Critical-Chains

Collaborative Project

Project Start Date 1st July 2019 Duration 36 Months

Deliverable D3.11

Joint report on conformance analysis, linking, mapping and synchronisation

Published by the Critical-Chains Consortium

Version 1.5 Date 28-06-2022

Project Coordinator: Professor Atta Badii (University of Reading)

Dissemination Level: Consortium Only

Work Package Task: WP3

Document Responsible: NETAS

Contributors: NETAS, EY, GT

Status: Final

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Abstract

This deliverable, D3.11 “Joint Report on Conformance Analysis, Linking, Mapping, and Synchronisation”, is concerned with the design and development of the integrating components of the Blockchain-based infrastructure for the Critical-Chains solution stack. As such, this deliverable is focused on the validation of the developed services, applications, and infrastructure and their compliance with the targeted deployment context.

Firstly, it addresses the synchronisation of the transactions in the blockchain environment over certain development iterations. Subsequently, it examines the mechanisms that govern Blockchain-based synchronisation of the participating nodes and arrives at the integrity mechanism to be deployed as a protocol across the framework. The linking and mapping cover the overall technical positioning of each mechanism of the Blockchain infrastructure. Moreover, smart contracts have been explored and new privacy-preserving attribute-based credentials mechanism integrated with the Critical-Chains blockchain. This mechanism has been developed to enable users to easily impose a fine-grained access control policy for their data while respecting multiple security policies, accountability, and transparency. Accordingly, the deliverable has conducted a series of tests and activities that ensured a well-defined process of development of blockchain interoperability mechanisms between the various components of the solution.

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Deliverable D3.11 Document History

Versioning			
Version Number	**Date**	**Contributing Partners**	**Changes**
V0.1	Through February	NETAS	Initial deliverable structure developed and shared with the contributors.
V0.2	03.05.2021	GT	REST standards, Blockchain and KSI standards provided. Diagrams for 3.3.2 and 3.3.3 explaining Quorum KSI integration.
V0.3	15.05.2021	NETAS	The reference contributions provided for the Conformance testing sections.
V0.4	9.06.2021	GT	Diagrams updated.
V0.5	15.06.2021	GT	3.3.1, 3.3.2, 3.3.3 explanation text added.
V0.6	23.06.2021	EY	EY contributions for Sec. 3 and 4 included.
V0.7	25.06.2021	EY	EY contributions for Sec. 5 included.
V0.8	29.06.2021	NETAS	Formatting polished.
V0.9	30.06.2021	All Contributors	Minor fixes contributed.
V1.0	05.07.2021	NETAS	The deliverable finalised
V1.1	07-08.07.21	UREAD	Minor edits throughout
V1.2	09.07.21	NETAS	Minor Fixes and re-compilation
V1.3	12.07.21	UREAD	Remaining edits and layout issues fixed
V1.4	14.07.21	NETAS	Further edits to make submission-ready
V1.4	14.07.21	UREAD	Final edits, quality check, spell-check and submit
V1.5	29.07.22	NETAS	Replacement of duagrams for enhanced quality
V1.5	30.07.22	UREAD	Final quality check and re- submit

Internal Review History

Internal Reviewer	Date	Comments
Kristo Klesment	02.07.2021	Feedback given for some sections.

External Review History

External Reviewer	Date	Comments
Atta Badii	07-08.07.21 & 12.07.21	Various minor edits needed, tabularisation lay-out issues etc. fixed and other edits recommended.

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Table of Contents

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Table of Figures

Figures:

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Table of Tables

<u>Tables:</u>

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

1 Executive Summary

This document, Deliverable 3.11, presents the progress made through Task 3.6 Conformance testing and Task 3.7 Linking, Mapping and Synchronisation. The deliverable focuses on the design and development of integrating components of the Blockchain-based infrastructure and services of the Critical-Chains framework. Accordingly, the deliverable presents a series of tests and activities to validate the compliance level of the developed services, applications, and the overall infrastructure.

The organisation of the deliverable is as follows:

Chapter 2: This section sets out the background highlighting the tasks of the Critical-Chains Main Framework as well as clarifying the scope of the analysis for this deliverable.

