D4.3 Design and specifications of the blockchain-based auditing mechanism v1

WP4 – SPHINX Toolkits

Version: 1.00

SPHINX

A Universal Cyber Security Toolkit for Health-Care Industry

Disclaimer

Copyright message

Document information

Grant Agreement Number	**826183**	**Acronym**	**SPHINX**
Full Title	A Universal Cyber Security Toolkit for Health-Care Industry		
Topic	SU-TDS-02-2018 Toolkit for assessing and reducing cyber risks in hospitals and care centres to protect privacy/data/infrastructures		
Funding scheme	RIA - Research and Innovation action		
Start Date	1st January 2019	**Duration**	36 months
Project URL	http://sphinx-project.eu/		
EU Project Officer	Reza RAZAVI (CNECT/H/03)		
Project Coordinator	National Technical University of Athens - NTUA		
Deliverable	D4.3 Design and specifications of the blockchain-based auditing mechanism v1		
Work Package	WP4 – SPHINX Toolkits		
Date of Delivery	**Contractual** M20	**Actual**	M20
Nature	R - Report	**Dissemination Level**	P - Public
Lead Beneficiary	TECNALIA		
Responsible Author	Santiago de Diego	**Email**	santiago.dediego@tecnalia.com
		Phone	
Reviewer(s):	TEC , AIDEAS		
Keywords	blockchain , IoT , identity management		

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Document History

Version	Issue Date	Stage	Changes	Contributor
0.10	10/06/2020	Draft	ToC	Santiago de Diego (TECNALIA)
0.20	22/06/2020	Draft	Added Content	Santiago de Diego (TECNALIA), Jason Xabier Mansell (TECNALIA)
0.30	14/06/2020	Draft	Finalised Content	Santiago de Diego (TECNALIA), Jason Xabier Mansell (TECNALIA)
0.40	21/08/2020	Draft	Internal Review 1	Talal Ous (TEC)
0.50	21/08/2020	Draft	Internal Review 2	Serafeim Moustakidis (AIDEAS)
0.60	24/08/2020	Pre - Final	Incorporated Review Comments	Santiago de Diego (TECNALIA), Jason Xabier Mansell (TECNALIA)
0.70	31/08/2020	Pre – Final	Quality Control	George Doukas (NTUA), Michael Kontoulis (NTUA)
1.00	31/08/2020	Final	Final	Christos Ntanos (NTUA)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Executive Summary

This document describes the architecture, design and implementation of the component: “Blockchain-Based Threat Registry” or BBTR, in the SPHINX project. This document goes deep into the BBTR component, first of all presenting the state of the art and the justification of the use of a Blockchain technology for implementing such component, and then describing the different steps to enable this component in the SPHINX Framework. Interfaces, which are the connection points with other components of the ecosystem are also presented in the document. Besides, as the component is related to three of the SPHINX use cases, these use cases are enumerated and it is also explained how this component interacts with these use cases.

The key innovation attained in this document is therefore the architectural design of the Blockchain-Based Threat Registry, providing details about its implementation and test cases.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Contents

1 Introduction 9
1.1 Purpose & Scope 9
1.2 Structure of the deliverable 9
1.3 Relation to other WPs & Tasks 9
2 Blockchain-Based Threat Registry (BBTR) 10
2.1 Background 10
2.1.1 Health and Blockchain technologies 10
2.1.2 IoT devices and Blockchain 11
2.2 Blockchain as a solution for a threat sharing platform 12
2.3 Public vs Private approach 13
2.4 Architecture and detailed design 13
2.4.1 Identity Management in the BBTR 16
2.4.2 Interfaces design 17
2.5 Implementation details 20
2.5.1 Test cases 23
3 Use-case validation: Sharing Threat Knowledge Between Healthcare Providers 24
4 References 25

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Table of Tables

Table 1: Initiatives on the use of blockchain for management of healthcare data & services 11
Table 2: Fields of the JSON structure 15
Table 3: BBTR Functional Specifications 18
Table 4: SPHINX BBTR Interface Specifications 19

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Table of Figures

Figure 1: BBTR architectural design 15
Figure 2: User Certificate in Hyperledger Fabric 16
Figure 3: Elements of the MSP 17
Figure 3: SPHINX BBTR Component Diagram 17
Figure 5: Authenticating against the BBTR 22
Figure 6: Submitting a new threat to the BBTR 22
Figure 7: Querying a group of threats matching a description 23

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Table of Abbreviations

BBTR - Blockchain-Based Threat Registry

P2P - Peer-to-peer

DApp - Decentralized Application

DLT - Distributed Ledger Technology

HIE - Health Information Exchange

HER - Electronic Health Record

IoT - Internet of Things

AI - Artificial Intelligence

SPoF - Single Point of Failure

API - Application Programming Interface

CRL - Certificate Revocation List

MSP - Membership Service Provider

PKI - Public Key Infrastructure

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

1 Introduction

1.1 Purpose & Scope

This document, named “Design and Implementation of the blockchain-based auditing mechanism”, elaborated as part of Task 4.3 - SPHINX tools, presents the first version of the SPHINX BBTR (Blockchain-Based Threat Registry) architecture and its detailed technical specifications.

The document also includes the State of the Art in Blockchain, Health and IoT as an introduction for the component description. The architecture of the BBTR has been defined following the Hyperledger Fabric design principles, as it has been the chosen technology for implementing the solution.

The final purpose of the document is to present the BBTR component to the reader and help him understand how it works, and how it can be deployed to be used. This Deliverable covers every aspect of the aforementioned component but the rest of the elements of the SPHINX Project are not disclosed in it.

1.2 Structure of the deliverable

In section 2, a State of the Art in terms of Blockchain, Health and IoT is presented to show the necessity of implementing a component like this. Section 3 describes the BBTR in deep detail; first of all, Blockchain is shown as a potential technology to be used in a threat sharing scenario and secondly, a justification of the use of a private approach, rather than a public one is included. In section 2, the BBTR design and architecture are shown as well as details about its implementation, interfaces with other components and test cases. Finally, the BBTR is included in some of the uses cases.

1.3 Relation to other WPs & Tasks

This work is related to D2.3 Use Case definition and requirements document, D2.4 SPHINX requirements and guidelines and D2.5 SPHINX Architecture. As part of the validation of the component a direct link to WP7 has been established in order to define the validation process and test cases that will be carried out in the use case validation within D7.1 Pilot plans including evaluation framework.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

2 Blockchain-Based Threat Registry (BBTR)

2.1 Background

2.1.1 Health and Blockchain technologies

The healthcare industry is one of the world's largest industries, consuming over 10% of gross domestic product (GDP) of the most developed nations including generalization and commercialization of goods and services to treat patients with curative, preventive, rehabilitative, and palliative care [1]. Being a complex system of interconnected entities operating under heavy regulation, patient data is highly fragmented and the cost of healthcare delivery is continuously rising due to inefficiencies in the system and dependence on several intermediaries. This has accentuated a need for an information technology system that can remove the middlemen and cut down costs while maintaining trust and transparency [2]. Blockchain is the technology behind Bitcoin [3], an open peer-to-peer (p2p) value transfer network that solved the problem of double spending for the first time. The blockchain is currently a revolutionary technology which can assist in solving the challenges of healthcare, as well as other use cases, by providing decentralized trust. Blockchain can be public or private depending on the permission level [4].

Trust and traceability are the two basic promises of the blockchain obtained out of the box which solves the generic trust problem on all public, federated, and organization levels. However, these traits are not always enough to provide a complete solution, which is why we often see blockchain paired with strong cryptographic protocols like zero-knowledge proofs [5]. This pairing ensures to provide trust, traceability, security, and control which are the core building blocks for critical solutions in several industries including health care. Data recorded in the blockchain cannot be changed or deleted without leaving a trace. This immutability and traceability of the data is a critical requirement for any health care system.

