC emits a data collection consent request. This request specifies which DS's personal data can be collected while using the aforementioned service and for how long this data can be kept.
3. If the DS agrees with DC's conditions, she generates a new *Collection Consent Smart Contract*. Otherwise, both parties must renegotiate the collecting polices, or the DS may finally refuse using the service offered by the DC.
4. If the *Collection Consent Smart Contract* fulfills DC's requirements, this entity validates the Smart Contract. As a result, DS gets access to DC's service and DC starts collecting DS's personal data.
5. This personal data collected by the DC is stored off-chain by the DR. In this way, there is no personal information stored in the Smart Contracts. They only hold the information regarding which categories of data can be collected. Note also that the DR can be the DC itself or an additional third party.

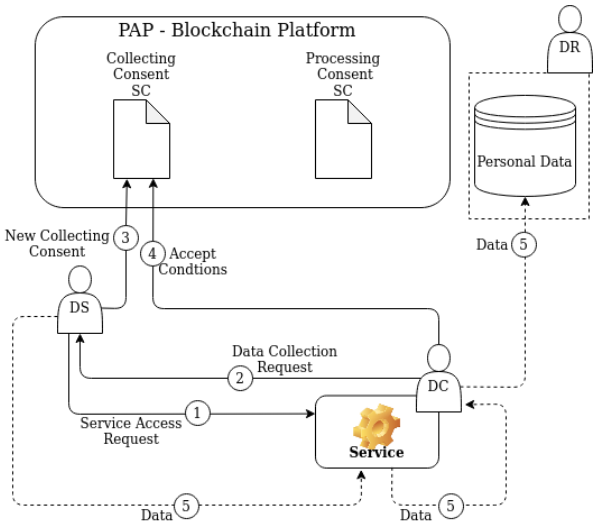


Figure 3: Collection Consent Smart Contract generation and data collection

During the aforementioned procedure, DS generates a new *Collection Consent Smart Contract*. In the new proposal, generating Smart Contracts and interacting with them is addressed by means of transactions which are digitally signed using public-key cryptography and the corresponding pair of secret/public keys. In order to improve DS's anonymity, for each new generated *Collection Consent Smart Contract*, the DS uses a new pair of keys. In this way, the chance of linkage attacks is mitigated. All the used key pairs are stored in a key storage system provided by the proposed service and directly controlled by the user.

3.4.2. Data processing request and Processing Consent Smart Contract generation

Once the *Collection Consent Smart Contract* is valid and DS's personal data has been collected and stored, a DP can request to process that data for certain purposes. Figure 4 depicts this procedure. In particular, the following steps apply:

1. A DP performs a data processing request to the DC regarding the personal data gathered from the DS.

2. The DC generates a new *Processing Consent Smart Contract* for that specific DP. If a *Processing Consent Smart Contract* bounded to DP already exists, DC only adds a new processing purpose to the actual Smart Contract. Whatever the case is, in the contract, the DC specifies which DS's personal data can be processed, the processing purpose and the period of time of this consent
3. The DC asks the DS to agree on the aspects of the new processing purpose and to validate the Smart Contract.
4. If the DP agrees also on the conditions of the new generated processing purpose, it can then request DS's personal data to the DR in order to process it.

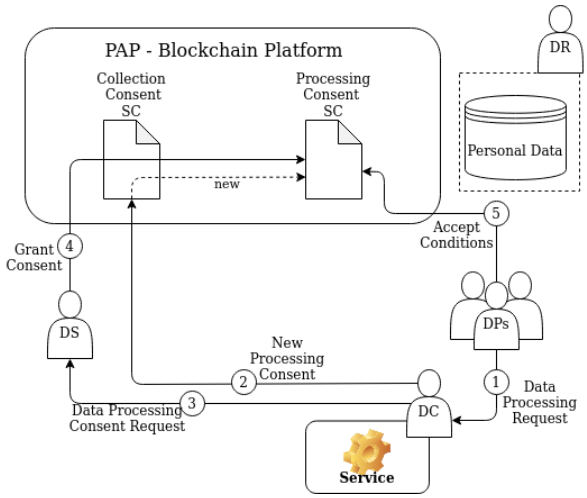


Figure 4: Data processing request and Processing Consent Smart Contract generation

Note that a *Collection Consent Smart Contract* may have a list of default purposes (i.e., *defaultPurposes* argument). In such manner, if a DP performs a request to process DS's personal data for a purpose which is already included in the *Collection Consent Smart Contract*, and the DP is not included in the *blacklist*, it is not needed that the DS validates the *Processing Consent Smart Contract*, therefore, steps 3 and 4 of this procedure are not required.

3.4.3. Data access and retrieval

Once the DS's personal data is collected, stored and guarded by the DR, different actors may be interested in accessing and retrieving it for different purposes. For example, a DP may be willing to process that data, or the DS, to whom the data belongs, may be interested in checking her own collected data. Whatever the case is, the procedure used to access DS's personal data is based on a XACML extension that enforces the access control over the protected data. Figure 4 shows this procedure. In particular, the following steps apply:

1. When an entity sends a data access request to the DR, it is intercepted by the Policy Enforcement Point (PEP) of the XACML-based access control system.
2. The PEP converts the data access request into a XACML authorization request that is forwarded to the Policy Decision Point (PDP).
3. The PDP evaluates the authorization request against the policies (i.e, Smart Contracts) it is configured with. Depending on who the entity requesting the data is, the *Collection Consent Smart Contract* of a *Processing Consent Smart Contract* will be used. In particular, if the request is performed by the DS, the *Collection Consent Smart Contract* will be used (step 3a); on the contrary, if the request is performed by a DP, the *Processing Consent Smart Contract* linked to that DP will be used (step 3b). Smart contracts are acquired via the Policy Administration Point (PAP) by searching them in the blockchain.
4. If needed, the PDP uses any underlying Policy Information Points (PIP) available to retrieve any additional attribute value which could be relevant to reach a final decision regarding the data access request.

5. Finally, the PDP reaches a decision (Permit / Deny / NotApplicable / Indeterminate), and returns it to the PEP, who will then give access or not to the requested data.

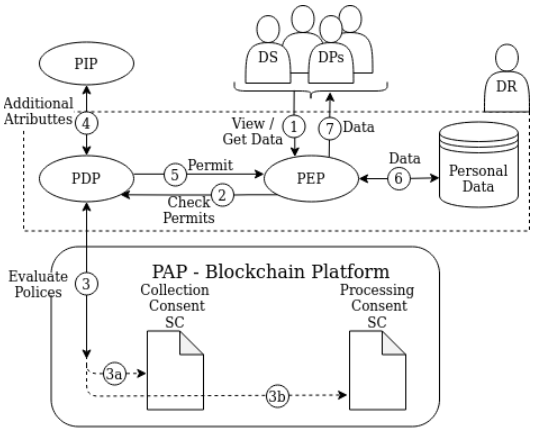


Figure 5: Data access and retrieval

3.4.4. Consent management

According to the requirements described in section 3.1, DSs have rights over their collected data, thus, at a certain point of time, they may be willing to modify their given consents, revoke them, request the collected data to be erased, etc. In the same way, DCs also have the right to revoke a given processing-consent to a DP, as the data processing is done under their responsibility. In the following, the set of actions that can be done over the different generated Smart Contracts are described:

- *Revoke Consent:* A DS can revoke her consent for collecting and processing her personal data at any time. In order to revoke a processing consent given to a certain DP, the DS must use the *revokeConsent()* method implemented in the *Processing Consent Smart Contract*. This method can also be used by the DC in charge. Similarly, for revoking a data-collecting consent given to a certain DC, the DS must use the *revokeConsent()* method implemented in the *Collection Consent Smart Contract*. When this method is called, the Smart Contract performs recursive calls to the *revokeConsent()* method of all the *Processing Consent Smart Contract* which are bound to it. Therefore, when a data-collecting consent is revoked, all the linked data-processing consents are also revoked.

- *Erase Data:* DSs can use the *Collection Consent Smart Contract*'s *eraseData()* method to remove their collected personal data. More specifically, this method sets the *Erasure* flag to "true", forcing the DC to erase the collected personal data without undue delay. The corresponding DC is fully responsible of this action, therefore, if the data is stored using a third party as DR, the DC must guarantee that the data destruction process is performed correctly. Note that running this method implies calling the revoke consent method too.

- *Modify Data:* DSs can modify at any time which of their personal data can be collected or processed for certain purposes by using the *Collection Consent and Processing Consent Smart Contracts*' *modifyData()* method. In particular, when a DS uses the *modifyData()* method of a *Collection Consent Smart Contract* to make the subset of personal data that can be collected less permissive than the previous one, in turn, the attached *Processing Consent Smart Contracts* get a recursive call to their *modifyData()* method to modify their respective subsets of personal data that can be processed.

3.4.5. Consent audit

A main objective of this new proposal is to provide Service Providers with a mechanism that enables them to demonstrate that they have the proper consents to collect and process DS's data in the case of an audit. These audits are carried out by Supervisory Authorities (SAs), this is "independent public authorities responsible of monitoring the application of the GDPR, in order to protect the fundamental rights and freedoms of natural persons in relation to processing and to facilitate the free flow of personal data within the Union" [Art. 51].