Chapter 3: This section focuses on the progress made through Linking, Mapping, and Synchronisation of the Blockchain components using a chronological approach. The first sub-chapter identifies the linking, mapping, and synchronisation of the first iteration of the development of the Blockchain network called Rinkeby. Moreover, this part focuses on the overall positioning of the Rinkeby Blockchain network with the Critical-Chains components. This is followed by the analysis that identifies the linking, mapping, and the synchronisation of the second iteration of the development with the Quorum Blockchain Network including the overall positioning of it with the respecting Critical-Chains components. The concluding part of this Chapter focuses on integrating components of the Critical-Chains target Blockchain-as-a-Service and presents the overall linking, mapping, and synchronisation of the two inner components, mainly the Keyless Signature Infrastructure (KSI) and Quorum Blockchain Network.

Chapter 4: This section focuses on the newly adopted Blockchain key mechanism that provides a holistic asymmetric key solution for the blockchain-based operations. This section firstly identifies the overall positioning of the solution over Critical-Chains Framework and then the synchronisation of the keys to replace the previously adopted, less secure wallet solutions. Moreover, this section highlights the linking and mapping between the Authentication service and the newly adopted blockchain-key solution.

Chapter 5: This section focuses on conformance testing which is to ensure that the Critical-Chains web applications are developed according to a set of standards. The set of standards are categorised for functional and non-functional testing. Functional tests include GUI testing and regression testing which are used to certify the features of the web applications. Non-functional tests include, web service standards, smart contract standards and overall blockchain infrastructure testing. this section is completed by highlighting the errors that are detected during the conformance testing.

Chapter 6: This chapter sets out the conclusions of the deliverable and is followed by the References section which includes the detailed referencing of the sources consulted.

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

2 Introduction

The Project Objectives are to develop an integrated effective, accessible, fast, secure, and privacy-preserving solution for financial contracts and transactions. This is to protect against illicit transactions, illegal money trafficking and fraud that can take place through the banking clearing system and financial transactions settlement processes. Thus, the objectives of the project are in the public interest.

The technologies to be deployed consist of:

- transaction and financial data flow analytics and modelling of the financial transactions clearing and claim settlement processes.
- secure and smart use of Blockchain for data integrity checking, by involving financial institutions in the distributed Blockchain network.
- cyber security protection of Inter-Banks and Internet Banking, insurance, and financial market infrastructures.
- Privacy protection through secure access supported by embedded systems and Internet-of-Things (IoT) security.
- Critical-Chains is to be validated using four case studies aligned with four critical sectors: banking, financial market infrastructures, the insurance sector, and the Highway Toll collection. The validation will include evaluating system reliability, usability, user-acceptance, social, privacy, ethical, environmental, and legal compliance by scrutiny of the geo-political and legal framework bridging the European economy to the rest of the world. The Consortium represents a strong chemistry of relevant expertise and an inclusive set of stakeholders comprising end-users (customers), CERTS, the financial sector (Banks & CCPs) and the Insurance consultant.

2.1 Scope of the Deliverable

The scope of this deliverable is centred on the design and development of the integration components of the Blockchain-based infrastructure for the Critical-Chains solution stack. According to the infrastructure, as defined in WP3, the individual information systems belonging to each transactional entity will be plugged in as nodes of the Blockchain-enabled platform, having access to, and participating in, the distributed ledger. The general agreements of technical standardisation as specified by the overall architecture and selection of Integration Profiles, sets the technical interoperability standard to be followed for this integration.

Synchronisation will address the development of the mechanisms that govern the Blockchain-based synchronisation of the participating nodes as an enforced protocol across the distributed ledger. The infrastructure provided in WP3 covers aspects including access control, authentication and authorisation management, privacy-preserving user input and logging of data usage. In this task smart contracts and privacy-preserving attribute-based credentials will be explored to enable users to easily impose a fine-grained access control policy for their data, while respecting multiple security policies. This will enable efficient data sharing and synchronisation amongst transactional units and provide controllable traceability and accountability of the shared data.

Further, a series of tests and activities will be defined and implemented in order to determine if the developed services, applications and infrastructure are compliant with the innovation objectives. These activities will enable a streamlined process of development of interoperability mechanisms between the various components of the solution.

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

3 Linking, Mapping and Synchronisation of the Blockchain Core Components

In this section, the development iterations of Blockchain core components are listed to present the overall positioning of the various blockchain solutions over the Critical-Chains development duration. The first subsection presents linking, mapping, and synchronisation of the public blockchain network called Rinkeby. Subsequently, the second subsection presents the linking, mapping, and the synchronisation of the more secure, fast, and private blockchain solution namely Quorum Blockchain. The last subsection presents the integration and the overall positioning of the Critical-Chains blockchain solution containing the Keyless Signature Infrastructure and Quorum Blockchain tailored for Critical-Chains.