Despite its massive potential, there are limitations with the current state of the blockchain. Currently, every node in the network processes every transaction making the blockchain rather slow and unsuitable to handle scenarios where many transactions per second are issued. This disparity highlights the scaling issues that blockchains must overcome for wider adoption across all industries. Moreover, with the growth in usage, the size of the blockchain is increasing enormously, making it difficult for nodes to have the full copy of this ledger.

In the last decade, we have observed the evolution of blockchain in different stages. Cryptocurrencies represented the first generation, which was initially designed as an alternative payment system. Then, decentralized applications based on smart contracts (DApps) emerged as the 2.0 applications providing business logic abstraction and execution on a trusted platform. Nowadays, there are other novel DLT technologies considering architectures different from Blockchain, for example Directed Acyclic Graphs.

Several projects are focusing on developing some form of blockchain-based Health Information Exchange (HIE) and providing data and services marketplace, as shown in Table 1. Among them, some are targeting electronic health records (EHR) data while others are specializing data modalities such as genomics and dermatology. For example, Medrec [6] is an open source blockchain platform for EHR management, which was recently tested in collaboration with Beth Israel Deaconess Center. Patientory [7] is developing a HIE powered by its own blockchain. HealthSuite Insights of Philips Healthcare is testing Verifiable Data Exchange Process, a product that enables the secure and traceable data exchange between the members inside a network of hospitals and universities [8]; all the data exchanges inside the network are stored in a blockchain alongside the identities of the people performing the exchanges to create an audit trail of the data exchange. Medshare [9] provides a blockchain based data sharing of electronic medical records among untrusted parties by introducing data provenance, auditing, and trailing on medical data. Utilizing smart contracts and an access control system, they claim that their system can effectively trace the behavior of the data and revoke access to violated rules and

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

permissions on the data. IRYO [10] is creating a global repository of health data in OpenEHR format [11]. Table 1 features other initiatives on the use of blockchain-based solutions for healthcare data management.

Multiple projects are focusing on particular modalities of data such as genomics and imaging. Genomics, in particular, has attracted significant interest from entrepreneurs and companies probably because of the recent popularity of personal genome sequencing, the importance of genomics data, and an immense possibility of its monetization. Personal genomics companies such as 23andMe and AncestryDNA monetize the genetic data by selling access to third parties such as labs and biotech companies. Several startups such as Encrypgen [12], Nebula Genomics [13], LunaDNA [14] are developing a blockchain based genomics data-exchange platform or network. With the blockchain-based platforms, they claim to reduce the cost of genome sequencing, to give control of the data to patients, and to share the value captured from the monetization of the data to patients.

Group	Projects
EHR	Medrec [15], Patientory [16], HealthSuite Insights Philips Healthcare [17], Gem Health [18], Medshare [19], Iryo [20], FHIR Chain [21], OMNI PHR [22], Medicalchain [23], Doc.ai [24], Hearthy [25]
Genomics	Encrypgen [26], Nebula Genomics [27], lunaDNA [14], Zenome [28], Genomes.io [29], Shivom [30]
Imaging	ETDB-Caltech [31], Patel et. al 2018 [32], OPU Labs[32], MedX Protocol[34], Dermonet [35], Akiri switch [36]

Table 1: Initiatives on the use of blockchain for management of healthcare data & services

2.1.2 IoT devices and Blockchain

The IoT paradigm provides solutions for a wide range of applications such as smart cities, traffic congestion, waste management, structural health, security, emergency services, logistics, retails, industrial control, and health care. [37] Medical care and health care represent one of the most attractive application areas for the IoT. [38] IoT has the potential to give rise to many medical applications such as remote health monitoring, fitness programs, personalized care, real-time monitoring, chronic diseases, and elderly care. Compliance with treatment and medication at home and by healthcare providers is another important potential application. IoT-based healthcare services are expected to reduce costs, increase the awareness about the progression of medical conditions and diseases, and improve the user's experience. From the perspective of healthcare providers, the IoT has the potential to reduce device downtime through remote provision. In addition, the IoT can correctly identify optimum times for replenishing supplies for various devices for their smooth and continuous operation.

Furthermore, the IoT provides an efficient use of limited resources by ensuring their best use and service of more patients. [39] Because of this vast range of applications and popularity, there have arisen many issues and challenges that need to be carefully addressed. In addition, IoT has been at the core of several cyber-security incidents, resulting from weak security policies and network vulnerabilities. Incidents include data theft, device malfunction and remote control [40]. The need of constant monitoring needs to be balanced with low power consumption protocols and technologies (implying limited and constrained capabilities). As already mentioned, the vast application existing in this ecosystem needs a scalable and secure network that can respond to the increasing need of different interconnected devices. But still some of the most serious issues are related to a difficult technological transition, which require the existence of backwards compatibility and flexibility in the integration of existing devices to ensure a smooth transition. Although technical solutions are required, it is also a political and social issue too. This issue is the data protection of the massive heterogeneous data produced by the IoT devices as well as other several user and patient data produced and acquired during the health care activities. This issue regards matters such as physical security and secure network routing as well as general storage techniques. Policy issues also play a major role regarding the transparency and handling of such data. Finally, an important part is finding resource efficient security solutions [41].

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

While blockchain is not the panacea for all the issues we have mentioned before, it can most certainly alleviate some technical issues and challenges that affect the IoT paradigm and its application in healthcare devices. The main aspects where blockchain can contribute are data security, integrity and ensuring the handling of sensitive medical data, only if it is done by authorized agents. [42] Adapting though blockchain into an IoT network isn't as straightforward. Traditional blockchain techniques, such as the Proof of Work consensus algorithm though providing a core mechanism of blockchain functionality, require high resource usage and cause scalability and latency issues. These characteristics cannot be supported by most, if not all, IoT devices [43]. In addition to the use of blockchain technology in an IoT network, it is imperative to optimize the constrained requirements of those IoT networks, like limited processing power, storage space and network usage.

The proposed system in [44] avoids these issues by sending the raw data to a master "smart device", typically a smartphone or tablet, for aggregation and formatting by the proposed application. Once completed, the formatted information is sent to the smart contract for full analysis along with customized threshold values, which will evaluate the provided data and issue alerts to both the patient and healthcare provider, as well as automated treatment instructions for the actuator nodes if desired. There is not confidential medical information stored in the blockchain because of data handling compliance reasons, the system only records in the ledger the fact that events happened. The measurements themselves are forwarded to a designated EHR storage database, while a new transaction is added to the blockchain stating that the data was successfully processed. This system will have a private blockchain, meaning that only authorized viewers can read the blocks and only designated nodes can execute smart contracts and verify new blocks. Compared to traditional systems, this one provides a much higher degree of transparency, privacy, traceability, immutability and availability with the cost of a slight reduction in processing speed. Other proposed systems use attribute-based encryption along with blockchain technology to address the issues of privacy and confidentiality of data produced by an IoT network [45].

2.2 Blockchain as a solution for a threat sharing platform

An interesting use case for blockchain is threat intelligence. As written in [46], threat intelligence is an advanced process which involves gathering valuable insights including mechanisms, context, indicators, actionable advice and implications about an emerging or existing cyberthreat. Threat intelligence processes must be adapted to a company ecosystem to integrate it properly.

One of the issues related to threat intelligent these days is that companies usually spend a lot of time researching the same threats, while others are unnoticed. As a consequence, new tendencies emerge with information sharing between different interested parties being crucial. Following this principle, different companies are able to share information about threats to benefit each other. In the end, a distributed ledger of shared information is the ultimate goal of the threat intelligence philosophy.