The proposed system fulfills this objective by means of three main points: i) all the agreements reached between DSs, DCs, and DPs are declared inside Smart Contracts; ii) Smart Contracts are published in the blockchain, hence, they are publicly available; and iii) due to the blockchain's persistence and immutability properties, Supervisory Authorities are able to check the life-cycle of a certain consent/agreement in the case it is required.

For example, if a SA suspects that a certain DC is collecting more personal data than she is allowed or she is collecting that data without consent, the SA may follow the next procedure (see Figure 6a):

1. The SA checks which data is being collected by the DC. This is achieved by requesting this information to the corresponding DR.
2. The SA looks for the related *Collection Consent Smart Contract* in the blockchain and verifies that it exists and that it is currently valid (using the Smart Contract's *verify()* method).
3. If the Smart Contract is not currently valid, the SA checks the life-cycle of the agreement in order to validate whether she had a previous consent that allowed the collection of the data when it was still valid.
4. If the *Collection Consent Smart Contract* is valid, the SA checks whether the DC has collected more data than it was authorized to. This is done by matching the stored data by the DR with the values stored in the *Collection Consent Smart Contract*'s *data* field.
5. Finally, if the SA detects that the DC has collected personal data without consent or that this entity is keeping data for which it does not have a valid consent any more, the SA will impose the corresponding administrative fines.

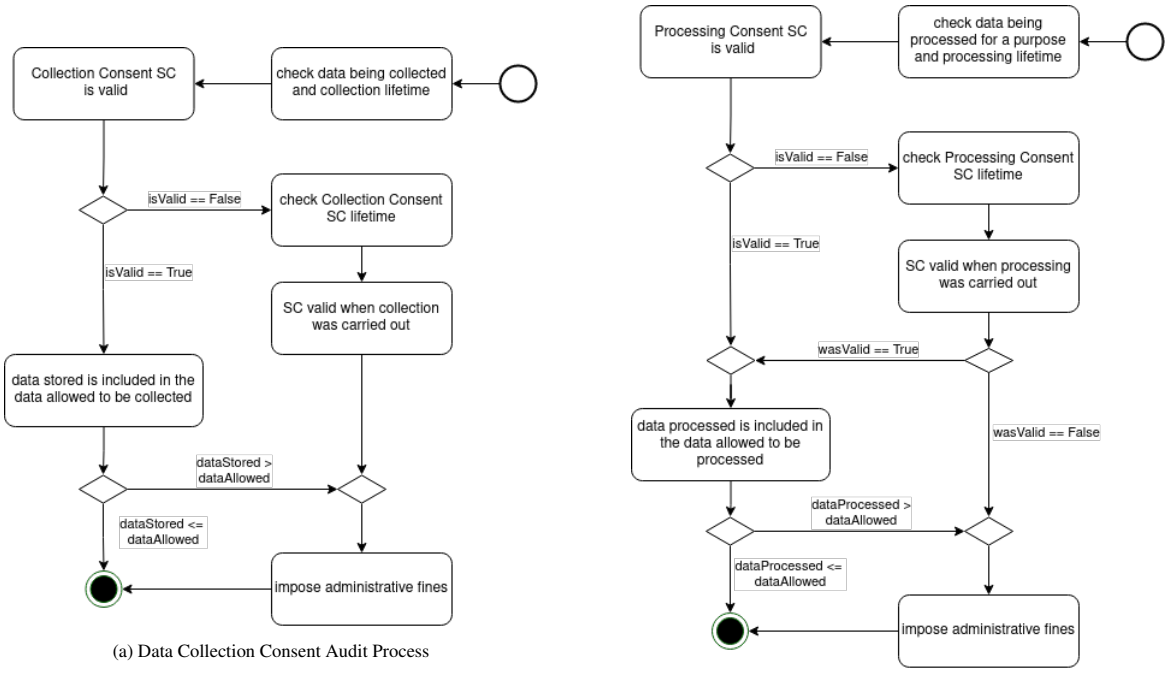
Similarly, if a Supervisory Authority suspects that a certain DP is misbehaving when processing some personal data, this entity may follow the next procedure (see Figure 6b):

1. The SA checks which data is being processed by the DP and for which purpose.
2. The SA looks for the corresponding *Processing Consent Smart Contract* in the blockchain and verifies that it exists and that it is currently valid (using the Smart Contract's *verify()* method).
3. If the Smart Contract is not currently valid, the SA checks the life-cycle of the agreement in order to validate whether the processing process was carried out when it was still valid.
4. If the *Processing Consent Smart Contract* is or was valid when the processing was carried out, the SA checks whether the DP has processed more data than it was authorized to. This is done by matching the processed data with the values stored in the *Processing Consent Smart Contract*'s *data* field.
5. Finally, If the SA detects that the DP has processed personal data without the proper consent, it will impose the corresponding administrative fines to this misbehaving entity.

3.5. Integration of the new proposal in a real-world setting

The new scheme presented in this work is the core component of a larger system intended to allow DSs to control and manage all the consents given to the SPs to collect and process their personal data, along with the collected data itself. Even though this extensive system is still to be defined, we next briefly introduce how we envisage it to be, in order to better contextualize the new proposed scheme.

A main point to be considered is that, in the expected large system, two software components are responsible of managing all the DSs' interactions. In Figure 7, we depict the specific case in which those two software components are a web browser plug-in and an smartphone application. More specifically, a DS is expected to: i) use her web browser (equipped with our plug-in) to consume the SPs' services; and ii) use her smartphone application to control



(a) Data Collection Consent Audit Process

(b) Data Processing Consent Audit Process

Figure 6: Consent audit process

and manage the access to her data and to her consents (i.e., smart contracts). Note that this behaviour is aligned with other common web-based scenarios such as buying a certain good in an electronic commerce while browsing the web and authorizing the corresponding payment by means of a bank's smartphone app.

According to this scenario, the following subsections detail the different steps that a DS must perform in order to use the new proposal. In particular, the user registration, service access, data processing and data/consent management steps are next discussed.

3.5.1. User registration

A DS willing to use this system must first install and set-up the smartphone application. During this process, the DS obtains a set of credentials, this is, a root secret and public key pair (SK_{DS_0}, PK_{DS_0}), which are used to derive new key pairs and to authenticate herself in the following interactions with the system. This root key pair is unique and might be stored in the smartphone's secure storage (e.g., Samsung's Knox Vault[2]).

Next, the DS must install the proposed plug-in in the different web browsers she plans to use. During this installation, a new master key pair (SK_{DS_i}, PK_{DS_i}) is generated from the root credentials (SK_{DS_0}, PK_{DS_0}) for each different web browser i. All these master key pairs (SK_{DS_i}, PK_{DS_i}) are stored and managed by a decentralized multi-platform *Key Storage System*, such as the one presented in [36]. In the last step of the plug-in set-up, the DS is requested to fix a set of pre-defined privacy preferences that are used to automatize the generation of collection consents with the different DCs. At the current point, the method that should be used to gather those privacy preferences is fully open and it requires further work in order to find the best option. In any case, a plausible possibility would be to show a list of data categories together with check-boxes, and allow the DS to select those type of elements that she wants to share with the DCs.

3.5.2. Service access

When a DS uses her web browser to consume a certain service offered by a DC, the installed plug-in, in a transparent way to the user, runs the *Collection Consent Smart Contract generation* procedure described in Section 3.4.1.

[2]Knox Vault: `https://docs.samsungknox.com/admin/whitepaper/kpe/knox-vault.htm`

