Tailored IoT & BigData Sandboxes and Testbeds for Smart, Autonomous and Personalized Services in the European Finance and Insurance Services Ecosystem

Infinitech

D4.7 – Permissioned Blockchain for Finance and Insurance- I

Lead Beneficiary	UBI
Due Date	2020-08-31
Delivered Date	2020-09-25
Revision Number	3.00
Dissemination Level	Public (PU)
Type	Report (R)
Document Status	Release
Review Status	Internally Reviewed and Quality Assurance Reviewed
Document Acceptance	WP Leader Accepted and Coordinator Accepted
EC Project Officer	Pierre-Paul Sondag

HORIZON 2020 - ICT-11-2018

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 856632

Contributing Partners

Partner Acronym	Role[1]	Name Surname[2]
UBI	Lead Beneficiary	Konstantinos Perakis, Dimitris Miltiadou
INNOV	Contributor	John Soldatos, Nikos Kapsoulis, Antonis Litke
IBM	Contributor	Fabiana Fournier, Inna Skarbovsky
AGRO	Internal Reviewer	Gregory Mygdakos, Stelios Kotsopoulos
GFT	Internal Reviewer	Maurizio Megliola
INNOV	Quality Assurance	Filia Filippou

Revision History

Version	Date	Partner(s)	Description
0.1	2020-07-28	UBI	ToC Version
0.2	2020-08-14	UBI	Initial contributions on Section 2, 4 and 5
0.21	2020-08-20	IBM	Contributions on Section 2, 3, 4 and 5
0.22	2020-08-21	INNOV	Contributions on Section 2, 4 and 5
0.30	2020-08-28	UBI, IBM, INNOV	Updated contributions on Section 2, 4, 5 and 6
0.40	2020-09-02	UBI, IBM, INNOV	Updated contributions on Section 2, 4, 5 and 6
0.45	2020-09-07	IBM	Update contribution on Section 3
0.50	2020-09-11	UBI, IBM, INNOV	Updated contributions on Section 2, 4, 5 and 6
0.60	2020-09-15	UBI	Finalisation of section 2, 3, 4, 5 and 6
1.0	2020-09-16	UBI	First Version for Internal Review
1.1	2020-09-22	AgroApps	Internal Review
1.2	2020-09-22	GFT	Internal Review
2.0	2020-09-23	UBI	Version for Quality Assurance
3.0	2020-09-25	UBI	Version for Submission

[1] Lead Beneficiary, Contributor, Internal Reviewer, Quality Assurance

[2] Can be left void

Executive Summary

The document at hand, entitled "D4.7 - "Permissioned Blockchain for Finance and Insurance - I", constitutes a report of the preliminary efforts and the produced outcomes of Task T4.3 "Distributed Ledger Technologies for Decentralized Data Sharing" of WP4. The purpose of this deliverable is a) to document the analysis of the blockchain technology and define of its role within the INFINITECH RA, b) to document the design of a new blockchain capability that can be leveraged within the context of the project, c) to deliver the design specifications of the blockchain applications which are tailored to the needs of the financial and insurance sector as well as the specifications of the INFINITECH blockchain network that will be utilised and finally d) to document the baseline technologies that will be utilised. Hence, the scope of the current report can be described in the following axes:

- To perform the comprehensive analysis of the key characteristics and offerings of the blockchain technology in order to define the role of the blockchain technology within the INFINITECH RA. In this analysis, the key characteristics are presented along with the main components of the blockchain technology. Furthermore, the different implementations that are based on the different approaches of blockchain technology followed, are presented. The role of the blockchain technology within the INFINITECH RA is defined taking into consideration the results of the analysis in conjunction with the requirements of the financial and insurance sectors.
- To document the approach and the design specifications of the new capability of the blockchain platform that will be introduced within the context of the project. The proposed new capability will be offered as a functionality "horizontally" at the level of the blockchain platform and will enable the blockchain application that can be developed to effectively address the requirements imposed by the EU GDPR. The high-level architecture and the design specifications of the proposed solution are elaborated.
- To present the initial design specifications of the blockchain applications that will be implemented within the context of the INFINITECH project. In total, two applications are presented, namely the Consent Management and the Know-Your-Customer (KYC) / Know-Your-Business (KYB). Each application is presented by documenting the business operation that it addresses and by presenting the key functionalities of the blockchain technology that are exploited for this purpose. Furthermore, the high-level architecture of each blockchain application is presented accompanied by the detailed use cases that the designed application addresses. Finally, the sequence diagrams that depict the interactions between the stakeholders and the involved components are documented.
- To document the design details of the INFINITECH blockchain network. Taking into consideration the role of the blockchain technology within the INFINITECH RA and the design specifications of the blockchain applications, the network topology is presented, defining the role of each node, the services hosted on each node and the interactions between the nodes. The INFINITECH blockchain network is formulated by eight nodes in total, four of them constituting the peer nodes of the blockchain network and the rest of them being the hosts of the external applications that interact with the blockchain network and the certificate authorities. The blockchain network is composed by three organizations interacting via four different channels, while three different smart contracts (chaincode) are deployed for the four different ledgers that are hosted by the four peers.
- To document the list of baseline technologies and tools that will be leveraged in the implementation phase of both the INFINITECH blockchain network and the designed blockchain applications. The list is composed by dominant open-source software, libraries and frameworks that have a high level of maturity, relevance to the design specifications and compatibility between them.

The current deliverable presents the first version of the INFINITECH blockchain network specifications and the design specifications of the blockchain applications. The outcomes of this deliverable will drive the implementation activities of Task 4.3. Nevertheless, the definition of the design specifications of both the blockchain network and the blockchain applications is a living process that will last until M27 as per the INFINITECH Description of Action. In this sense, the deliverable D4.7 constitutes a living document, and it will be continuously updated based on the analysis of the feedback that will be collected from the pilots of the project and the stakeholders of the platform. The derived optimisations and enhancements will be documented in the upcoming versions of the deliverable.

Table of Contents

List of Figures

List of Tables

Abbreviations

AES	Advanced Encryption Standard
AI	Artificial Intelligence
AML	Anti-Money Laundering
API	Application Programming Interface
BFT	Byzantine Fault-Tolerant
CA	Certificate Authority
CFT	Crash Fault-Tolerant
CRUD	Create Read Update Delete
DLT	Distributed Ledger Technology
DoA	Description of Action
GDPR	General Data Protection Regulation
IoT	Internet of Things
KYB	Know-Your-Business
KYC	Know-Your-Customer
M	Month
MSP	Membership Services Provider
RDBMS	Relational Database Management System
RA	Reference Architecture
UUID	Universally Unique Identifier

1 Introduction

The scope of deliverable D4.7 "Permissioned Blockchain for Finance and Insurance - I" is to document the preliminary efforts undertaken within the context of T4.3 "Distributed Ledger Technologies for Decentralized Data Sharing" of WP4. The deliverable D4.7 is prepared in accordance with the INFINITECH Description of Action and constitutes the first iteration of the work performed under Task 4.3. It provides the initial specifications of the permissioned blockchain infrastructure that will be exploited in the INFINITECH platform, as well as the initial design specifications of the blockchain applications that will be developed on top of this infrastructure within the context of the project, aiming to support the requirements and effectively address the needs of the financial sector. Finally, the deliverable provides the design details of a proposed new blockchain capability which will extend the list of the existing blockchain capabilities and will be introduced within the context of the project.

The blockchain technology constitutes one of the main ingredients of the INFINITECH platform and within the context of the INFINITECH project, the tremendous potential of the blockchain technology will be leveraged towards the execution of blockchain empowered scenarios for the financial and insurance sectors. Through the exploitation of the blockchain technology, the financial institutions will gain access to innovative solutions that will optimise and enhance their current core operational services and processes with significant cost reductions and increased performance, while at the same time, will be empowered to deliver new services, products and offerings to their clients. To this end, within the context of Task 4.3 the consortium has analysed the key characteristics of the blockchain technology in order to formulate the suitable blockchain infrastructure, that will be integrated and implemented as part of the INFINITECH RA and will be leveraged towards the design and implementation of blockchain-enabled solutions that will be exploited by the INFINITECH project's pilots, as well as the stakeholders of the financial sector. In addition to this, the consortium recognised the need for an additional capability, related to the requirements imposed by the EU General Data Protection Regulation (GDPR), that will be introduced in the blockchain from which the aspired INFINITECH scenarios can benefit.

1.1. Objective of the Deliverable

The purpose of this deliverable is to report the outcomes of the work performed within the context of Task 4.3 at this phase of the project (M12). In this first iteration, the work that has been performed was mainly focused on the analysis of the blockchain technology and the definition of its role within the INFINITECH RA, the design of a new blockchain capability that can be leveraged within the context of the project, the design of blockchain applications tailored to the needs of the financial and insurance sectors, the design of the suitable blockchain infrastructure and finally the definition of the baseline technologies that will be utilised.

Hence, the main objectives of the deliverable at hand are as follows:

- **To perform a thorough analysis of the key characteristics and offerings of the blockchain technology.** In this analysis, the key characteristics of the blockchain technology are documented and the core parts of the technology are presented, highlighting their role and scope in the blockchain technology. The different approaches of the blockchain technology's implementations are also documented. Furthermore, the role of the blockchain technology within the INFINITECH RA is presented taking into consideration the results of the performed analysis, the differences in the approaches adopted in the various available implementations and the requirements of the financial and insurance sectors.
- **To present a new proposed capability of the blockchain that the scenarios of the project can leverage.** The new capability will be offered as a functionality "horizontally" at the level of the

blockchain platform and any application that requires the new capability will be able to utilise it and build on top of it. The high-level architecture as well as the design details of the proposed capability are presented in detail.

- **To document the initial design specifications of the blockchain applications that will be implemented.** For each application, a thorough description of the addressed business case is presented, the main characteristics of the blockchain technology which are leveraged are highlighted and the high-level architecture of the application is presented. Furthermore, the detailed use cases that each application addresses are documented along with the respective sequence diagrams that illustrate the interactions between the involved stakeholders and the components.
- **To present the details of the blockchain network that will be utilised within the context of INFINITECH project and on which the designed blockchain applications will be deployed.** The main components of the blockchain network are documented and the formulated network topology is presented. Additionally, the role of each node in the network is documented and the interactions between the nodes are described. Finally, the services and hardware resources that are required for each node are documented.
- **To document the baseline technologies and tools that will be used in the implementation phase of the described blockchain network and applications.** The list is composed by a set of well-established open-source software, libraries and frameworks suitable for the needs of the designed network and applications.

It should be noted that according to the INFINITECH Description of Action Task 4.3 lasts until M27, and therefore, two more versions of the deliverable will be released on M20 and M27 with deliverables D4.8 and D4.9. Hence, the upcoming iterations of the deliverable at hand will extend the content of this document with the necessary updates and optimisations, where needed, taking into consideration the evolvement of the project and the blockchain technology, the implementation details of the proposed new functionality, the presented blockchain applications and blockchain network, whose implementation will be driven by the presented design specifications.

1.2. Insights from other Tasks and Deliverables

The deliverable D4.7 is released in the scope of WP4 “Interoperable Data Exchange and Semantic Interoperability” activities and documents the preliminary outcomes of the work performed within the context of T4.3 “Distributed Ledger Technologies for Decentralized Data Sharing”. The task is tightly interconnected with the outcomes of WP2 “Vision and Specifications for Autonomous, Intelligent and Personalized Services” in which the overall requirements of the INFINITECH platform are defined. In detail, the outcomes of Task 2.1 that as presented in deliverable D2.1 that reported the collected user stories of pilots of the project and the extracted user requirements, are provided as input in T4.3. Furthermore, the specification of the technologies that constitute the fundamental building blocks of the INFINITECH platform and the elicited technical requirements that are linked to these building blocks, as reported by the outcomes of T2.3 in deliverable D2.5, are also provided as input in T4.3. Last but not least, the outcomes of T2.7 that formulated the INFINITECH Reference Architecture (INFINITECH RA) and that serve as the blueprint for the development, deployment and operation of Big Data, AI and IoT in the finance and insurance sectors are directly related with the work performed in this task, as the reported outcomes of this report related to the blockchain network and the blockchain applications are integral parts of the INFINITECH platform. Finally, the work reported in this deliverable is tightly connected with the work performed in T4.4 and T4.5 of WP4 as the output of Task 4.3 and especially the designed blockchain network serves as the basis in the activities of both T4.4 and T4.5.

1.3. Structure

This document is structured as follows: Section **Error! Reference source not found.** introduces the document, describing the context of the outcomes of the work performed within the task and highlights its relation to the rest of tasks of the project and deliverables of the project. Section 2 provides an analysis of the blockchain technology, highlighting the key characteristics and main components of the technology, and presents the role of the blockchain technology in the INFINITECH RA. Section 3 presents the details of the proposed new capability of the blockchain platform which will further extend the capabilities of the blockchain platform. Section 4 presents the designed specifications of the blockchain applications, the use cases addressed by each application and the corresponding sequence diagrams of each use case. Section 5 documents the details of the blockchain network that will be leveraged, by presenting the network topology along with the services of each node and their interactions. Section 6 presents the list of baseline technologies and tools that will be utilised in the implementation of the described network and applications. Finally, section 7 concludes the document.

2 The Blockchain technology

2.1 An overview of the blockchain technology

Blockchain is a distributed digital ledger of cryptographically signed transactions that are grouped into blocks, which in turn are cryptographically linked to each other after validation and undergoing a consensus decision. The addition of new blocks increases the tamper resistance of the older ones and are replicated across the copies of the ledger within the network, resolving automatically any possible conflicts through a set of established rules [1]. In this sense, blockchain is a continuously growing, distributed, shared ledger of uniquely identified, linked transaction records organised in blocks that are sealed cryptographically with a digital fingerprint generated by a hashing function and are sequentially chained through a reference to their hash value [2]. Blockchain technology became extremely popular as the basis of cryptocurrencies and its implementations such as Bitcoin and Ethereum, which are digital currencies that were designed to work as medium of exchange incorporating secure and verifiable cryptographically signed transactions and controlled cryptocurrency unit generation.

In general, the blockchain technology is composed by multiple technologies related to cryptography, peer-to-peer networks, identity management, network security, transaction processing, (distributed) algorithms and more, that are all leveraged in order to formulate an immutable transaction ledger which is maintained by a distributed network of peer nodes formulating the blockchain network. The key characteristics of the blockchain technology can be grouped as follows [3]:

- *Decentralised:* One of the core characteristics of the blockchain is its decentralised and distributed nature across multiple number of nodes (peers) that provides extensibility, scalability, confidentiality, flexibility and resilience to attacks or misuse.
- *Immutable:* Any transaction record is immutable and reserved forever. Hence, full transactional history is maintained and all records are cryptographically secure. This fact safeguards that the underlying data cannot be tampered and are attestable.
- *Transparent:* The transaction data that formulate the blocks are transparent to each node and each node can introduce an update, based on a set of a rules, increasing the transparency and trustworthiness of the technology.
- *Autonomy*: One of the core characteristics of the blockchain is the autonomy offered within the peer network that is regulated by the consensus protocols, where each node can safely and securely transfer and update data. Since the ledger (or actually a copy of the ledger) is shared among multiple nodes (peers), the transparency and trustworthiness is also increased.
- *Open Source*: The blockchain technology is an open source technology with multiple blockchain implementations and variations being available; sustained by various communities and ecosystems, that can be leveraged upon needs.

At a high level, the blockchain technology exploits well-established computer network mechanisms and cryptographic primitives such as cryptographic hash functions, digital signatures, asymmetric-key cryptography, certificate authority mixed with record keeping concepts (such as append only ledgers) [1]. Nevertheless, the blockchain technology has a set of key concepts that includes the distributed ledger that is composed by blocks containing the transaction records, the smart contracts or chaincode and the consensus model.

The heart of the blockchain is the distributed ledger in which all transaction records that are published within the blockchain network are stored in the form of blocks. Being by nature decentralized, the technology takes advantage of both the distributed ownership and the distributed physical

architecture of distributed ledger. Each peer maintains its own copy of the ledger, ensuring that is synced and updated with the same data. As the blockchain network is designed and is operating in a peer-to-peer mode, the blockchain network has an increased resilience to the loss of any node. Every new transaction is checked and verified among all peers before it is accepted and inserted into the ledger and it is referenced to the previous block, enabling an integrity check of invalid transmitted transactions or nodes. Finally, with the utilisation of the cryptographic mechanisms the distributed ledger is tamper evident and tamper resistant.

The nodes that are participating to the blockchain network can be characterised as publishing and non-publishing nodes. The candidate transactions are submitted to the blockchain network via its user through the interacting applications and services. However, it is the role of the publishing node(s) only to publish a block in the blockchain network that contains these transactions via the gossip data dissemination protocol, once they have been validated and authenticated, that will be received by the rest of the (publishing and non-publishing) nodes which in turn will validate and authenticate the received block and accept it in order to be inserted finally in the ledger.