The decentralization in threat management ecosystem is not new at all, previous works as [47],[48] study decentralization strategies applied to threat intelligence use cases. This approach manages threat sharing in a traditional way sharing web hosted feeds that collect the threats. Related to the health sector, [49] directly proposes a Blockchain solution to share healthcare data. Finally, focusing on AI and Blockchain, the reference work is mainly [50], where AI and Blockchain interaction is widely discussed.

Considering the previous state of the art in Blockchain and health ecosystems, our approach is slightly different. One of the premises of using Blockchain is to share a single view of data between different involved parties. As we have seen in the introductory section, this is exactly what Blockchain delivers. Our final goal is to set up a proof of concept of the first Blockchain-based threat management system.

In terms of security, Blockchain is fundamental to avoid a Single Point of Failure (SPoF) as well as it works as an anti-tampering mechanism. Privacy can also be guaranteed by applying ciphering over the communications.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Synchronization between different parties is also a crucial requirement when we are using a shared Threat Registry.

2.3 Public vs Private approach

This section tries to justify the use of a private Blockchain for the BBTR component in the SPHINX project, rather than a public one. First of all, in a public Blockchain, all users are allowed to interact with the network, so all of them can issue transaction freely. By contrast, in a private Blockchain, only the authorized users can issue transactions, and these transactions are only visible from inside of the network, in other words, people not participating in the network cannot see the transactions because they are not visible from the outside.

The SPHINX project needs a tool that is able to notice several medical centers, subscribed to a Blockchain network. In case a threat is produced in one of them, the rest can take measures to protect their systems. It is extremely important that this information about a potential threat is not leaked and known by other entities different from the medical systems. if this situation happens, a possible attacker could cause a lot of trouble in medical centres, so, in other words, this information must remain private, being only accessible by the interested parties. This reason could be sufficient to choose a private approach for the BBTR design. Also it could be preferable to restrict other parties, even within the SPHINX project, from accessing some of this information, that is why permissions are needed.

Furthermore, performance is another aspect to take into account to choose between a private or a public approach. First of all, it is well known that transactions are validated much faster in private Blockchains than in public ones. This is mainly because the consensus in public Blockchains is more complex due to the fact that it is necessary to protect the network from untrusted nodes, so extra comprobations and operations must take place, while in private Blockchain it doesn´t happen because we control the nodes who are in the network. Also, it logically takes more time to syncronise a network and reach consensus when more nodes are involved in the consensus process, a finding that leads us to the next consideration: permissioned vs permissionless Blockchain. In a permissioned Blockchain, only a few nodes are allowed to participate in the consensus, while in a permissionless approach, all nodes can participate for the consensus: Hyperledger Fabric would be an example of the first one, while Ethereum [51] will be of the second one. As a consequence, a permissioned Blockchain should offer improved performance, and this is another reason why we are using Hyperledger Fabric as a technology for the SPHINX Blockchain.

However, private Blockchains are not the panacea for this use case, as public ones can involve many actors in the solution so ultimately, it will reach more people. Mass adoption is one of the main advantajes of public Blockchains, so it could be interesting to have a public Blockchain for this use case if the final goal was to share information about threats with everyone. However, as we have seen, this is not the case, so we clearly need a private approach.

2.4 Architecture and detailed design

Focusing first on an identity management perspective, it is important to manage the identity of the different entities who are going to submit information to this shared registry in order to prevent impersonation of these entities by an attacker. This aspect is also important to achieve non-repudiation in the submission of a new entry in the registry. Blockchain is suitable to manage these identities in a very straightforward way using common certificate strategies as done in traditional systems but embedded into the infrastructure. More details about this aspect will be discussed in the “Identity Management” subsection. However, as an additional note, identity should not be always linked with a submitter, because sometimes the objective will be an anonymization of these submitters. Collaborative pseudo-anonymous Blockchains as Ethereum can achieve this goal in a very straightforward way, although this is out of the scope of this work.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

A threat registry is normally made up by simple data types, as shown in “use-case validation” section. One of the drawbacks when using Blockchain is the disk space as Blockchain is not suitable for storing high amounts of information, like multimedia. Therefore, the envisioned scenario does not foresee using this kind of data, allowing us to use Blockchain without disk-space issues. Another important drawback comes with latency, but again it is overcomed using a private network.

In addition to the aforementioned advantages related to privacy, integrity and availability, one of the main purposes of our solution is for us to be able to keep the traceability of the different threats. Furthermore, it is also important to guarantee the auditability of the whole system, that is why Blockchain is so suitable for this particular scenario. Further analysis can be performed by an auditing team or an incident response team, so an infrastructure like Blockchain is really helpful to make this task easier for these teams.

The architecture of the proposed distributed threat registry, herein named Blockchain Based Threat Registry (BBTR) is shown in Fig.1. Discontinuous lines represent boundaries or confidence zones. In this scenario, we have 3 different nodes located in 3 different zones (Fabric Organizations). Each node has the same copy of the shared ledger so all of them have a single view of the data, fully synchronized thanks to Blockchain. In the middle of the schema, it is shown the Ordering service, which in our design is composed by 3 nodes, corresponding to the different zones. This is a special subset of nodes used by Hyperledger Fabric to manage the communications in the network. In total, the minimum design is composed by 6 nodes (3 peer nodes plus 3 orderer nodes). However, this minimum configuration does not have much sense because at least 2 medical centres are necessary to share information about threats, but we have used this diagram for its simplicity, so it will be 4 peer nodes plus 4 orderer nodes, which is a total of 8 nodes. New medical centres can be easily added just adding a new organisation and peer nodes to the diagram.

There are two different actors in this diagram, the auditor and the medical centers. The medical center submits information about threats affecting their infrastructure in JSON format (more information about the specific information submitted to the Blockchain can be read at the end of this section). Arrow heads are not casual, because they represent read-write operations in our BBTR (incoming arrows denote read operations and outcoming arrows write operations, respectively). An actor with two arrowheads can act as a writer and reader at the same time. All connections are made between different parties by using secure protocols, as HTTPS, in order to prevent sniffing attempts. As shown in the diagram, actors act as Blockchain users communicating with a BaaS (Blockchain as a Service) through an API node.

As an example, the DSS component has been included to the diagram as an actor to show how external actors can connect to the BBTR to get useful information for them, in this case the DSS component uses the BBTR to feed its decision support engine. This actor will have read-only access to the BBTR, not being able to write any information.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

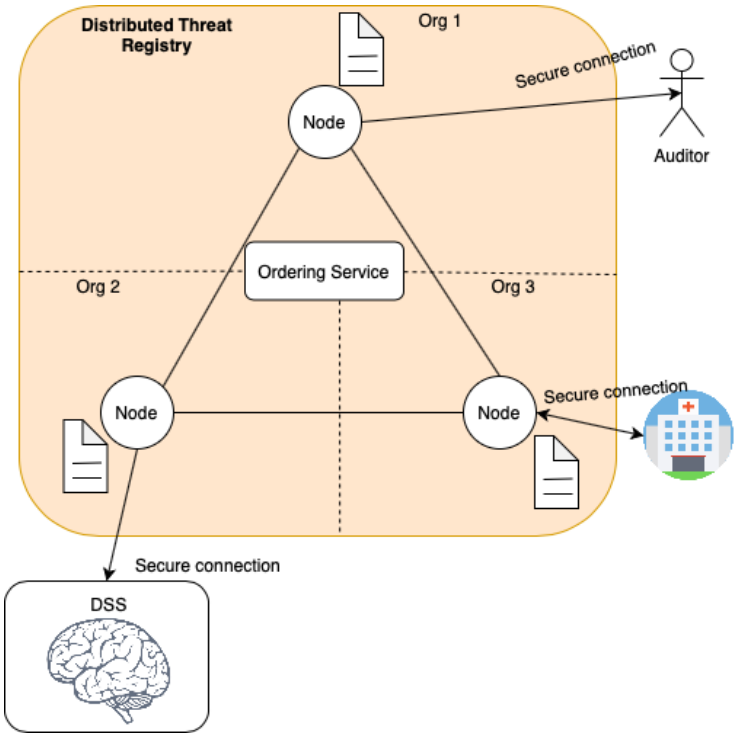

Figure 1: BBTR architectural design

Connections between client applications and the Blockchain nodes are made using a secure API REST communication. At an implementation level, this is basically an SSL over HTTP (HTTPS) connection using X509 certificates. As explained before, each node has its own certificate which identifies it in the network.