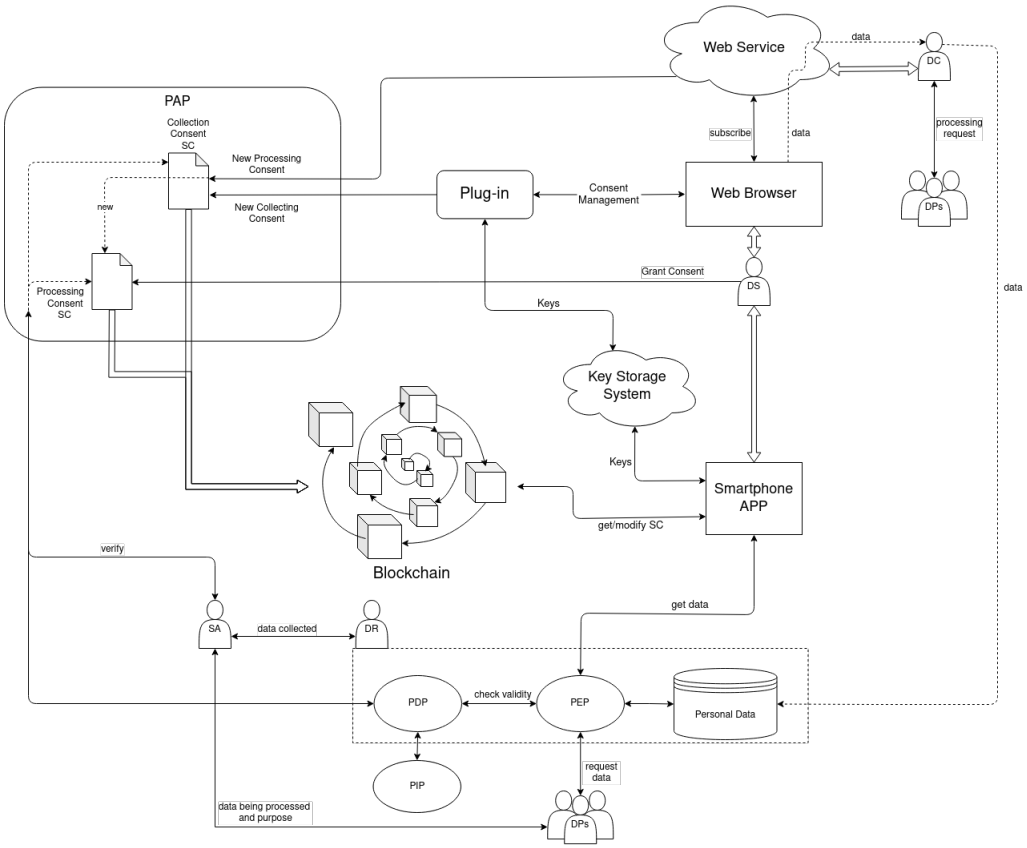


Figure 7: Integration of the new proposal in a real-world setting

By means of this procedure, the DC asks first for permission to collect DS's personal data in exchange for accessing the requested service. This data collection consent request is filled with a set of conditions that the plug-in checks with the set of DS's pre-defined privacy preferences. If those conditions match, the plug-in generates a new key pair $(SK_{DS_{i,j}}, PK_{DS_{i,j}})$ from the stored master key pair (SK_{DS_i}, PK_{DS_i}). The new key pair is stored in the *Key Storage System*.

Next, the plug-in uses the newly generated secret key $SK_{DS_{i,j}}$ to spawn a fresh *Collection Consent Smart Contract*. Then, the DC verifies whether this smart contract fulfills its requirements. If that verification succeeds, the DC starts collecting DS's personal data, and the DS starts consuming the service.

In the case that the data collection consent conditions are not matched with the DS's pre-defined privacy preferences, the plug-in shows those conditions to the DS, and she is required to accept them explicitly before continuing with the process.

3.5.3. Data processing

Once a *data-collection consent* has been agreed between a DC and a DS (and the linked *Collection Consent Smart Contract* has been published in the blockchain), a certain DP can request the corresponding DC to process some of the collected data. This data processing request issued by the DP is driven by the *Processing Consent Smart Contract generation* procedure described in Section 3.4.2.

By means of this procedure, the DC builds a *Processing Consent Smart Contract* for that specific DP that states which data can be processed and for which purpose. If the smart contract linking the DC with the DP already exists, it only adds a new processing purpose to it. Whatever the case is, this *Processing Consent Smart Contract* verifies that the processing purpose of the *data processing request* issued by the DP is included in its *default-purposes* list before allowing the DP to gather any data. If this is the case, the *valid flag* of the smart contract is automatically set to "true".

In the case that the DP's purpose is not found in the smart contract's *default-purposes* list, the DC requires the affected DS to validate the new processing purpose by means of her smartphone application. A possible way of achieving this validation would be to set the smartphone app to periodically perform push requests to the blockchain in order to find all the *Processing Consent Smart Contracts* that are linked to the DS and have their *valid flag* set to "false". For each non-valid *Processing Consent Smart Contract* found, the smartphone application would show a message to the DS asking her to explicitly grant or deny that *processing consent*.

3.5.4. Data and consent management

The smartphone application is expected to allow the DSs to keep track of all the consents that they have agreed with third parties, and the data that those parties have collected and processed while backed by the corresponding consents. The procedures that apply to these two cases are described in Sections 3.4.3 and 3.4.4.

Focusing on the data management procedure, a DS uses her smartphone application to identify herself as the data owner and request the DR to retrieve all her personal data which was collected by a certain DC. In order to do that, the following steps are followed: i) the smartphone application obtains from the *Key Storage System* the cryptographic keys ($SK_{DS_{i,j}}, PK_{DS_{i,j}}$) that were used for deploying the *Data-Collection Consent Smart Contract*; ii) the smartphone application uses the retrieved secret key $SK_{DS_{i,j}}$ to compute a digitally signed temporal proof; iii) the smartphone builds a data access request containing the digitally signed temporal proof and sends it to the DR through the proposed XACML extension; iv) the *Policy Enforcement Point (PEP)* receives the data access request and asks the *Policy Decision Point (PDP)* for a grant/deny access decision; v) the PDP uses the *Policy Administration Point (PAP)* to get from the blockchain the corresponding *Collection Consent Smart Contract*; and vi) the PDP, in a two-step operation, checks whether the smart contract contains the corresponding $PK_{DS_{i,j}}$, and it ascertains whether the requesting entity has the proper rights to access the protected data.

Regarding the consent management procedure, a DS can use her smartphone application to interact with the different consents/smart contracts that she has created. For this purpose, the smartphone application is able to retrieve all the required cryptographic keys from the *Key Storage System*, use those keys to find the corresponding smart contracts that have been published on the blockchain, and interact with them by means of the methods they offer.

3.5.5. Consent audit and legislation enforcement

When a DS (or the SA itself) suspects that a certain DC/DR or DP has misbehaved with the consent given, she can request the SA to perform an auditing process. The SA will then follow the steps described in Section 3.4.5.

If the suspicious entity is a DP, the auditing process will first lead the SA to check which data has been processed by the DP and for which purpose. Then, it will retrieve and use the related *Processing Consent Smart Contract* to verify whether the consent was valid when the processing of the data was performed. In the same way, if the DR is believed to be illegally keeping some personal data on behalf of a DC, the SA will check the validity and extend of the given consent by means of the related *Collection Consent Smart Contract*.

In both cases, if after completing the auditing process the SA detects that personal data has been collected or processed without the proper consent, she will impose the corresponding administrative fine to the DC and/or the DP.

4. Discussion

This section provides an analysis and discussion of the described platform. This includes analyzing GDPR-compliance, functionality, security and privacy, performance and, finally, studying the costs that the use of Smart Contracts will introduce.

4.1. GDPR-Compliance analysis

In section 3.1.1, we defined the main requirements that the proposed system should successfully fulfill in order to be GDPR-compliant. Now, in this section, those requirements are put to the test. The provided discussion consists of four propositions with several claims that support their fulfillment.

4.1.1. Proposition-1: A data-collection consent represents the agreement between DS and DC regarding the collection of DS's personal data

When a DS requests access to a certain service, a data-collection consent contract is created between this entity and the DC that represents the service provider. This contract specifies: i) which personal data can be collected; ii) the period of time it can be kept; iii) the identity of the DC; and iv) the identities of the DRs. This contract is valid as proof of the DS's explicit consent given to the DC to collect her personal data.

This proposition is directly related to the GDPR requirements R1.1 and R1.2, which state that a DC must have the DS's explicit consent in order to collect her personal data. It is supported by the following claim.

Claim 1. *A new Collection Consent Smart Contract is created every time a DS requests access to a service which collects her personal data.*

Proof. When a DS request access to a certain service, the service provider (i.e., the DC) may request that some of DS's personal data be collected in exchange. In this case, the proposed system allows the DS to generate a new Smart Contract that specifies which personal data the DC will be able to collect and for how long it will be kept. The Smart Contract also holds the identities of the DC and all the DRs that will store the collected data. In this way, the created Smart Contract represents the explicit consent given by the DS to a certain DC to collect her personal data. □

4.1.2. Proposition-2: A data-processing consent represents the agreement between DS, DC, and DP regarding the processing of DS's personal data

When a DP request to process DS's collected personal data, the DC in charge of collecting that data can create a new data processing consent contract specifying: i) which of DS's personal data can be processed and for which purpose; ii) the period of time it can be kept; and iii) the identities of the DC and the DPs. This contract requires that the DS gives her explicit or implicit consent to be valid.

This proposition is directly related to the GDPR requirements R1.3 and R1.4, which state that a DP must have the consent of the DSs in order to process their personal data. It is supported by the following two claims.

Claim 2. *A new Processing Consent Smart Contract is created every time a new DP requests to process some DS's personal data collected by a DC. A reference to this new smart contract is stored into the Collection Consent Smart Contract that encompasses the whole data management process.*

Proof. When a DP requests to process DS's personal data for the first time to a DC, the DC creates a *Processing Consent Smart Contract* that holds the consent for processing a subset of DS's personal data for a specific purpose and period of time. This Smart Contract is deployed through the *Collection Consent Smart Contract* that was previously agreed between the DS and the DC by using the *newPurposeConsent()* method. This method adds the address linked to this new Smart Contract into the *Purposes* list of the *Collection Consent Smart Contract*. Once a *Processing Consent Smart Contract* binding a certain DP with the *Collection Consent Smart Contract* exists, future data processing requests made by the same DP will be added to the existing Smart Contract. □

Claim 3. *A Processing Consent Smart Contract requires DS's explicit or implicit approval in order to be valid.*

Proof. When the DC creates or adds a new processing purpose in a *Processing Consent Smart Contract*, that specific processing purpose may be already recorded in the *defaultPurposes* list of the linked *Collection Consent Smart Contract*. If that is the case, the DP gets DS's implicit consent for processing her personal data for that specific purpose. Otherwise, the DP requires that the DS explicitly approves the data processing request by running the *grantConsent* method of the *Processing Consent Smart Contract*. □

4.1.3. Proposition-3: A DS can interact with her data-collection and data-processing consents according to the GDPR's rights

The proposed system allows DSs to interact with the generated data-collection and data-processing consents, which are backed by means of smart contracts, in the following ways:

- A DS may request to remove all her collected personal data at any time (Art. 17 Right to erasure).