To facilitate all the operations performed in the ledger within the blockchain network, the smart contracts (or chaincode) are leveraged. Smart contracts are the trusted distributed applications that are deployed within the nodes of the blockchain network and encapsulate the business logic of the blockchain applications. Smart contracts include the agreements that the participants of the blockchain network have formulated with regards to the generation of new facts that are added to the ledger and that will update the current and historical state of the facts that are already stored in the ledger. In this sense, the smart contracts enable the creation of new transactions by the users of the blockchain network by invoking the smart contracts' functions. The smart contracts are facilitating the controlled access to the ledger, offering a layer of abstraction on top of the aspired transactions, encapsulating and simplifying all the relevant information while also ensuring their compliance with the underlying legal agreements, as well as the automation of the several aspects of the transactions. The implementation and execution of smart contracts varies depending on the blockchain implementation with most popular cases being the Ethereum's smart contracts and Hyperledger Fabric's chaincode.

One of the most critical concepts of blockchain technology is the consensus model that is utilised in order to validate a transaction and to keep the ledger transactions synchronized across the blockchain network. Hence, the consensus model undertakes the validation and approval of the candidate transactions and ensures that the copies of the ledger that are kept within the nodes of the blockchain network are updated with the same transactions and in the same order. As the blockchain network is composed by multiple nodes, it is very likely that many publishing nodes will compete at the same time to publish new nodes. Additionally, conflicts might be created by nodes publishing new block at approximately the same time. Hence, it is evident that a method is required to ensure that transactions will be written to the ledger at the same order as they generated, as well as that malformed or malicious transactions are rejected. For this reason, the blockchains depending on their implementation specifications exploit different consensus models that are available in computer science such as the CFT (crash fault-tolerant) or BFT (byzantine fault-tolerant) ordering, while at the same time large research effort is spent on this topic towards the definition of further alternative consensus models capable of better addressing this issue with less trade-offs.

The blockchain implementations can be characterised and grouped into two major high-level categories based on the permission model applied on the blockchain network, the *permissionless blockchain networks* and the *permissioned blockchain networks*. Permissionless blockchains are based

on open and public blockchain networks where anyone is capable of publishing new blocks or read the blocks of the blockchain anonymously and without grading any permission from any authority. Hence, the implementation of the permissionless blockchains dictates that they are open and available to anyone and anyone can issue new transactions in new blocks and read the transactions included in the existing blocks, thus write and read the ledger. To prevent the malicious usage of the blockchain, these implementations employ a consensus model that requires from the participants to expend or maintain resources through a "mined" native cryptocurrency or through transaction fees when it comes to publishing new blocks. The most common consensus models employed are the "proof of work" or "proof of stake" that are rewarding the participants of new blocks that conform the consensus protocol with a native cryptocurrency. The well-established examples of permissionless blockchains are Bitcoin and Ethereum.

On the other hand, the permissioned blockchain networks are regulated blockchain networks where only authorised users, by a specific authority as defined within the network specifics, are able to maintain the underlying blockchain, while read access and publishing of new transactions are also restricted and regulated. In this sense, the permissioned blockchain networks are formulated only by a set of known, identified and verified participants whose access rights and roles are regulated by an agreed governance model defined by the participants of the networks providing a certain degree of trust and security for all generated transaction records. As the identities of the participants of the network are known and trusted, the consensus models that are employed for publishing new blocks do not require the expense or maintenance of resources. In permissioned blockchain networks the consensus models exploited are usually faster and less computationally expensive, as the mining operations are eliminated and more traditional consensus protocols are adopted, such as the CFT or BFT protocols. Permissioned blockchain networks allow the tight control and protection of the underlying blockchain, and the level of trust between each participant of the network can be reflected on the consensus model that will be used or regulated by the access rights to the data that each participant can obtain. Furthermore, the authorisation of each participant can be revoked in the case of misuse or withdrawal of trust.

Blockchain technology has an enormous potential that has been noticed by several industries that are looking forward to exploit its various advantages and offerings in order to introduce new services, products and offerings to their clients or to rejuvenate their internal processes and legacy systems towards a better performance, reduction of financial costs and increase of trust between the partners involved in business transactions. The mostly adopted area is the cryptocurrencies area where Bitcoin and Ethereum are the most notable cases. Additionally, blockchain is adopted in financial services, insurance services, supply chain, energy trading, sales, digital music, anti-counterfeiting, domain name services and videogames, among others. For the financial and insurance services in particular, the blockchain technology has found many potential use cases for providing blockchain-enabled banking and insurance services that optimise various back-office processes, removing various intermediaries and disrupting various operational processes towards the financial cost reductions and the expansion of the portfolio of offered services to their customers.

2.2 The role of blockchain technology in INFINITECH RA

Within the INFINITECH Reference Architecture (INFINITECH RA), as presented in deliverable D2.13, the blockchain technology has a dual presence and can be exploited in different ways, depending on the scope of the use case that INFINITECH RA aims to address.

Hence, on the one hand, blockchain can be considered as an additional data source type at the infrastructure layer, from which data are accessed and collected with the use of the respective

sophisticated data collection mechanisms. On the other hand, blockchain can be considered as a cross-cutting service positioned in the central layer of the INFINITECH RA, in which decentralised applications tailored to the needs of the financial and insurance sectors can be developed and exploited. The blockchain-enabled decentralised applications can be utilised to realise use cases that can optimise, and even revolutionize, core operational processes of the financial or insurance institutions, decreasing their costs and increasing radically their performance and efficiency by reducing or eliminating the need for manual processing or manipulation, while at the same time augmenting their level of trust. Within the context of the WP4, the focus is on the development of such decentralised applications that will showcase the potentials of the blockchain technology.

As with many other industrial sectors, the financial and insurance sectors are highly regulated with strict legislations and processes in which security and trust are fundamentals aspects. While the blockchain technology is considered as the dominant candidate to disrupt all of the financial and insurance sectors' processes, as it is promising to mitigate the cost of trust and increment the security level in these processes and even business models [4], not all blockchain implementations are deemed as ideal candidates. The reason for this lays on the specific characteristics offered by the two major approaches followed in the blockchain technology, the permissionless blockchain and the permissioned blockchain.

While the permissionless blockchain, with public open networks to which anyone can participate and interact in an anonymous manner, has been adopted in the case of cryptocurrencies, when it comes to more enterprise-orient use cases, such as the banking institutions or other financial institutions, different requirements arise and are related mainly to the privacy and confidentiality of the data, as well as the underlying business or financial transactions stored in the blockchain, the regulated and strictly controlled access to the blockchain network, the performance of the network with high throughput and low latency, and most importantly the hard requirement of the identifiable and pre-approved identity of the participants of the blockchain network. For all these reasons, from the two major approaches only the permissioned blockchain technology is considered as the appropriate candidate solution and will be exploited within the IRA.

Towards this end, the consortium decided to exploit the Hyperledger Fabric open source enterprise-grade permissioned distributed ledger technology (DLT) platform [5] that is one of the most active projects of the Hyperledger project founded by the Linux Foundation. Hyperledger Fabric has been designed specifically for enterprise use and delivers a set of key differentiating capabilities over other popular distributed ledger or blockchain platforms. One of the main differentiations of Fabric is its highly modular and configurable architecture that promotes the innovation, versatility and optimization for a broad range of industry use cases including banking, finance and insurance [6]. Based on its modular and configurable architecture, Fabric guarantees high level of confidentiality, resiliency, flexibility, and scalability.

Through its pluggable ordering service, consensus can be achieved with multiple implementations, such as the CFT or BFT and more, based on the requirements of a specific use case or deployment. Fabric offers a private and permissioned blockchain network where all participants can be enrolled based on their cryptographic entities, through a set of pluggable trusted membership service providers that are supported, and can be tailored to the needs of the deployment. Fabric provides its own ordering service implementation named Raft. Raft ordering service is a CFT that is based on an implementation of Raft protocol in the etcd distributed key-value store and it constitutes the first step toward Fabric's development of a BFT ordering service.

Although Hyperledger Fabric's Raft ordering service is CFT based on an implementation of Raft protocol in etcd, it constitutes the first step toward Fabric's development of a byzantine fault tolerant (BFT) ordering service.

In Fabric, smart contracts, referred as chaincode, are operating in an isolated container environment, such as Docker, and can be written in standard programming languages, such as Go and Node.js. Smart contracts offer the required interfaces that are exploited by applications outside of the blockchain network in order to interact with distributed ledger providing the required level of abstraction, as well as increased level of privacy and confidentiality. To further promote the privacy and confidentiality, Fabric enables the creation of channels in which the participants own a separate ledger of transactions from the rest of the blockchain network that is visible only to the participants of the channel. Finally, it provides the feature of private data, where collections of data can be only be visible and accessible to a portion of the participants of a specific channel.

It is acknowledged that the combination of all these key differentiating capabilities, sets Fabric as one of the better performing platforms in transaction processing and transaction confirmation latency platform. Its pluggable architecture enables its exploitation in a variety of different use cases of the financial and insurance sectors that are characterized as highly complex and restrictive sectors.

Towards this end, to address the needs of the financial and insurance sectors, a new blockchain capability which will extend the existing capabilities is proposed and a set of trusted distributed applications in the form of chaincode will be developed. The aim of the new blockchain capability will enhance the blockchain for the needs of these sectors. On the other hand, the design distributed applications will address core use cases of the financial and insurance sectors exploiting the benefits of the permissioned blockchain technology. In this sense, the designed distributed applications will effectively leverage the underlying permissioned blockchain infrastructure provided by Fabric that is designed in accordance to the needs and requirements of the stakeholders of the specific sectors. The details of the proposed blockchain capability, that are currently under formulation, are described in Section 3. The design specifications of the designed distributed applications are presented in detail in Section 4 of the current deliverable, while the details of the designed underlying permissioned blockchain network that these distributed applications will be deployed are documented in Section 5.

3 INFINITECH Blockchain Capabilities

Blockchain based solutions are applications built on top of the selected blockchain platform technology exploiting the provided tools of the blockchain platform, such as smart contracts and client SDKs. Generally speaking, when some use cases require a certain capability currently not existing in the selected blockchain platform, two main approaches can be followed to cope with the specific requirement. The first approach includes the extension of the current platform capabilities with additional ones that are suitable for the needs of the use case, while the second approach requires the design and implementation of a solution at the application layer that the scenarios requiring the specific capability can leverage, if possible.

In the scope of the INFINITECH project complying with the General Data Protection Regulation (GDPR) [7] is an example of such needed capability as described henceforth.

3.1 Addressing GDPR in Blockchain

3.1.1 Motivation

One of blockchain stated benefits includes its immutability, i.e. once the data is written on the ledger it cannot be changed or deleted. This assures complete trustworthiness of the ledger's content and supports provenance and non-repudiation use cases. Nevertheless, this seemingly comes in contrast to and "the right to be forgotten" which is a basic principle of GDPR.

Highly regulated sectors, such as healthcare, banking, and insurance, need to support GDPR privacy articles, such as a "right to be forgotten", and therefore, require an approach which can allow to remove personal data from the chain on demand without violating the blockchain's consistency.

It became evident at the early stages of the project that providing such capability can benefit the aspired INFINITECH scenarios. The aim is to provide such functionality "horizontally" at the level of the blockchain platform and not at the application level, so applications requiring this capability could be built on top of this "extended" platform. Such approach implies a departure from the current Fabric implementation as described in the following sections. The following subsection presents a high-level overview and the design behind the proposed solution. In the future versions of this deliverable, more mature versions of the proposed approach will be presented in detail.

3.1.2 Description of the solution

Many approaches have been proposed for dealing with this problem (e.g.; [8] and [9]). Some approaches rely on not storing personal data on the blockchain at all, the data is stored offchain and only the hashes of the data onchain. Such an approach, while solving the problem of putting personal data on immutable ledger, poses other problems such as performance penalties for complete end-to-end latencies, and consistency and availability challenges in case of multiple administrative domains for this data. Other models have been proposed, based on the fact that encrypted data cannot be retrieved within a reasonable period of time without proper authorization, and therefore, concluding that it can be stored directly in the blockchain. This notion is not robust enough to guarantee the GDPR compliance for the right to be forgotten in systems where the data may be used for more extended periods of time. Forward secrecy is unclear at the quantum computing age.

Our proposed solution is based on changing the transaction content stored at the level of the block, the transaction envelope content, and the way transactions are validated at the Hyperledger Fabric infrastructure level.

The main steps of the approach are as follows (Figure 1):

- Store values and their hashes in the ledger
- Construct block hash from the values' hashes only (not from the values themselves)
- When validating a transaction, ensure hash matches the computed value's hash
- Keep the actual values alongside the signed transaction
- Introduce a new type of transaction – the *redaction transaction*

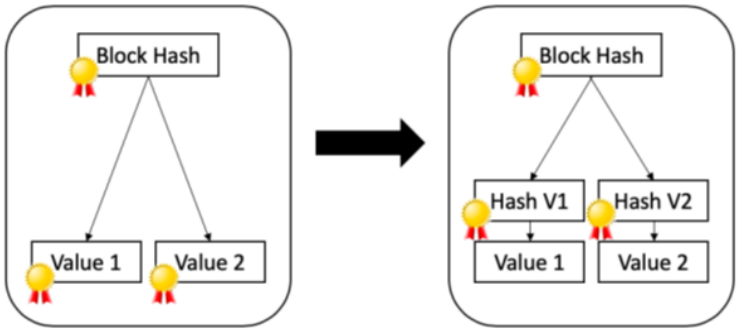

Figure 1: Storing hashes and values on the ledger

The redaction transaction would redact the value itself (set bits to zero), not its hash. The redaction does not entail a change in the value's hash nor in the hash-chain (which is only dependent on the hash value of the original value and not on the value itself). This way the consistency of the chain and accountability is preserved (see Figure 2).

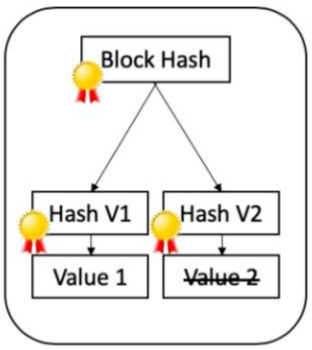

Figure 2: Reduction of transaction value

It should be noted that in this deliverable our aim is to describe the design of our approach. As the project evolves, the details of this approach will be further extended, and the presented description will be further elaborated.

3.1.3 Use Cases

The following sections provide the initial documentation of all the use cases encapsulated in this extension, describing in detail all information of each use case.

3.1.3.1 Use Case GDPR-1: New transaction submission

As a blockchain client, I would like to submit a transaction proposal to a blockchain supporting value reduction and make sure the data is properly written on the ledger.

Table 1: Use Case GDPR-1: New transaction submission

Stakeholders involved:	Transaction invoker
Pre-conditions:	1. The requesting party is a legal end-user in the network and has write access
Post-conditions:	1. The transaction is written on the blockchain, the data is available for reading
Data Attributes	2. A value to write
Normal Flow	1. The authentication and access roles for the requester are determined 2. In case write access is allowed, the hash of the value being stored on the blockchain is calculated 3. The value and the hash are stored on the blockchain, hash as part of the transactions read write set, pre-image value in a separate structure
Pass Metrics	1. The transaction is available on the chain; value can be read
Fail Metrics	1. Requested information could not be written: -the invoker has no access

3.1.3.2 Use Case GDPR-2: Value reduction

As a blockchain client, I would like to redact a value previously stored by me on the ledger.

Table 2: Use Case GDPR-2: Value reduction

Stakeholders involved:	Transaction invoker
Pre-conditions:	1. The requesting party is a legal end-user in the network and has write access and has previously written a value
Post-conditions:	1. The redacted value (bits set to zero) are written on the blockchain, the original hash is kept.
Data Attributes	1. A value to redact
Normal Flow	1. The authentication and access roles for the requester are determined 2. In case write access is allowed, the existence of the value to redact is checked 3. The redacted value is stored on chain
Pass Metrics	1. A value is redacted, and a notification is returned to the invoker
Fail Metrics	1. Value could not be redacted: -the invoker has no access -the requested value does not exist

3.1.3.3 Use Case GDPR-3: Read redacted value

As a blockchain client, I would like to make sure that an attempt to read the value after reduction will return the redacted (all bits "0") and not the original value.