In terms of scalability, we can reach any desired level, because additional nodes can be added as needed. For our use case, we have implemented four nodes, which are more than enough for our testing purposes. At the enterprise level, the storage required by a public Blockchain is not practical given the amount of information that it contains. By using a permissioned paradigm, the obtained distributed registry limits the access to the appropriate entities, thus the stored data will be minimized, resulting in an acceptable trade-off between obtained features and storage/cost.

Initially, the data stored in the Blockchain follow the JSON structure presented in Table 2. Mandatory fields are written in bold type, whereas the rest are optional. Moreover, the structure can be adapted to the requirements of each particular use-case. The proposed structure can be used to classify different threats using simple fields to avoid escalating the disk usage in the Blockchain. If additional complementary information is required, an auxiliary database can be used to store this information, leaving the hash of the external information in the blockchain in order to guarantee its integrity.

Field	Data type	Description
Threat_id	String	Id of the threat
Description	String	Description of the threat
Priority	String	Priority of the threat
Source	String	Source of the threat, such an IP address
Responsible_parties	Array of String	People responsible for managing the threat

Table 2: Fields of the JSON structure

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

JSON format has been chosen because its simplicity and its quality as a standart format for communicating services.

2.4.1 Identity Management in the BBTR

One of the advantages in using a permissioned Blockchain is that it provides a solid Identity Management mechanism. In our case, each actor interacting with the BBTR has its own X.509 certificate with a private-public keypair. Each identity must be checked before any transaction can be accepted. Each element of the DSS has also its own identity to interact with our Blockchain, as it happens with the actors. Our BBTR is also compatible with CRLs (Certificate Revocation Lists) schemes, being possible to revoke old or compromised certificates if required. One of these user certificates, extracted from the Hyperledger Fabric documentation is the following one:

Figure 2: User Certificate in Hyperledger Fabric

Identity Management in the BBTR is managed by the MSP (Membership Service Provider) of Hyperledger Fabric. This component determines the participants of the network in Hyperledger Fabric, as well as the members who will take part in a channel. The MSP is able to assign specific roles to the different actors defining the access permissions in a network and channel. The configuration of the MSP is announced to every channel where the different organisations of the network participate. These organisations are formed by peers nodes and several of these organisations can coexist in the same channel. The different peers, orderers and users keep their own local MSP, which authenticates the messages of the differente members out of the context of a channel. In the next figure, extracted from the Hyperledger Fabric Documentation, it is possible to see the elements forming this MSP:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

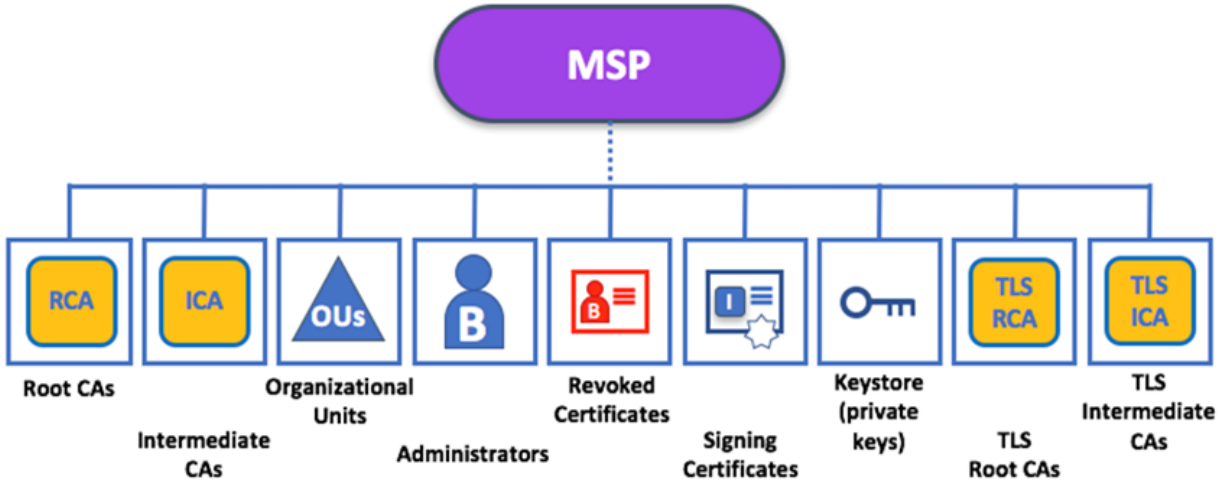

Figure 3: Elements of the MSP

By contrast, it is possible to anonymize the user in public Blockchains like Ethereum, but in our use-case we are interested in the opposite aspect, which is the identification of each user interacting with the Blockchain.

2.4.2 Interfaces design

2.4.2.1 BBTR relevance in SPHINX

The BBTR is an important component within the SPHINX Framework, because it acts as a notification and interconnection tool, which allows to transmit the intelligence, gathered by the other SPHINX tools, to other components and interested actors, so it acts as a transmitter of information. In particular, this component fits the necessity of sharing information about active threats in near-real time between the different interested parties. As a consequence, it should be well connected with the other SPHINX tools using proper interfaces, as we are about to see in the next section.

2.4.2.2 BBTR interfaces

The information in this subsection has been widely discussed in the Deliverable D2.3 in the BBTR section and it has been extracted from this deliverable, but is also important to explain it here, because one of the crucial aspects of the BBTR is its connectivity with other components in the SPHINX Project. In the next figure we can see the different components interacting with the BBTR.

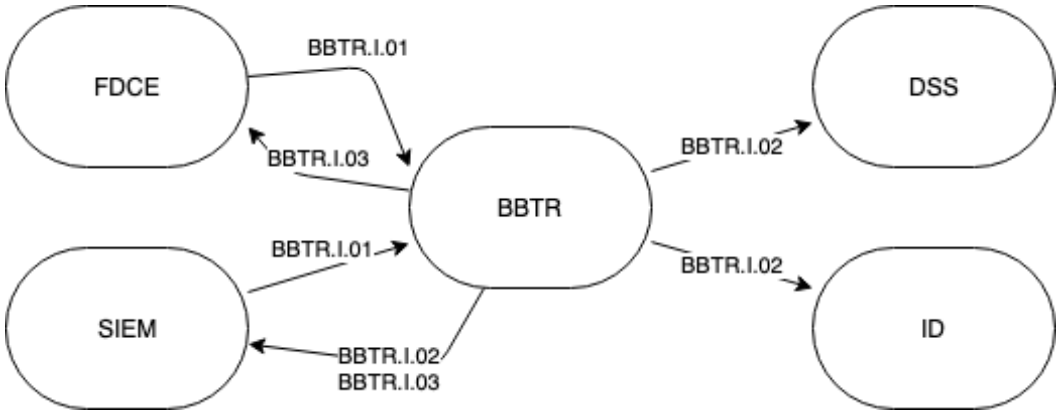

Figure 4: SPHINX BBTR Component Diagram

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Detailed Technical Specifications

Based on the VOLERE methodology as adapted by the SPHINX Action, the technical requirements/specifications for the BBTR component are as follows.