- A DS may limit which of her personal data can be collected, which can be processed, and for which purposes (Art. 18 Right to restriction of processing).
- A DS may revoke her consent at any time (Art. 21 Right to Object).

According to the presented proofs in Claims 1 and 2, data-collection and data-processing consents are stored into Smart Contracts. Furthermore, these Smart Contracts implement a set of methods that allow DSs to interact with them allowing to exercise their rights over their personal data.

This proposition is directly related to the GDPR requirements R1.5.1 and R1.5.5, which state that DSs must be able to apply their rights over their personal data anytime. It is supported by the following claim.

Claim 4. *Collection and Processing Consent Smart Contracts implement a set of methods that allow DSs to exercise their rights over the reached agreements.*

Proof. As shown in sections 3.3.1 and 3.3.2., *Collection and Processing Consent Smart Contracts* implement a set of methods allowing DSs to:

- Revoke an specific data-collection or data-processing consent at any time (*revokeConsent()*).
- Modify which personal data can be collected or processed (*modifyData()*).
- Request the DC to erase their collected personal data (*eraseData()*).
- Revoke the processing consent for an specific purpose and remove it from the *defaultPurposes* list (*revokeConsentPurpose()*).

□

4.1.4. Proposition-4: A DC can revoke at any time all the data-processing consents that this entity has issued

According to the presented proof in Claim 2, when a DP request to process DS's personal data, the DC that has collected that data generates a processing consent which is stored into a Smart Contract. In the same way, at anytime, the DC can revoke her consent given to the DP to process DS's data and overrule the previous agreement.

This proposition is directly related to the GDPR requirement R1.7, which states that DC must be able to revoke their consents to DPs to process DS's personal data anytime. It is supported by the following claim.

Claim 5. *Processing Consent Smart Contract implements an specific method that allows the DC in charge to revoke her consent for processing the collected data.*

Proof. The DC that has generated a certain *Processing Consent Smart Contract* may use the *revokeConsent()* method to revoke its consent. □

4.2. Functionality analysis

Functional requirements R2.2, R2.3 and R2.4 presented in section 3.1.2 are studied in this subsection. Remaining functional requirements R2.1 and R2.5 will be tackled in further subsections. The following discussion consists of a set of propositions, each one addressing a certain considered requirement. Each proposition has a set of claims to support its fulfillment.

4.2.1. Proposition-5: A DS can access all the data-collection and data-processing consents that she has agreed from a single resilient access point

A DS can use the asymmetric key pairs generated and stored in the key storage system to find and access all her published Collection Consent Smart Contracts and all the corresponding Processing Consent Smart Contracts. In this way, the proposed system allow DSs to find and retrieve all the collection and processing consents that they have given. This proposition is directly related to the functional requirement R2.2, which states that the proposed scheme must provide a single resilient point of access to all the consents that the DS has given. In this case, this access point is the blockchain. This proposition is supported by the following four claims.

Claim 6. *A public blockchain is used to store all the data-collection and data-processing consents generated by the proposed system. Public blockchains are resilient by design, provide access transparency, and individuals can interact with them by means of asymmetric key pairs.*

Proof. Each DS has a key storage system that generates and stores all the asymmetric key pairs ($SK_{DS_{i,j}}$,$PK_{DS_{i,j}}$) required to interact with the public blockchain. Blockchains are fully distributed and, hence, resilient by design. Moreover, they work as a single access point by means of the access transparency they provide to users. □

Claim 7. *All the data-collection and data-processing consents which are agreed between the different parties are stored in the public blockchain by means of public and immutable smart contracts.*

Proof. In the proposed system, data-collection and data-processing consents are represented by *Collection Consent Smart Contracts* and *Processing Consent Smart Contract* respectively. Smart contracts are deployed into the public blockchain by means of transactions. Transactions are immutable records in the blockchain digitally signed by means of a public-key algorithm with signing capabilities and an asymmetric secret key $SK_{DS_{i,j}}$. No party can change or tamper with a transaction after it has been recorded into the shared ledger. □

Claim 8. *A DS can find and access a certain smart contract published on the blockchain by knowing the corresponding public key $PK_{DS_{i,j}}$ of the asymmetric secret key $SK_{DS_{i,j}}$ which was used to perform its deployment.*

Proof. All Smart Contracts deployed on the blockchain are public and anyone can access them. As stated previously, Smart Contracts are published by means of digitally signed transactions (see Claim 7). DSs employ asymmetric key pairs ($SK_{DS_{i,j}}$, $PK_{DS_{i,j}}$) to interact with the blockchain. In particular, the private key $SK_{DS_{i,j}}$ is used to digitally sign a transaction to be deployed, while the linked public key $PK_{DS_{i,j}}$ can be used to find the corresponding smart contract on the ledger. □

Claim 9. *Collection Consent Smart Contracts are bound to Processing Consent Smart Contracts by means of the "Purposes" field of the former. In this way, when a certain Collection Consent Smart Contract is found an retrieved, it is straightforward to find and retrieve all the Processing Consent Smart Contracts which are bound to it.*

Proof. Collection Consent Smart Contracts keep track of all *Processing Consent Smart Contracts* bound to them. An entity that retrieves a certain *Collection Consent Smart Contract* is then able to read the *Purposes* list where the addresses of the *Processing Consent Smart Contracts* are stored. All *Processing Consent Smart Contracts* are published in a public blockchain, therefore, by knowing their addresses, they can be straightforwardly found and accessed. □

4.2.2. Proposition-6: The proofs linked to the consents agreed between DSs and DCs/DPs are stored in a fully distributed way

According to the proofs presented in Claims 1 and 7, all the agreements made between DSs, DCs and DPs are encoded into Smart Contracts that are then published on a public blockchain. The Blockchain technology is hosted by a set of anonymous and fully distributed nodes. In this way, all these agreements are not kept in a unique central server. Instead of that, they are shared between all the nodes that maintain the network. This proposition is directly related to the functional requirement R2.3, which states that the proposed scheme must work in a distributed way.

4.2.3. Proposition-7: DSs, DCs, and DPs can interact with all the published data-collection and data-processing consents

All agreements made between DSs and DCs/DPs are encoded into Smart Contracts which are then published in a public blockchain. These Smart Contracts implement methods which can be invoked by the different entities by using transactions digitally signed transactions on the blockchain. This proposition is directly related to the functional requirement R2.4, which states that the proposed scheme must allow DSs, DCs and DPs to interact with and modify all the agreements already made between them. This proposition is supported by the following claim.

Claim 10. *Entities can interact with the published agreements by sending digitally signed transactions to the blockchain.*

Proof. As explained in Section 4.1, Smart Contracts provide methods to interact with them. By design, interacting with a method offered by a smart contract deployed into a public blockchain requires the use digitally signed transactions. In particular, to run a certain method, the calling entity must publish a digitally signed transaction on the blockchain that, in consequence, will be recorded in the ledger and become immutable. □

4.3. Security and privacy analysis

This section focuses on studying the security and privacy requirements of the provided system. Moreover, the functional requirement R2.5 is also addressed here. Like in previous sections, the discussion is organized as a set of propositions, where each proposition may have several claims to support its fulfillment.

4.3.1. Proposition-8: Only authenticated and authorized entities can interact with the consents made between DSs, DCs, and DPs

Entities can interact with the Smart Contracts by using transactions. These transactions are digitally signed and the Smart Contract stores the identifying public keys of the involved entities. As a result, smart contracts can verify whether the entity that is willing to perform a certain action has the proper rights, according to her role and the GDPR rules, to run the corresponding method.

This proposition is directly related to requirement R3.1, which states that an entity must not be able to carry out an action over an agreement for which it has not permission; and to requirement R.3.2, which states that entities must be properly authenticated, which, in turn, is achieved by digitally signing the blockchain transactions. It is supported by the following three claims.