3.1 1st Iteration: Overall Positioning of Rinkeby Blockchain Network

The Critical-Chains first phase pilot use-cases highly depended on the complex business logic that requires multiple parties' approval or judgment of some cases (such as the Crop Insurance case). In this sense, the first iteration development of the blockchain-based applications required a platform empowered with the smart contracts which could contain and support complex business logic. Moreover, this so-called platform required a community-based secure digital wallet that could be integrated with the apps in a native sense and that could run on the client-side for scalability. Hence, the Critical-Chains first iteration (1st part development of Phase 1) development has been made with the Ethereum Public Blockchain Network, Rinkeby Blockchain Network to be precise.

In this sense, this section presents the reasoning of the usage of the Ethereum Platform and Rinkeby Network, overall linking and mapping of this network, and finally, the synchronisation of the positioned blockchain network with the Critical-Chains components and applications.

3.1.1 Linking and Mapping of the Critical-Chains Components with Rinkeby

3.1.1.1 Ethereum and Rinkeby Blockchain Network

Ethereum is a blockchain-based software platform that is primarily used to support the world's second-largest cryptocurrency by market capitalisation after Bitcoin namely Ether. Like other cryptocurrencies, Ethereum can be used for sending and receiving value globally and without a third party watching or stepping in unexpectedly. However, the Ethereum Platform is not limited to crypto transactions, it is more of a community-build technology platform that enables users to develop blockchain-based applications that could run over the Ethereum Virtual Machine. Moreover, with Ethereum, centralised servers are replaced by thousands of so-called "nodes" run by volunteers all over the world thus forming a "world computer." The hope is that one day, anyone in the world will be able to use it. It means that one could build applications that no one is in the charge of, in the sense of the data. The idea of Ethereum is to change how apps on the internet work today, awarding users more control by replacing intermediaries with smart contracts that execute rules automatically. (Coindesk 2020).

Smart contracts are tools that can automatically execute transactions if certain conditions are met without requiring the help of an intermediary company or entity. They are often associated with Ethereum, a blockchain that was designed to accommodate smart contracts, but the idea is not restricted to any platform or network. Smart contracts are made possible by blockchains, a network of computers that work together to enforce rules on the network without requiring the help of an intermediary. (Coindesk 2020). The fundamental reasoning of using Ethereum Platform in general is the Ethereum is the largest decentralised software app that helps to build smart contracts and decentralised applications without any downtime or

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

any third-party interference. As stated, Ethereum allows the developer to create and publish next-generation distributed applications based on an actual running blockchain network.

In the first iteration, the Rinkeby Blockchain Test Network has been used in the Ethereum Environment. Rinkeby is an Ethereum network (or test network). Test networks are typically used by developers to run "tests" for their application or software. Currency on test networks is valueless. The Rinkeby test network, unlike the Ethereum main network, is a proof-of-authority network, as opposed to a proof-of-work network such as the Ethereum main network. Some of the characteristics of the Rinkeby Network are:

- Proof of Authority (PoA) & (Clique)
- Supported by Geth, Besu, Nethermind, and OpenEthereum
- Chaindata size 6 GB (April 2018)
- PoA test network started by the Ethereum team. (Uses Clique PoA consensus protocol.)
- Started in April 2017. Named after a metro station in Stockholm.
- Immune to spam attacks (Ether supply and transactions are controlled by trusted parties with PoA)
- Supported by Geth only
- Does not fully reproduce the current production environment as it uses PoA.
- Ether cannot be mined. (It must be requested from a tap)
- Network id: 4
- Block time: 15 seconds

In terms of the Critical-Chains context, the Rinkeby Blockchain Test Network used since the network itself is an exact replicate of the Ethereum Main Network environment in terms of the computing, but the consensus algorithm is more suitable for the Critical-Chains context. It enables the developed systems to be tested in a real-time blockchain environment without any financial burden. Moreover, it has been a good first step in the sense of the Critical-Chains pilot applications in terms of the technical and user aspects. In this context, the blockchain transaction approval delays and confirmations have been studied and presented in the applications as in user-intimate interfaces. Accordingly, technical expertise has been gained with the network and smart-contracts since the conventional backend-based business logic switched to blockchain – *smart contract* – based business logic.

