International Journal of Information Security (2024) 23:1021–1036
https://doi.org/10.1007/s10207-023-00782-z

REGULAR CONTRIBUTION

Olympus: a GDPR compliant blockchain system

Ricardo Martins Gonçalves[1] · Miguel Mira da Silva[1] · Paulo Rupino da Cunha[2]

Published online: 17 November 2023

Abstract
Blockchain has been gaining significant interest in several domains. However, this technology also raises relevant challenges, namely in terms of data protection. After the General Data Protection Regulation (GDPR) has been published by the European Union, companies worldwide changed the way they process personal data. This project provides a model and implementation of a blockchain system to store personal data complying with GDPR. We examine the advantages and challenges and evaluate the system. We use Hyperledger Fabric as blockchain, Interplanetary File System to store personal data off-chain, and a Django REST API to interact with both the blockchain and the distributed file system. Olympus has three possible types of users: Data Subjects, Data Processors and Data Controllers and a fourth participant, Supervisor Authority, that, despite not being an explicit role, can perform all verifications that GDPR mandates. We conclude that it is possible to create a system that overcomes the major challenges of storing personal data in a blockchain (Right to be Forgotten and Right to Rectification), while maintaining its desirable characteristics (auditability, verifiability, tamper resistance, distributed—remove single points of failure) and complying with GDPR.

Keywords Blockchain · GDPR · Data privacy · Developing · Hyperledger fabric · IPFS

1 Introduction

Blockchain has been the subject of great enthusiasm in several domains [1–7]. The technology's versatility has led to blockchain being used for different applications, from cloud authentication [2] to secure sharing of health records [3]. Recent users include incentive mechanisms for machine learning [5], data sharing frameworks for IoT [7], control and traceability of food supply chains [4], and storing transaction details of pets' adoption process [6]. However, the hype also leads to misuse, which motivated several authors to propose criteria for deciding whether to use blockchain. For example, the National Institute of Standards and Technology [8] suggests criteria such as the need for a distributed network and consistent records.

Another trend in the last decades has been the online collection and processing of huge amounts of personal data, which led to the introduction of privacy-oriented legislation such as the European Union's general data protection regulation (GDPR) [9]. For the purposes of this article, personal data is defined as "any information relating to an identified or identifiable natural person" [9]. GDPR entitles the users to request the deletion of their personal data (barring some exceptions) which is at odds with the immutability of blockchain, a key pillar for enabling trust in those systems. This conflict raises the need for investigating solutions to use blockchain for storing personal data without violating the law.

1.1 Problem statement

Storing personal data in the blockchain have many benefits, including integrity and tamper resistance (that make blockchain easier to audit), distribution (that removes of Single Points of Failure) and almost real-time record updating [10]. However, to benefit from this characteristic is mandatory to comply with the law.

✉ Ricardo Martins Gonçalves
ricardo.martins.goncalves@tecnico.ulisboa.pt

Miguel Mira da Silva
mms@tecnico.ulisboa.pt

Paulo Rupino da Cunha
rupino@dei.uc.pt

[1] Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal

[2] Faculty of Science and Technology, University of Coimbra, Coimbra, Portugal

GDPR is one of the world's leading personal data protection regulations and, since the regulation makes storing personal data directly on the blockchain impractical, is important to search for solutions that have the characteristics of blockchain while complying with the GDPR.

We propose a GDPR blockchain system with off-chain storage to verify if it is practical to use blockchain to store personal data while complying with GDPR.

Although there are several projects using blockchain to store personal data, there is a lack of implementation and most of them do not address GDPR directly [10]. We notice the necessity of a more GDPR-focused solution to store personal data in blockchain.

This project presents the conceptual model of such a system and its respective implementation. Possible applications include customer data processing by companies, employee data processing by employers, students' data processing by schools/universities and hospitals that want to store patient data.

1.2 Our contribution

We designed and developed a blockchain system to store personal data complying with the GDPR using Hyperledger Fabric (HLF) as blockchain and IPFS (as off-chain storage). The system stores personal data off-chain and an pointer to the personal data in the blockchain. The system was designed considering the participants in GDPR and the different functions and authorisations for each role. The evaluation was made in terms of compliance with the GDPR and performance.

1.3 Paper structure

The remainder of this paper is organised as follows: in the next section, we provide some common ground on key topics. Then, in Sect. 3, we describe the conceptual proposal in detail. Sections 4 and 5 have a similar structure. We describe the technologies (Hyperledger Fabric and IPFS) and how they were integrated into the system. Section 6 is a description of an API created for interaction with the system. Next, we provide an evaluation concerning GDPR compliance and performance of the system. Finally, we summarise related work before closing with the conclusion.

2 Background

This section provides some common ground required throughout the paper, namely blockchain, a system characterised by great auditability and immutable records that can be used to store a variety of data; General Data Protection Regulation (GDPR), one of the world's leading personal data protection regulations; and challenges of using blockchain to store personal data without violating the GDPR.

2.1 Blockchain

Blockchain (BC) was introduced by Satoshi Nakamoto [11] to support Bitcoin. Blockchain technology consists of a distributed and tamper-resistant ledger shared by a network of nodes. This ledger is append-only, which means that once information is entered it can neither be deleted nor modified. Blockchain is categorised into permissioned and permissionless. In the former, everyone can join and maintain the network (publish blocks); in the latter, only authorised nodes can publish blocks [8].

There are four types of blockchain systems: public, private, consortium and hybrid [12]. A public blockchain is an open platform which anyone can access. All participating nodes have the same authority to verify transactions and validate blocks. Private blockchains are closed networks, owned by an entity or organisation and restricted to specific nodes, i.e. new nodes can only join the network if the owner of the blockchain accepts them. Controlling nodes are set by the owner. The Consortium mode rests on a community that enables more than one organisation to manage a private blockchain. Finally, a hybrid blockchain is a combination of public and private that allows users to decide who can participate and which transactions should be made public [13].

There are several technologies for each type of blockchain. Bitcoin, Ethereum, Dash, and Litecoin are examples of public blockchain systems. Monox and MultiChain are examples of private ones. EWF, R3, Corda and B3i are consortium blockchains, and Dragonchain is an example of a hybrid blockchain [12].

One of the key aspects of blockchain is the use of consensus algorithms that ensure that "all the nodes in the network agree upon a consistent global state" [14]. There are several and distinct consensus algorithms [14]. For example, proof-of-work (PoW) consists in solving a mathematical puzzle. Proof-of-stake (PoS) uses the quantity of cryptocurrencies (stake) to select the next node to publish a block. The nodes that invest more in the network have a higher probability of being selected, and there are several algorithms that can be used to choose the next node to publish a block (e.g. random). Proof-of-authority (PoA) relies on the trust of publishing nodes through their know link to real-world identities. Proof-of-elapsed-time relies on secure hardware to determine the next publisher, as each publishing node requests a wait time from a secure hardware source, sleeps during that time, and then awakes to create and publish a block alerting the other nodes [8].

Among the key properties of blockchain are immutability, transparency, availability, privacy, and consistency [15].

2.2 General data protection regulation

The GDPR took effect across all member states of the European Union (EU) in May 2018 [16]. This regulation includes strong sanctions that can reach a limit of 20 million Euros or 4% of the global revenue of a company (whichever is higher) according to Article 83 [9].

GDPR applies to the processing of personal data for all companies that process personal data of European citizens, regardless of where they are headquartered [9].

GDPR identifies four stakeholders [17]: the Data Subject, identified or identifiable natural person to whom personal data relates directly or indirectly [9]; the Data Controller, "natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data" [9]; the Data Processor, "a natural or legal person, public authority, agency or other body which processes personal data on behalf of the controller" [9]; the supervisor authority, "an independent public authority which is established by a Member State [...] to be responsible for monitoring the application of GDPR" [9].

The GDPR's purpose is to "protect the rights, privacy and freedoms of natural persons in the EU" and to reduce "barriers to business by facilitating the free movement of data throughout the EU" [17].

2.3 Blockchain and GDPR

There are several compliance considerations to keep in mind concerning personal information. GDPR makes it impractical to store certain information in the blockchain.

Lately, there have been several cases where personal information is recorded on blockchain-based systems. For example, for a data sharing framework [18], managing data in insurance [19], mobile communication [20], and creating log systems [21]. Compliance with the regulation requires finding solutions to address and solve the challenges of storing personal data in the blockchain.