Claim 11. *Smarts contracts safely store the public keys of all the entities involved in the represented agreements.*

Proof. When a new *Collection Consent Smart Contract* representing a certain data collection agreement is created, the public keys of the DS ($PK_{DS_{i,j}}$) and the DC (PK_{DC_j}) involved are stored in its internal state. In the same way, when a new *Processing Consent Smart Contract* is created, the public keys of the DS, DC, and the corresponding DP (PK_{DP_k}) are also stored in its internal state. □

Claim 12. *Each public key belongs to a unique entity and cannot be tampered.*

Proof. Each public key belongs to an asymmetric public/secret key pair that is uniquely linked to a specific entity. For example, the key pair of DS i used to sign the *Collection Consent Smart Contract* with the DC j is the pair ($PK_{DS_{i,j}}$, $SK_{DS_{i,j}}$). Only the party that knows the secret key $SK_{DS_{i,j}}$ of the aforementioned key pair can prove to be the owner of the corresponding public key. Moreover, due to the fact that the transactions recorded on the blockchain are immutable, public keys stored in smart contracts cannot be tampered with. □

Claim 13. *Only authorized entities can perform actions over a smart contract.*

Proof. Each method of a smart contract has a set of authorized roles that limit who can perform that specific action. In this way, before running any action, the smart contract verifies whether the corresponding entity has the proper rights to perform that action according to its role. This process is twofold: first, the smart contract checks that the key pair which is being used to sign the transaction of the specific method call (see Claim 10) matches the stored public key; second, the smart contract checks that this public key is linked to a role that grants access to the required action. □

4.3.2. Proposition-9: An entity that participates in the system cannot falsely claim that they have not performed a certain action (i.e., non-repudiation)

As it has been explained previously, actions carried out over smart contracts are done by means of digitally signed transactions (see Claim 10). Digital signatures, if they are generated correctly using a sound asymmetric cryptosystem and the cryptographic material is kept safe, are assumed to ensure the non-repudiation property (explained in Section 2.3). In this way, a DS may digitally sign a transaction by means of her secret key $SK_{DS_{i,j}}$ and a public-key algorithm with signing capabilities, for example, the ECDSA algorithm, which is included in the current Digital Signature Standard (DSS) issued by NIST[3]. ECDSA is assumed to be secure provided that the signer uses a proper random

[3]Digital Signature Standard (DSS): `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf`

nonce to generate the corresponding digital signature[37]. Regarding the security of the signing key, it is only known by the DS, who is responsible of keeping it secure in the key storage system in order that no adversary can retrieve it. According to this, an entity willing to perform a certain action over a smart contract must digitally sign a blockchain transaction linked to that specific action. Due to the fact that digital signatures ensure the non-repudiation property, it can be stated that no entity can falsely claim that it has not performed a certain action.

This proposition is directly related to requirement R3.3.

4.3.3. Proposition-10: All the consents and all the actions applied to them that may modify their state are generated as permanent and publicly accessible proofs that allow the public tracking of all events

As it has been explained previously, all agreements between DSs, DCs and DPs are encoded into Smart Contracts which are published on a public blockchain. According to the persistence property of this technology, explained in Section 2.3, for extension, these agreements become permanent and publicly accessible.So, according to proofs presented in Claim 10, all interactions with agreements are registered on a public blockchain by means of transactions, which make them permanent and publicly accessible too. This fact allows any entity to reconstruct the life-cycle of any agreement.

This proposition is directly related to requirement R3.4, which states that the system system must provide immutable evidences of all the events which have been carried out. Moreover, this proposition is also directly related to the functional requirement R2.5, which states that the system must keep all actions related to a consent allowing any external entity to reconstruct that consent life-cycle.

4.3.4. Proposition-11: No entity can link a certain consent agreement with a certain DS (i.e., anonymity)

No DS's personal information is stored in the Smart Contracts so it is unfeasible to directly re-identify a certain DS by means of the agreements generated by the proposed system. However, an identity disclosure attack may be possible by using multiple generated agreements as quasi-identifiers. In order to avoid this issue, the proposed scheme uses a new asymmetric key pair each time a DS sets a new agreement with a DC. As a result of that, an adversary cannot link different agreements together and she cannot obtain any additional information that allows her to re-identify the DS.

This proposition is directly related to requirement R3.5., which states that the system must not publicly store DSs' personal data neither provide any way to re-identify them. It is supported by the following three claims.

Claim 14. *DS's personal data is never stored in the Smart Contracts generated by the proposed system.*

Proof. The *Collection and Processing Consent Smart Contracts* employed in the proposed system only store the public key of the asymmetric key pair used to digitally sign the corresponding blockchain transaction. Those Smart Contracts do not store any piece of personal data belonging to the involved DS. □

Claim 15. *The agreements made between DCs and DSs are considered to be Quasi-Identifiers.*

Proof. The agreements between a DS i and a DC j may contain information about DS's interests (e.g., location, political views, etc.). On the one hand, it is not possible to re-identify a certain DS just by analyzing a single agreement made by her. On the other hand, a capable adversary may be successful re-identifying a certain DS if she has access to multiple agreements made between this DS and different DCs. Due to this situation, the agreements made by DSs are considered to be quasi-identifiers. □

Claim 16. *The proposed scheme uses a new asymmetric key pair each time a DS sets a new agreement with a DC.*

Proof. As explained in Claim 6, DSs use a key storage system to generate and store all the crytographic material required to interact with the public blockchain. By using this key storage system, a DS may generate a new asymmetric key pair whenever she wants. In particular, the proposed scheme generates a new pair for each new agreement with a DC. □

4.3.5. Proposition-12: Only authorized entities can obtain DS's collected personal data

The DR is the entity in charge of managing the access requests to the protected personal data. This entity uses a XACML extension to drive all access control process. By using this extension, the DR, before granting access to the requesting actor, verifies that the data requester owns the proper access rights on the blockchain. Those access rights are encoded in the blockchain by means of smart contracts which represent the agreements made between DSs, DCs and DPs (see Proposition-6).

This proposition is directly related to requirement R3.6., which states that the system must guarantee that only properly authorized actors should have access to DS's collected personal data. It is supported by the following claim.

Claim 17. *The DR manage the access to the DSs' collected data by means of the proposed XACML extension.*

Proof. As shown in the architecture of the proposed scheme, the DR is the entity in charge of keeping safe the personal data collected by the DC and managing any access to it by means of the proposed XACML extension. In order to achieve this, the XACML extension uses the blockchain technology as the *Policy Administration Point (PAP)* in the access control architecture. Then, when the *Policy Enforcement Point (PEP)* receives a data access request from an entity, it asks the *Policy Decision Point (PDP)* for an grant/deny access decision. The data access request contains a digitally signed temporal proof using the entity's secret key SK_e that validates the right ownership of the corresponding public key PK_e. Next, the PDP uses the PAP to get from the blockchain the corresponding consent smart contract (i.e., a *Collection Consent Smart Contract* if the requesting entity is a DS/DC or a *Processing Consent Smart Contract* if it is a DP). After that, in a two-step operation, the PDP first checks whether the obtained smart contract contains the corresponding PK_e; and, finally, it ascertains whether the requesting entity has the proper rights to access the protected data. □

4.4. Performance analysis

Functional requirement R2.1 states that the new proposal must be fast and efficient. In particular, it should be more lightweight than existing ones. In order to validate this requirement, the new scheme is compared with those solutions in the literature that: i) are properly documented so performing a comparison is possible; and, ii) whose main objectives and functionality are similar enough to the goals considered in this paper.

Table 4 focuses on the functionalities offered by the current schemes in the literature that effectively consider the GDPR compliance. In particular, this table summarizes whether the different aspects of the GDPR regulation are tackled by each proposal or not. As it can be seen, only two works differentiate between data-collection and data-processing consents. These are Truong et al. and Nessie et al. Next, from these two, only the former fulfills all the functionalities that we have established as essential in this paper. Regarding Nessie et al., it is not clear whether the collected data can be erased or not, due to the fact that the data is sent to each processor who has the consent to process it and no further explanations are given in this regard. For this reason, the corresponding property is left as unknown in this case.

Proposal	Grant Collection Consent	Revoke Collection Consent	Grant Processing Consent	Revoke Processing Consent	Access Data	Modify Data	Delete Data	Update Polices
Our Proposal	✓	✓	✓	✓	✓	✓	✓	✓
Truong et al. [5]	✓	✓	✓	✓	✓	✓	✓	✓
Nessie et al. [9]	✓	✓	✓	✓	✓	✓	?	✓
Wirth and Kolain [4]	X	X	✓	X	✓	X	X	X
Barati et al. [24]	X	X	✓	X	✓	✓	X	X
Wang et al. [10]	X	X	✓	✓	✓	✓	✓	✓

Table 4: Comparison of functionalities

Focusing now on the performance analysis, it is worth mentioning that most of the schemes in the literature do not provide an experimental performance analysis at all, or, if they do, it differs considerably from the study that we have carried out on the new proposal. Therefore, in order to fairly compare the performance of our scheme with the rest of the solutions present in the literature, our performance analysis focuses only on the three most relevant operations

among the eight indicated above. More specifically, we compare the number of steps required to perform each one of these three main operations.

The three integral operations being considered are: i) the "get processing consent" process, from which a DP obtains the authorization of the DS and the DC to process DS's personal data; ii) the "revoke processing consent" operation, which is used by a DS or DC to revoke a processing consent previously given; and iii) the "access data" method used by the DP and the DS to get access to the collected data. These three procedures have been selected due to the fact that they are covered by most systems in the literature and, also, these are the procedures explained with a significantly better level of detail in those proposals.