3.1.1.2 Overall Positioning of the Rinkeby Blockchain Network with Critical-Chains

The specification of the components focused on providing the fundamental functionality with the highest possible security while considering privacy-preservation in the first development iteration.

In this sense, the first component identified was the Authentication-as-a-Service which developed with Keycloak Authentication provider. The linking of the component has been made using the Critical-Chains web applications boilerplate project. Critical-Chains applications rely on the React-based projects and Node.Js-based backend system. It composes of a web server and the application beneath it. The linking of the Keycloak server has been made using the Keycloak JavaScript library and its corresponding API that enables direct linkage between the Authentication Server and the web server. The mapping of these two-components has been in the infrastructure layer of the cloud. The Keycloak Server deployment and application deployments have been made in different subnets placed under one virtual network secured with the Network Security Groups. The connections have been secured using the HTTPS connections and valid certificates including SSL (communications via TLS and HTTPS protocols).

Accordingly, the following linkage has been made for the interoperability and integration capability. For this matter, a set of RESTful web services were developed for various data operations. Endpoints have been connected both with the user interfaces in the frontend and the blockchain interfaces for secure yet operable

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

data registration, retrieval, and transactions. The mapping has been completed with the Frontend module, backend module, and the blockchain interfaces (Web3.js and Infura API).

In terms of the Blockchain component linking, initially, the smart contracts have been developed according to the business logic of the Pilot use-cases. Afterward, the smart contracts have been deployed to the Rinkeby Blockchain Network for the Ethereum Virtual Machine (EVM) to keep the meaningful code for the use-case usage. In simple terms when the expected certain set of rules are satisfied, the expected transactions are sent to the blockchain. Afterwards, some interfaces have been integrated and mapped with the business-specific smart contracts for the Critical-Chains applications to communicate securely via the applications. In this sense, the web3.js library and the corresponding API Infura have been used for the interoperability between the public blockchain system with the applications. The Infura API is powered by a cutting-edge micro-service-driven architecture that dynamically scales to support the APIs. It enables connection to Ethereum and IPFS via HTTPS and WebSocket, where request-response times are up to 20 times faster than other services and self-hosted solutions. In order to scale the connections, the Infura API has been enhanced. The overall interfaces provide a capability to convert the POST and GET operations of the REST architecture to SEND and CALL operations in which these methods are the only meaningful instructions that the blockchain system may understand since the system itself could only allow a data registration or data retrieval, none of the other operations are supported in the blockchain system.

Finally, in the first iterations, the Metamask wallet provider has been used so that Critical-Chains users could complete their transactions by signing them. Metamask enables the execution of the Ethereum operations, including identity management, signing transactions, and managing multiple accounts over multiple networks (including the Mainnet, Ropsten, Kovan, and Rinkeby). The wallet has been used via the HD Wallet Provider library. The Metamask wallet provides features as asymmetric key pairing and storing. Also, it introduces Web3 interface within the browser apps to complete the overall transaction life cycle within the blockchain environment. Moreover, it provides a suitable environment for various test networks to validate the functionality of Blockchain-based applications and enables users to conduct transactions through the browser which provides further stability and scalability since all the signing operations are carried out on the client-side.