Table 3: Use Case GDPR-3: Read redacted value

Stakeholders involved:	Any query invoker
Pre-conditions:	1. The requesting party is a legal end-user in the network and has read access
Post-conditions:	1. The redacted value (bits set to zero) and not the original value is returned
Data Attributes	1. The value to read
Normal Flow	1. The authentication and access roles (read) for the requester are determined 2. In case read access is allowed, the redacted transaction payload is read (bits set to zero) and returned to the invoker
Pass Metrics	1. The redacted value is returned to the invoker
Fail Metrics	1. Requested query could not be performed: -the invoker has no access -the value does not exist

4 INFINITECH Blockchain Applications

The benefits of the blockchain technology are leveraged with the development of trusted distributed applications that are deployed within the nodes of the blockchain network. In the blockchain technology, the trusted distributed applications are developed as smart contracts or chaincode in the Fabric terminology. The role of a chaincode is to generate the new facts that will be added to the ledger, which maintains all the facts related to the current and historical state of a set of business objects, in order to introduce the appropriate changes in both the current and historical state. In this sense, the chaincode defines the transaction logic that is responsible for the evolution of the state of the business objects. Furthermore, the chaincode provides a set of interfaces that are utilised by the applications outside of the blockchain network in order to interact with the distributed ledger.

Hence, in the design of any blockchain-enabled solution that effectively addresses the needs of a specific business operation, process or service, the main components included are the external to the blockchain network application and the chaincode that is deployed on the blockchain network. In the design specifications of both components, a specific business logic is encapsulated whose aspects are clearly defining the permitted and required operations which are executed for a use case of the business operation.

With regards to the design of the chaincode, it is usually written in the GO programming language and it explicitly defines the governance rules for any type of business object utilised in the use case. The structuring of the source code of the chaincode can vary as there is no strict norm of how the source code should be constructed, rather than best practises formulated by the blockchain development communities. However, the basis of the source code is the definition of the business objects on top of which this chaincode will perform all the operations and the functions that will undertake the execution of these operations. Thus, as for every specific business operation different business objects are defined, the whole chaincode is tailored to the needs of each business operation.

Nevertheless, with the context of the INFINITECH project and specifically Task 4.3, the developed chaincode will adhere to the following structuring of the source code that is based on the logical schema of the Data Processing Components as defined in the deliverable D2.5:

- *Blockchain Reader*: The main purpose of this component is to enable the fetching of the requested data from the blockchain ledger. Depending on the underlying use case, the design specifications may vary. However the context of the functions of the component will remain the same.
- *Blockchain Writer*: The main purpose of this component is to facilitate the submission of new transactions to the blockchain ledger. As with the blockchain reader, the design specification for each use case may vary, nevertheless the context of the functions of the component will remain unaffected.
- *Smart Contract Executor*: The main purpose of this component is to encapsulate the business logic of the designed use case and execute the smart contracts on the blockchain ledger. Hence, the design specifications of each use case are tailored to the needs of the aspired operation and service and that will orchestrate the use case execution.
- *Blockchain Authenticator*: The main purpose of this component is to perform the authentication of the blockchain network user in order to grant access to a specific channel of the blockchain network. The implementation of the specific component is almost generic and will be utilised across all the implemented use cases.
- *Blockchain Encryptor:* The main purpose of this component is to perform the encryption of the data that are involved and produced within the smart contract execution utilising the AES 256

encryption. The implementation of the specific component is almost generic and will be utilised across all the implemented use cases.

- *Blockchain Decryptor:* The main purpose of this component is to perform the decryption of the data that were encrypted by the Blockchain Encryptor. Again, the implementation of the specific component is almost generic and will be utilised across all the implemented use cases.

In the following sections, the design specifications of the blockchain applications that will be developed within the context of Task 4.3 are presented in detail. In total, two different blockchain applications are presented. For these two applications, namely the Consent Management and the Know-Your-Customer (KYC) / Know-Your-Business (KYB), a thorough description of the addressed business operation is presented, highlighting the use of the blockchain technology on each application accompanied by the high-level architecture of the application. Furthermore, the details of each specific use case of the business operation that is supported by the blockchain application, is presented. Finally, for each use case of the business operation, the corresponding sequence diagram that illustrates the interactions between the involved stakeholders and the components is documented. The section concludes with a short description of the application related to tokenization use case that is currently formulated within the context of Task 4.4. However the thorough documentation of this specific use case will be documented within the context of the deliverable D4.10 in accordance with the INFINITECH Description of Action (DoA).

It should be noted that the list of applications that will be presented might be further expanded with additional applications, as well as that the existing applications might be further enriched with additional uses cases and functionalities, as the project evolves and new requirements may arise as the result of the feedback that will collected by the pilots of the project and the stakeholders of the platform. These updates will be reported in the next iterations of the current deliverable as planned by the Description of Action of the INFINITECH project.

4.1 Consent Management

4.1.1 Description of the solution

Collaborative data sharing between customers, banks and other organizations, enables application of advanced analytics over particular datasets and intelligent support tools for better understanding customers and their financial relationships, thus being critically important in today's financial markets. However, the development of intelligence support tools that will support new customer services that solve business problems such as improved Know-Your-Customer (KYC) processes and consequently Anti Money Laundering (AML), credit scoring and fraud detection services that can be built upon them, is highly dependent on the customers permission to share data. As a result, the requirement for a trusted and secure sharing mechanism of customer consent arises. In this sense, banks also identify granular permission consent as a key enabler of trust which is vital to maximise data sharing and ensure customers are comfortable with sharing data.

In this blockchain application, we aim at exploiting the blockchain technology and specifically the permissioned blockchain in order to develop a decentralised and robust consent management mechanism, that will enable the sharing of the customers' consent to exchange and utilise their customer data across different banking institutions. Blockchain technology, and its latest advancements, appears as a compelling technology to overcome the underlying challenges of trust improvement due to its decentralised nature and immutability, due to impossibility of ledger falsification. The integrity of customer data processing consents and their immutable versioning control are protected by the blockchain infrastructure. The blockchain-enabled consent management mechanism will enable the financial institutions to effectively manage and share their customers'

consents in a transparent and unambiguous manner, enabling them to inform the customers at any time about:

a) any customer data they are managing upon their consent,
b) the status of their consent (active or revoked/withdrawn),
c) the recipients (financial institutions or peers) of their customer data upon their consent,
d) the purpose (or even legal basis) and time period of their customer data sharing to the recipient (financial institutions or peers)

In the same transparent and unambiguous manner, the customers of the financial institution will:

a) be constantly informed for all the requests for sharing of their customer data
b) be able to activate or revoke their consents
c) be constantly aware of the active consents they have given to each specific recipient.

The blockchain-enabled consent management mechanism guarantees the integrity of the consents through the usage of the blockchain technology, as well as cryptographic techniques and digital signatures incorporated in it. Besides the consent records and their integrity, the blockchain technology will be used to securely maintain the consent history providing the complete consents' versioning. Through the blockchain technology, immutable versioning control is provided that is capable of obtaining the latest version of the consent, as well as the previous versions of the consent and their valid periods, in an indisputable manner. Thus, the blockchain is capable of storing the consents and their complete update history in a secure and trusted manner. In this way, both the financial institutions, as well as their customers are protected as both the consents' integrity and validity periods are secured.

With regards to the consents and their validity period, two different approaches are considered. The first approach provides the ability of an "once off" consent, in which the validity period of the consent is a predefined period. Once this period expires, the consent is automatically revoked and new consent (or an updated consent) is required in order for the recipient to be able to access the customer's data. An example of the cases considered for this consent type is the customer's consent for sharing of KYC data between two financial institutions (banks) for the scenario of a loan origination / application or an account opening. The second approach provides the ability of a permanent (or "regular") consent that has an infinite validity period and it is only invalidated if the consent is withdrawn. An example of the cases considered for this consent type is the Peer to Peer (including Person to Person, Person to Organisation, Organisation to Organisation cases) customer data sharing consent in which a customer utilises an interface of a mobile application to select a set of its specific customer data (such as specific accounts, specific transactions or alerts) that they would like to share with an individual person or a group of persons. It should be noted that the consent management mechanism does not save or distribute any customer data. Customer data are exchanged using the respective secured APIs of the financial institution. Instead, the blockchain based consent management mechanism maintains the minimum information that is required in order to formulate a consent agreement between the customer and the respective financial institution.

Figure 3 depicts a high-level architecture of the proposed solution. As depicted, the core elements of the proposed solution are the *Consent Management System*, the *File Storage* and the *Blockchain infrastructure*, while the key stakeholders that are involved and are interacting with the proposed solution are the *internal financial institution* that collects and has access to its customer data, *the customer of internal financial institution* whose data are collected by the internal financial institution and finally the *external financial institution or peer* that aspires to obtain the data of the customer of internal financial institution upon the formulation of a valid and legitimate consent that is formulated between the three parties.

In this architecture, the Consent Management System is the mediator between the involved parties. It receives and processes the requests for a consent formulation from the external financial institution or peer, to the internal financial institution and consequently the customer of the internal institution. By interacting with all three involved parties and upon acceptance of the details and terms of the consent by all of them, the final consent receipt is formulated and stored in digital form within the File Storage. At the last and most crucial step, the minimum information that is required from the formulated consent form is entered in the blockchain infrastructure by invoking the deployed chaincode that is responsible for the generation of a new transaction in the underlying ledger. Additionally, the Consent Management System, is consulting and retrieving the information stored in the blockchain infrastructure, by invoking the deployed chaincode in order to formulate an access control decision depending on the existence and validity of a consent when access to customer data via the internal financial institution's APIs is requested or to retrieve the complete history of consents for a customer upon request.

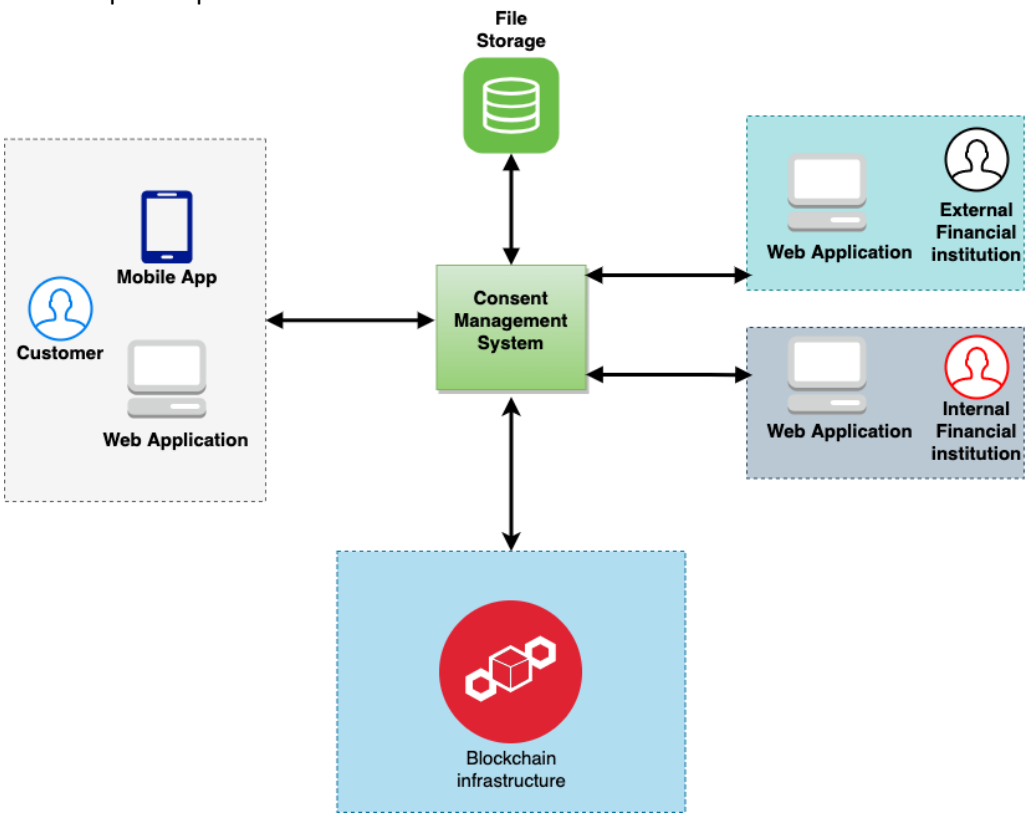

Figure 3: High-level architecture of the Consent Management solution

One of the core aspects of the design specifications is the definition of the business objects for which the current and historical state will be maintained and updated through the functions of chaincode. To this end, for the Consent Management application the initial data schema which defines the core business objects has been defined. The definition was based on the Consent Receipt specification that is proposed by the Kantara Initiative [10] which has been adapted in terms of terminology in order to be aligned with the EU GDPR legislation. Figure 4 depicts the initial data schema of the Consent Management application, while the details of all the entities are documented in tables Table 4 to Table 7.

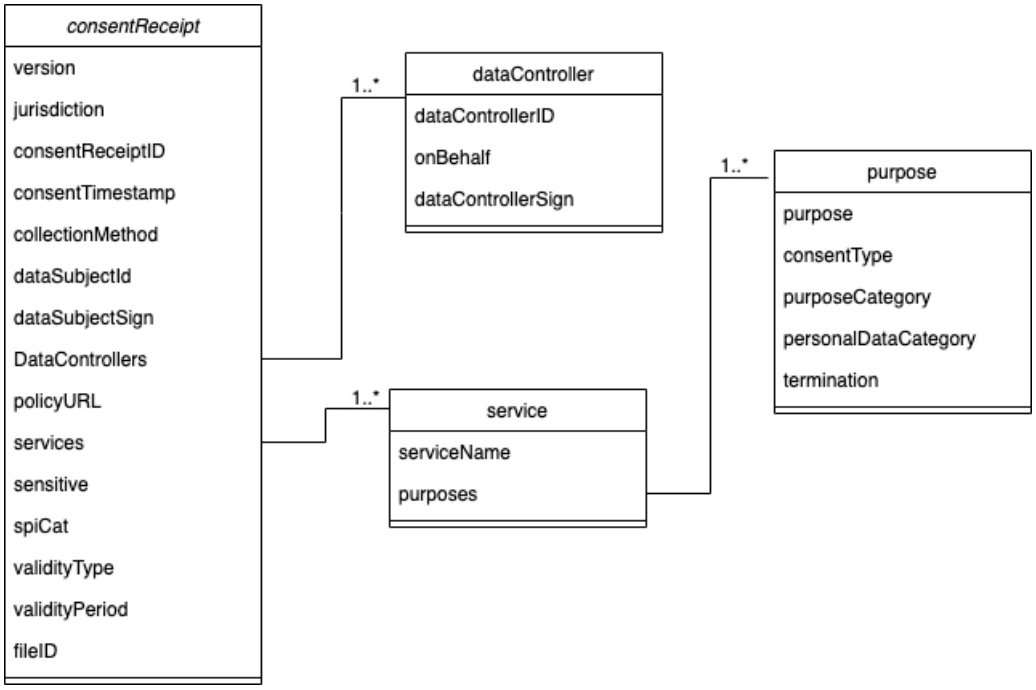

Figure 4: Consent Management Data Schema

Table 4: Consent Management Data Schema (1)

consentReceipt		
Name	**Type**	**Short description**
version	String	REQUIRED: The version of the consent specification to which a receipt conforms
jurisdiction	String	REQUIRED: The jurisdiction(s) applicable to this transaction i.e. EU and/or any EU-national
consentReceiptID	String	REQUIRED: The unique identifier of each Consent Receipt in UUID-4 format
consentTimestamp	String	REQUIRED: Date and time of the consent transaction in ISO 8601 format
collectionMethod	String	REQUIRED: A description of the method by which consent was obtained i.e. the Consent Management System
dataSubjectID	String	REQUIRED: The unique identifier of the Data Subject in UUID-4 format
dataSubjectSign	String	REQUIRED: The Data Subject's digital signature in base64 format.
dataControllers	Array [dataController]	REQUIRED: An array of dataController items (see Table 5)
policyURL	String	REQUIRED: A link to the DataController's privacy statement/policy and applicable terms of use in effect when the consent was obtained, and the receipt was issued.
services	Array [service]	REQUIRED: An array of Service items (see Table 6)

sensitive	Boolean	REQUIRED: Indicates whether the consent interaction contains personal data that is designated sensitive or not sensitive
spiCat	Array [categories]	REQUIRED: A listing of categories where personal data collected is sensitive. The array must contain the categories if sensitive is TRUE
validityType	String	REQUIRED: The validity type of the consent, either ONCE_OFF or PERMANENT.
validityPeriod	String	REQUIRED: Date and time of the consent expiration in ISO 8601 format. Required when validityType is ONCE_OFF.
fileID	String	REQUIRED: A reference to the file stored in the File Storage

Table 5: Consent Management Data Schema (2)

dataController		
Name	**Type**	**Short description**
dataControllerID	String	REQUIRED: The unique identifier of the data controller in UUID-4 format
onBehalf	Boolean	OPTIONAL: True if a data processor is acting on behalf of a data controller.
dataControllerSign	String	REQUIRED: The Data Controller's digital signature in base64 format.