Note that, in our work, we have stressed the importance of the data-collection consent which is established between a DS and a DC, a contract that must be valid in order to start collecting DS's personal data. Nevertheless, as it can be shown in Table 4, most schemes do not distinguish between the data-collection and data-processing consents, and they do not differentiate between a DC and a DP either. Due to this situation, in this analysis we do not evaluate the Grant and Revoke data-collection consent processes, among others.

Proposal	**Get Processing Consent**								
	DS-DP	DS-DC	DS-BC	DP-BC	DC-BC	DC-DP	DS-DR	DC-DR	DP-DR
Our Proposal	0	0-1	0-1	1	1	1	0	0	0
Truong et al. [5]	2	2	0	0	4	0	0	0	0
Nessie et al. [9]	2	1	1	0	2	2	0	0-2	0-2
Wirth and Kolain [4]	1	0	2	1	0	2	0	0	0
Barati et al. [24]	0	2	1	1	2	4	0	0	0
Wang et al. [10]	2	0	5-6	0	0	0	2	0	0

Table 5: Comparison of Get Processing Consent workflow steps

Centering the study on the *Get Processing Consent* operation, Table 5 shows the number of steps that each entity (i.e., DS, DC, DR and DP) has to perform in order to complete this procedure.

In order to fill this table, certain assumptions regarding some proposals have been made. Those are next detailed:

- Nessie et al. [9]: In this work, DCs are supposed to store DSs' personal data. Therefore, when a DP request to process the collected data, this data is transferred to the DP, the DS is informed, and a new identity and a new usage contract is created by the DS. If an external DR were involved in this process, this would imply adding four more steps for each run of this operation. Since the authors do not specify this point, Table 5 depicts both possibilities.

- Wang et al. [10]: This work belongs to the category of proposals in which DSs are willing to share her personal data with other parties. On the other hand, our new proposal runs under the assumption that a service provider collects the personal data of a DS for further processing in exchange for using a product that it offers. Due to that particularity, Wang et al. requires DSs to deploy a first Smart Contract through which they will add new DPs, add new data or revoke consents in order to control the access to their data. This situation does not happen in our new proposal. In order to compare both proposals, Table 5 depicts both the original solution presented by the authors, and it also depicts the case in which this first Smart Contract is not generated.

- Barati et al. [24]: This scheme introduces a new entity named "Contract Activator", which is charge of deploying all Smart Contracts and interact with the other entities. In our study, we consider that this entity acts as DC, due to its similarities with the DC depicted in our proposal. Moreover, in this work the personal data to be processed is not sent to DPs, it is processed in containers instead. In consequence, in our study we consider that the entity that runs these containers acts as DR.

As shown in Table 5, in all the analyzed proposals the DS is performing three or more interactions with the system (see the three first columns) than in our new scheme. In particular, the new proposal only requires two interactions in the worst case (i.e., when a DP requests to process DS's personal data for a purpose not included in the default purposes list of the *Collection Consent Smart Contract*), and zero interactions in the best case (i.e., when the processing purpose

already is in the aforementioned list). This aspect is of paramount importance due to the fact that the lower the number of interactions, the more lightweight the system is from the point of view of the user.

On the other hand, economical costs are directly related to the number of interactions between any entity and the blockchain. Taking this into account, [4] and [9] require three blockchain operations. Our new proposal also requires three blockchain operations in the worst case, but it requires one less blockchain operation in the best case. The rest of the studied schemes require four or more blockchain operations.

Summarizing all the points above, this analysis shows that the *Get Processing Consent* operation designed for our new scheme is the most efficient in terms of overhead at the DS's side and theoretical economical costs among the current schemes in the literature.

Proposal	**Revoke Processing Consent**	
	DS-BC	DC-BC
Our Proposal	1	1
Truong et al. [5]	4	4
Nessie et al. [9]	1	-
Wirth and Kolain [4]	-	-
Barati et al. [24]	-	-
Wang et al. [10]	1	-

Table 6: Comparison of Revoke Processing Consent workflow steps

Regarding the *Revoke Processing Consent* operation, this method only requires the intervention of one entity. This entity can be a DS who does not want her personal data to be processed anymore, or it can be the DC in charge of the collected data, which does not want that a certain DP processes it anymore. According to that, table 6 depicts the number of steps an entity must follow in order to complete by their own the revoke processing consent process.

As it can be seen in this table, only [5] and our new proposal allow both, the DC and the DS, to revoke a processing consent. The rest of the studied schemes do not consider this operation. Going deeper into this point, [5] requires four steps to complete this operation, while our new proposal requires only one step.

Proposal	**Access Data**					
	DP-DR	DP-DC	DP-BC	DC-BC	DR-BC	DC-DS
Our Proposal	2	0	0	0	1	0
Truong et al. [5]	2	0	2	0	2	0
Nessie et al. [9]	0-2	0	2	0	0-2	0
Wirth and Kolain [4] 0	2	2	0	0	0	0
Barati et al. [24]	2	3	1	1	0	1
Wang et al. [10]	2	0	6	0	0	0

Table 7: Comparison of Access Data workflow steps

Regarding the *Access Data* operation, Table 7 shows the number of steps each entity has to follow in order to allow a DP to get access to the collected personal data. As it can be seen, half of the proposals require at least 6 steps to complete the full process. Among the schemes that require less steps, there are: [9] with 2 steps (note that, if an external DR is involved, it then requires 6 steps); [4] with 4 steps; and our new proposal with only 3 steps.

Proposal	**Get Processing Consent**	**Revoke Processing Consent**	**Access Data**
Our Proposal	3-5	1	3
Truong et al. [5]	8	4	6
Nessie et al. [9]	8-12	1	2-6
Wirth and Kolain [4]	6	-	4
Barati et al. [24]	10	-	8
Wang et al. [10]	8-9	1	8

Table 8: Comparison summary

Finally, in order to summarize all the findings of this study, Table 8 depicts the corresponding comparison results. As this table reflects, the total number of steps required by our new scheme is lower than in the rest of the analyzed proposals. This shows that the new proposal is the most lightweight in terms of entity interactions and theoretical efficiency.

4.5. Smart contract costs

This section evaluates the costs linked to the use the Smart Contracts. Analyzing those costs is of paramount importance due to the fact that they have a direct incidence on the feasibility of the proposed system when deployed in a real scenario.

In order to perform this evaluation in a plausible setting, the new scheme has been implemented, and the personal blockchain *Ganache*[4], which is widely used for fast and easy Ethereum distributed application development, has been used. Ganache allows developers to deploy Smart Contracts, run commands, and inspect the state of the data while controlling how the chain operates.

The different Smart Contracts in use by the proposed system have been coded using the Solidity language[5] and they have been compiled by means of the *Truffle suite*[6]. This is a well-known ecosystem for Web3 development that provides a development environment, asset pipeline, and testing framework for developing Smart Contracts using the Ethereum Virtual Machine (EVM). In particular, the aforementioned personal blockchain *Ganache* is part of the *Truffle suite*. Also, note that Web3[7] is a modular, reactive, type safe Java and Android library for working with Smart Contracts and integrating with clients (nodes) on the Ethereum network. This tool is used to interact with the Smart Contracts and the blockchain.

In the following, we test and validate the correctness of the new proposed scheme on the introduced blockchain infrastructure. This infrastructure was deployed in a single laptop. The performed evaluation comprises: i) studying some relevant EVM compatible networks in terms of efficiency and cost to run the proposed scheme on one of them; and ii) evaluating the cost in gas of deploying the Smart Contracts and the cost of performing their related operations.

4.5.1. Analysis of EVM compatible networks

Every transaction that is used to deploy a contract or invoke a function of a Smart Contract requires the payment of a fee to compensate the mining node for running the transaction and saving it on the blockchain. EVM compatible networks use gas to represent this fee. Users can purchase gas from the mining nodes by paying with the network token (in the Ethereum network, this token is the *Ether*). The gas and the network token are two distinct terms. In particular, the gas indicates a constant cost of performing an operation on a blockchain network. On the other hand, the network token is a volatile virtual currency, which is used to pay for the network resources.

This section studies some of the most relevant EVM compatible networks in terms of latency, throughput, and gas cost. Due to the difference in the token price and gas cost between the different networks, in order to evaluate the transaction cost among them, we use the transaction "base fee" cost, which is of 21000 gas[8]. This gas covers the cost of an elliptic curve operation to recover the sender address from the signature as well as the disk and bandwidth space for storing the transaction.

Table 9 shows the comparison of the studied blockchain platforms considering their key features. As it can be seen, the most efficient blockchain, in terms of cost, is the *BitTorrent Chain*. However, it is still a young network (it was deployed on 9th December, 2021), and, currently, this network suffers from a clear lack of information and usage statistics. Due to that, this network is not considered a valid option in this study. In the same way, *Ethereum Mainnet*, *Binance Smart Chain*, *Avalanche Chain*, and *Phantom Opera* are not considered valid options either, due to the fact that they generate very high costs when deploying Smart Contracts.