The BBTR shall provide a blockchain threat registry to register new real-time threats.	
Requirement ID	BBTR-F-010
Requirement Type	Functional Specifications
Dependencies	STA-F-330
Customer Value	5
Description and Rationale	BBTR provides secure mechanisms to build a chain of evidence, allowing information pertaining to threats to be stored and distributed across nodes. This enables all connected hospitals using the SPHINX solution to host a blockchain node and thus a synchronised threat registry.

The BBTR shall provide a blockchain threat registry allowing all nodes to be informed of new real-time threats.	
Requirement ID	BBTR-F-020
Requirement Type	Functional Specifications
Dependencies	STA-F-340
Customer Value	5
Description and Rationale	When a node is attacked, the threat registry is updated. This causes all nodes to be informed (notified) through the BBTR mechanism.

The BBTR shall allow retrieving registered threats.	
Requirement ID	BBTR-F-030
Requirement Type	Functional Specifications
Dependencies	STA-F-330; STA-F-340
Customer Value	5
Description and Rationale	BBTR allows modules to retrieve registered threats.

Table 3: BBTR Functional Specifications

Interface Specifications

The interfaces applicable to the SPHINX BBTR component are:

- **BBTR.I.01: Insert New Threat Registry Interface**
  This interface allows the BBTR to receive information regarding new threats and attack types.
  - **Input**: New attack type information and metadata;
  - **Output**: Not applicable.

  Related Interfaces: FDCE.I.04; SIEM.I.01.
- **BBTR.I.02: Threat Registry Notification Interface**
  This interface is used by the BBTR to notify SPHINX components that a new threat was registered.
  - **Input**: Not applicable;
  - **Output**: New attack type information and metadata.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

Related Interfaces: FDCE.I.01; DSS.I.05; ID.I.02.

- **BBTR.I.03: Retrieve Threat Registry Interface**
  This interface allows the BBTR to provide information concerning registered threats.
  - **Input**: Threat select criteria (e.g., selected types of threats and time scale) (or all if no criteria is provided);
  - **Output**: List of threats meeting the provided criteria.

  Related Interfaces: FDCE.I.04; SIEM.I.01.

Component Interfaces			
Interface ID	**Involved Components**	**Components Relation**	**Interface Content**
BBTR.I.01	BBTR and SIEM	SIEM generates log entries of new threats and logs them in the BBTR.	New threat registry (attack type information and metadata).
BBTR.I.01	BBTR and FDCE	FDCE identifies new threats and logs them in the BBTR.	New threat registry (attack type information and metadata).
BBTR.I.02	BBTR and FDCE	FDCE receives updates regarding new threats from BBTR.	New threat registry (attack type information and metadata).
BBTR.I.02	BBTR and DSS	As part of the decision process, DSS is notified in case new threats are identified.	New threat registry (attack type information and metadata).
BBTR.I.02	BBTR and ID	ID receives information in case a new threat is identified so that users can be notified.	New threat registry (attack type information and metadata).
BBTR.I.03	BBTR and SIEM	SIEM retrieves the list of threats from BBTR.	List of threats.
BBTR.I.03	BBTR and FDCE	FDCE retrieves the list of threats from BBTR.	List of threats.

Table 4: SPHINX BBTR Interface Specifications

Third-party APIs

The following third-party APIs will be made accessible:

- **BBTR.API.01: Insert New Threat Registry Interface**
  This interface allows the BBTR to receive information from third parties regarding new threats and attack types.
  - **Input**: New attack type information and metadata;
  - **Output**: Not applicable.

  Related Interface: BBTR.I.01.

- **BBTR.API.02: Threat Registry Notification Interface**
  This interface is used by the BBTR to notify third parties that a new threat was registered.
  - **Input**: Not applicable;
  - **Output**: New attack type information and metadata.

  Related Interface: BBTR.I.02.

- **BBTR.API.03: Retrieve Threat Registry Interface**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

This interface allows the BBTR to provide information concerning registered threats to third parties.

- **Input**: Threat select criteria (e.g., selected types of threats and time scale) (or all if no criteria is provided);
- **Output**: List of threats meeting the provided criteria.

Related Interface: BBTR.I.03.

2.4.2.3 Swagger API

The API has some methods to access to the Blockchain through HTTPS. These methods have been coded into Postman, as it will be seen in the next section, but they can be invoked directly from the terminal using *curl*, for example. Before any method calling can even take place, it is necessary to authenticate with the API service from the client. Once this process has been done, the client can start calling methods of the API. As an example, the "Register threat" method and the API answer are is shown in the next section. As the reader may notice, all of them are POST methods instead of GETs, and the answer is because it is required to use a JSON as a parameter, so it is necessary to use POST calls.

2.5 Implementation details

First of all, it is clear for previous sections that a private Blockchain technology is required for this particular use case, rather than a public one. However, it is still required to provide information about why Hyperledger Fabric has been chosen amongst other private alternatives, such as Corda o Quorum, which are the most famous ones. On the one side, Corda [52] is a private Blockchain technology but it doesn´t suit for the use case because it has been mainly used for financial applications. Another candidate technology was Quorum [53]. Quorum, which is a permissioned, private, Ethereum-based Blockchain was an initial candidate for deploying our BBTR. However, when it comes to permissions, Hyperledger Fabric provides a much fine-grain permissions, even more than Quorum and the user-access control was a crucial requisite in the SPHINX BBTR. In addition, Hyperledger Fabric is more easily compatible with existing PKI (Public Key Infrastructure) and in general, a more mature technology.

Hyperledger Fabric is a Blockchain technology which uses Go as a programming language for the Smart Contracts. It is also possible to implement these Smart Contracts in other languajes, but Go is the main programming language for Smart Contracts in Fabric. That is why Go has been the chosen language. This code is the Smart Contract which will be deployed in the network once it has been initialized and it is stored in the sphinx-cc repository as it will be explained later.

Docker has been used for deploying the different components of the Blockchain network and different bash scripts have been developed to create, start, stop, restart and clean the network. The sphinx network repository also includes the configuration and the required crypto material for the network to work properly.

For the installation process, it is necessary to download the repository where the BBTR is stored. This is: https://sphinx-repo.intracom-telecom.com/blockchain/ . In this repository, there are three sub-repositories, where the different elements of the BBTR are stored:

- BBTR network: https://sphinx-repo.intracom-telecom.com/blockchain/sphinx-network
- BBTR Smart Contract: https://sphinx-repo.intracom-telecom.com/blockchain/cc-sphinx
- BBTR API REST: https://sphinx-repo.intracom-telecom.com/blockchain/sphinx-rest

The first step is to clone these repositories using *git.* Then, in each one of these repositories there is a README.me file, where the different prerrequisites are shown. Docker is going to be used to deploy in several containers the different components of the Blockchain network. The structure of the network has been defined in the file *docker-compose.yaml,* in the *sphinx-network* repository.