Table 6: Consent Management Data Schema (3)

service		
Name	**Type**	**Short description**
serviceName	String	REQUIRED: The service or group of services being provided for which personal data is collected. The ID of the service for which consent for the collection, use, and disclosure of personal data is being provided.
purposes	Array [purpose]	REQUIRED: An array of Purpose items (see Table 7)

Table 7: Consent Management Data Schema (4)

purpose		
Name	**Type**	**Short description**
purpose	String	OPTIONAL: A short, clear explanation of why the personal data is required
consentType	String	REQUIRED: The type of the consent used by the Data Controller as their authority to collect, use or disclose personal data. The accepted values are EXPLICIT or IMPLICIT
purposeCategory	String	REQUIRED: The reason the Data Controller is collecting the personal data. The acceptable values are based on the (CISWG) Wiki page [11]

personalDataCategory	String	REQUIRED: A list of defined personal data categories. Personal data category should reflect the category that will be shared as understood by the Data Subject. The acceptable values are based on the (CISWG) Wiki page [11]
termination	String	REQUIRED: Conditions for the termination of consent. Link to policy defining how consent or purpose is terminated.

Taking into consideration the described data schema, the main functions that are foreseen on a component-level are the following:

- Blockchain Reader:
  - *QueryConsent():* The function is responsible for retrieving the consent receipt from the ledger based on the ID that each specific consent receipt has been saved in the ledger. It utilises the Decrypt() function to decrypt the data once they have been retrieved from the ledger.
- Blockchain Writer:
  - *CreateConsent():* The function is responsible for the introduction of the new consent receipt into the ledger. It utilises the Encrypt() function to encrypt the data prior to their insertion in the ledger.
- Smart Contract Executor:
  - *Contract_SubmitTransaction():* The function is responsible for initiating the execution of the smart contract for the creation of a new consent receipt and specifically the CreateConsent() function with the necessary variables.
  - *Contract_evaluateTransaction():* The function is responsible for initiating the execution of the smart contract for the retrieval of a consent receipt and specifically the QueryConsent() function with the necessary variables.
- Blockchain Authenticator:
  - *Authenticate():* The function is responsible for the authentication of the user before the access to a specific channel is granted utilising the appropriate keys and certificates.
- Blockchain Encryptor:
  - *Encrypt():* The function is responsible for the encryption of the data utilising the AES 256 encryption before they are inserted into the ledger. The randomly generated encryption key is kept in a Vault.
- Blockchain Decryptor:
  - *Decrypt():* The function is responsible for the decryption of the encrypted data that are fetched from the ledger. It utilises the corresponding encryption key that is kept in a Vault.

4.1.2 Use cases

The following sections provide the detailed documentation of all the use cases encapsulated in this blockchain application describing in detail all information of each use case.

4.1.2.1 Use case CMS-1: Register Customer in the Consent Management System

In order for a customer to be able to receive consent requests, it is mandatory that they are registered in the Consent Management System, creating their profile that will be used in order to receive consent requests. The customer profile shall include the minimum customer discovery and communication details required in order to receive a consent request. The profile information and any consequent private information is not disclosed to any interested party and is not saved in the blockchain infrastructure.

Table 8: Consent Management Use Case CMS-1

Stakeholders involved:	Customer, Internal Financial Institution
Pre-conditions:	A customer willing to provide his/her consent to share his/her customer data upon his/her approval
Post-conditions:	A customer is registered in the Consent Management System, his /her profile is created and is able to receive consent requests from Internal Financial Institution.
Data Attributes	Required data based on the Consent Management Data Schema
Normal Flow	1. The customer (or Internal Financial Institution on behalf of the customer) registers to the Consent Management System filling in the required information explained in the Data Attributes above. 2. The customer provides his / her approval to receive new consent requests.
Pass Metrics	1. Customer profile is available in the Consent Management System
Fail Metrics	1. No customer profile is available in the Consent Management System

In the same manner, the external financial institution or the peer, registers in the Consent Management System in order to be able to initiate consent request to a customer of the Internal Financial Institution.

Table 9: Consent Management Use Case CMS-1 (2)

Stakeholders involved:	Customer, Internal Financial Institution
Pre-conditions:	A customer willing to provide his/her consent to share his/her customer data upon his/her approval
Post-conditions:	A customer is registered in the Consent Management System, his /her profile is created and is able to receive consent requests from Internal Financial Institution.
Data Attributes	Required data based on the Consent Management Data Schema
Normal Flow	1. The customer (or Internal Financial Institution on behalf of the customer) registers to the Consent Management System filling in the required information explained in the Data Attributes above. 2. The customer provides his / her approval to receive new consent requests.
Pass Metrics	1. Customer profile is available in the Consent Management System
Fail Metrics	1. No customer profile is available in the Consent Management System

4.1.2.2 Use Case CMS-2: Customer receives a request to provide new consent for sharing his/her customer data

The stakeholder wishing to gain access to the customer data issues a new consent request to the Consent Management System specifying the details of the requested customer data (i.e. specific data, period of usage). The customer receives a notification with all the required information for the consent request in a proper way that it allows him/her to review the request.

Table 10: Consent Management Use Case CMS-2

Stakeholders involved:	Customer, Internal Financial Institution, Banking institutions, Peers, Financial organisations
Pre-conditions:	1. A customer profile is available in the Consent Management System capable of receiving new consent requests 2. A new request for customer data is issued by a banking institution or financial organisation to Internal Financial Institution
Post-conditions:	The customer is notified for the new consent request in order to review and decide about giving his consent or denying the access
Data Attributes	Required data based on the Consent Management Data Schema
Normal Flow	1. Internal Financial Institution generates the new request for consent to the customer including all the required information described in the data attributes above 2. Customer receives the new request in a proper format (e.g., through his/her mobile phone app or a web-based interface provided by the Internal Financial Institution) so that he/she can review and decide whether to provide his/her consent or deny the access to his/her customer data.
Pass Metrics	1. The customer is informed for the new consent request and is able to formulate a decision.
Fail Metrics	1. The customer is not informed for the new consent request

4.1.2.3 Use Case CMS-3: Definition of the consent

The customer reviews and possibly alters the details and conditions of the consent request before formulating his/her decision to give his consent or deny the access. In the case of approval, the final consent is defined by the customer and it is submitted to the Consent Management System. In the case of the denial, the request is blocked and the interested party is informed and the processing is finished.

Table 11: Consent Management Use Case CMS-3

Stakeholders involved:	Customer, Internal Financial Institution
Pre-conditions:	A new consent request from an interested party is provided to the customer

Post-conditions:	The customer formulates the consent form that is ready to be signed by both parties
Data Attributes	Required data based on the Consent Management Data Schema
Normal Flow	1. The customer reviews, edits and finalises the consent form details (based on the data attributes above). He/she is able to check and alter the proposed attributes and details of the request in a user-friendly way (e.g., through his/her mobile phone app or a web-based interface provided by the Internal Financial Institution). 2. **In case of approval:** The customer provides his/her consent form to the Consent Management System in order to be signed. 3. **In case of denial:** The customer denies the request; the interested stakeholder is informed and the process is finished.
Pass Metrics	1. **In case of approval:** The consent form is ready to be signed by both parties 2. **In case of denial:** The interested stakeholder is informed and the process is finished
Fail Metrics	1. **In case of approval:** No consent form is available 2. **In case of denial:** The process is still open

4.1.2.4 Use Case CMS-4: Signing of the consent by the interested parties

Once the consent form is ready, the Consent Management System provide it to both parties (the customer and the interested stakeholder) in order to be digitally signed. The Consent Management System collects the digitally signed from both parties' consent form.

Table 12: Consent Management Use Case CMS-4

Stakeholders involved:	Customer, Internal Financial Institution, Banking institutions, Peers, Financial organisations
Pre-conditions:	A valid consent form from a customer is available
Post-conditions:	The signed from both parties' consent form
Data Attributes	Required data based on the Consent Management Data Schema
Normal Flow	1. The consent form is sent to the customer by the Consent Management System (e.g., through his/her mobile phone app or a web-based interface provided by the Internal Financial Institution) in order to be digitally signed 2. The customer provides the digitally signed consent form to the Consent Management System 3. The consent form is sent to the interested party by the Consent Management System in order to be digitally signed.

	4. The interested party provides the digitally signed consent form to the Consent Management System
Pass Metrics	1. The Consent Management System obtains the digitally signed from both parties' consent form
Fail Metrics	1. The Consent Management System cannot obtain the digitally signed from both parties' consent form

4.1.2.5 Use Case CMS-5: Consent form is entered into blockchain

Once the digitally singed consent form is available, the Consent Management System interacts with the blockchain infrastructure in order to create a new transaction that will be introduced in the blockchain.

In the case of the "once off" consent, where a specific validity period is defined, the Consent Management System is internally handling the validation of consent time period by creating and monitoring the specific timer in order to perform the validation of consent time period. Use Case 1.9 describes the handling performed when the consent time period expires.

Table 13: Consent Management Use Case CMS-5

Stakeholders involved:	N/A
Pre-conditions:	The digitally signed from both parties' consent form is available.
Post-conditions:	A new transaction containing all the information of the consent form is available in the blockchain infrastructure
Data Attributes	Required data based on the Consent Management Data Schema
Normal Flow	1. The Consent Management System retrieves the digitally signed from both parties' consent form. 2. The Consent Management System interacts with the blockchain infrastructure and enters the new consent form in the ledger in the form of a new transaction. 3. In the case of the "once off" consent the proper timer is started in the Consent Management System.
Pass Metrics	1. The new transaction containing all the information of the consent form is available in the blockchain infrastructure
Fail Metrics	1. No new transaction is available in the blockchain infrastructure

4.1.2.6 Use Case CMS-6: Consent update or withdrawal

The consent form can be updated or withdrawn at any time by the customer side. In the case of the update the previously described steps Use Case 1.3 to Use Case 1.5 are re-executed and the status of the consent remains "active". On the other hand, the withdrawal of the consent is internally translated

to “invalidation” of the smart contract and the status of the consent is set to “withdrawn”. In the case of the “once off” consent, the associate timer is restarted when the consent is updated. On the other hand, when the consent is withdrawn the associated timer is stopped.

In all these processes, the consent history is maintained in the blockchain infrastructure with the context of the smart contract. Through the transactions all versions of the consents and their complete update history is maintained.

Table 14: Consent Management Use Case CMS-6

Stakeholders involved:	Customer, Internal Financial Institution, Banking institutions, Peers, Financial organisations
Pre-conditions:	A transaction with status active containing all the information of the consent form is available in the blockchain infrastructure
Post-conditions:	A new transaction containing the latest information of the consent form is available in the blockchain infrastructure
Data Attributes	Required data based on the Consent Management Data Schema
Normal Flow	1. The customer initiates the consent update process by defining the consent form (in the case of withdrawal the corresponding status is defined) 2. The Consent Management System retrieves the new consent form and sends it to both the customer and the interested stakeholders 3. The consent form is digitally signed by both parties and provided to the Consent Management System. 4. The Consent Management interacts with the blockchain infrastructure and introduces the updates with a new transaction. 5. In the case of the “once off” consent, the timer is either restarted (update) or stopped (withdrawal) in the Consent Management System.
Pass Metrics	1. A new transaction containing the latest information of the consent form is available in the blockchain infrastructure
Fail Metrics	1. There is no new transaction with updated consent information

4.1.2.7 Use Case CMS-7: Access Control based on the consent forms

During a data access request to the underlying data management system, the data management system is consulting the Consent Management System in order to validate the consent status between the requesting party and the customer whose data are requested. By utilising the transactions formulated in the previous steps, the Consent Management System is able to formulate the access control decision.

Table 15: Consent Management Use Case CMS-7

Stakeholders involved:	Internal Financial Institution, Banking institutions, Peers, Financial organisations

Pre-conditions:	A transaction containing all the information of the consent form is available in the blockchain infrastructure
Post-conditions:	The access control is formulated based on the consent status contained in the latest transaction
Data Attributes	Required data based on the Consent Management Data Schema
Normal Flow	1. Upon a data access request, the data management system initiates a request to the Consent Management System to check the consent status between the customer and the requesting party 2. The Consent Management System retrieves the relevant transaction from the blockchain infrastructure in order to validate the status of the consent for the requesting party. 3. The Consent Management System formulates the approval or denial decision and informs the data management system
Pass Metrics	1. The data access requests are correctly validated
Fail Metrics	1. The data access requests are incorrectly validated

4.1.2.8 Use Case CMS-8: Retrieve complete history of consents

The customer is able to be constantly informed for all the consents that they have given to each specific recipient, as well as the complete history of the consents. The transaction performed contain all the different versions of the consents of the customer besides the latest one. In the sense, the customer will be able to retrieve at any time his/her consent history per specific stakeholder, for all stakeholders or the all consents given by the customers for a stakeholder.

Table 16: Consent Management Use Case CMS-8

Stakeholders involved:	Customer, Internal Financial Institution
Pre-conditions:	At least a transaction containing all the information of the consent form is available in the blockchain infrastructure
Post-conditions:	The customer is able to retrieve the complete history of consents by the Consent Management System
Data Attributes	Required data based on the Consent Management Data Schema
Normal Flow	1. The customer initiates a request to the Consent Management System to retrieve the active consents per specific stakeholder along with their history or for a specific stakeholder. 2. The Consent Management System interacts with the blockchain infrastructure and retrieves the relevant transactions in order to compile a list of consents that customer has granted

	3. The customer is able to check the requested list of consents in a user-friendly way (e.g., through his/her mobile phone app or a web-based interface provided by the Internal Financial Institution).
Pass Metrics	1. The customer retrieves the request list of consent he/she has granted.
Fail Metrics	1. The customer is able to retrieve the request list of consent he/she has granted

4.1.2.9 Use Case CMS-9: Expiration of the validity period

In the case of the "once off" consent, the validity period is set to a predefined time period. The moment that a transaction is created in the blockchain infrastructure for this specific type of consent, the Consent Management System creates and monitors the specific timer in order to perform the validation of consent time period. Once this timer is expired, the Consent Management System create a new transaction for the specific consent with the status set to "expired".

Table 17: Consent Management Use Case CMS-9

Stakeholders involved:	N/A
Pre-conditions:	A transaction with status active containing all the information of the consent form is available in the blockchain infrastructure whose timer has expired
Post-conditions:	A new transaction containing the latest information of the consent form is available in the blockchain infrastructure with status set to "expired".
Data Attributes	Required data based on the Consent Management Data Schema
Normal Flow	1. Upon the expiration of the validity timer, the Consent Management System retrieves the latest transaction for the specific consent by interacting with the blockchain infrastructure. 2. The Consent Management introduces the updates with a new transaction where the status is set to "expired".
Pass Metrics	1. A new transaction containing the latest information of the consent form is available in the blockchain infrastructure
Fail Metrics	1. There is no new transaction with updated consent information

4.1.3 Sequence Diagrams

In the previous section all the relevant use cases of the designed blockchain application were documented. The following sections present the sequence diagrams for each use case, depicting the interactions between the stakeholders and the components of the designed solution, as well as the interactions between the various components of the designed solution.

4.1.3.1 Register Customer in the Consent Management System

In Use Case CMS-1, the customer of the internal financial institution registers in the Consent Management System in order to create the associated profile that will receive consent requests by

interacting with the web interface of the Consent Management System and providing the profile details. In the same manner, the representative of the external financial institution or the peer registers in the Consent Management System in order to create the associated profile that will initiate consent requests to the internal financial institution in order to get access to the internal financial institution's customer's data.

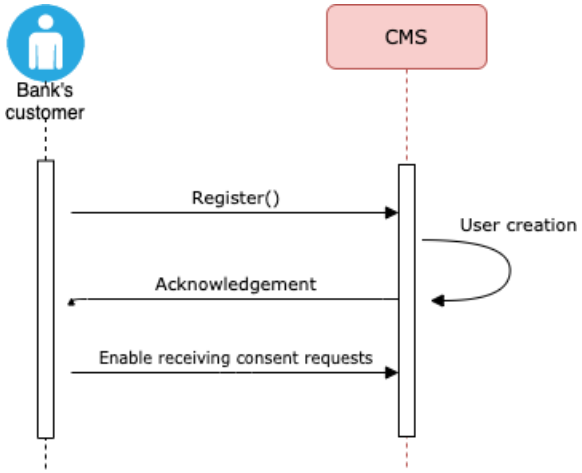

Figure 5: Use Case CMS-1 sequence diagram (customer)

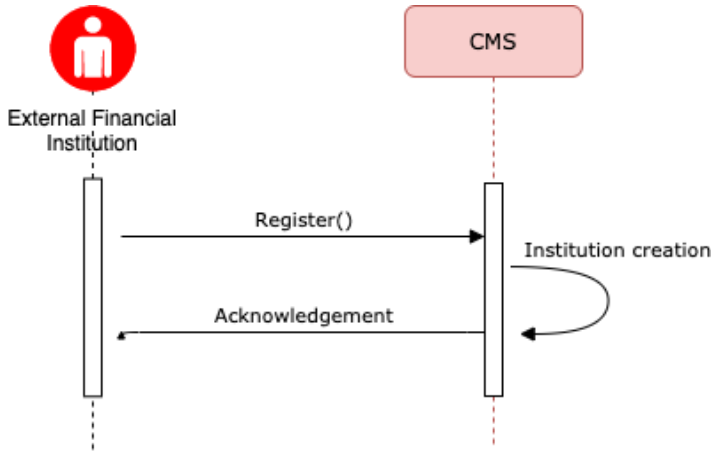

Figure 6: Use Case CMS-1 sequence diagram (external financial institution)

4.1.3.2 Customer receives a request to provide new consent for sharing his/her customer data

In Use Case CMS-2, the external financial institution or peer initiates a request to receive the customer's consent in order to access his/her data. Upon the approval of the internal financial institution's administrator, the Consent Management System interacts with blockchain components in order to check the existence of a consent in the ledger by querying and reading the query results upon decryption. In the case of absence of a valid and active consent, the Consent Management System is informed to transmit the new request to the customer. In case of existence of a valid and active consent, the Consent Management System is informed and approves the request to access the requested data.