[4] https://trufflesuite.com/ganache/
[5] https://soliditylang.org/
[6] https://trufflesuite.com/truffle/
[7] https://docs.web3j.io/
[8] https://eth.wiki/en/fundamentals/design-rationale

Regarding the two blockchain networks left, *Polygon Chain* and *HECO Chain*, according to *Polygon Chain*'s documentation[9], this network supports up to 10,000 tps, having a peak of 106 tps (16th of June, 2021). On the other hand, *HECO Chain*, according to its documentation[10], only supports up to 2000 tps, having a peak of 50 tps (10th of May, 2021). Also, the block mining time of *Polygon Chain* is 2s, while *HECO Chain*'s is 3s, making *Polygon Chain* the most efficient network.

Network	Consensus Protocols	Token	Mining Block Time	Transaction Throughput	Gas Cost	Token Price	Tx Base Fee Cost
Ethereum Mainnet [11]	PoW	ETH	13s	14 tps	35 Gwei	3146.36 $	2.31 $
Binance Smart Chain [12]	DPoS	BNB	3s	54 tps	6.5 Gwei	416.23 $	0.06 $
Polygon Chain [13]	PoS	MATIC	2s	30 tps	60 Gwei	1.61 $	0.002 $
Fantom Opera Chain [14]	Lachesis	FTM	1s	9.5 tps	360.5 Gwei	1.40 $	0.01 $
BitTorrent Chain [15]	PoS	BTT	2s	0.04 tps	304015 Gwei	$1.94x10^{-6}$$	$1.24x10^{-5}$$
Avalanche Chain [16]	Snowman	AVAX	2s	9.4 tps	42.5 nAVAX	86.0 $	0.077 $
HECO Chain [17]	HPoS	HT	3s	5.32 tps	4.40 Gwei	9,0 $	0.00083 $

Table 9: Comparison of EVM compatible blockchains.

4.5.2. Gas Consumption

This section evaluates the cost, in terms of gas, of deploying on the personal blockchain *Ganache* the different Smart Contracts considered in the new scheme, and the cost of performing the different allowed actions on them. According to this information, we then use the *Polygon Chain* network to calculate the cost in terms of tokens and USD of the aforementioned operations. During the analysis, an average gas value of 60 Gwei (60x10-9 MATIC) and 1 MATIC = 1.61 USD is observed.

Table 10 shows that the proposed *Collection Consent and Processing Consent Smart Contracts* require 3239226 and 1629698 gas for deployment respectively (by means of "newCollectionConsent" and "newProcessingConsent" operations in each case). Therefore, creating and deploying them on the *Polygon Chain* network require 0.313 and 0.157 USD respectively.

Regarding the costs of the operations to be applied on the deployed Smart Contracts, as Table 10 depicts, all functions, except the *newPurpose* one, have a cost of less than 0.005 USD. The *newPurpose* method has a cost of 0.02 USD because it generates a new structure on the *Processing Consent Smart Contract* (as explained in section 3.3.2). On the other hand, the other operations simply modify a single variable of the state of the Smart Contracts, or, in the case of the *revokeConsentPurpose* and *revokeConsentProcessor* functions, they make recursive modifications of a single variable depending on the number of processors that process data for a certain purpose, or the number of purposes for which a processor processes the collected data respectively.

[9] `https://docs.polygon.technology/docs/home/polygon-basics/what-is-polygon`
[10] `https://www.hecochain.com/developer.133bd45.pdf`
[11] `https://etherscan.io` (accessed on 26th of March, 2022)
[12] `https://bscscan.com` (accessed on 26th of March, 2022)
[13] `https://polygonscan.com` (accessed on 26th of March, 2022)
[14] `https://ftmscan.com` (accessed on 26th of March, 2022)
[15] `https://bttcscan.com` (accessed on 26th of March, 2022)
[16] `https://snowtrace.io` (accessed on 26th of March, 2022)
[17] `https://hecoinfo.com` (accessed on 26th of March, 2022)

In order to put these costs into perspective, they are compared with the results provided by three works in the literature that provide empirical results. These are [10, 14, 24]. First, the costs of performing actions on already deployed Smart Contracts are reviewed; next, the costs of deploying the Smart Contracts are discussed too.

Regarding the operations to be run on deployed Smart Contracts, [10, 14] show numbers in the range between 13000 and 50000 of gas, while [24] reports an average cost of 600000 gas. In comparison, the operations performed on the proposed *Collection Consent and Processing Consent Smart Contracts* are in the range between 18685 and 42946 (221707 if the costly "newPurpose" operation is considered), with an average cost of 28798 of gas (again, 44874 of gas if the "newPurpose" operation is considered). Whatever the case is, the costs of the new proposal are equivalent to the costs reported by its counterparts in the literature.

Finally, regarding the average cost of deploying a Smart Contract, in [10, 14, 24] this operation requires around 1500000 of gas. In comparison, the proposed *Processing Consent Smart Contract* matches this value with a cost of 1629698 of gas. Nevertheless, the proposed *Collection Consent Smart Contract* doubles this average with a cost of 3239226 of gas. This higher cost was expected, due to the fact that the number of methods, their complexity and, also, the data structures used in Smart Contracts directly increase their deployment costs. In this way, the proposed *Collection Consent Smart Contract* contains more methods and a higher complexity than other works in the literature in order to provide additional functionalities. It is worth mentioning that the aforementioned cost in gas implies an affordable cost in USD (i.e., 0.31 USD), due to the fact that the real economical cost linked to the deployment operation directly depends on the practical blockchain being used and the price of its token. As it has been previously mentioned, in our case we have used the *Polygon chain* network.

According to the findings of this study, we argue the following points: i) the enhanced functionalities of our proposal justify spending more gas in deploying the *Collection Consent Smart Contract*; ii) the rest of operations of the two Smart Contracts used in our proposal are expected to be used more frequently than the costly one, and they have a cost in gas equivalent to the other works in the literature; and iii) the cost in USD of running those operations depends on the blockchain in use, in this way, the results that we have obtained in the *Polygon Chain* network show that all of them are affordable.

Smart Contract	Operation	Actor	Gas used	Cost (Token)	Cost (USD-$)
Collection Consent	newCollectionConsent	DS	3239226	0.1944	0.3129
Collection Consent	grantConsent	DC	29813	0.0018	0.0029
Collection Consent	revokeConsent	DS	27397	0.0016	0.0026
Collection Consent	modifyData	DS	42946	0.0026	0.0041
Collection Consent	eraseData	DS	29868	0.0018	0.0029
Collection Consent	revokeConsentPurpose	DS	23748	0.0014	0.0023
Collection Consent	revokeConsentProcessor	DS	21715	0.0013	0.0021
Processing Consent	newProcessingConsent	DC	1629698	0.0978	0.1574
Processing Consent	newPurpose	DC	221707	0.0133	0.0214
Processing Consent	grantConsent	DS	28880	0.0017	0.0028
Processing Consent	grantConsent	DP	33751	0.0020	0.0033
Processing Consent	modifyData	DS	27647	0.0017	0.0027
Processing Consent	revokeConsent	DS	31964	0.0019	0.0031
Processing Consent	revokeConsent	DC	30182	0.0018	0.0029
Processing Consent	revokeConsent	DP	18685	0.0011	0.0018

Table 10: Cost of the different functions of the provided Smart Contracts.

5. Conclusions and future work

In this work, we have proposed a new system that exploits smart contracts and the blockchain to provide publicly accessible and immutable evidences that show the agreements made between a Data Subjects and Service Providers regarding the gathering and processing of the personal data belonging to the former.

These immutable evidences are a key aspect due to the fact that they allow to manage the personal data of the individuals by fulfilling the GDPR main requirements. In this way, the proposed system allows Data Controllers and Data Processors to prove that they have the authorization to collect/process the personal data of the Data Subjects in the case of an audit. Moreover, it also allows Data Subjects to be aware at any time about which data is being collected, who is processing it, and for which purposes; in addition to that, they can also modify the details of their consents anytime.

In order to evaluate the new scheme, we have implemented it, and we have carried out a complete analysis and discussion of the described platform. This detailed analysis includes: GDPR-compliance, provided functionality, security and privacy issues, and the cost in terms of gas and USD of the different operations to be run on the blockchain.

Regarding future work, we aim to design and implement the complete system that has been briefly introduced in Section 3.5. The proposal that has been presented in this work is the core component of this complete system that is envisaged to allow data subjects to easily manage all the consents agreed with service providers to collect and process their personal data. In order to achieve this, the next steps will be considered: i) design the software components required to manage all the interactions between data subjects and the complete system; ii) design a compatible key pair management scheme that allows the data subjects to generate fresh cryptographic key pairs for each new *Collection Consent Smart Contract* and manage them in a transparent way; and iii) deploy the proposed system on an existing blockchain and study all the incurred costs in a real setting.