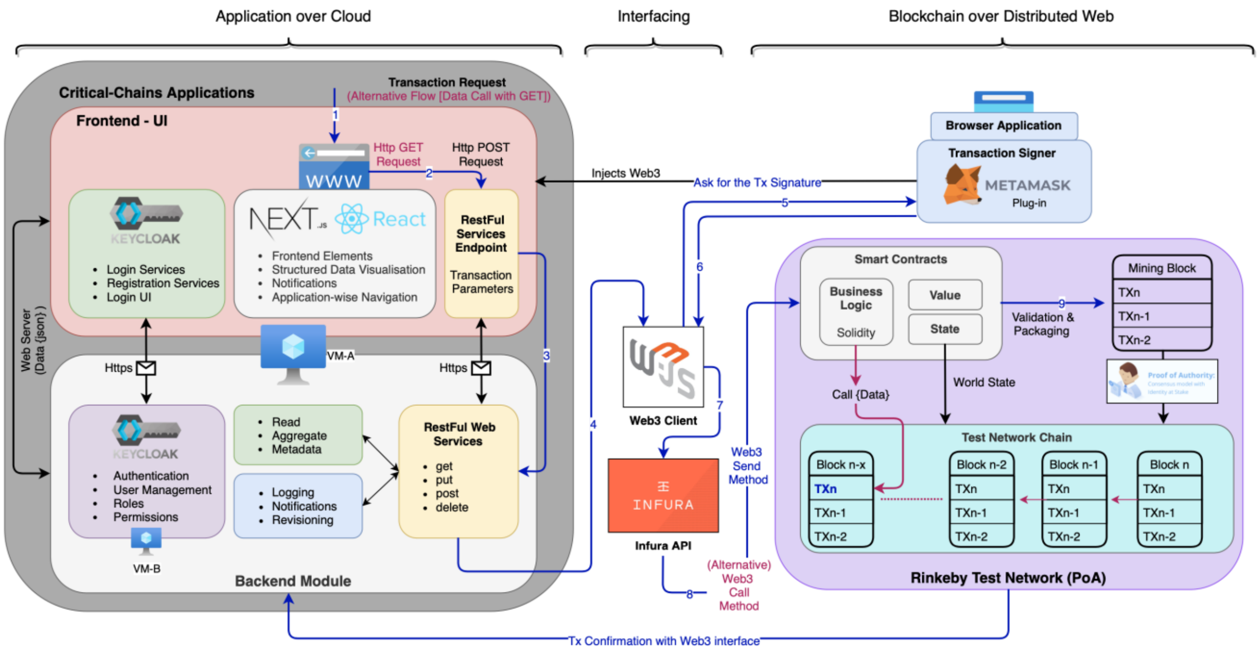


Figure 1. The First Iteration Blockchain Mapping and Linking

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

In order to visually represent the overall linking and mapping of the blockchain components with the Critical-Chains components, Figure 1 diagram has been created. The architecture represents a layered presentation of the first iteration components and their positioning over the web and Critical-Chains environment. The architecture diagram depicts the flow linkages in terms of a transaction request (data registration) that could occur. The blue flow represents the transaction request and the purple colour in some inner parts represents the alternative flow of a data call (retrieval) in the applications.

3.1.2 Synchronisation of the Transactions to the Rinkeby

The activity begins by an externally owned account (refers to an account maintained by a human, rather than a contract) is referred to an Ethereum transaction. For instance, If Bob transfers Alice 1 ETH, Bob's account must be debited and Alice's account must be credited. Within a transaction, this state-changing operation occurs.

In the sense of gas (the cost of the computing in the Ethereum Blockchain-based Computing Environment called EVM), once the transaction has been submitted the following flow is realised:

1. Once transaction sent, cryptography generates a transaction with hash: 0x97d99bc7729211111a21b12c933c949d4f31684f1d6954ff477d0477538ff017
2. The transaction is then broadcast to the network and included in a pool with lots of other transactions.
3. A miner must pick the transaction and include it in a block in order to verify the transaction and consider it "successful".
4. It may end up waiting at this stage if the network is busy and miners are not able to keep up. Miners will always prioritise transactions with higher GASPRICE because they get to keep the fees.
5. The transaction will also get a block confirmation number. This is the number of blocks created following the block that included a given transaction. The higher the number, the greater the certainty that the transaction was processed and recognised by the network. This is because sometimes the block the transaction was included in may not have made it into the chain.
6. The larger the block confirmation number the more immutable the transaction is. So, for higher value transactions, more block confirmations may be desired. (Ethereum 2021)

In terms of preventing a bifurcation of the blockchain and / or transactions in each block of the chain, this is worked on by many network participants or miner nodes. The one with the highest score is chosen over the others due to the inclusion of a specific method for scoring alternative versions of the history (in addition to safe hash-based history). Orphan blocks are blocks that were not chosen for inclusion in the specified chains or that are unfinished. Though blockchain is an iterated process of forming a chain of blocks, a fork in the chain could be created and expected under certain conditions, such as when two peer or miner nodes on the network find and broadcast a new block reference similar to that of the previous one in an environment with little to no permissions. Different versions of history, such as the highest-scoring version known to them, may be recorded in the local chain data storage of miner or peer nodes running on the same network. These nodes will overwrite their own local storage, resend, and broadcast these enhancements to their peer miner nodes whenever a new version with a better score is released, such as the insertion of a new block to the chain. (Nakamoto 2009) (Yiu 2021)