In a previously executed systematic literature review (SLR) [10], we identified the main advantages and challenges to overcome when storing personal data in blockchain systems. The major advantages to store personal data in BC are verifiability, integrity, and tamper resistance of records. Other advantages, such as the removal of single points of failure, strong consensus algorithms, and anonymity, are also mentioned in the literature.

The major challenges detected in the SLR are consequences of compliance with articles 17 and 16 of GDPR. The right to erasure or the right to be forgotten, Article 17 [9] states that "the data subject has the right to have his/her personal data erased where it is being held under someone else custody without any justifying grounds at all" [22]. This entitlement conflicts with the immutable nature of records in blockchain systems. On the other hand, article 16, Right to Rectification, states that "data subjects have the right to rectify inaccurate personal data" [23], which conflicts with the exact same property of blockchains. Other challenges related to GDPR are outlined in the literature, such as defining accountability, identifying the data controller, and defining responsibilities.

Many solutions to overcome these challenges involve combining the type of blockchain (private/public) and the category (permissioned/permissionless) with the way the data are stored. These combinations are made because the challenges involved in using a private and permissioned blockchain for storing personal data are different from the challenges involved in using a public and permissionless one.

Although there are several projects addressing the challenges of storing personal data in a blockchain, there are not many implementations, and most of them are related to a specific area/problem. Thus, we identified the need to create a more generic, domain-independent system that could be adapted to a large number of applications.

3 Related work

This section presents an overview of other projects about blockchain-based systems to store personal records and be compliant with GDPR, including both technological implementations and conceptual architectures.

Some solutions to deploy blockchain systems to store personal data are based on the creation of new blockchain technologies. For example, Onik et al. [24] created a privacy-aware blockchain for sharing and tracking personally identifiable information (PII), which is defined as "any subset of attribute values of an individual person which sufficiently identifies this individual person within any set of persons" [24]. This paper describes a scheme that is decentralised and can be used for trusted PII sharing and tracking. Like most of the other projects, BcPIIMS uses off-chain storage to reduce blockchain storage space and private blockchain to limit personal data leaking. However, the off-chain storage is local, not distributed like IPFS.

The system stores all personal information in the local DB and the rest (user id, controller id and Non-Personally identifiable information (NPII)) in the blockchain. They used the Australian privacy law's definition for NPII: "non-identifiable data, which have never been labelled with individual identifiers or from which identifiers have been permanently removed, and by means of which no specific individual can be identified" [24]. The proposed architecture forces all the information inserted by the user to pass the data controller before reaching the local database of the processor. The controller separates the data into PII and NPII, generates

the hash, and adds the information to the blockchain. When a DS wants to modify or delete data, needs to inform the controller and processor that he wants to perform the operation. After that, the process is carried out by the DP and the DC. However, existing technologies are better known and easy to use.

There is also international initiatives, the European Block chain Services Infrastructure (EBSI) [25] is a blockchain-based public services infrastructure. The project uses a permissioned blockchain system to help solving business problems. However, EBSI does not have a native off-chain decentralise storage, and therefore, it does not address the challenge of storing personal data in the blockchain directly.

Barati et al. [26] address the problem of storing personal data in blockchain in a specific domain: storing vaccine passports during the COVID-19 pandemic. They designed a system to create, store and verify digital vaccine certifications, also based on IPFS as off-chain storage. This option guarantees no single points of failure and allows data to be securely distributed globally. Since consent and data privacy are major aspects of GDPR compliance, this project ensures user consent, data encryption, data erasure, and accountability. However, the authors used Ethereum, a blockchain that has major disadvantages when compared with Hyperledger Fabric, including the cost of transactions, and time of mining/ordering.

Daudén-Esmel et al. [27] propose a system that provides public access to immutable evidence while also allowing the DS to manage his personal data. The DS must grant explicit and time-limited consent for the collection and processing of his data. The data record contains the id of DC, id of DS, and consent lifetime. Each DP and the user consent must be publicly available for the DS to manage. The DS are able to modify, erasure and revoke consent, and any actor must prove his identity before performing an operation.

There are projects that store data locally. For example, Chiu et al. [28] propose an architecture in which data are stored in a location that the user can trust and has control, such as local storage. Since the main focus of this paper is not GDPR, the roles in the system are slightly different. The authors also emphasise the advantages of using off-chain storage, such as scalability of the blockchain (since only the hash of the data is stored), and enabling deletion and modification of information.

Truong and Lee [29] propose a design concept with technical mechanisms to create a blockchain-based system compliant with GDPR for personal data management. They use the same blockchain system as proposed in this paper: Hyperledger Fabric. The system has three different roles: end-user (Data Subject), Service Provider (an entity that collects and manages personal data) and a third party (an entity that provides a service to end-users). The main role of blockchain is to grant authorisation tokens when an entity wants to perform an operation on personal data. The entity asks for a token through a smart contract and then requests information directly to the resource server. The personal data are stored off-chain to improve scalability, efficiency and GDPR compliance. When a DP accesses personal information, the DS performs an active role by signing an acceptance message.

The COVID-19 pandemic inspired several blockchain projects. The need to process vaccine information and certificates worldwide motivated several authors to create distributed and secure systems. Abid at al. [30] developed a system (NovidChain) that uses blockchain, self-sovereign identity, encryption and W3C credentials standards to help preserve privacy while sharing health certificates. Since NovidChain uses Ethereum blockchain, the authors chose uPort, a self-sovereign identity and user-centric data platform to manage identification. Novidchain relies on Ethereum private and permissioned Blockchain, and uses IPFS as off-chain storage. Since NovidChain was designed to store medical information, the users do not hold full control of their information since the certificates are not generated or managed by the users. Ethereum also makes the network more costly, having an approximate $2.18 cost for issuing a COVID-19 certificate [30].

Several of the authors that created a blockchain system to store personal data are from the medical and health sectors. Agbo and Mahmoud [31] designed and implemented a blockchain framework for health consent management. The authors proposed to store personal medical records on the blockchain, and access the ledger from smart contracts that are used as an access–control interface. The DS is capable of granting and revoking permissions to his data.

Another possible application of blockchain to store personal data is human resources management. Rotondi et al. [32] created a system to store work hours using Ethereum and a off-chain storage. The information about the worker attendance is stored off-chain, and the hash is stored on the blockchain.

Although there are several authors that proposed solutions based on BC to store personal data, the majority is highly connected to a type of industry or problem. For example, healthcare and COVID-19 have inspired several projects [26, 30, 31]. There is also interested in using blockchain systems to e-governance. The Glass project [33, 34] used a similar architecture to the one we propose. The authors used an HLF as blockchain and IPFS as off-chain storage. However, the access to the IPFS files in our solution is more restricted (only a few nodes) and our model separate more clearly the functions of the Data Subject, Data Controller, and Data Processor.

We propose a solution to store general data, in which the user keeps full control of his data. Our system can be used

to store client information among enterprises, to store health care information, social network information, among others.

One of the major advantages of using a permissioned blockchain (HLF in this case) is the lower costs in terms of time and money. In contrast to other HLF solutions, we do not use the HLF authentication system to allow greater agility, since adding the user directly to HLF implies a manual process.

4 Conceptual proposal

Since, according to articles 16 and 17 of GDPR, corrections and deletions of personal data are mandatory, this information cannot be directly written to the immutable blockchain. Thus, we store it instead off-chain, where the required operations are possible. However, to ensure the tamper resistance and integrity of the data, its hash is calculated and stored in the (immutable) blockchain.

When it is required to delete personal data, the information in the off-chain is completely deleted; however, the hash in blockchain is not modified. All the data previously written can be verified using the hash code in the blockchain; however, it is not possible to retrieve the original information. The modification process is simple when a deletion function is provided. To modify information, it is required to delete the previous record off-chain and then create a new one, after that, its calculated a new hash code and stored in the blockchain.

Storing information off-chain makes it possible to tamper the information; however, since the hash of all data saved off-chain is added to the blockchain, any modification can be easily detected. To detect alterations made to the data, compare the hash written in the BC with the hash code of the actual file.

Our design thus consists of three main parts: the blockchain, the off-chain storage and a REST API to facilitate interactions with other systems. There are several technologies to implement blockchain systems, off-chain storage and REST APIs. We chose Hyperledger Fabric (HLF) to implement the blockchain, Interplanetary File System (IPFS) as off-chain storage, and Django to build the REST API.

In our proof-of-concept, the blockchain consists of three organisations (Alpha, Beta and Omega), Alpha and Beta are peer organisations (each with two peers) and Omega is an orderer organisation (with three orderers).