Acknowledgments and disclaimer

This research is supported by the European Union Regional Development Fund within the framework of the ERDF Operational Program of Catalonia 2014-2020 with a grant of 50% of the total cost eligible, under the "FEM-IOT" project [001-P-001682]; by the EU's European Regional Development Fund (ERDF), through the "ERDF Catalonia Operational Programme 2014-2020, investment priority for the creation of jobs and sustainable growth", under the Territorial Specialisation and Competitiveness Project (PECT) "Cuidem el que ens uneix - Sensòrica" project [PR15-020174]; projects RTI2018-095094-B-C21 "Consent" and PID2021-125962OB-C32 "SECURING/DATA" funded by MCIN/AEI /10.13039/501100011033/ FEDER, UE; project HERMES funded by INCIBE and the European Union NextGenerationEU/PRTR; and is part of the research project "Comenersys: Comunidades energéticas, mercados locales y redes: perspectivas y problemas" with reference TED2021-131840B-I00, funded by MCIN/AEI/10.13039/501100011033 and the European Union NextGenerationEU/PRT. The first author is also supported by the Spanish Government under an FPU grant (ref. FPU20/03254).

References

[1] A. Esteve, The business of personal data: Google, Facebook, and privacy issues in the EU and the USA, International Data Privacy Law 7 (1) (2017) 36–47. arXiv:https://academic.oup.com/idpl/article-pdf/7/1/36/14043496/ipw026.pdf, doi:10.1093/idpl/ipw026.
URL https://doi.org/10.1093/idpl/ipw026

[2] K. Houser, W. Voss, Gdpr: The end of google and facebook or a new paradigm in data privacy?, SSRN Electronic Journal (07 2018). doi:10.2139/ssrn.3212210.

[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 (2016) 1–88.
URL https://eur-lex.europa.eu/eli/reg/2016/679/oj

[4] C. Wirth, M. 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.

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

[6] S. Bu-Pasha, A. Alen-Savikko, J.-S. Mäkinen, R. Guinness, P. Korpisaari, Eu law perspectives on location data privacy in smartphones and informed consent for transparency, European Data Protection Law Review 2 (3/2016) (2016) 312–323.

[7] L. A. Linn, M. 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, 2016, pp. 1–10.

[8] A. Azaria, A. Ekblaw, T. Vieira, A. Lippman, Medrec: Using blockchain for medical data access and permission management, in: 2016 2nd International Conference on Open and Big Data (OBD), 2016, pp. 25–30. doi:10.1109/OBD.2016.11.

[9] R. Neisse, G. Steri, I. 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, 2017, pp. 1–10. doi:10.1145/3098954.3098958.

[10] S. Wang, Y. Zhang, Y. Zhang, A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems, IEEE Access 6 (2018) 38437–38450. doi:10.1109/ACCESS.2018.2851611.
[11] M. Chowdhury, A. Colman, A. Kabir, J. Han, P. 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), 2018, pp. 1330–1335. doi:10.1109/TrustCom/BigDataSE.2018.00183.
[12] B. Faber, G. Michelet, N. Weidmann, R. R. Mukkamala, R. Vatrapu, Bpdims:a blockchain-based personal data and identity management system, in: Conference: Hawaii International Conference on System Sciences, 2019. doi:10.24251/HICSS.2019.821.
[13] J. Liang, W. Han, Z. Guo, Y. Chen, C. Cao, X. Wang, F. Li, Desc: enabling secure data exchange based on smart contracts, Science China Information Sciences 61 (04 2018). doi:10.1007/s11432-017-9245-1.
[14] M. U. Rahman, F. Baiardi, B. Guidi, L. Ricci, Protecting Personal Data using Smart Contracts, arXiv e-prints (2019) arXiv:1910.12298arXiv:1910.12298.
[15] G. Zhao, H. He, B. Di, Design and implementation of the digital education resources authentication system based on blockchain, in: Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy, ICCSP 2020, Association for Computing Machinery, New York, NY, USA, 2020, p. 100–104. doi:10.1145/3377644.3377663.
URL https://doi.org/10.1145/3377644.3377663
[16] H. Wang, Y. Yuan, F. Yang, A personal data determination method based on blockchain technology and smart contract, in: Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy, ICCSP 2020, Association for Computing Machinery, New York, NY, USA, 2020, p. 89–94. doi:10.1145/3377644.3377656.
URL https://doi.org/10.1145/3377644.3377656
[17] G. Zyskind, O. Nathan, A. . Pentland, Decentralizing privacy: Using blockchain to protect personal data, in: 2015 IEEE Security and Privacy Workshops, 2015, pp. 180–184. doi:10.1109/SPW.2015.27.
[18] S. Kirkman, A data movement policy framework for improving trust in the cloud using smart contracts and blockchains, in: 2018 IEEE International Conference on Cloud Engineering (IC2E), 2018, pp. 270–273. doi:10.1109/IC2E.2018.00054.
[19] O. Choudhury, H. Sarker, N. Rudolph, M. Foreman, N. Fay, M. Dhuliawala, I. Sylla, N. Fairoza, A. K. Das, Enforcing human subject regulations using blockchain and smart contracts, Blockchain in Healthcare Today 1 (Mar. 2018). doi:10.30953/bhty.v1.10.
URL https://blockchainhealthcaretoday.com/index.php/journal/article/view/10
[20] N. Duong-Trung, H. X. Son, H. T. Le, T. T. Phan, On components of a patient-centered healthcare system using smart contract, in: Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy, ICCSP 2020, Association for Computing Machinery, New York, NY, USA, 2020, p. 31–35. doi:10.1145/3377644.3377668.
URL https://doi.org/10.1145/3377644.3377668
[21] N. Duong-Trung, H. X. Son, H. T. Le, T. T. Phan, Smart care: Integrating blockchain technology into the design of patient-centered healthcare systems, in: Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy, ICCSP 2020, Association for Computing Machinery, New York, NY, USA, 2020, p. 105–109. doi:10.1145/3377644.3377667.
URL https://doi.org/10.1145/3377644.3377667
[22] M. Zichichi, S. Ferretti, G. D'Angelo, V. Rodríguez-Doncel, Personal data access control through distributed authorization, in: 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), 2020, pp. 1–4. doi:10.1109/NCA51143.2020.9306721.
[23] M. Davari, E. Bertino, Access control model extensions to support data privacy protection based on gdpr, in: 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 4017–4024. doi:10.1109/BigData47090.2019.9006455.
[24] M. Barati, O. Rana, Tracking gdpr compliance in cloud-based service delivery, IEEE Transactions on Services Computing (2020) 1–1doi:10.1109/TSC.2020.2999559.
[25] M. M. Merlec, Y. K. Lee, S.-P. Hong, H. P. In, A smart contract-based dynamic consent management system for personal data usage under gdpr, Sensors 21 (23) (2021). doi:10.3390/s21237994.
[26] S.-S. Jung, S.-J. Lee, I.-C. Euom, Delegation-based personal data processing request notarization framework for gdpr based on private blockchain, Applied Sciences 11 (22) (2021). doi:10.3390/app112210574.
URL https://www.mdpi.com/2076-3417/11/22/10574
[27] C. Daudén-Esmel, J. Castellà-Roca, A. Viejo, J. Domingo-Ferrer, Lightweight blockchain-based platform for gdpr-compliant personal data management, in: 5th IEEE International Conference on Cryptography, Security and Privacy, CSP 2021, Zhuhai, China, January 8-10, 2021, 2021, pp. 68–73.
[28] S. Haber, W. S. Stornetta, How to time-stamp a digital document, in: A. J. Menezes, S. A. Vanstone (Eds.), Advances in Cryptology-CRYPTO 90, Springer Berlin Heidelberg, Berlin, Heidelberg, 1991, pp. 437–455.
[29] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system (2008).
URL http://www.bitcoin.org/bitcoin.pdf
[30] R. C. Merkle, Secrecy, authentication, and public key systems., Ph.D. thesis, Stanford, CA, USA, aAI8001972 (1979).
[31] Z. Zheng, S. Xie, H. Dai, X. Chen, H. Wang, An overview of blockchain technology: Architecture, consensus, and future trends, 2017 IEEE International Congress on Big Data (BigData Congress) (2017) 557–564.
[32] R. Kaur, A. Kaur, Digital signature, in: Proceedings of the 2012 International Conference on Computing Sciences, ICCS '12, IEEE Computer Society, USA, 2012, p. 295–301. doi:10.1109/ICCS.2012.25.
URL https://doi.org/10.1109/ICCS.2012.25
[33] V. Buterin, Ethereum: A next-generation smart contract and decentralized application platform (2014).
URL https://github.com/ethereum/wiki/wiki/White-Paper
[34] B. Mohanta, S. Panda, D. Jena, An overview of smart contract and use cases in blockchain technology, 2018. doi:10.1109/ICCCNT.2018.8494045.
[35] H. L. Bill Parducci, Oasis: extensible access control markup language (xacml) version 3.0, http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html#_Toc325047066 (January 2013).
[36] G. Gutoski, D. Stebila, Hierarchical deterministic bitcoin wallets that tolerate key leakage, in: Financial Cryptography and Data Security: 19th

International Conference, FC 2015, San Juan, Puerto Rico, January 26-30, 2015, Revised Selected Papers 19, Springer, 2015, pp. 497–504.
[37] J. Katz, Y. Lindell, Introduction to modern cryptography, CRC press, 2020.