The blockchain technology is built on a peer-to-peer (P2P) architecture and with the Proof of Work consensus algorithm, at the approval of block stage every node in the network has the same rights. Each peer is a node, and they could be distinguished based on their role as follows:

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

- Validation: A validation node is a node that validates information, verifies its accuracy, and passes it on to other nodes, enabling monetary value to be transferred from point A to point B. Because every mining node is also a validation node, mining nodes are a subset of validation nodes.
- Mining: A mining node is a validation node that uses the hardware system to guess the number and letter combinations required to validate and verify a block. To boost the chances of guessing, a mining node can join forces with other nodes and transmit guesses to a common pool (pool mining), although it still counts as only one node.

In simple terms, mining nodes are responsible for guessing the combination required to mark a block as legitimate, seal it, and confirm it, whereas validation nodes are responsible for validating and ensuring that the process is correct. The information is then passed on to other nodes. (Davies 2020)

On the other hand, the Proof of Authority (PoA, as used in the first iteration development) consensus algorithm is a blockchain method that uses a consensus mechanism based on the identity of a stakeholder to produce comparatively fast transactions (Hasan, Brunie and Bertino 2021). In this algorithm, validators, or approved accounts, validate transactions and blocks in PoA-based networks (Parity Technologies 2017). Validators use software to organise transactions into blocks. The process is automated, so validators do not have to monitor their computers constantly. It does, however, necessitate keeping the computer (the authority node) secure.

Gavin Wood, co-founder of Ethereum and Parity Technologies, created the term “Proof of Authority” (Wood 2015). Individuals earn the right to become validators through PoA, thus they have an incentive to keep the position they have earned. Validators are incentivised to support the transaction process by attaching a reputation to their identities, since they do not want their identities to be associated with a negative reputation. This is thought to be more secure than PoS (Proof-of-Stake), which, though a stake between two parties may be equal, does not account for each party's entire holdings. As a result, incentives may be uneven. Proof of Authority, on the other hand, only permits one validator to approve non-consecutive blocks, limiting the potential of catastrophic damage to the authority node. Proof of Authority is best suited to private blockchains where the identification of authority nodes may be established and revealed within the network. Some argue that PoA is unsuitable for public blockchains however it is one of the early adopted consensus algorithms for the private blockchains.

In conclusion, the synchronisation of a transaction covers the overall timeline from the transaction request that has been signed by the users to the transaction approval, registration in the block, and finally the block approval and validation period. Accordingly, this process differs in each consensus algorithm regardless of the Blockchain network.

3.2 2nd Iteration: Overall Positioning of Quorum Blockchain Network

Blockchain as a Service layer contributes to the Critical-Chains framework by providing Blockchain and Blockchain Integrity capabilities, by combining Quorum Blockchain technology, based on Ethereum, and the Guardtime Keyless Signature Infrastructure (KSI). The Quorum Blockchain is used to execute business logics on the Blockchain through the means of the smart contracts, and to develop use cases based on the use of Blockchain technology that involves for example the exchange of cryptocurrencies. KSI is used to sign transactions in platform level, protecting the integrity in which KSI is using a private blockchain.

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

3.2.1 Linking and Mapping of the Nodes and the Consensus Algorithm

Quorum is an Ethereum based distributed-ledger technology and open-source protocol layer that enables enterprises to leverage Ethereum for private production blockchain applications. It provides a permissioned implementation of Ethereum which supports transactions and contract privacy.

The Quorum and Ethereum platforms function similarly, but with a few key differences:

- Network and peer permissions management - Quorum is a permissioned system, where only authorised parties are given access to the platform network. Indeed, Quorum has a permissioned chain of people in the system; exchanges take place between participants who are pre-approved by a designated authority.
- Enhanced transaction and contract privacy - Quorum differentiates between public and private transactions. Open transactions are similar to those taking place on the Ethereum platform; whereas, private transactions are confidential, insofar as data relating to the transactions are not exposed to the public.
- Better performance - due to its simple consensus mechanism, Quorum can easily provide more than 100 transactions per second, higher than Bitcoin and Ethereum.

With no need for Proof of Work (PoW) in a permissioned network, Quorum implements multiple consensus mechanisms that are more appropriate for consortium chains. Quorum Istanbul BFT is a state machine replication algorithm, where each validator maintains a state machine replica in order to reach block consensus. A schematic for the operation of the system (in the case of adding blocks to the blockchain) is provided in Figure 2.