For ease of maintenance, the whole setup was built using virtual machines (VMs) that act as nodes for both the HLF blockchain and the IPFS network. The virtual machines used as peer nodes for HLF also serve as nodes in the IPFS network.

All personal data are stored in a private IPFS network protected by a key, to ensure that only authorised nodes can read, edit and delete that information. In a classical IPFS system, each node stores a independent set of files, which makes it impossible to ensure that a particular file is deleted across the network when requested. To address this issue, we propose the use of IPFS cluster [35], in order to maintain the same file list in every node and, consequently, be able to perform erasure and rectification in all nodes.

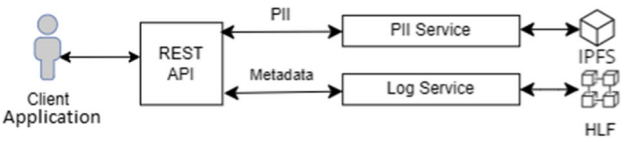

Fig. 1 Microservice design

To interact with the system, the client applications use a REST API. It can be accessed through a command line or, if a graphical interface is needed, by a web browser. The REST API has the options to create, read, update and delete information.

When a user performs an action, the API separates the personal information from metadata, using IPFS to address operations relative to personal information and HLF to address operations relative to metadata.

We propose a microservice architecture [36]. Microservices are autonomous services and can be deployed independently for a clear and defined purpose. This type of architecture also allows each system to be developed and tested independently and more easily maintenance [37]. Figure 1 is a simplified representation of the system.

5 HLF blockchain component

Hyperledger Fabric is an open-source permissioned blockc hain platform, designed for enterprise contexts. Among the main characteristics of HLF, the highly modular and configurable architecture stands out. This architecture allows innovation, versatility and optimisation for a wide range of industry and enterprise use cases [38].

Fabric supports smart contracts (chaincode) written in different programming languages, such as Java, Go, and Node.js and supports different consensus protocols [38]. To fully understand the architecture of HLF, it is important to understand the different roles in an HLF network. The main stockholders of Hyperledger Fabric are Certification Authorities (CA), Organisations, Identities, Membership Service Providers (MSP), orderers and peers [38]. The HLF documentation provides a tutorial on how to deploy a production network divided into 5 steps: deciding network configuration, setting up the cluster, setting up certification authorities, creating identities and MSP and deploying peer and ordering nodes.

5.1 Certification authorities and membership service providers

Certification authorities (CA) associate certificates with each node and identity and create MSP structures that are used to check the permissions that each node/identity has in the network. Any organisation in an HLF network should have at least two CAs each, with distinct objectives. The first one, Enrolment CA, is used to enrol admins, MSP and nodes of an organisation; the second, TLS CA, generates and manages TLS certificates.

The identities "determine the permissions over resources and access the information" [38]. Since HLF is permissioned, all participants need to prove their identity to the rest of the network. Although the CA generates key pairs, is the MSP that recognises the identity and verifies if a given user can perform an action or endorse a specific transaction.

Although the name "Membership Service Provider" suggests an active subject, MSP is actually a file structure that contains a list of permissioned identities, held by each entity/node.

5.2 Ordering service

In a permissionless blockchain system, any node can participate in the consensus process (ordering transactions and distributing them among blocks). Conversely, a permissioned blockchain like HLF has specific nodes that sort and separate the transactions into blocks. These nodes form the ordering service and rely on deterministic consensus algorithms.

Our proof-of-concept, Olympus, has a particular organisation, Omega, to which all orderer nodes belong. Like other organisations in an HLF network, Omega has two CAs (Enrol and TLS). The three orderers (Cronus, Atlas and Rhea) work together to maintain the ledger and enforce access control for channels. A channel is a private subnet of communications between two or more network members. They are a crucial aspect of HLF because they enable conducting private and confidential transactions. Channels are defined by organisations and configured in the first block of that channel, the genesis block. This block stores the policies, members and peers of the channel.

The transaction flow consists of three phases, transaction proposal and endorsement (when a client application proposes a transaction); submission and ordering (orderers sort transitions and create the block); and transaction validation and commitment (distribution of ordered and package blocks to ensure that all the nodes have the same transactions).

HLF provides three main consensus protocols, Raft, Kafka and Solo. However, the last two were deprecated after the release of version 2 of Fabric. The Raft protocol (used in Olympus) is a crash fault-tolerant (CFT) that has "leader and follower" model. Each channel elects a leader that makes decisions and then replicates them among the followers.

5.3 Peer nodes

The peer nodes are a fundamental element in HLF networks; they host ledgers and the chaincode (smart contracts).

To ensure auditability, integrity, tamper resistance in a permissioned blockchain, it is crucial that the network has at least two organisations that interact with the ledger. If the totality of the BC network is controlled by just one organisation, it is not possible to do a full scrutiny of the actions, since all the transactions and alterations to the chaincode are approved by just one entity. Olympus has one orderer organisation and two peer organisations, Alpha and Beta, each one with two peers: Zeus and Poseidon, and Hera and Demeter, respectively. After deploying the CA, and the peer nodes it is possible to interact with the blockchain system through any of these peers by using the peer client. All peers belong to the same channel and have access to the same information in the Olympus network.

5.4 Chaincode

Chaincode (CC) is a program that initialises the ledger and manages its states through transactions. Chaincode can be written in Go, node.js or Java, and since it handles functions agreed by the members of a network, it can be considered a "smart contract". States of the ledger are dependent on the chaincode and created in a specific scope. This makes it impossible to access a state created by a chaincode directly from another chaincode. However, it is possible for a chaincode to call another chaincode, if it has the necessary permissions and both are in the same network [38].

The process to deploy chaincode is divided into several stages. After writing the chaincode, it is necessary to label it and create a package with all the code and metadata (name, label, programming language). After packaging, the CC is installed in each peer, so that it can be approved by each organisation. After analysing the chaincode, the administrator of each organisation decides if he approves it to be committed across his organisation. After the chaincode is approved by every organisation that will use it, it can be committed in the channel [38]. After being committed, the chaincode can be invoked via the command line or using the Olympus REST API.

Olympus has two CC (written in go) running simultaneously. The first CC stores the metadata of the personal data inserted in the system: Timestamp, Pointer of the IPFS file, SHA256 hash code, client ID, flag indicating if the IPFS file was deleted and consent agreement. The second CC is used to authenticate and manage the authorisation of Olympus administrators.

ID	cid	consents	deleted	hash	timestamp
1	QmRYAV2x4R...	True	false	11958a011e18...	2022-07-29 10:...
2	QmXkE7aGNZ...	True	false	4fa66132032dc...	2022-07-29 10:...
3	QmW6dJ2kygR...	True	false	9420586a0685...	2022-07-29 10:...
4	QmNzstXmUge...	True	false	c744678140a8...	2022-07-29 10:...
5	QmNmfhw8UB...	True	false	76bd20cfe83ca...	2022-07-29 10:...

Fig. 2 CouchDB with 5 user records

Hyperledger Fabric provides two options to use as State Database (DB) and support chaincode operations. LevelDB is the default; however, CouchDB can perform JSON queries and be used for more complex actions. For this reason, Olympus uses CouchDB as state DB. Figure 2 shows a possible state of the database, with 5 entries. Each line represents a user, and each column to a different field: (1) user ID; (2) pointer to the IPFS file of the user, also called Content ID; (3) consent given when the API requested to process the personal data of the user; (4) a flag indicating if the IPFS file of that user was deleted; (5) SHA256 of the IPFS file; (6) timestamp of the user record's creation.

5.5 Network

The full network is composed of three organisations (Alpha, Beta and Omega) and respective CAs. The ordering service is formed by three orderers connected with each other. Conversely, peer nodes only connect with other peers inside the same organisation. The four peers (Alpha: Zeus, Poseidon; Beta: Hera, Demeter) can use any of the orderer nodes to connect to the ordering service as a whole. To interact with the ledger, the user can invoke the chaincode using any one of the four peers that will communicate that transaction to the ordering service. Figure 3 illustrates the architecture of HLF network in our Olympus proof-of-concept. The "direct connection" represent machines that communicate between each other without any intermediate; the "indirect connections" are connections to the ordering service itself, a peer node can use to any of the orderers to connect to the ordering service as a whole.

Every machine represented in the network diagram is a virtual machine with Ubuntu 20.04, running in Virtual Box, hosted by a hardware running Windows 11 with 128GB of Ram, 1 TB of Memory and an Intel Core i7-12700 2.10GHz.