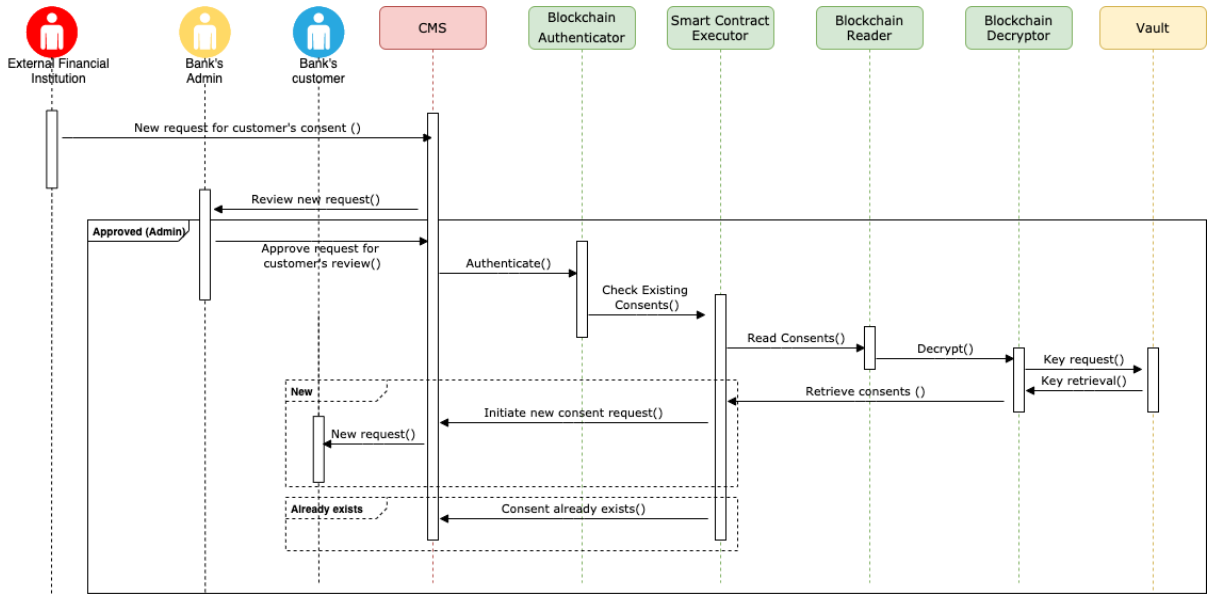

Figure 7: Use Case CMS-2 sequence diagram

4.1.3.3 Definition of the consent

In Use Case CMS-3, the customer receives and reviews the request and in the case of approval, defines the details and conditions of the consent in the Consent Management System and the candidate consent form is created. In the case of denial, the Consent Management System inform the external financial institution.

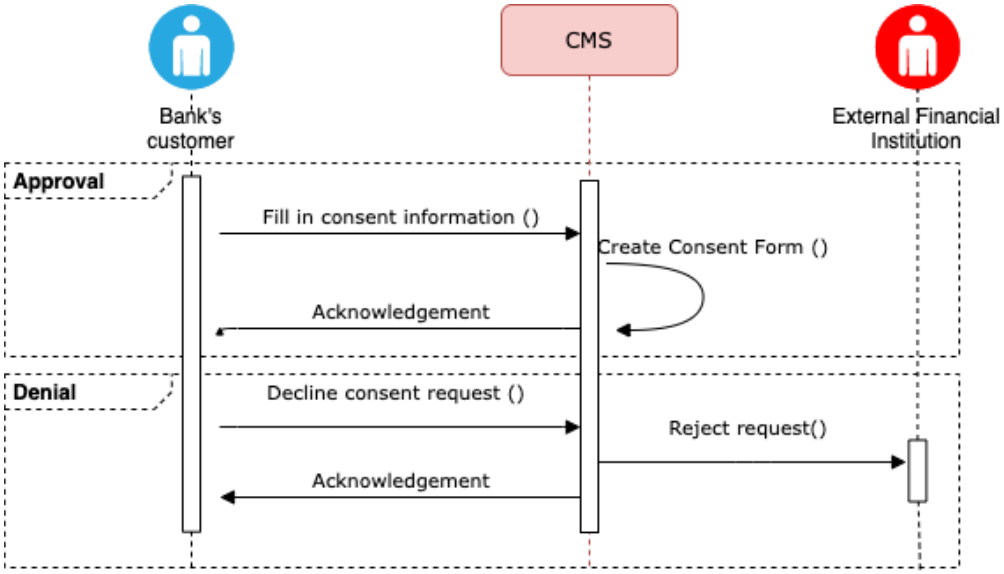

Figure 8: Use Case CMS-3 sequence diagram

4.1.3.4 Signing of the consent by the interested parties

In Use Case CMS-4, the Consent Management System provides the candidate consent form to both the customer and external financial institution and collects the digitally signed consent from both parties.

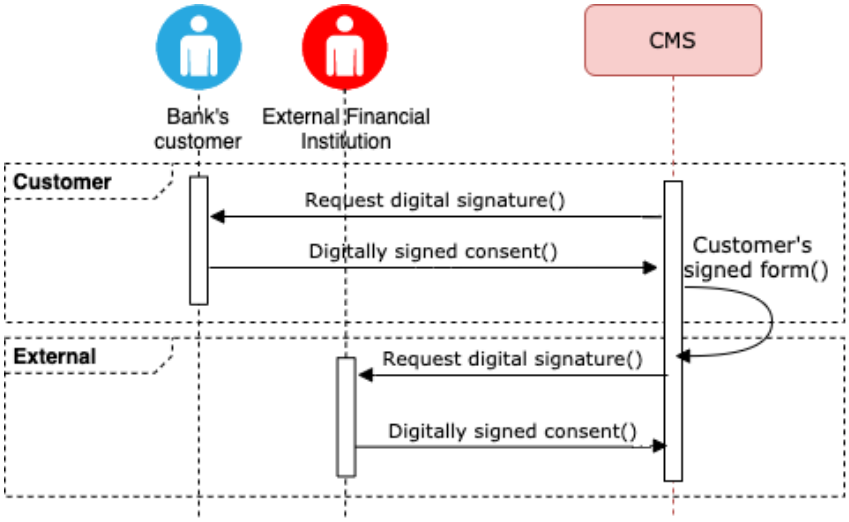

Figure 9: Use Case CMS-4 sequence diagram

4.1.3.5 Consent form is entered into blockchain

In Use Case CMS-5, the Consent Management System interacts with the blockchain components in order to write the new consent by executing the respective chaincode, encrypting the transaction and writing the new transaction in the ledger. In the case of "once off" consent, the Consent Management System initiates the internal timer. Finally, the Consent Management System stores the formulated consent form in the File Storage.

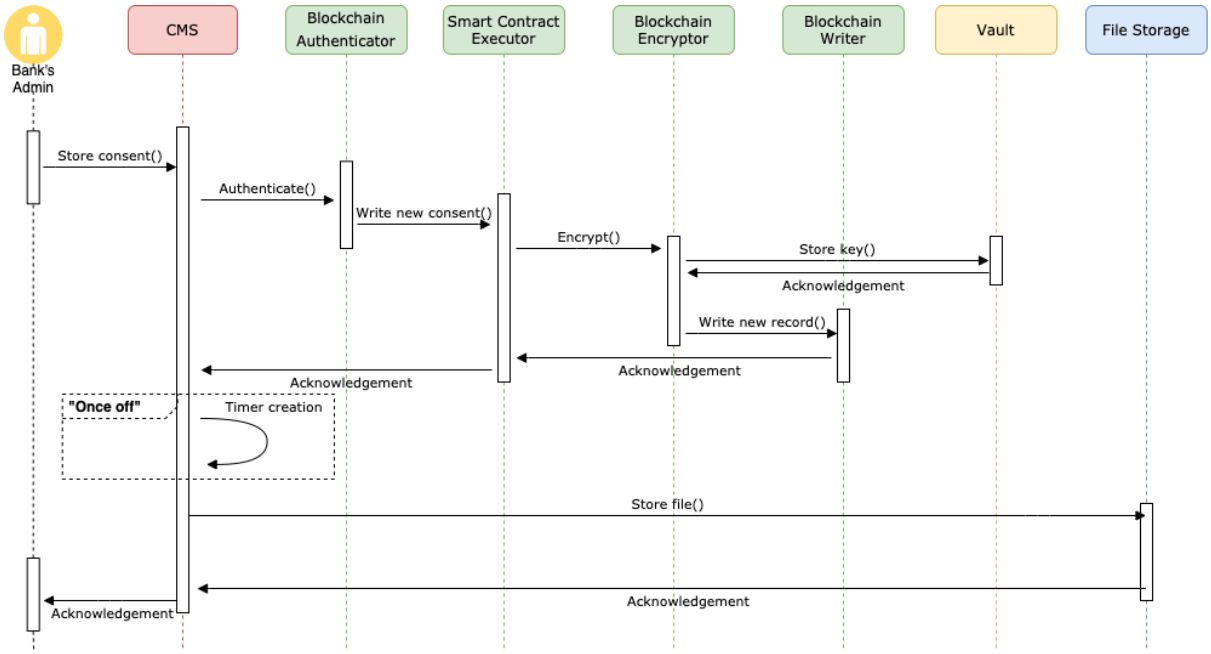

Figure 10: Use Case CMS-5 sequence diagram

4.1.3.6 Consent update or withdrawal

In Use Case CMS-6, the update or withdrawal at any time by the customer side of the consent is handled. In the case of an update, the Consent Management System orchestrates the execution of the sequence diagrams described for Use Case CMS-3 till Use Case CMS-5 in order to generate a new transaction based on the updated consent form that is digitally signed by both parties. In the case of

withdrawal, the Consent Management System interacts with the blockchain components in order to retrieve the existing consent from the ledger and update the consent status through a new transaction in the ledger that depicts the new status. Both cases are presented in the sequence diagrams below.

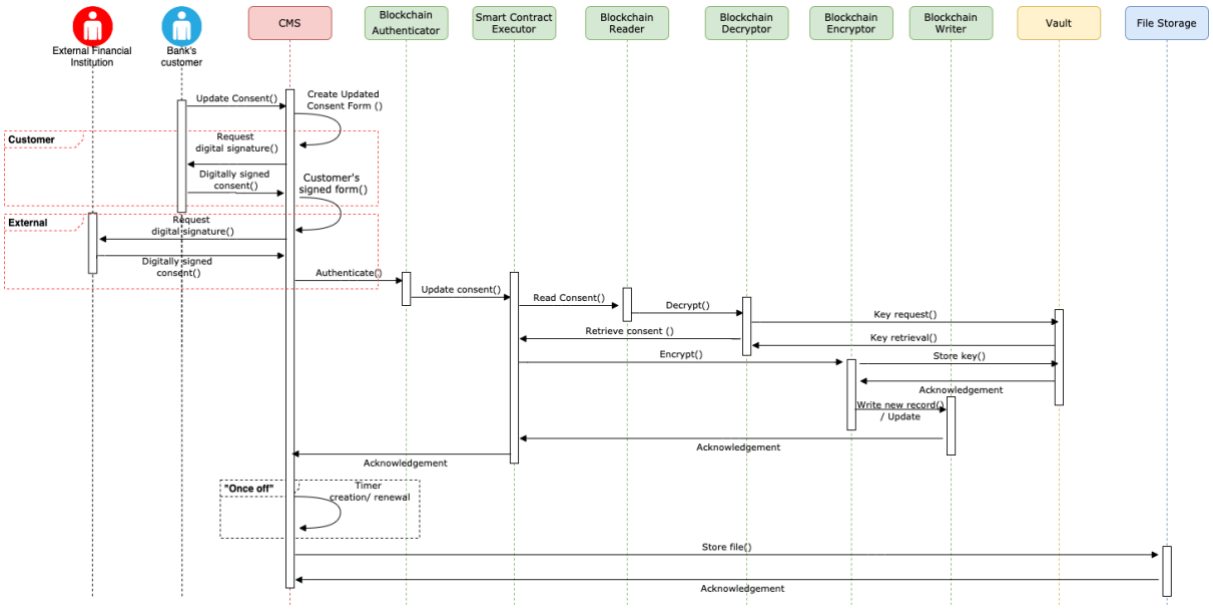

Figure 11: Use Case CMS-6 sequence diagram (update)

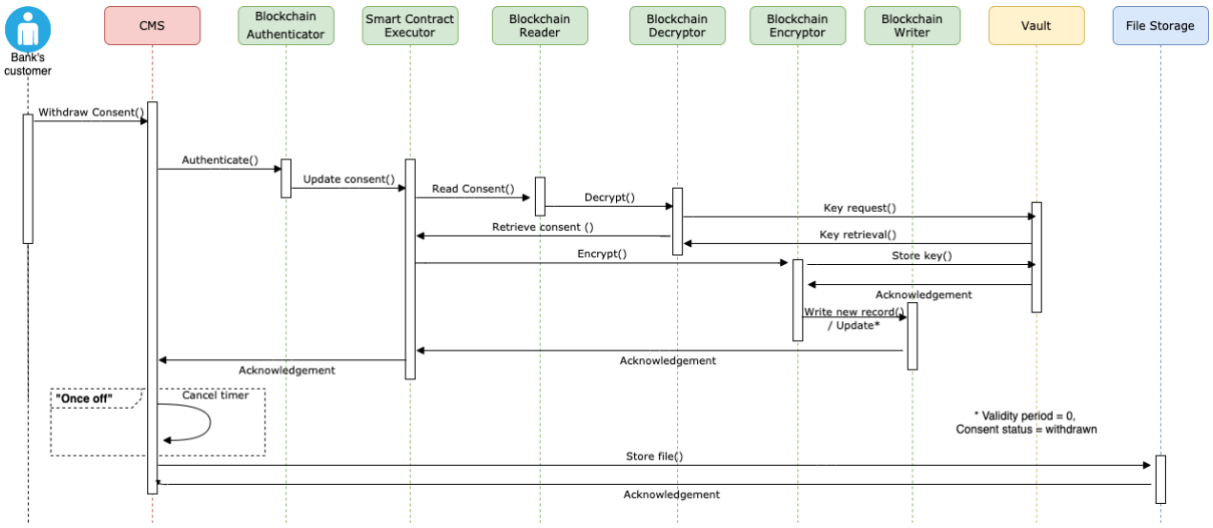

Figure 12: Use Case CMS-6 sequence diagram (withdrawal)

4.1.3.7 Access Control based on the consent forms

In Use Case CMS-7, the Consent Management System receives a new data access request and formulates an access control decision based on the existence of a valid and active consent. To achieve this, it interacts with the blockchain components in order to query the ledger for the existence of a consent between the involved parties for the specific data.