First of all, it is necessary to run the network. For this task, we move to the *sphinx-network* folder and we run: *yarn generate && yarn start.* The network should start working automatically, even when the computer

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

restarts. Then, we need to log in TECNALIA artifact by doing: *docker login blockchain-docker.artifact.tecnalia.com/*

Once the network has been started, it is necessary to start the network Explorer, which can be done by running: *yarn explorer:start*

Then, it is time to install the the API Rest, which is stored in SPHINX artifact repository, so we write: *docker login registry.sphinx-repo.intracom-telecom.com/blockchain/sphinx-rest/fabric-rest-api* and then we run *yarn rest: start*

Another important component is the SDK of Hyperledger Fabric. To install it, we must do it with npm, by running: *npm install @fabric/blockchain-sdk-fabric.* However, before that, it is required to log in the artifactory where this library is located, which is: *https://artifact.tecnalia.com/artifactory/api/npm/sphinx-npm/* with a valid user. Then, it is required to include the following text in the file *.npmrc* inside the user directory:

```
@fabric:registry=https://artifact.tecnalia.com/artifactory/api/npm/sphinx-npm/
//artifact.tecnalia.com/artifactory/api/npm/sphinx-npm/:_password=<BASE64_PASSWORD>
//artifact.tecnalia.com/artifactory/api/npm/sphinx-npm/:username=<USERNAME>
//artifact.tecnalia.com/artifactory/api/npm/sphinx-npm/:email=youremail@email.com
//artifact.tecnalia.com/artifactory/api/npm/sphinx-npm/:always-auth=true
```

Now, that the network is running with no errors, it is time to instantiate the Smart Contract in the peer nodes of the network. All the stuff related to this aspect can be located in the repository *sphinx-cc*. Some scripts have been developed to make this process easier and faster. To do that, we must run: *./install.sh && ./instantiate.sh*. If there are no errors, then the Smart Contract will be correctly installed and instantiated in the peers of the Blockchain. If the reader wants to learn the logic behind these scripts, they can be found in the repository.

It is recommended to use Postman to work with the API REST due to its simplicity. As a consequence, Postman must be installed as a prerequisite in the system. Once it has been installed, we must import the collection into Postman. The collection file is available in the *sphinx-cc* repository, under the name of: *SPHINX API REST.postman_collection.json.* This collection includes the methods for:

- Authenticating in the Blockchain
- Getting a threat by any field (normally threat_id)
- Getting threats by description (including regular expressions)
- Storing a threat in the Blockchain

Postman has been used because of its simplicity. However, other software can be used if it supports working with HTTPS APIs. Even a *curl* call using the Unix terminal is possible to be used. In the next figure we can see the Postman interface, where the authentication process is taking place. The SignIn method is called and a token is issued by the BBTR to perform further operations against the platform.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

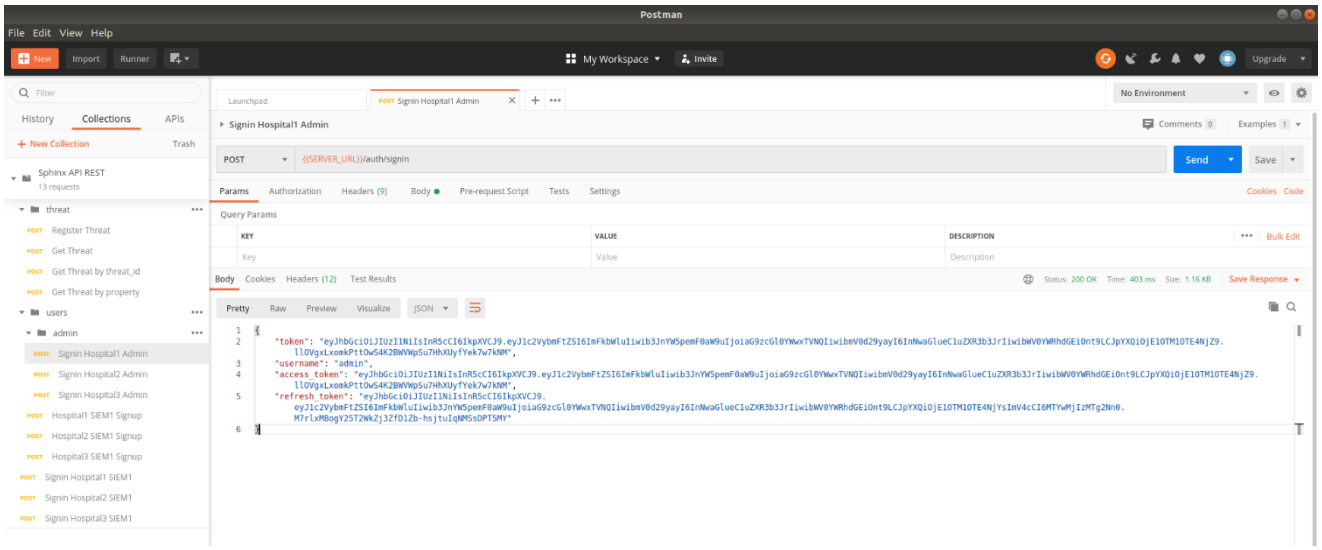

Figure 5: Authenticating against the BBTR

Once the authentication has been successfully made, it is possible to interact with the BBTR, like for example to submit a new threat, as can be seen in the next figure.