6 IPFS component

The Interplanetary File System (IPFS) is a peer-to-peer protocol designed to create a permanent, decentralised, efficient, and robust data storage and distribution [39]. IPFS combines several different technologies to achieve low latency and a content-addressable network. It combines distributed

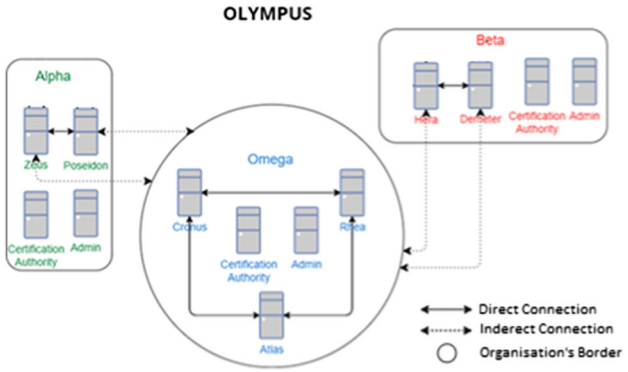

Fig. 3 Olympus: HLF network

hash tables (DHT) to coordinate and maintain metadata and BitSwap (a protocol inspired by BitTorrent) to coordinate networks of untrusting peers and a cryptographically authenticated data structure, similar to Git, to support file versioning [39].

All files in an IPFS network are addressed using a cryptographic hash (Content ID) code that results in tamper-resistant files (since the hash code changes if the file was tampered) and prevents duplication (since two equal files result in the same IPFS object). Network nodes store a list of hashed files in a local store. When a node wants to transfer a file, it starts by finding providers (nodes that have the file) and then transfers the document. Every node has an independent file collection that is publicly available across the network [40]. IPFS can store files, websites, applications, and general data. When a node adds a file, it is split into smaller chunks, a hash is calculated for each, and given a Content ID (CID) that is used to verify and retrieve the file. When another node requests the file, it provides the CID, downloads and stores a copy of the document. This makes the second node also a provider of the original file. To avoid losing the file, the second node can pin the content. When a file is pinned, its CID is added to a list of CIDs whose files cannot be deleted, the pinset. If the second node does not pin the file, it will be deleted after a predetermined amount of time or when a specified amount of storage is used. If a second version of the file is created, a new CID is generated, instead of overwriting the previous document [40]. These properties ensure good resilience, speed, and great censorship resistance [40].

6.1 Private network

IPFS private networks allow the nodes to connect only to other nodes that have the same shared key and reject communications from nodes outside that network [41]. Creating a private network in IPFS is a simple process involving a few steps: first it is necessary to create a key in one of the peers and share it among the selected peers; second, delete the bootstrap

list (each node has a individual bootstrap list that enumerates the nodes to which the current node is directly connected) from all the nodes of the network; third, it is necessary to add the selected nodes to the bootstrap list. To ensure that the nodes only connect to a private network, it is possible to set an environment variable.

6.2 Cluster

IPFS provides the option to build private networks using IPFS cluster to coordinate pinsets across peers with the same secret keys. This enables collective pinning and unpinning and, consequently, deletion of a file from all peers [39]. IPFS cluster is a distributed application that provides data orchestration across a private network. This is achieved by "allocating, replicating and tracking a global pinset distributed among multiple peers" [35]. Cluster works side by side with the IPFS peer creating a total replication of data across the network, enabling deletion of files and, like the main protocol, ensuring a distributed network, since there is no central server [35].

Any node can delete a file from the network.

6.3 Network

Olympus's private network is formed by the same virtual machines that support the HLF peer nodes. The network consists of four peers that share a private key and a set of files. All peers are part of a cluster, to make it possible to have a consistent file list across the network. Figure 4 illustrates the Olympus IPFS network. Four peers (Zeus, Poseidon, Hera and Demeter) connected among them and sharing a private key to enable a private network and a cluster. Besides being possible to delete a file, it is also possible to recover any file if a peer is lost.

Performing operations on the data is relatively simple from a user's perspective. To add a file, the user only needs to execute a command that replicates the file across the network and add the CID of the file to the pinset of each node. To list the pinset of the network, the user only needs to invoke a command. To delete a file from the network, the user runs two commands, one to delete the CID of a file from the shared pinset and the second to execute the garbage collector across the network to delete every unpinned file.

7 REST API

This section explains the design of the API used to communicate with our system and the stakeholders it supports. When a user wants to perform an operation, he can call the REST API from the command line or use the web interface.

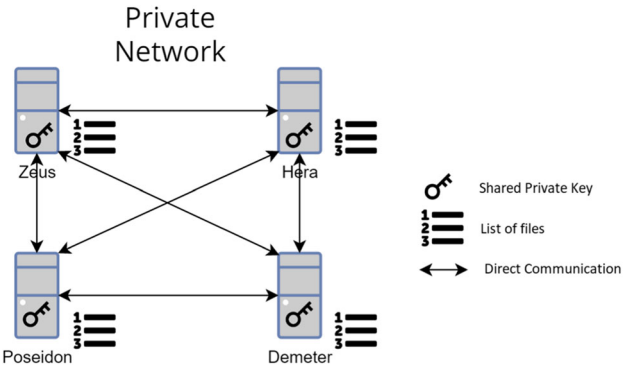

Fig. 4 Olympus: IPFS network

The API manages all the required interactions with HLF and IPFS to securely store the data across both networks.

7.1 Users

The API supports three types of users, based on the above-mentioned GDPR stakeholders. There is no specific role for the Supervisor Authority in the API since this entity is not always present in a organisation (only in audits) and, however, is possible to guarantee all the functions granting specific authorisations in the system.

The roles supported by the API are independent from the blockchain and IPFS system. In other words, the separation of roles in GDPR is more exhaustive and complex, an orderer can be considered a data processor and an administrator in HLF can be considered a data controller; however, none of them has a explicitly role in the API. This section is about the specific user types of the API, not identifying the stakeholder of GDPR.

7.2 Authorisation

The main difference between a DS and a DP or DC is the authentication and authorisation process. The authorisation of each role has a similar mechanism; however, the keys are stored in different systems. The authentication is made by verifying signed messages. When a participant wants to execute a function, he needs to send a signed message with his id and the desired operation. In the proof-of-concept, this is achieved by providing his private key to the API. In a production environment, the signature can be done by executing a script in his browser, and a timestamp can be added to ensure signature freshness [42]. Then, the system verifies whether the signature is valid, checks the type of user, and whether he has permission to execute that operation.

The difference between data subjects and administrators (data processor and data controller) is that, since GDPR makes impractical to store personal data in the blockchain (because it is impossible to delete), the public keys of DS are