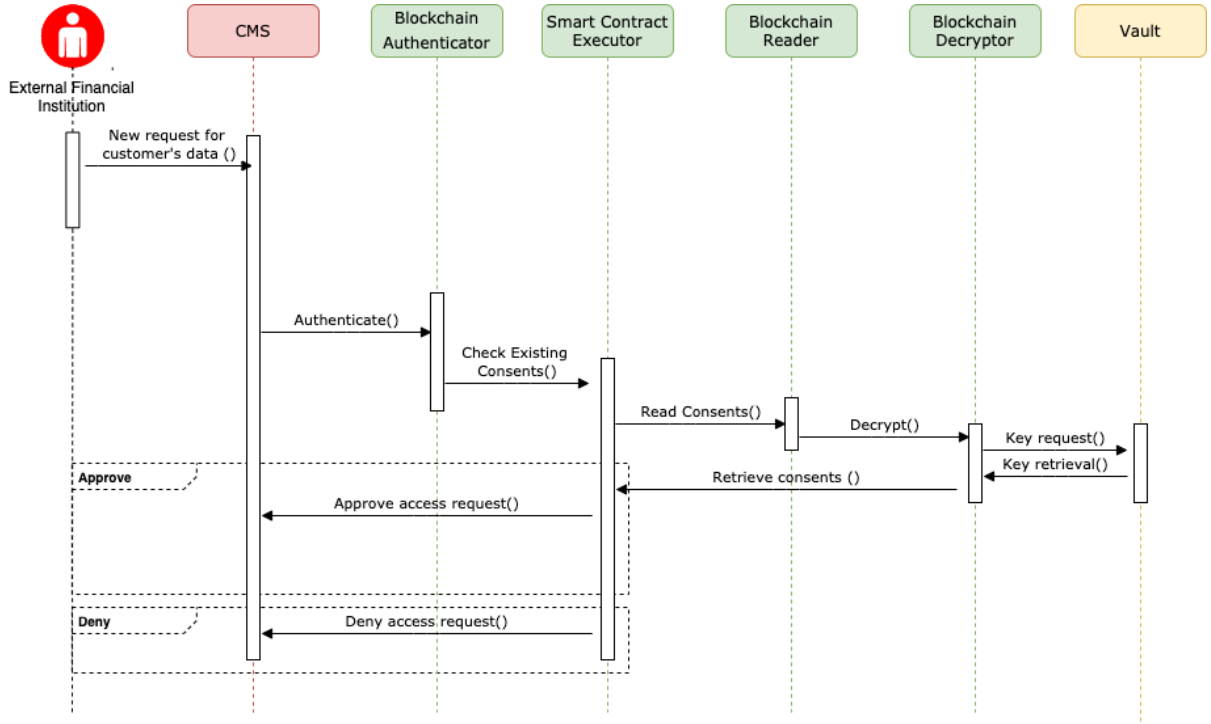

Figure 13: Use Case CMS-7 sequence diagram

4.1.3.8 Retrieve complete history of consents

In Use Case CMS-8, the Consent Management System receives a new request to retrieve all the consents that a customer has given to each specific recipient of his/her data along with their completed history. To achieve this, the Consent Management System with the blockchain components in order to query the ledger and retrieve the required information.

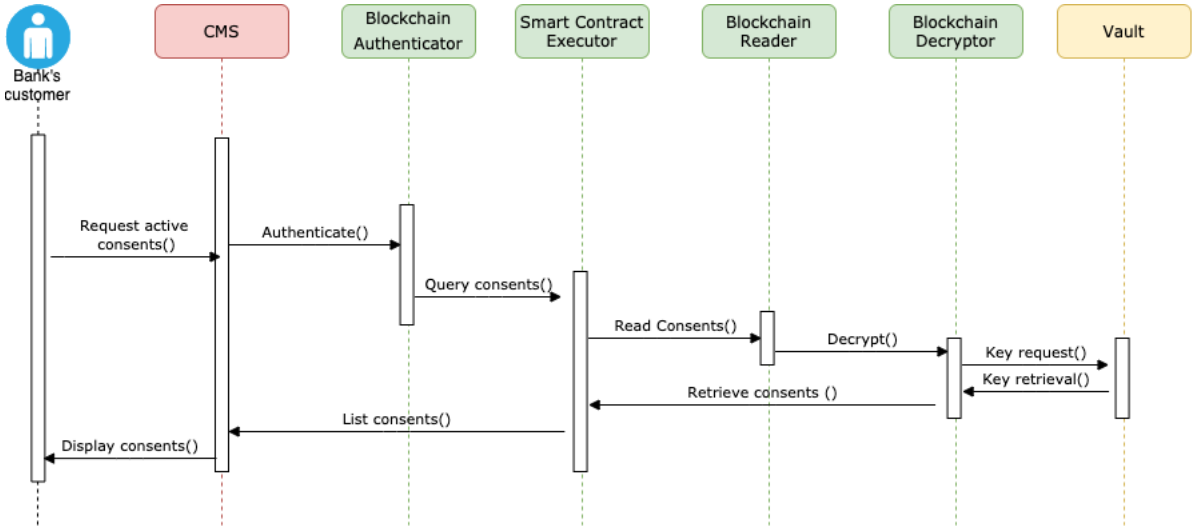

Figure 14: Use Case CMS-8 sequence diagram

4.1.3.9 Expiration of the validity period

In Use Case CMS-9, before the validity period of a "once off" consent expires, the Consent Management System informs the customer in order to initiate an update of the consent as described in Use Case CMS-6. In the case where the timer expires, the Consent Management System interacts

with the blockchain components to retrieve the consent and update the consent status via a new transaction.

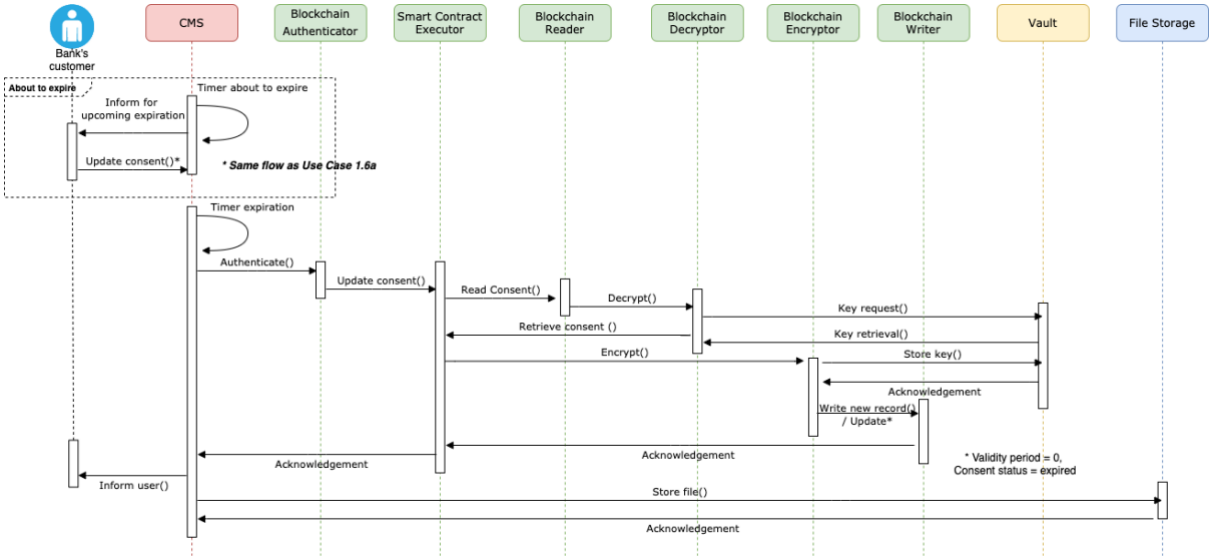

Figure 15: Use Case CMS-9 sequence diagram

4.2 Know Your Customer / Know Your Business

4.2.1 Description of the solution

The Know Your Customer (KYC) / Know Your Business (KYB) policies in state-of-the-art financial relations expect that customer parties, either individuals or entire corporations, endeavour verification of their identity. Thus, each financial organization is able to estimate the risks involved with sustaining a new business-customer partnership. In this context, together with the wider Finance field of Anti-Money Laundering (AML) procedures, every financial institution establishes KYC and KYB operations at the time they register a new customer [12].

The KYC/KYB blockchain application resolves the implementation of such industry mechanisms by utilizing blockchain technology as a basic background infrastructure. Security, immutability and controlled transparency rule inside large enterprise blockchain networks while they offer efficiency of tasks and effectiveness of transactions within the corporation and its members. Ultimately, blockchain technology simplifies the emerging use cases and directly addresses any possible issues that are created.

Particularly, a KYC/KYB mechanism ensures that the identification and verification of a customer occurs against national and international regulations and laws set by governments, commissions, central banks and financial associations. As both the customer profile information and the relevant laws and rules are subject to changes over time, their update and maintenance become complicated. Moreover, their centralized systems are exposed to data protection and cyber-security risks, which become cheaper to launch while they are led by more sophisticated adversaries year by year [13].

Blockchain technology and particularly permissioned blockchain networks are capable of providing security to the KYC and KYB processes through decentralization. The concept of decentralization mainly exploits the idea that the information is replicated across all network nodes, while sabotaging one or more nodes cannot harm the information integrity and a single point of failure is avoided. In particular, the permissioned blockchain technology promises to keep that sensitive information inside

a private network where only privileged parties can access it with an insider invitation. Thus, the customer information is kept safe on a private ledger that offers transparency to a privileged group of legal network participants. Both the customer and the organization are able to perform create, read, write, delete (CRUD) operations on the data under pre-defined access control policies. The various features of permissioned blockchains enable different policies applications that are able to, for instance, separate legal parties into a higher privacy network running inside the initial private one. Improved privacy control and data immutability rule inside the aforementioned technological scenario while they ensure legitimate customer data protection and management together with proper administration of this data by financial enterprises [14].

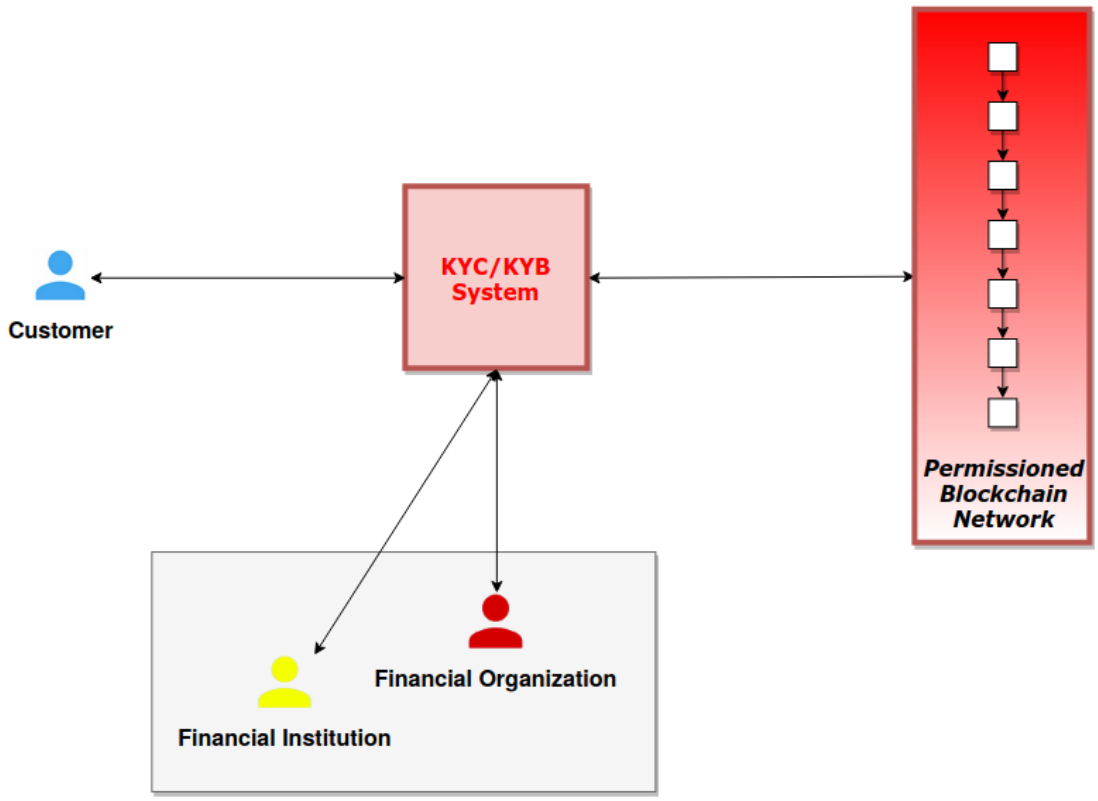

Figure 16: High-level architecture of the KYC/KYB solution

Figure 16 depicts a high-level architecture of the proposed solution. In general, the customer participant (light blue actor), being either an individual or an entire organization, needs to acquire financial services that are offered by the financial institution (yellow actor). For the completion of this transaction, the financial institution requests that the customer identity information is documented in KYC/KYB data after it is legally verified. In that case, the customer participant uploads their KYC/KYB documentation to the KYC/KYB System that interacts with the permissioned blockchain network and stores this information on-chain. When another financial organization (red actor) requires to start business relations with the same customer (light blue actor), the latter's KYC/KYB information is easily and time-efficiently obtained through the access rights provided by the financial institution (yellow actor) which is already having business partnerships with this customer. It is important to clarify that the customer using this kind of technology has declared their consent of sharing their information among privileged participants of the network, while their information privacy is guaranteed. In this system, data security and efficient data management and maintenance rule, for the united underlying blockchain infrastructure with common data structures offers stability, security, sustainability and time-efficiency.

The design specifications of any blockchain application is tightly related to the definition of the business objects for which the relevant functions of the chaincode will be implemented towards the maintenance and update of their current and historical state. In this sense, to facilitate the implementation of the KYC/KYB application the initial data schema has been defined. The initial data schema is depicted in Figure 17, while the details of all the entities are documented in tables Table 18 and Table 19.

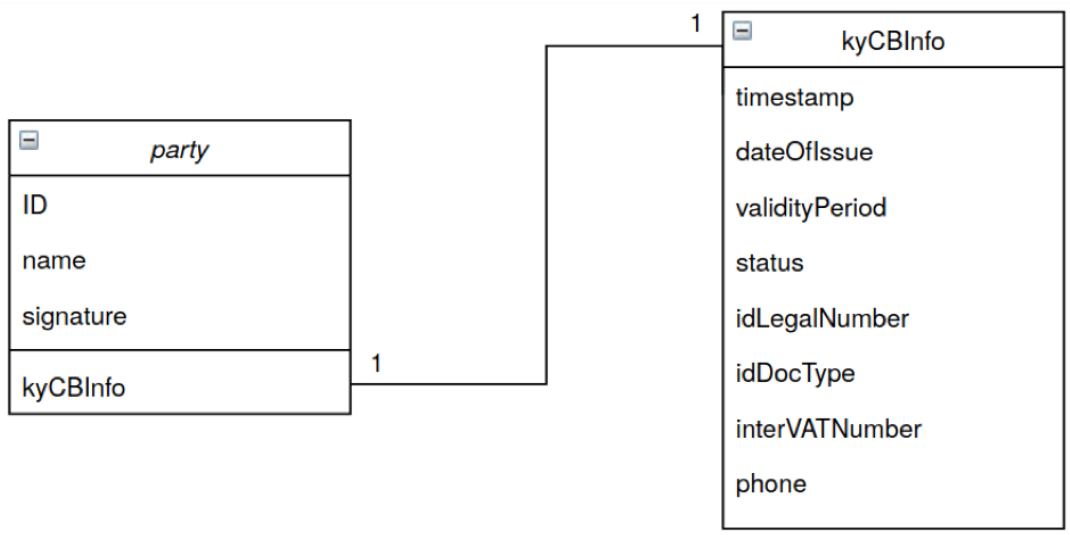

Figure 17: KYC/KYB Data Schema

Table 18: KYC/KYB Data Schema (1)

party		
Name	**Type**	**Short description**
ID	String	REQUIRED: The identification number used to uniquely identify network participants.
name	String	REQUIRED: The name of the participant.
signature	String	REQUIRED: The digital signature of the participant.
kyCBInfo	String	REQUIRED: The KYC/KYB pointer to the participant's KYC/KYB record.

Table 19: KYC/KYB Data Schema (2)

kyCBInfo		
Name	**Type**	**Short description**
kyCBInfo	Integer	REQUIRED: The KYC/KYB documentation pointer number.
timestamp	String	REQUIRED: The date the KYC/KYB documentation is modified.
dateOfIssue	String	REQUIRED: The date the KYC/KYB documentation is first accepted.
validityPeriod	String	REQUIRED: The period until which the KYC/KYB information is valid.

status	Boolean	REQUIRED: The validity of the KYC/KYB documents (either valid or invalid).
idLegalNumber	String	REQUIRED: The alphanumeric number of the legal identity of the participant.
idDocType	String	REQUIRED: The document type with which the party is legally approved.
interVATNumber	String	REQUIRED: The international VAT identification number of the party in alphanumeric number form.
phone	Integer	REQUIRED: The communication phone to contact a participant representative.

Taking into consideration the above data schema, the main functions that are foreseen on a component-level are the following:

- Blockchain Reader:
  - *ReadInfo():* The function fetches the KYC/KYB information of a party from the ledger, based on the pointer number (kyCBInfo) of each specific KYC/KYB information.
- Blockchain Writer:
  - *WriteInfo():* The function writes the KYC/KYB information of a participant on the ledger.
- Smart Contract Executor:
  - *Contract_SubmitTransaction():* The function is responsible for initiating the execution of the smart contract for the on-chain recording of a new KYC/KYB information.
  - *Contract_EvaluateTransaction():* The function is responsible for initiating the execution of the smart contract for the retrieval of KYC/KYB documents of a party.
  - *Contract_ApproveAccess():* The function is responsible for initiating the execution of the smart contract for the access approval of the KYC/KYB documents of a party from one financial organization to another.
- Blockchain Authenticator:
  - *Authenticate():* The function is responsible for the authentication of the user before the access to a specific channel is granted utilising the appropriate keys and certificates.
- Blockchain Encryptor:
  - *Encrypt():* The function is responsible for the encryption of the data utilising the AES 256 encryption before they are inserted into the ledger.
- Blockchain Decryptor:
  - *Decrypt():* The function is responsible for the decryption of the encrypted data that are fetched from the ledger.