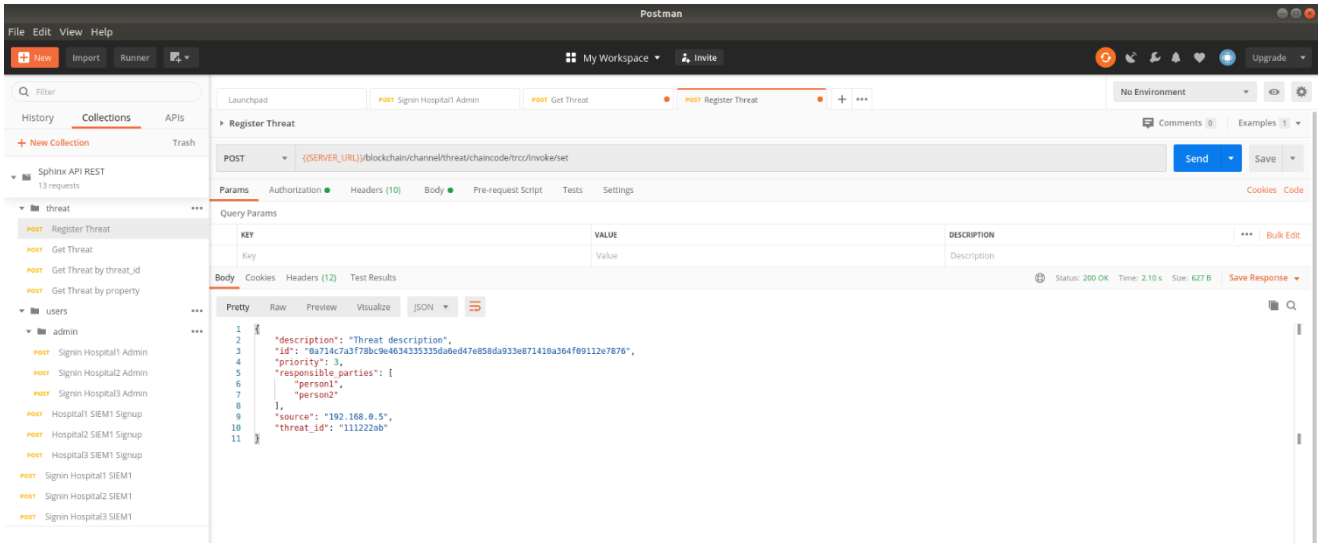

Figure 6: Submitting a new threat to the BBTR

This method returns a 200 code if the process was successful. When issueing a threat, the format is in JSON file. As an example, a threat is presented below in JSON format:

```
{
 "threat_id": "aabb112233",
 "description": "description",
 "priority"": "3",
 "source"": "source_of_the_threat",
 "responsible_parties": ["person1","person2"]
}
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

The same can be done to retrieve a group of threats which match with a query. In the next figure, we can see the “get by description” method as an example, which allows us to get a subset of threats that match with the regular expression put in the description field in the query.

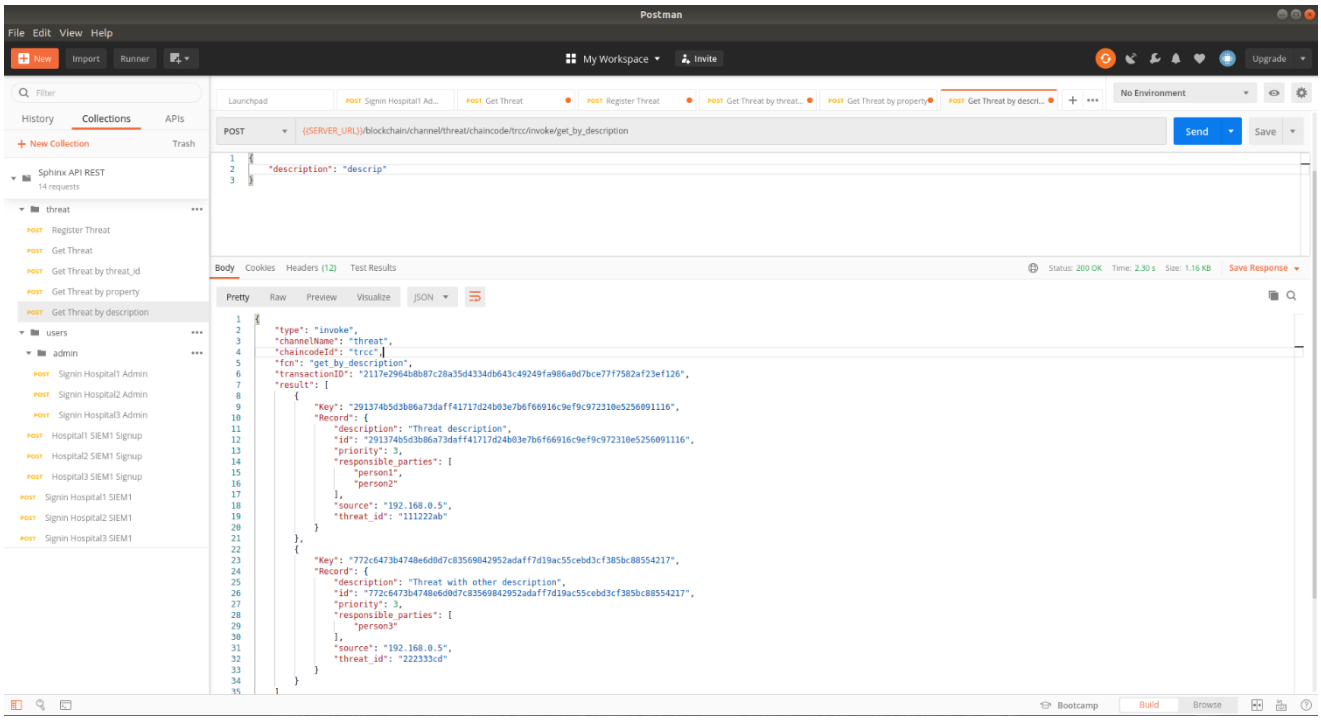

Figure 7: Querying a group of threats matching a description

2.5.1 Test cases

In order to test the correct operation of the BBTR, several test cases have been defined during the installation process. These test cases will allow us to know if everything worked as expected during the installation and setup process. The table with this information can be found in the Deliverable D7.1, so it won´t be mentioned here.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

3 Use-case validation: Sharing Threat Knowledge Between Healthcare Providers

Healthcare institutions are a living process of modernisation involving digitalisation of most of their processes and services, relying on information and communications technologies. Moreover, in order to improve the quality of healthcare delivery while at the same time reducing costs, healthcare institutions are incorporating connected devices providing medical diagnoses, whose results go directly to the doctor's desktop or even to remote locations.

Unfortunately, and similarly to other sectors, the digitalisation and networking process in healthcare institutions is subject to being exploited by cyberattacks that aim at disrupting their operations. Cyberattacks in healthcare may affect equipment (damaging or even rendering them inoperational), personal records (stealing, obfuscating, altering or deleting data) or services, thus might seriously impact patients' health and wellbeing, besides causing substantial financial damage.

Healthcare institutions share common types of vulnerabilities and exposures to cyberattacks, because they operate similar equipment and software across institutions. In addition, the occurrence of a "zero-day attack" (i.e., an attack that occurs at the same time a vulnerability is found, and no fix is yet available) defies even the best protected institution, thus urging the provision of a mechanism that can register and distribute information concerning incidents and threats.

The BBTR will be validated with some of the use cases defined in the Deliverable D2.4, where all the use cases of the SPHINX Project are included. Particularly, the use cases which will validate the BBTR are:

- UC3: Rootkit Malware Attack in a Cancer Treatment Institute
- UC4: Theft of Health Data by Exploiting Vulnerable Software
- UC5: Tampering with Medical Devices

In all of these use cases, the BBTR acts as a post-incident tool, whose objective is to share information about a materialized threat with other medical centres, so they can prevent their systems before anything bad happens. This notification process is the last step in the SPHINX tools process for protecting medical equipment. Further information about how the BBTR empowers each one of these use cases can be found in the Deliverable D2.4.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

4 References

[1] OECD. Health spending. 2018. doi: https://doi.org/https://doi.org/10.1787/8643de7e-en. URL https://www.oecd-ilibrary.org/content/data/8643de7e- en.

[2] Elizabeth A Bell, Lucila Ohno-Machado, and M Adela Grando. Sharing my health data: a survey of data sharing preferences of healthy individuals. In AMIA Annual Symposium Proceedings, volume 2014, page 1699. American Medical Informatics Association, 2014.

[3] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[4] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In Big Data (BigData Congress), 2017 IEEE International Congress on, pages 557–564. IEEE, 2017.

[5] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. Journal of Cryptology, 7(1): 1–32, 1994.

[6] Ariel Ekblaw and Asaf Azaria. MedRec: Medical Data Management on the Blockchain. Viral Communications, 12 2017. URL https://viral.media.mit.edu/pub/medrec.

[7] Chrissa Mcfarlane, Michael Beer, Jesse Brown, and Nelson Prendergast. Patientory - Whitepaper. (May):1–19, 2017.

[8] The Next Web. Philips will challenge tech giants to bring blockchain to healthcare, 2018 (accessed October 18, 2018). https://thenextweb.com/blockchain/2018/10/17/philips-solve-healthcare-data-breaches-with- blockchain/.

[9] Qi Xia, Emmanuel Boateng Sifah, Kwame Omono Asamoah, Jianbin Gao, Xiaojiang Du, and Mohsen Guizani. MeDShare: Trust-Less Medical Data Sharing among Cloud Service Providers via Blockchain. IEEE Access, 5:14757–14767, 2017. ISSN 21693536. doi: 10.1109/ACCESS.2017.2730843.

[10] IRYO Network Technical Whitepaper, 2018(accessed October 18, 2018). https://iryo.io/iryo_whitepaper.pdf.

[11] openEHR, 2018(accessed November 6, 2018). www.openehr.org.

[12] The Clinical and Investment Potential in the Gene-Chain Project The Unprecedented Growth of Genomic Data, 2018 (accessed October 10, 2018). https://icotimeline.com/wp-content/uploads/2017/07/Gene-Chain-Whitepaper.pdf.

[13] Dennis Grishin, Kamal Obbad, Preston Estep, Mirza Cifric, Yining Zhao, and George Church. Nebula Genomics: Blockchain- enabled genomic data sharing and analysis platform, 2018(accessed November 6, 2018). https://www.nebula.org/ assets/Nebula_Genomics_Whitepaper.pdf.

[14] LunaDNA, 2018 (accessed Nov 5, 2018). https://lunadna.com/.

[15] Ariel Ekblaw and Asaf Azaria. MedRec: Medical Data Management on the Blockchain. Viral Communications, 12 2017. URL https://viral.media.mit.edu/pub/medrec.

[16] Chrissa Mcfarlane, Michael Beer, Jesse Brown, and Nelson Prendergast. Patientory - Whitepaper. (May):1–19, 2017.

[17] The Next Web. Philips will challenge tech giants to bring blockchain to healthcare, 2018 (accessed October 18, 2018). https://thenextweb.com/blockchain/2018/10/17/philips-solve-healthcare-data-breaches-with- blockchain/.