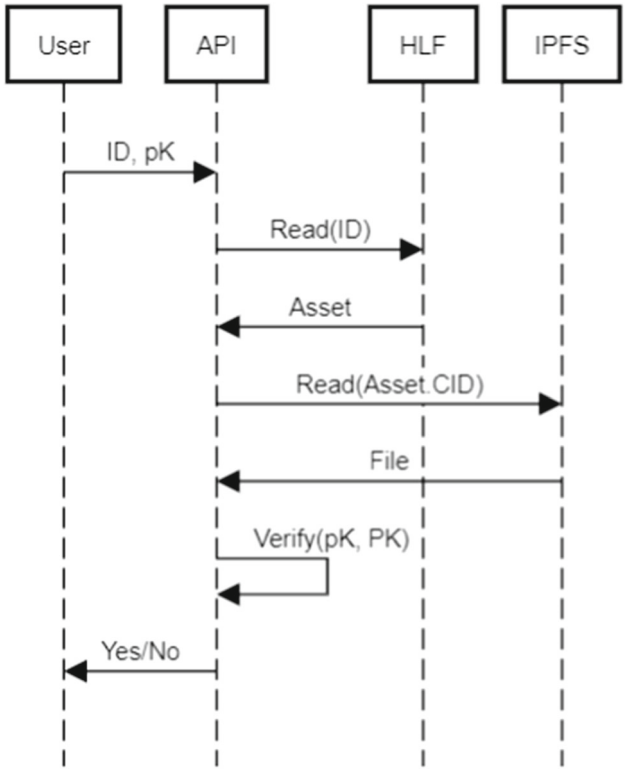

Fig. 5 User authentication

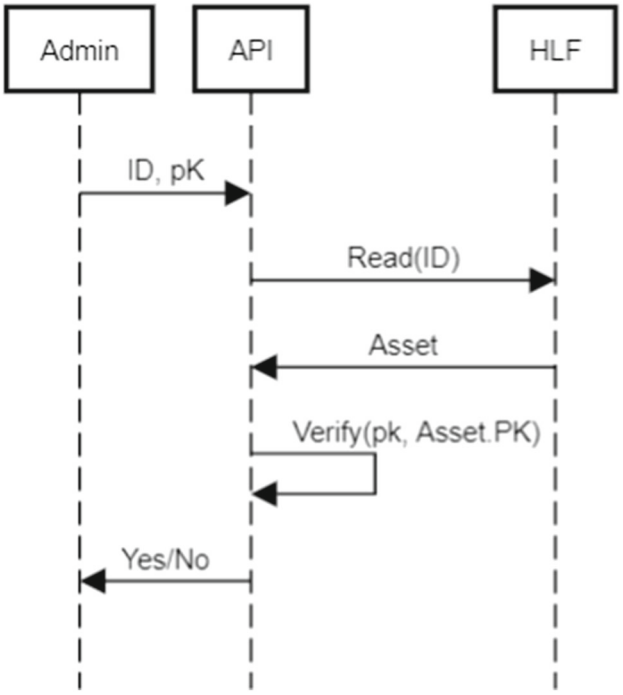

Fig. 6 Admin authentication

stored with the personal data in a single IPFS file while the public keys of the administrators are stored directly in the blockchain. Each data subject has only one IPFS file containing all the personal information provided by the DP and a public key for authorisation purposes.

To authenticate and authorise administrators, there is a separate chaincode that stores all the administrator public keys and the correspondent roles (DP or DC).

Figure 5 illustrates the users' authentication process. First, the user sends his ID and private key to the API; second, the API requests to HLF the user asset (that contains the CID of the user IPFS file); third, the API requests to IPFS the user file (with users private key) in order to verify the private key. The administrator authentication process, as shown in Fig. 6, is simpler. First, the administrator sends his ID and private key (pK) to the API; second, the API requests to the HLF the asset of the respective administrator; third, verify if the private key that corresponds to the Admin ID.

Hyperledger Fabric provides authentication and authorisation mechanisms; however, they were designed for systems with few users interacting directly with the blockchain. In HLF, any new user must be added manually by the CA admin before enrolling in the network. Since this process is time-consuming and needs manual actions, we decided to use an alternative mechanism that completely decouples the development and operation of the HLF from the roles in the API.

7.3 Functions

Each of the different user types can perform different operations namely:

- *Data Subject* Create, Read, Update, Delete (CRUD) his own record, Update own key pair;
- *Data Processor* Read DS record, Read information by field, List all users;
- *Data Controller* CRUD operations on DS, Update DS and other admins key pairs, List all users.

Olympus can be used in more realistic scenarios. However, to simplify (in terms of implementation, understanding and analysis), we have decided to list and implement only the fundamental functions of each role. The combination of these can be used in real-case scenarios. (The source code of each algorithm is available online [43].)

All algorithms: start with an authorisation check (based on RSA 2048 signatures); encrypt all files (using AES 256) before storing them in the IPFS network; decrypt them (using AES 256) when reading. These steps are omitted below to simplify the comprehension of the functions. The cryptografic algorithms were chosen based on security, performance and widespread use.

Algorithm 1 shows the creation of a user: (1) the API creates a key pair; (2) the API verifies whether that user ID is already used; (3) write all the personal information in an IPFS file; (4) calculate a hash code of the file; (5) the hash and con-

tent id are then written in the blockchain to ensure auditability and verifiability; (6) the private key is then returned to the user.

Algorithm 1 Create Data Subject

```
Input: PersonalInfo, ID
(pK, PK) ← CreateKeyPair()
exists ← HLF.ExistsAsset(ID)
if not exists then
    IPFSFile, CID ← CreateIPFSFile(PersonalInfo, PK)
    hash ← SHA256(IPFSFile)
    HLF.Write(ID, CID, hash)
    return pK
end if
return None
```

Since different parts of the user records are stored in the HLF blockchain and IPFS, the read function needs to interact with both to retrieve the information. As shown in Algorithm 2: (1) getting the user asset present in the HLF; (2) checks the "deleted" flag; (3) if the file was not already deleted, the API read the IPFS file; (4) calculates the hash code that compares with the hash code present in the HLF. After finishing the process, all the information in HLF, IPFS is returned to the user along with a tampered flag that became true if the information has been tampered.

Algorithm 2 Read Data Subject

```
Input: ID
asset ← HLF.GetAsset(ID)
if not asset.Deleted then
    file ← IPFS.Read(Asset.CID)
    hash ← SHA256(file)
    tampered ← hash ≠ Asset.hash
    return Asset, file, tampered
end if
return None
```

Figure 7 is a screenshot of the JSON object returned by the read function, as displayed in a web page. The API identifies and presents the origin (BC or IPFS) of all data. In the blockchain, the user ID, consent, timestamp, CID, hash and deleted flag, in the IPFS, were stored all the personal information (Address, Birthday, Email, Name, Phone Number) and the user ID, consent and private key.

The delete function, which is mandatory for GDPR compliance, consists of three steps: (1) get the asset from HLF using the ID; (2) delete the user's IPFS file and set the asset's deleted flag to true. Algorithm 3 illustrates this process.

The update function enables the system to be compliant with GDPR's right to rectification, by allowing the modification of stored information. The method to update DS information consists of five steps: (1) get the asset from the HLF; (2) use the CID present in the asset, delete the IPFS file;

Fig. 7 Successfully read user information

Algorithm 3 Delete Data Subject

```
Input: ID
asset ← HLF.GetAsset(ID)
if not asset.Deleted then
    IPFS.Delete(Asset.CID)
    asset.setDeleted(True)
    return true
end if
return false
```

(3) create a new one; (4) calculate a new hash; (5) update the CID and hash in the BC record. The original record is never deleted; however, this is not problematic, since personal data are not stored there. The BC only contains validation hash of the personal data stored off-chain, and it is not possible to reconstitute the personal information from its hash.

Changing the public key is similar to update information: (1) create a new key pair; (2) delete the previous IPFS file; (3) create a new one with the new public key; (4) calculate the hash; (5) write the new CID and hash in the BC; (6) return the new private key.

Algorithm 4 is a simplification of the one used. The algorithm showed only receives as input one field, while the actual algorithm supports one or more fields as input. To list user records by field: (1) use the function "list all users" provided by the chaincode; (2) iterate over the list; (3) add the requested field to an array. The real function has auxiliary arrays to which add the other fields.

Algorithm 4 List by Field

```
Input: Field
assetsList ← HLF.GetAllAssets()
Array ← Ø
for all asset ∈ assetsList do
  if not asset.Deleted then
    file ← IPFS.read(asset.CID)
    Array.append(file.Field)
  end if
end for
return Array
```

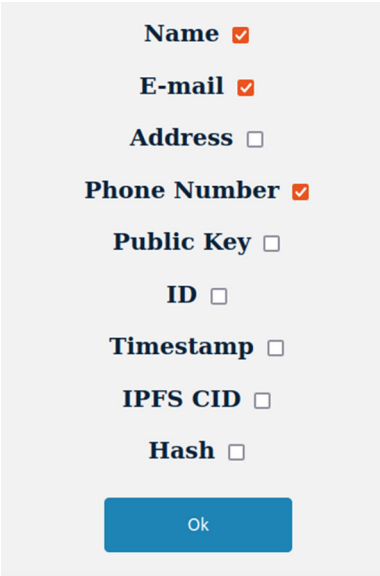

Fig. 8 Field choose page

Fig. 9 Read information by field

Figure 8 is a screenshot of the web page where the DP can choose the fields for consultation. Figure 9 shows the JSON object returned by this operation, displayed in a web page (DP chosen to consult name, email and phone number).

The function that returns the list of users is very similar to the Algorithm 4. However, instead of adding only the information of a field to the array, the "list all" algorithm appends the whole user information (HLF record and IPFS file).

8 Evaluation

The evaluation of the tool is relative to the success of overcoming the challenges of GDPR. This section enumerates the main challenges found in the literature and explains how each one was addressed, and how a blockchain system can be designed to be fully compliant with GDPR.

8.1 Right to be forgotten

The main challenge found in the literature is the immutable nature of records in blockchain (one of the core properties of blockchain) versus the Article 17 of GDPR which states that "The data subject shall have the right to obtain from the controller the erasure of personal data concerning him or her without undue delay" [9]. The Right to Erasure (a.k.a Right to be Forgotten) is incompatible with a classical approach to store data in the blockchain system.