4.2.2 Use cases

The following sections provide the detailed documentation of all the use cases encapsulated in this blockchain application describing in detail all information of each use case.

4.2.2.1 Use Case KYC/KYB-1: Assertion of customer identity documents

In this use case, the customer asserts their identity documents to the financial organization through the KYC/KYB process. In particular, a customer which represents either an individual or an entire corporation, acknowledges their KYC or KYB information in order to initiate business relations with the financial institutions and organizations of the network. In this context, each enterprise participant of the permissioned blockchain network ensures the legitimacy of their customer and, thus, new business relationships are established.

Table 20: KYC/KYB Use Case KYC/KYB-1

Stakeholders involved:	Customer
Pre-conditions:	1. A private blockchain network is set up.
Post-conditions:	1. The Customer's KYC/KYB information is stored on-chain.
Data Attributes	1. The Customer's KYC/KYB documents.
Normal Flow	1. The Customer's uploads their KYC/KYB documentation on the blockchain ledger. 2. The documentation information is successfully stored on-chain.
Pass Metrics	1. The documentation information is successfully stored on-chain.
Fail Metrics	2. There is no Customer to upload their KYC/KYB documents.

4.2.2.2 Use Case KYC/KYB-2: Read access to customer identification information

In this use case, as a legal network participant, a financial organization has *read access* to any customer's KYC/KYB documents information. In particular, inside the permissioned blockchain network, each party is able to read the corresponding ledgers. In this context, each financial organization has *read access* to the data that is stored on-chain. By a simple *read access* request, the financial organization can fetch the information of a customer they are interested in. Additionally, upon using the system, each customer approves that their data may be accessed by the financial organization they will initiate business relations with.

Table 21: KYC/KYB Use Case KYC/KYB-2

Stakeholders involved:	Financial organization
Pre-conditions:	1. The financial organization is a legal participant of the network.
Post-conditions:	1. The financial organization has read access control over the requested KYC/KYB information.
Data Attributes	1. The financial organization's network attributes. 2. The requested KYC/KYB documents.
Normal Flow	1. The financial organization requests for read access of a specific customer's KYC/KYB documentation. 3. The requested documentation information access is successfully granted.
Pass Metrics	1. The requested documentation information is successfully granted.
Fail Metrics	1. The financial organization is not eligible to access the requested documentation information.

4.2.2.3 Use Case KYC/KYB-3: Sharing of customer KYC/KYB documents

In this use case, the financial organization A shares the KYC/KYB document information of customer B with the financial organization C through the secure and private blockchain network. In particular, each of the financial organizations participating in the blockchain network is eligible for accessing the data stored on-chain. However, depending on the different access control rights granted by the time of joining the network, there exist the case where different organizations are qualified to access different data. In this context, together with initial customer consent, a financial organization may grant access to another organization or institution for a specific customer KYC/KYB documentation.

Table 22: KYC/KYB Use Case KYC/KYB-3

Stakeholders involved:	Financial organization
Pre-conditions:	1. The financial organizations A and C are legal participants of the network. 2. The requested customer B has submitted their KYC/KYB document information.
Post-conditions:	1. The financial organization C has read access control over the requested KYC/KYB information.
Data Attributes	1. The financial organizations' A and C network attributes. 2. The requested KYC/KYB documents.
Normal Flow	1. The financial organization A requests for sharing of a specific customer's KYC/KYB documentation with a financial organization C. 2. The requested documentation information access is successfully granted.
Pass Metrics	1. The requested documentation information is successfully granted.
Fail Metrics	1. The financial organization A is not eligible to share the requested documentation information.

4.2.3 Sequence Diagrams

In the previous section all the relevant use cases of the designed blockchain application were documented. The following sections present the sequence diagrams for each use case, depicting the interactions between the stakeholders and the components of the designed solution, as well as the interactions between the various components of the designed solution.

4.2.3.1 Assertion of customer identity documents

In Use Case KYC/KYB-1, a customer provides their documentation information in order to be eligible for business partnerships with the participating financial organizations. In this use case, a customer actor, which is either an individual or an entire enterprise, acknowledges the requested KYC or KYB documents in the KYC/KYB System by uploading them. This action is translated to an internal request that propagates the information inside the blockchain network, though the different blockchain components. Finally, the data is safely stored on-chain, i.e. on the private ledger, and further actions and use cases can emerge afterwards (see Use Case KYC/KYB-2 and Use Case KYC/KYB-3).

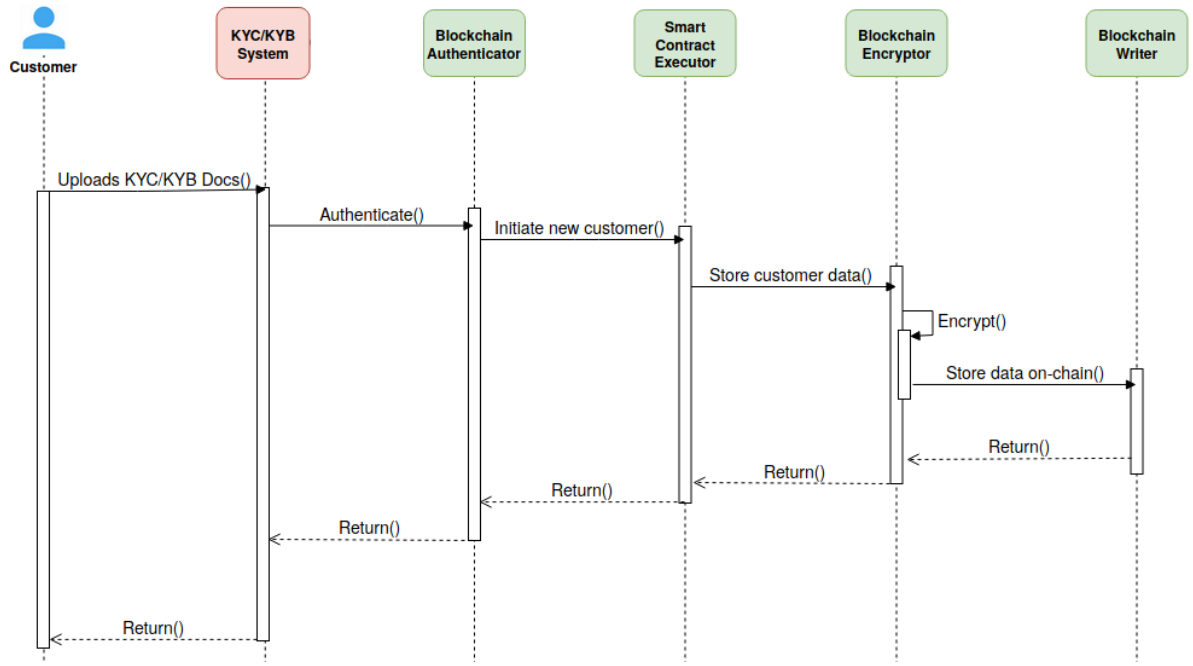

Figure 18: Use Case KYC/KYB-1 sequence diagram

4.2.3.2 Read access to customer identification information

In Use Case KYC/KYB-2, the blockchain network parties are constituted by financial organizations and are able to inspect the KYC/KYB documentation information of customers. Through the private and secure network, the different organizations obtain access to a customer's KYC/KYB submitted data in order to estimate whether to establish business relationships with them or not. The customer data submission is already accompanied by the customer's consent of using the KYC/KYB information by the legal parties of the network, i.e. the financial organizations. As in Figure 19, through the sequence of the blockchain components the *read access* is propagated to the financial organization that requested it.

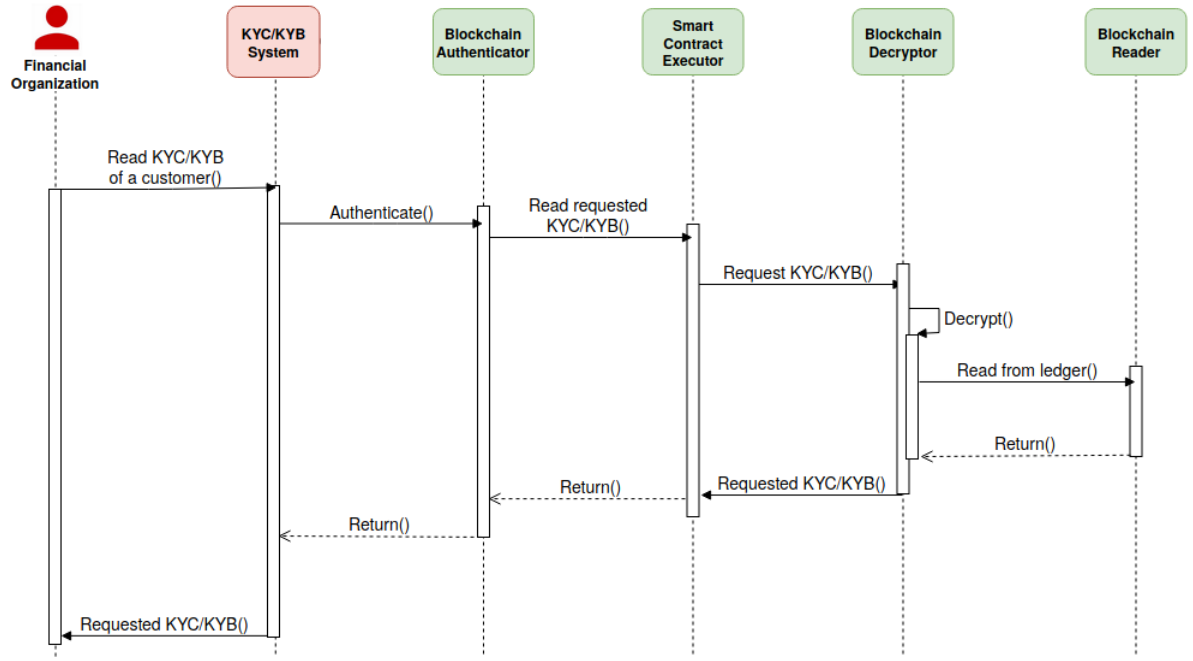

Figure 19: Use Case KYC/KYB-2 sequence diagram

4.2.3.3 Sharing of customer KYC/KYB documents

In Use Case KYC/KYB-3, a sharing of a customer (B) information among participant organizations (A and C) takes place. Organization C obtains customer's B information through the cooperation of organization A. Since the customer's data already exists inside the secure and private blockchain network, organization C can request it indirectly. Organization A responds to the request by granting *read access* of customer B information to organization C. In such an efficient system, the customer and the financial organizations benefit from this kind of sharing, since the customer avoids re-entering their KYC/KYB data to a different system of a different organization, while the financial organizations speedily obtain customer information and make decisions upon new business relations establishment.

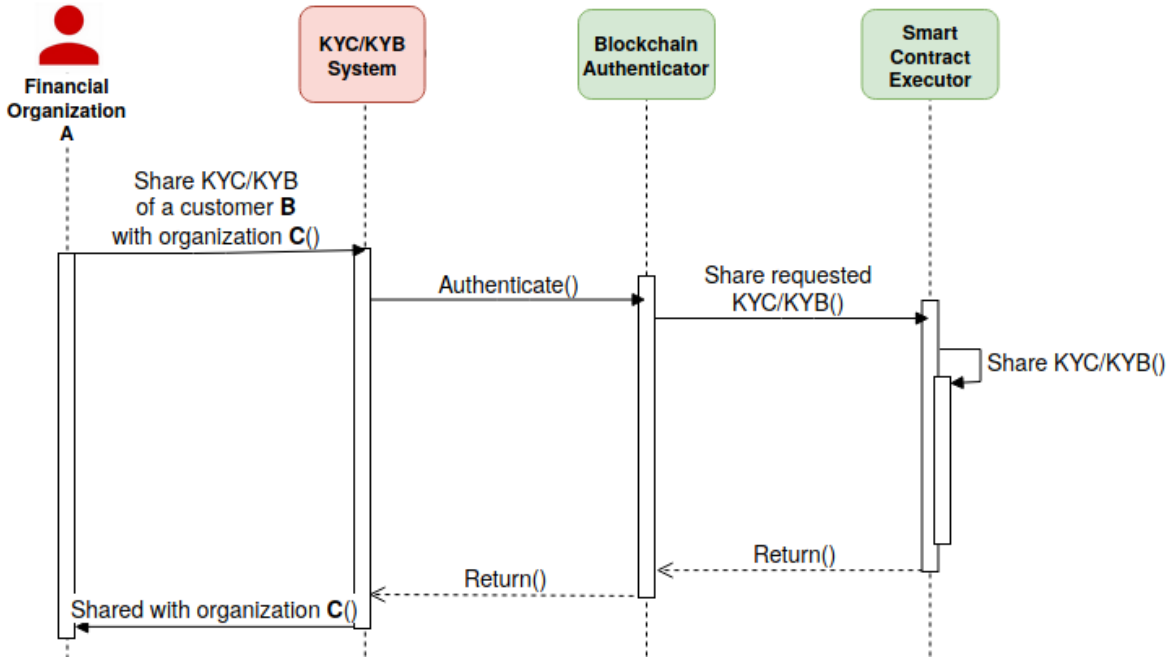

Figure 20: Use Case. KYC/KYB-3 sequence diagram

4.3 Tokenization

4.3.1 Description of the solution

One important aspect of any blockchain, whether public or private is asset tokenization. This refers to the representation of any physical asset into its digital form for trading which can later be bought, sold, exchanged or redeemed for any other digital or physical value. Assets can be anything tangible or intangible and can vary from luxury wines to gold or data, as well as fiat money such as USD and EUR called stable coins.

Representing assets as tokens allows using the blockchain ledger to establish the unique state and ownership of an item, and the transfer of ownership using a consensus mechanism that is trusted by multiple parties. As long as the ledger is secured, the asset is immutable and cannot be transferred without the owner's consent.

Tokens can represent tangible assets, such as goods moving through a supply chain or a financial assets being traded. Tokens can also represent intangible assets such as loyalty points. Because tokens cannot be transferred without the consent of the owner, and transactions are validated on a distributed ledger. Representing assets as tokens allows reducing the risk and difficulty of transferring assets across multiple parties.

Generally speaking, the work on tokens will rely on the ERC-20 standard [15] from Ethereum for the tokenization of assets, which is the de-facto standard in many financial and other blockchain applications for fungible tokens. The specific blockchain application will focus on implementation of ERC20 as a chaincode on Hyperledger Fabric, further extending the demonstration with broader scope involving the use of tokens for data trading and extending the ERC20 implementation with additional functionality such as mintable tokens and the generalization of ERC20 to ERC1155 [16] multi-token standard which allows representation of combinations of fungible tokens and non-fungible tokens within the same contract.

Since the described use case is the outcome of the work performed in Task 4.4 and currently under formulation, the detailed documentation of the specific use case will be documented in detail in "D4.10 - Blockchain Tokenization and Smart Contracts – I" on M14 per the INFINITECH Description of Action.

5 The INFINITECH Blockchain Network

The cornerstone of every blockchain deployment is the underlying blockchain network in which the immutable transaction ledger is maintained by a distributed network of peer nodes. As explained in Section 2, within the context of the INFINITECH project, the Hyperledger Fabric will be exploited in order to provide the required permissioned blockchain network on which the distributed applications presented in Section 4 will be deployed.

The main elements of the blockchain network are as follows [17] :

- **Peer nodes**: Peers are the nodes that are formulating the blockchain network, hosting the distributed ledger(s) and the smart contract(s) (chaincode), as well as other services of the network.
- **Organisations**: The peers are owned by different organisations that they are contributing to the blockchain network as the members of the network. Each peer has a digital identity in the form of a digital certificate issued by the Certificate Authority owned by each organisation.
- **Channel:** A channel refers to the isolated logical structure of a collection of peers which establish a sub-network. In a channel, only its members can see the particular transactions of the specified ledger and can have access to the deployed smart contract (chaincode) as well as all the data being transacted. Each channel is regulated by a specific channel policy as defined by the participants of the channel and is completely isolated as they cannot communicate with other channels. Furthermore, each channel contains all the configurations of communication between the peers along with the list of peers and which peers are endorsing, anchor and leader peers.
- **Ledger:** The distributed shared ledger where the current and historical state of the facts are maintained. Each peer has a copy of the ledger that they share through a channel.
- **Smart Contract (chaincode):** The trusted distributed applications that contain the business logic of the system. They are responsible for all the interactions with the ledger and can be invoked by all the external applications that have access to them.
- **Ordering service:** The purpose of the ordering service, deployed on a peer or a set of peers, is to formulate the transactions into blocks and ensure the delivery of the blocks to the peers of the channel. Furthermore, it guarantees that the transactions are included in the proper order and that all peers have the same updated ledger. The most common ordering services are Solo, Kafka, and Raft. Just like peers, ordering nodes belong to an organization. And similarly to peers, a separate Certificate Authority should be used for each organization.
- **Membership Services Provider (MSP):** The MSP acts as a Certificate Authority (CA) that issues and manages the certificates utilised in order to authenticate the identity and roles of the members of the network. As Fabric implements permissioned blockchain networks, all participants of the network must be authenticated in order to perform any kind of operation. Everything that interacts with a blockchain network, including peers, applications, administrators and orderers, acquires their organizational identity from their digital certificate and their Membership Service Provider (MSP) definition.
- **Network Policy:** The blockchain network is created and regulated by a network policy that is applied across the whole network with permissions that are determined prior to the network creation by the involved organisations.