[18] Gem Health, 2018(accessed November 6, 2018). https://enterprise.gem.co/health/.

[19] Qi Xia, Emmanuel Boateng Sifah, Kwame Omono Asamoah, Jianbin Gao, Xiaojiang Du, and Mohsen Guizani. MeDShare: Trust-Less Medical Data Sharing among Cloud Service Providers via Blockchain. IEEE Access, 5:14757–14767, 2017. ISSN 21693536. doi: 10.1109/ACCESS.2017.2730843.

[20] IRYO Network Technical Whitepaper, 2018(accessed October 18, 2018). https://iryo.io/iryo_whitepaper.pdf.

[21] Peng Zhang, Jules White, Douglas C Schmidt, Gunther Lenz, and S Trent Rosenbloom. FHIRChain : Applying Blockchain to Securely and Scalably Share Clinical Data. Computational and Structural Biotechnology Journal, 16:267–278, 2018. ISSN 2001-0370. doi: 10.1016/j.csbj.2018.07.004. URL https://doi.org/10.1016/j.csbj.2018.07.004.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

[22] Alex Roehrs, Cristiano André, and Rosa Righi. OmniPHR : A distributed architecture model to integrate personal health records. Journal of Biomedical Informatics, 71:70–81, 2017. ISSN 1532-0464. doi: 10.1016/j.jbi.2017.05.012. URL http://dx.doi.org/10.1016/j.jbi.2017.05.012.

[23] Medicalchain Whitepaper 2.1, 2018(accessed November 6, 2018). Whitepaper-EN.pdf.

[24] Doc.ai, 2018(accessed November 6, 2018). https://doc.ai/.

[25] Hearthy Foundation, 2018(accessed November 6, 2018). http://hearthy.co/assets/images/Hearthy-whitepaper.pdf

[26] Encrypgen, 2018(accessed Nov 5, 2018). https://encrypgen.com/.

[27] Dennis Grishin, Kamal Obbad, Preston Estep, Mirza Cifric, Yining Zhao, and George Church. Nebula Genomics: Blockchain- enabled genomic data sharing and analysis platform, 2018(accessed November 6, 2018). https://www.nebula.org/ assets/Nebula_Genomics_Whitepaper.pdf.

[28] Nikolay Kulemin, Sergey Popov, and Alexey Gorbachev. The Zenome Project: Whitepaper blockchain-based genomic ecosystem, 2018(accessed Nov 5, 2018). https://zenome.io/download/whitepaper.pdf.

[29] Mark Hahnel. The Genomes.io Lightpaper- Blockchain enabled genome security from the moment it is sequenced, 2018(accessed Nov 5, 2018). https://genomes.io/assets/uploads/Genomes-whitepaper.pdf.

[30] Shivom, 2018(accessed Nov 5, 2018). https://shivom.io//.

[31] Davi R Ortega, Catherine M. Oikonomou, H. Jane Ding, Prudence Rees-Lee, Alexandria, and Grant J Jensen. ETDB-Caltech: a blockchain-based distributed public database for electron tomography, 2018(accessed November 6, 2018). https://www.biorxiv.org/content/early/2018/10/25/453662.

[32] Vishal Patel. A framework for secure and decentralized sharing of medical imaging data via blockchain consensus. Health Informatics Journal, 0(0):1460458218769699, 2018. doi: 10.1177/1460458218769699. URL https://doi.org/10.1177/ 1460458218769699. PMID: 29692204.

[33] OPU Labs. OPU whitepaper 2018, 2018(accessed November 7, 2018). https://ico.opu.ai/wp-content/uploads/ 2018/04/Opucoin_Whitepaper_v2.0.pdf?x38818.

[34] MedX Protocol. 2018(accessed November 7, 2018). https://medxprotocol.com.

[35] A Blockchain Approach Applied to a Teledermatology Platform in the Sardinian Region(Italy), 2018 (accessed November 7, 2018). http://www.mdpi.com/2078-2489/9/2/44/pdf.

[36] Akiri Switch, 2018(accessed November 6, 2018). https://akiri.com.

[37] J. Höller, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand, andD. Boyle,From Machine-to-Machine to the Internet of Things: Introduc-tion to a New Age of Intelligence. Amsterdam, The Netherlands: Elsevier,2014

[38] Z. Pang, ‘‘Technologies and architectures of the Internet-of-Things (IoT)for health and well-being,’’ M.S. thesis, Dept. Electron. Comput. Syst.,KTH-Roy. Inst. Technol., Stockholm, Sweden, Jan. 2013

[39] K. Vasanth and J. Sbert. Creating solutions for health throughtechnology innovation. Texas Instruments. [Online]. Available:http://www.ti.com/lit/wp/sszy006/sszy006.pdf, accessed Dec. 7, 2014

[40] Lu, Y., & Da Xu, L. (2018). Internet of Things (IoT) cybersecurity research: a review of current research topics. IEEE Internet of Things Journal, 6(2), 2103-2115

[41] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain and K. Kwak, "The Internet of Things for Health Care: A Comprehensive Survey," in IEEE Access, vol. 3, pp. 678-708, 2015.

doi: 10.1109/ACCESS.2015.2437951

[42] Xiao Yue, Huiju Wang, Dawei Jin, Mingqiang Li, and Wei Jiang. 2016. HealthcareData Gateways: Found Healthcare Intelligence on Blockchain with Novel PrivacyRisk Control.Journal of medical systems40, 10 (2016), 218.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.

[43] Ali Dorri, Salil S. Kanhere, and Raja Jurdak. 2017. Towards an Optimized BlockChain for IoT. In Proceedings of the Second International Conference on Internet-of-Things Design and Implementation (IoTDI '17). ACM, New York, NY, USA, 173-178. DOI: https://doi.org/10.1145/3054977.3055003

[44] Griggs, K.N., Ossipova, O., Kohlios, C.P. et al. J Med Syst (2018) 42: 130. https://doi.org/10.1007/s10916-018-0982-x

[45] Y. Rahulamathavan, R. C. -. Phan, M. Rajarajan, S. Misra and A. Kondoz, "Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption," 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Bhubaneswar, 2017, pp. 1-6. doi: 10.1109/ANTS.2017.8384164

[46] Shahare R. "Blockchain, for Threat Intelligence Maybe?" [Internet]. 2019. Available from: https://www.cpomagazine.com/cyber-security/blockchain-for-threat-intelligencemaybe/

[47] Wagner, Cynthia & Dulaunoy, Alexandre & Wagener, Gérard & Iklody, Andras. (2016). MISP -The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform. 10.1145/2994539.2994542.

[48] Burger EW, Goodman MD, Kampanakis P, Zhu KA. Taxonomy Model for Cyber Threat Intelligence Information Exchange Technologies. 2014. pp. 51-60. DOI:10.1145/2663876.2663883

[49] Peterson, K.J., Deeduvanu, R., Kanjamala, P., & Mayo, K.B. (2016). A Blockchain-Based Approach to Health Information Exchange Networks.

[50] Salah, Khaled & H Rehman, M & Nizamuddin, Nishara & Al-Fuqaha, Ala. (2018). Blockchain for AI: Review and Open Research Challenges. IEEE Access.

[51] Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper, 151(2014), 1-32.

[52] Hearn, M. (2016). Corda: A distributed ledger. Corda Technical White Paper, 2016.

[53] Morgan, J. P. (2016). Quorum whitepaper. New York: JP Morgan Chase.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826183 - Digital Society, Trust & Cyber Security E-Health, Well-being and Ageing.