There are several approaches to address this problem. We chose to add to the blockchain another storage system. Adding an off-chain storage enables to have with the necessary security to protect the data, while keeping the qualities of blockchain, such as tamper-proof and the immutability of records.

Some authors suggested to change the blockchain structure. For example, using chameleon hash [22, 44, 45], although that alternative is not very reliable because it breaks one of the biggest properties of the blockchain, the immutability of records. Other papers propose to add updated information blocks to the BC, but the previous information would not be totally erased, and so the system will not be in accordance with the GDPR.

Another possible solution to this problem would be to encrypt the information and destroy the key [32, 46–48]. However, independently of the encryption technology, it is not possible to know for how many years that technology will continue to be secure, especially considering the advances that quantum cryptography has had in the last few years. That means, personal information that would be erased could be revealed later when technology allows the decryption.

We chose to use another storage system to hold the sensitive data. This allows data to be completely erased (off-chain) while maintaining a hash code of the information that is also a pointer to where the information is stored.

We propose to use IPFS, a distributed and low latency system that, like blockchain, does not rely on a central authority, and all the nodes have the same control and authority over the network.

This proposal was designed to allow fully deletion of personal data. All personal information is stored in the same file in the IPFS network. When a DS wants to erase his information from the system, the API deletes the IPFS file. The only information that is left in the system relative to the personal data is the hash and the CID. The hash can be used to verify logs and to audit the blockchain, but not to recover the deleted information.

8.2 Right to rectification

According to Article 16 (Right to Rectification) "data subjects have the right to rectify inaccurate personal data" [23]. This means that controllers must ensure that the data subject can rectify inaccurate or incomplete information [23], but this article goes against the essence of blockchain for the same reasons as the previous challenge: blockchain records are unalterable.

This challenge was solved with the same approach as the right to be erasure. Since a file can be deleted, to rectify any information, delete the previous file is deleted and create a new file created with the updated information. This approach guarantees that the respective metadata in the HLF are updated. Since the previous file is deleted, it is impossible to access the previous information. Like deleting, the updating process leaves the previous hash and CID in the blockchain.

8.3 Right to data portability

Right to Data Portability (Article 20) states that every DS should receive (upon request) his "own personal data from the DC in a structured and commonly used machine-readable format" [23]. This right is ensured because all personal information relative to a DS is stored in a single file in the IPFS network. So this requirement is met by the read function in the API.

The DS may get all his information in two different ways. The first way is to use the read function present in the API. When the read function is called, the API returns the personal information and metadata. The second way is to ask the data controller to invoke the same function on his record and send the file to the DS.

8.4 Identification of roles and accountability

The identification of roles and accountability is not straightforward. There is more than one system storing and processing information, and more than one hierarchy architecture. To address this challenge, this section is divided into three parts: roles and accountability in the blockchain, in the IPFS, and in the API.

GDPR identifies four stakeholders, the data subject (the person that the data refers to), the data controller (who determines the purpose and the processing that will be done), the data processor (who indeed processes the data) and the supervisor authority (an independent public authority that is responsible for the enforcement of the GDPR) [17].

Blockchain Hyperledger Fabric is a permissioned block chain that has an authorisation and authentication system as well as a hierarchy. HLF allows to distinguish several stakeholders, orderers, peers, CA admins and organisation admins. In HLF, orderers maintain the ledger and can be considered data processors, peers (who invoke functions on the ledger) are considered data controllers and data processors (since, beyond storing information, they decide what assets are stored).

IPFS In an IPFS network, all the nodes have access to the same data and, using cluster, all the nodes have privileges to create, read and update the files. Given these characteristics, all nodes in an IPFS network can be considered both data processors and data controllers. The IPFS nodes correspond to the same machines as HLF peers.

API There are specific roles in the API to DS, DP and DC. The only stakeholders that should process and control data are the DC and DP. This makes possible to define an accountable group for each action. The accountability for the action should fall on the DC and DP users.

The Supervisor Authority (SA) has access to HLF and to the IPFS network. However, SA has no access to the key used to cipher the information before storing the data in the IPFS. The SA can verify that a DS has been erased by checking whether the correspondent IPFS file is not available. To ensure that the information about a DS has been rectified, the SA should try to retrieve the previous IPFS file and check that only the new one exists. All these verifications are performed without having access to the user information, only to the hash codes and ciphered information.

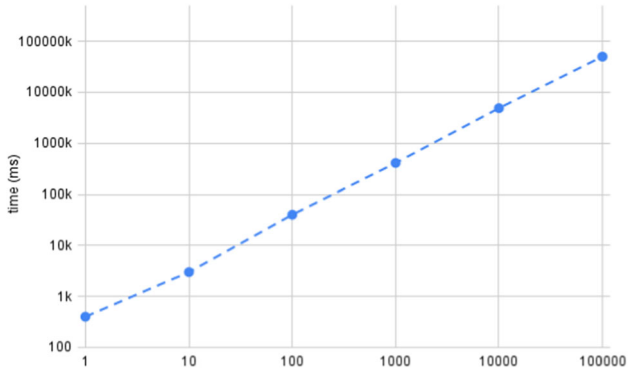

Fig. 10 Create multiple records

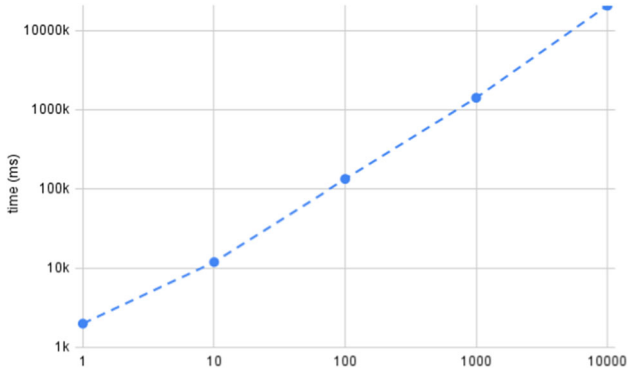

Fig. 11 Delete multiple records

8.5 Performance

The performance tests were made by applying a use case with fictional records of personal data. The records contained name, email, cellphone number, birthday and address and had the following structure: "Rhonda Givens; rhonda@email.com; 907604624; 2017-03-31; Rhonda's Street"

We performed tests in the most important functions of the system: create, read, delete, list all users, and list one field of each user. Each one of the tests was performed using a bash script that made requests to the API. Figure 10 illustrates the time taken to create 1, 10, 100, 1000, 10.000 and 100.000 personal records, both in HLF and in IPFS. The time to delete records is illustrated in Fig. 11 for the same values, except 100.000. In both cases, the time increase is linear.

The read function is independent of the number of records stored in the system. While the number of records stored in the system increased, the time required to read a random record remained similar. This can be verified in Fig. 12.

Figure 13 reports the time to list all personal data in the system, while Fig. 14 shows the time taken to get all the information relative to a field. To perform this test, we increased the number of records while reading always the same field (name). Both graphs show linear growth.