To facilitate the implementation of the trusted distributed applications that were presented in Section 4, the consortium designed the suitable blockchain network that is capable of hosting all three applications that will developed in the context of Task 4.3, as well as the tokenization use case that will be developed within the context of Task 4.4. It should be noted that the designed network can be

easily adjusted and further expanded in the case where more applications will be designed and developed as the project evolves.

Figure 21 depicts the initial testbed topology of the Hyperledger Fabric network. In total, eight (8) nodes will be set up in order to serve all the applications. Four (4) out of eight (8) nodes will be used purely as the foundation of the blockchain network, hosting the ledger, the chaincodes and the peers together with the single orderer that will be used initially, while the other four (4) will be mainly hosting the external applications that interact with the blockchain network and they will also be providing resources for the required certificate authorities.

In particular, each of the eight (8) nodes will correspond to a single VM, which interacts with the entire blockchain network N. Three (3) organizations (FO1, FO2, FO3) will be created in total, with two peers for FO1 (P1, P4) and one peer for FO2 and FO3 (P2, P3) respectively, while nodes 1, 2, 3 and 4 will be hosting one peer each as well. In this initial topology of the network, there will exist four (4) channels (C1, C2, C3, C4) with their own copy of the ledger (L1, L2, L3, L4). While (P1, P2, P3) will be added in all three channels and will be able to deploy smart contracts inside the first three channels (S1, S2, S3), P4 will be added in the C4 channel and will be able to deploy smart contracts inside the corresponding fourth channel (S4). Finally, in node 1, the single orderer (O) will be included.

Regarding nodes 5, 6, 7 and 8, they will be mainly hosting the external applications as well as a Certificate Authority each (CA1, CA2, CA3). Besides Certificate Authority CA1, node 5 will contain the entire KYC/KYB external application, which will be interacting with channels C1 and C2. Two distinct KYC/KYB applications will be developed in order to operate with the two distinguished channels C1 and C2. In a similar manner, besides Certificate Authority CA2, node 6 will be hosting the whole Tokenization external application, which will be interacting with channel C3. The external application for the Consent Management will be hosted in node 7, which will include CA1 and will be interacting with channel C4. Finally, the external application for the GDPR will be developed in node 8, which will include CA3.

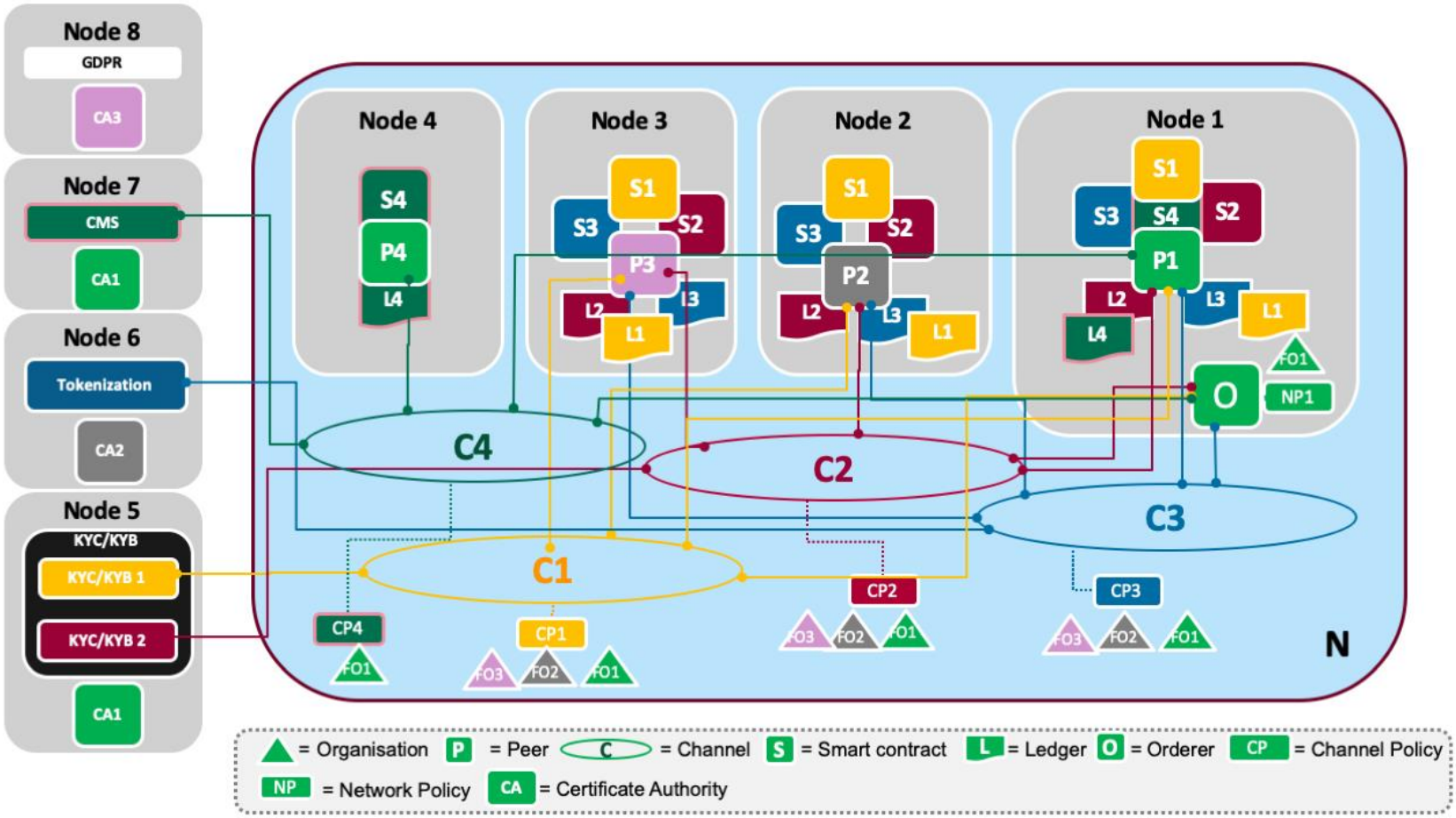

Figure 21: The INFINITECH Blockchain network

Regarding the node interactions, the deployment and the network topology, the following table illustrates the number of nodes, the hosted services and the overall hardware resources needed per node.

Table 23: INFINITECH Blockchain network details

Node Name / Service	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
Orderer	Yes							
Peer	Yes	Yes	Yes	Yes				
CA					Yes	Yes	Yes	Yes
KYC/KYB					Yes			
Tokenization						Yes		
CMS							Yes	
GDPR								Yes
Hardware Resources	2 vCPUs / 4GB RAM / 120 GB Disk	2 vCPUs / 4GB RAM / 100 GB Disk	2 vCPUs / 4GB RAM / 100 GB Disk	2 vCPUs / 4GB RAM / 100 GB Disk	4 vCPUs / 8GB RAM / 220 GB Disk	4 vCPUs / 8GB RAM / 220 GB Disk	4 vCPUs / 8GB RAM / 220 GB Disk	4 vCPUs / 8GB RAM / 220 GB Disk
Total Hardware Resources	22 vCPUs, 48 GB RAM, 1.32 TB Disk storage							

6 Baseline Technologies and Tools

The development of the presented applications is based on a set of carefully selected technologies and tools that will facilitate the development teams to deliver the described functionalities and use cases. Towards this end, the consortium decided to exploit a set of well-established technologies and tools that includes open-source software, libraries and frameworks which will enable the smooth and effortless implementation, as well as integration, of the designed use cases. The criteria during the selection process were their level of maturity, their applicability to the designed use cases and the compatibility of each selected technology and tool with the rest ones.

The following table presents the list of core technologies and tools that will be leveraged during the implementation phase of the presented applications.

Table 24: Baseline Technologies and Tools

Software Name	Short description	Version
Hyperledger Fabric	The open source enterprise-grade permissioned distributed ledger technology (DLT) platform.	2.2.0
Fabric - CA	The Hyperledger Fabric CA is a Certificate Authority (CA) for Hyperledger Fabric that will be used to authorise the various components and users.	1.4.8
Vault	Vault provides high-level policy management, secret leasing, audit logging, and automatic revocation to protect sensitive information	1.5.3
Consul	The Consul storage backend is used to retain Vault's data in Consul's key-value store	1.8.0
PostgreSQL	PostgreSQL is the RDBMS that will be utilised to store the certificates information by the Fabric CA	12.0

7 Conclusions

The purpose of the deliverable at hand entitled "D4.7 - "Permissioned Blockchain for Finance and Insurance - I" was to report the preliminary outcomes of the work performed within the context of T4.3 "Distributed Ledger Technologies for Decentralized Data Sharing" of WP4. To this end, the deliverable documented the initial specifications of the INFINITECH blockchain network which constitutes an integral part of the INFINITECH platform, the proposed new capability that will be introduced in the blockchain, as well as the detailed design specifications of the blockchain applications that will be implemented on top of the INFINITECH blockchain network which will address core needs and requirements of the financial and insurance sector.

At first, an in-depth analysis of the key characteristics and offerings of the blockchain technology was presented. Within this analysis the key characteristics were presented and the main components of the blockchain technology were documented focusing on the scope and context of each component within the blockchain technology. Moreover, the different approaches on the implementation of the blockchain technology were presented explaining their main differences and their application in different use cases. The results of this analysis were utilised in the definition of the role of the blockchain technology within the INFINITECH RA, in conjunction with the differences in the approaches, their relevance and their applicability to the requirements of the financial and insurance sectors.

From this analysis, the need to introduce a new capability in the blockchain platform which will address the needs of the financial and insurance sectors towards the compliance with EU GDPR, has been identified. To this end, a new approach was proposed aiming to provide such functionality "horizontally" at the level of the blockchain platform. To this end, a high-level overview of the proposed solution and the design specifications behind this proposed solution were elaborated.

Furthermore, the deliverable presented the initial design specifications of the blockchain applications that will be implemented within the context of the INFINITECH project. In total two different application were presented, namely the Consent Management and the Know-Your-Customer (KYC) / Know-Your-Business (KYB). For each application, a comprehensive description of the addressed business operation was presented, highlighting the key functionalities of the blockchain technology that are leveraged and presenting the high-level architecture of the application. Moreover, the detailed documentation of the use cases that the designed application addresses were presented for all applications. Finally, the sequence diagrams that depict the interactions between the stakeholders and the involved components were presented, for both the Consent Management and the Know-Your-Customer (KYC) / Know-Your-Business (KYB) applications.

In accordance with the role of the blockchain technology within the INFINITECH RA and the design specifications of the blockchain applications, the design details of the INFINITECH blockchain network were presented. The network topology of the designed INFINITECH blockchain network was presented, describing in detail the role of each node in the network, the interactions between the nodes of the network, as well as the list of services on each node and the required hardware resources. In total, the INFINITECH blockchain network is composed by eight nodes, with four of them being used for the blockchain network (peers) and the rest four nodes hosting the external applications that interact with the blockchain network and the certificate authorities. Within the INFINITECH blockchain network, three organizations are created that are interacting via four different channels, three different smart contracts (chaincode) deployed for the four different ledgers that are hosted by the four peers.

Finally, the list of baseline technologies and tools that will be exploited for the deployment of the INFINITECH blockchain network, as well as the development of the designed blockchain applications, was presented. The list was composed by the consortium in order to facilitate the implementation phase of the aforementioned and during the selection phase, multiple criteria such as their level of maturity, applicability to the designed use cases and compatibility with the rest of the technologies were taken into consideration.

It should be stressed out at this point that the current deliverable presents the first version of the INFINITECH blockchain network specifications and the design specifications of the blockchain applications. The outcomes of this deliverable will drive the implementation activities that will be performed within the context of T4.3. However, as the project evolves and since the definition of the design specifications of both the blockchain network and the blockchain applications is a living process that will last until M27, new requirements may arise as the result of the feedback that will collected by the pilots of the project and the stakeholders of the platform. Hence, the design specifications documented in this deliverable are subject to changes that will be documented in the upcoming versions of the deliverable which will be released on M20 and M27 with deliverables D4.8 and D4.9.

8 Appendix A: Literature

[1] D. Yaga, P. Mell, N. Roby and K. Scarfone, Blockchain technology overview, National Institute of Standards and Technology, 2018.

[2] P. Treleaven, R. Gendal Brown and D. Yang, "Blockchain Technology in Finance," *Computer,* vol. 50, no. 9, pp. 14-17, 2017.

[3] M. Niranjanamurthy, B. N. Nithya and S. Jagannatha, "Analysis of Blockchain technology: pros, cons and SWOT," *Cluster Computing,* vol. 22, no. S6, pp. 14743-14757, 2018.

[4] M. Casey, J. Crane, G. Gensler, S. Johnson, N. Narula and C. A. Wyplosz, The impact of blockchain technology on finance, Geneva: International Center for Monetary and Banking Studies (ICMB), 2018.

[5] "Hyperledger Fabric – Hyperledger," 2020. [Online]. Available: https://www.hyperledger.org/use/fabric. [Accessed 5 September 2020].

[6] "Introduction — hyperledger-fabricdocs master documentation," Hyperledger, 2020. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/whatis.html. [Accessed 03 September 2020].

[7] "EU data protection rules," European Commission, 2020. [Online]. Available: https://ec.europa.eu/info/law/law-topic/data-protection/eu-data-protection-rules_en. [Accessed 27 August 2020].

[8] D. Deuber, B. Magri and S. A. K. Thyagarajan, "Redactable Blockchain in the Permissionless Setting," *2019 IEEE Symposium on Security and Privacy (SP),* pp. 124-138, 2019.

[9] G. A. Stevens, B. Magri, D. Venturi and E. Andrade, "Redactable Blockchain – or – Rewriting History in Bitcoin and Friends," *2017 IEEE European Symposium on Security and Privacy (EuroS&P),* pp. 111-126, 2017.

[10] "Consent Receipt Specification – Kantara Initiative," Kantara Initiative, 2020. [Online]. Available: https://kantarainitiative.org/download/7902/. [Accessed 1 September 2020].

[11] "Supporting Material - Archived Groups - Kantara Initiative," Kantarainitiative.org, 2020. [Online]. Available: https://kantarainitiative.org/confluence/display/archive/1+-+Supporting+Material. [Accessed 2 September 2020].

[12] N. Kapsoulis, A. Psychas, G. Palaiokrassas, A. Marinakis, A. Litke and T. Varvarigou, "Know Your Customer (KYC) Implementation with Smart Contracts on a Privacy-Oriented Decentralized Architecture," *Future Internet,* vol. 12, no. 2, p. 41, 2020.

[13] A. Polyviou, P. Velanas and J. Soldatos, "Blockchain Technology: Financial Sector Applications Beyond Cryptocurrencies," *Proceedings,* vol. 28, no. 1, p. 7, 2019.

[14] I. Karagiannis, K. Mavrogiannis, J. Soldatos, D. Drakoulis, E. Troiano and A. Polyviou, "Blockchain Based Sharing of Security Information for Critical Infrastructures of the Finance Sector," *Computer Security Lecture Notes in Computer Science, Computer Security. IOSEC 2019, MSTEC 2019, FINSEC 2019,* vol. 11981, pp. 226-241, 2020.

[15] "EIP-20: ERC-20 Token Standard," 2020. [Online]. Available: https://eips.ethereum.org/EIPS/eip-20. [Accessed 1 September 2020].

[16] "EIP-1155: ERC-1155 Multi Token Standard," Ethereum Improvement Proposals, 2020. [Online]. Available: https://eips.ethereum.org/EIPS/eip-1155. [Accessed 10 September 2020].

[17] "Blockchain basics: Hyperledger Fabric," IBM Developer, 2020. [Online]. Available: https://developer.ibm.com/technologies/blockchain/articles/blockchain-basics-hyperledger-fabric/. [Accessed 29 August 2020].