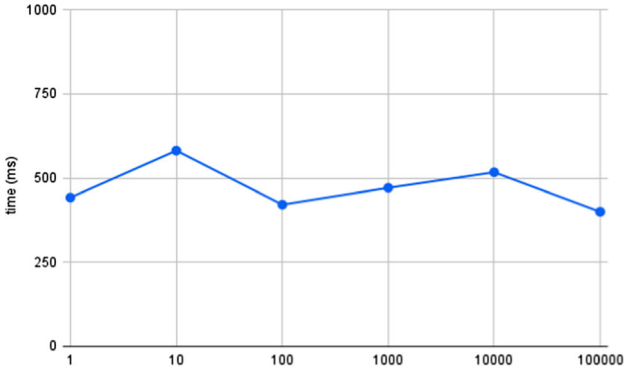

Fig. 12 Read one record depending on the total number of records

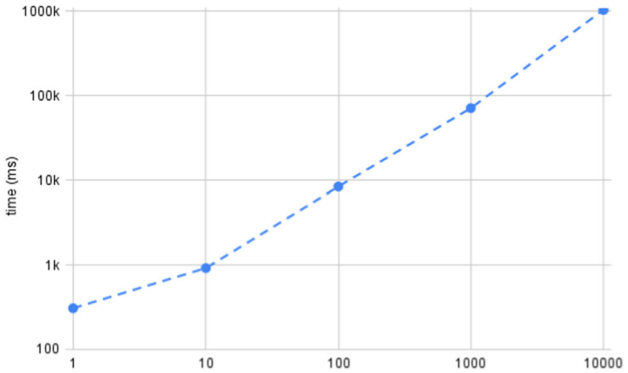

Fig. 13 List all records

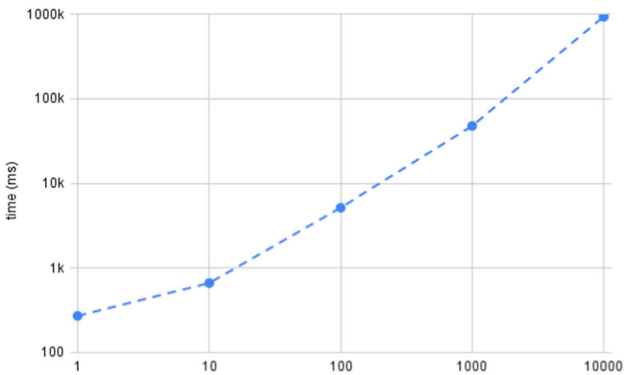

Fig. 14 List one specific information

For the previous tests, the scripts were run in just one virtual machine, and checked in the other three. For the following tests, we invoked the same script in all four machines at the same time. Figures 15 and 16 show that the time to create and delete records decreases by a factor of four (the number of VMs), but remains linear.

8.6 Main limitations

In a distributed system, it is always difficult to make a clear identification of data controllers and data processors.

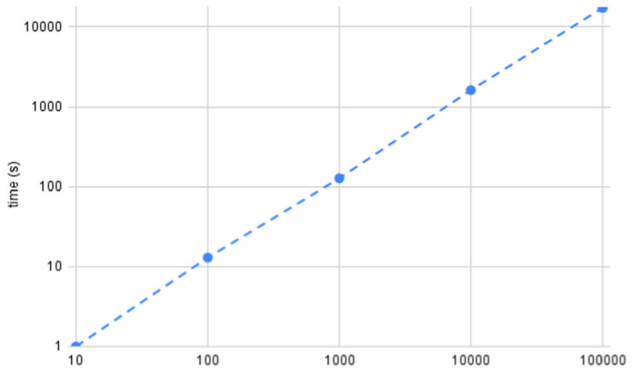

Fig. 15 Create in 4 VMs

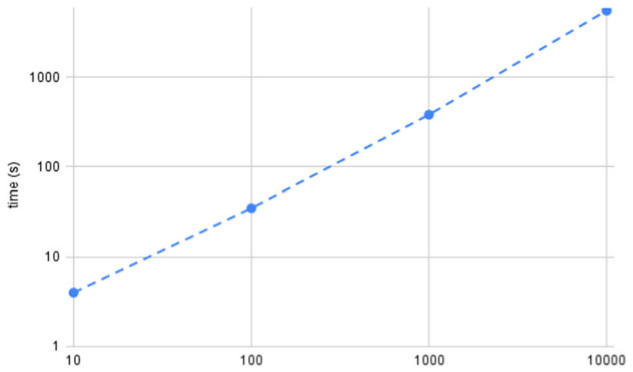

Fig. 16 Delete in 4 VMs

Although we take this challenge into account, the process to identify these two roles remains a challenge.

There exists a substantial delay in some functions (List and Delete) when the number of users escalates, although there is a simple solution to speed up each of these operations. Regarding List All, the users' data can be distributed over different web pages (instead of displaying everything in just one) and, regarding deletion, instead of invoking the IPFS garbage collector after each elimination, it can be invoked periodically. However, we have not implemented and tested this solutions.

9 Conclusion

In this paper, we designed and implemented a blockchain system to store personal data, complying with GDPR. We proposed a model and described each step of the implementation, along with the technologies. We evaluated the system in terms of performance and compliance with GDPR to perceive the weaknesses and strengths of using blockchain to store personal data, and to know which techniques must be used to comply with GDPR. Our system keeps the major benefits of storing data in the blockchain (verifiability integrity, tamper resistance, no single point of failure, availability, among others) while complying with the most challenging articles of GDPR concerning the storage of personal data in the blockchain, a immutable system (right to be forgotten, right to rectification, right to data portability).

9.1 Research contributions

This paper presents a system to store personal data in blockchain complying with GDPR, along with functional system implementation. The paper shows the results of testing the main operations on data with authentication and authorisation. The system implementation includes a REST API with a graphic interface that can be used through a web browser or directly through the console. In addition, we also have written a full (and easy to follow) tutorial to create a Hyperledger Fabric network with three organisations, three orderer nodes and four peer nodes [49].

9.2 Future work

We propose three possible directions for future work: improve the system functionalities, add new functionalities, and further evaluate the current tool.

Regarding the improvement of the system, it would be important to do a systematic and deep study about the identification of data processors and data controllers in a blockchain system with off-chain storage. It would be also important to reduce the time of the "List All" and "Delete" functions, for example, using the methods mentioned in the previous section.

Regarding additional functionalities, we are currently developing a new survey feature that allows the data controller to create free-response surveys. These surveys have a specific consent and deadline date, after which all information relating to the survey is deleted. The data processor can access the survey data without having access to the identification of the users that responded. Users can create, edit and delete responses and can only respond once to each survey.

Finally, for a more extensive evaluation of the system, we propose to do a performance test with a benchmark on the system using IPFS and Cloud as off-chain storage and compare results. Also, it would be very interesting to have GDPR experts evaluate the model to verify any possible flaws in the GDPR compliance.

Author Contributions R.M.G. and P.R.C. did conceptualisation and software; R.M.G, M.M.S. and P.R.C. done methodology; R.M.G. done writing—original draft; M.M.S. and P.R.C. were involved in writing—review and editing; M.M.S. and P.R.C. supervised the study; M.M.S. done project administration. All authors have read and agreed to the published version of the manuscript.

Funding Open access funding provided by FCT|FCCN (b-on). This research was funded by Guest Intelligence Chain: ALG-01-0247-

FEDER-047399 and Fundação para a Ciência e Tecnologia: CISUC R&D Unit-UIDB/00326/2020, project code UIDP/00326/2020.

Data availability The algorithms and smart contracts used in this paper can be found at https://github.com/ricardo-martins-goncalves/olympus.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Vasylkovskyi, V., Guerreiro, S., Sequeira, J.S.: Blockrobot: increasing privacy in human robot interaction by using blockchain. In: 2020 IEEE international conference on blockchain (Blockchain) p. 106–115 (2020). https://doi.org/10.1109/blockchain50366.2020.00021
2. Deep, G., Mohana, R., Nayyar, A., Sanjeevikumar, P., Hossain, E.: Authentication protocol for cloud databases using blockchain mechanism. Sensors (2019). https://doi.org/10.3390/s19204444
3. Abouali, M., Sharma, K., Ajayi, O., Saadawi, T.: Blockchain framework for secured on-demand patient health records sharing. In: 2021 IEEE 12th annual ubiquitous computing, electronics mobile communication conference (UEMCON) pp. 0035–0040 (2021). https://doi.org/10.1109/UEMCON53757.2021.9666482
4. Pandey, V., Pant, M., Snasel, V.: Blockchain technology in food supply chains: review and bibliometric analysis. Technol. Soc. **69**, 101954 (2022). https://doi.org/10.1016/j.techsoc.2022.101954
5. Gao, L., Li, L., Chen, Y., Xu, C., Xu, M.: Fgfl: a blockchain-based fair incentive governor for federated learning. J. Parallel Distrib. Comput. **163**, 283 (2022). https://doi.org/10.1016/j.jpdc.2022.01.019
6. Gururaj, H., Manoj, A.A., Kumar, A.A., Nagarajath, S., Kumar, V.R.: Adoption of pets in distributed network using blockchain technology. Int. J. Blockchains Cryptocurr. **1**(2), 107 (2020)
7. Yang, L., Zou, W., Wang, J., Tang, Z.: Edgeshare: a blockchain-based edge data-sharing framework for industrial internet of things. Neurocomputing **485**, 219 (2022). https://doi.org/10.1016/j.neucom.2021.01.147
8. National Institute of Standards and Technology, Blockchain technology overview. Tech. Rep. Federal Information Processing Standards Publications (FIPS PUBS),2018, U.S. Department of Commerce, Washington, D.C. (2018). https://doi.org/10.6028/nist.if.8202
9. European Commission, Eu general data protection regulation (gdpr): 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), Official Journal of the European Union (2016)
10. Gonçalves, R.M., da Silva, M.M., da Cunha, P.R.: Using blockchain to store personal information: a systematic literature review. Int. J. Blockchains Cryptocurr. **3**(3), 235 (2022)
11. Nakamoto, S.: A peer-to-peer electronic cash system, Bitcoin.–URL: https://bitcoin.org/bitcoin.pdf**4** (2008)
12. Kaur, A., Nayyar, A., Singh, P.: Blockchain: a path to the future, Cryptocurrencies and Blockchain Technology Applications pp. 25–42 (2020)
13. Javed, I.T., Alharbi, F., Margaria, T., Crespi, N., Qureshi, K.N.: Petchain: a blockchain-based privacy enhancing technology. IEEE Access **9**, 41129–41143 (2021). https://doi.org/10.1109/access.2021.3064896
14. Lone, A.H., Mir, R.N.: Consensus protocols as a model of trust in blockchains. Int. J. Blockchains Cryptocurr. **1**(1), 7 (2019)
15. Karthika, V., Jaganathan, S.: A quick synopsis of blockchain technology. Int. J. Blockchains Cryptocurr. **1**(1), 54 (2019)
16. Rieger, A., Guggenmos, F., Lockl, J., Fridgen, G., Urbach, N.: Building a blockchain application that complies with the EU general data protection regulation. MIS Q. Exec. **18**(4), 263–279 (2019). https://doi.org/10.17705/2msqe.00020
17. Schwerin, S.: Blockchain and privacy protection in the case of the European general data protection regulation (GDPR): a Delphi study. J. Br. Blockchain Assoc. **1**(1), 1–77 (2018). https://doi.org/10.31585/jbba-1-1-(4)2018
18. Chowdhury, M.J.M., Colman, A., Kabir, M.A., Han, J., Sarda, P.: Blockchain as a notarization service for data sharing with personal data store. In: 17th IEEE international conference on trust, security and privacy in computing and communications/ 12th ieee international conference on big data science and engineering pp. 1330–1335 (2018)
19. Vo, H.T., Mehedy, L., Mohania, M Abebe, E.: Blockchain-based data management and analytics for micro-insurance applications. In: CIKM '17: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management p. 2539–2542 (2017). https://doi.org/10.1145/3132847.3133172
20. Yan, X., An, X., Ye, W., Zhao, M., Wu, J.: A blockchain-based subscriber data management scheme for 6g mobile communication system. In: 2021 IEEE Globecom Workshops (GC Wkshps) pp. 1–6 (2021). https://doi.org/10.1109/GCWkshps52748.2021.9682154
21. Aslan, U., Şen, B.: Gdpr compliant audit log management system with blockchain: Gdpr uyumlu denetim günlüğü yönetim sistemi, IEEE Access pp. 1–3 (2021)
22. Tatar, U., Gokce, Y., Nussbaum, B.: Law versus technology: blockchain, GDPR, and tough tradeoffs. Comput. Law Secur. Rev. (2020). https://doi.org/10.1016/j.clsr.2020.105454
23. Mirchandani, A.: The GDPR-blockchain paradox: exempting permissioned blockchains from the GDPR. Fordham Intellect. Prop. Media Entertain. Law J. **29**(4), 1199–1241 (2019)
24. Onik, M.M.H., Kim, C.S., Lee, N.Y., Yang, J.: Privacy-aware blockchain for personal data sharing and tracking. Open Comput. Sci. **9**(1), 80 (2019)
25. European Commision. What is ebsi? https://ec.europa.eu/digital-building-blocks/wikis/display/EBSI/Home
26. Barati, M., Buchanan, W.J., Lo, O., Rana, O.: A privacy-preserving platform for covid-19 vaccine passports. In: Proceedings of the 14th IEEE/ACM international conference on utility and cloud computing companion (2021). https://doi.org/10.1145/3492323.3495626
27. Dauden-Esmel, C., Castella-Roca, J., Viejo, A., Domingo-Ferrer, J.: 2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP) (IEEE, 2021), p. 68–73. https://doi.org/10.1109/csp51677.2021.9357602

28. Chiu, W.Y., Meng, W., Jensen, C.D.: My data, my control: a secure data sharing and access scheme over blockchain. J. Inf. Secur. Appl. **63**, 103020 (2021)
29. Truong, N.B., Sun, K., Lee, G.M., Guo, Y.: GDPR-compliant personal data management: a blockchain-based solution. IEEE Trans. Inf. Forensics Secur. **15**, 1746–1761 (2020). https://doi.org/10.1109/tifs.2019.2948287
30. Abid, A., Cheikhrouhou, S., Kallel, S., Jmaiel, M.: NovidChain: blockchain-based privacy-preserving platform for COVID-19 test/vaccine certificates. Software Pract. Exp. **52**(4), 841–867 (2022)
31. Agbo, C.C., Mahmoud, Q.H.: 2020 IEEE international conference on systems, man, and cybernetics (smc) (IEEE, 2020), pp. 812–817
32. Rotondi, D., Saltarella, M., Giordano, G., Pellecchia, F.: Distributed ledger technology and European union general data protection regulation compliance in a flexible working context. Internet Technol. Lett. (2019). https://doi.org/10.1002/itl2.127
33. Lo, O., Buchanan, W.J., Sayeed, S., Papadopoulos, P., Pitropakis, N., Chrysoulas, C.: Glass: a citizen-centric distributed data-sharing model within an e-governance architecture. Sensors **22**(6), 2291 (2022). https://doi.org/10.3390/s22062291
34. Chrysoulas, C., Thomson, A., Pitropakis, N., Papadopoulos, P., Lo, O., Buchanan, W.J., Domalis, G., Karacapilidis, N., Tsakalidis, D., Tsolis, D.: Computer Security. ESORICS 2021 International Workshops. (Springer International Publishing, 2022), pp. 40–57
35. IPFS Community. Ipfs cluster. https://ipfscluster.io/
36. Wolff, E.: Microservices: Flexible Software Architecture. Pearson Education Inc., London (2017)
37. Auer, F., Lenarduzzi, V., Felderer, M., Taibi, D.: From monolithic systems to microservices: an assessment framework. Inf. Softw. Technol. **137**, 106600 (2021). https://doi.org/10.1016/j.infsof.2021.106600
38. Hyperledger Fabric Community . A blockchain platform for the enterprise (2020). https://hyperledger-fabric.readthedocs.io/en/release-2.2/
39. Politou, E., Alepis, E., Patsakis, C., Casino, F., Alazab, M.: Delegated content erasure in IPFS. Futur. Gener. Comput. Syst. **112**, 956–964 (2020). https://doi.org/10.1016/j.future.2020.06.037
40. IPFS Community. IPFS powers the distributed web. https://ipfs.tech/
41. IPFS Community. Experimental features of go IPFS (2021). https://github.com/ipfs/kubo/blob/release-v0.9.0/docs/experimental-features.md#private-networks
42. Yadav, K.A., Vijayakumar, P.: Lppsa: an efficient lightweight privacy-preserving signature-based authentication protocol for a vehicular ad hoc network. Ann. Telecommun. **77**(7), 473 (2022)
43. Martins Gonçalves, R.: Olympus source code. https://github.com/ricardo-martins-goncalves/olympus
44. Wulff, C.M.: The right to be forgotten in post-google Spain case law: an example of legal interpretivism in action? Compar. Law Rev. **26**, 255–279 (2021). https://doi.org/10.12775/clr.2020.010
45. Jiménez-Gomez, B.S.: Risks of blockchain for data protection: A European approach. Santa Clara Hight Technol. Law J. **36**(3), 280–343 (2020)
46. Hofman, D., Lemieux, V.L., Joo, A., Batista, D.A.: The margin between the edge of the world and infinite possibility. Rec. Manag. J. **29**(1/2), 240–257 (2019). https://doi.org/10.1108/rmj-12-2018-0045
47. Mahindrakar, A., Joshi, K.P.: 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS) (IEEE, 2020), p. 86–93. https://doi.org/10.1109/bigdatasecurity-hpsc-ids49724.2020.00026
48. Teperjian, R.: The puzzle of squaring blockchain with the general data protection regulation. Jurimetrics **60**(3), 253 (2020)
49. Martins Gonçalves, R.: Guide to deploy production network hyperledger fabric (2022). http://web.tecnico.ulisboa.pt/ist198668/hlf_tutorial.pdf