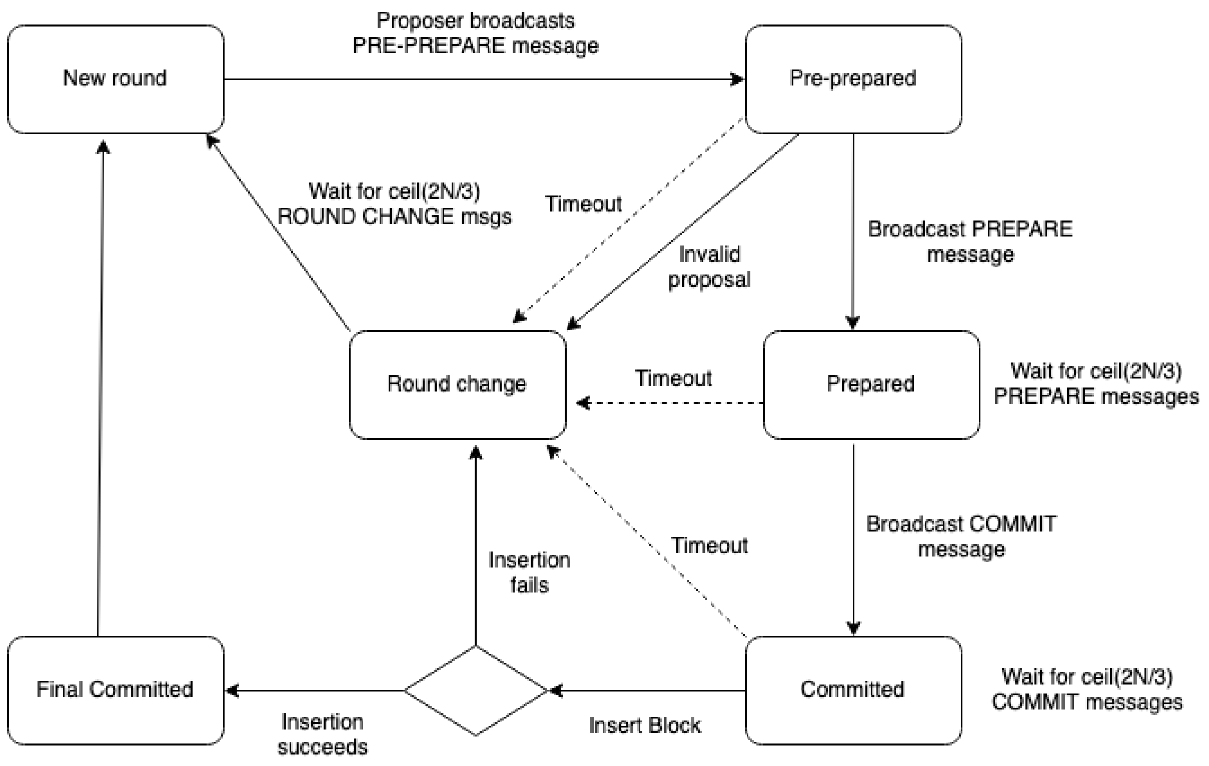

New round
Proposer broadcasts PRE-PREPARE message
Pre-prepared
Wait for ceil(2N/3) ROUND CHANGE msgs
Timeout
Invalid proposal
Broadcast PREPARE message
Round change
Timeout
Prepared
Wait for ceil(2N/3) PREPARE messages
Insertion fails
Timeout
Broadcast COMMIT message
Final Committed
Insertion succeeds
Insert Block
Committed
Wait for ceil(2N/3) COMMIT messages

Figure 2. GoQuorum schematic for block addition

The Quorum Blockchain is a simple privacy supporting network for both private/ public transactions and smart contracts. In fact, all public and private smart contracts and state derived blockchain of transactions

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

are validated by every node in the network exposing details of private transactions and contracts to relevant parties. With this permissioned implementation of Ethereum supporting data privacy, the network is able to process hundreds of transactions per second, depending on system configuration, enough to support institutional volumes as happens for financial services.

The Quorum architecture in Figure 3 is composed by the following elements (JP Morgan 2016):

- **Transaction Manager**: enables access to encrypted transaction data for private transactions, manages local data store and communication with other Transaction Managers.
- **Crypto Enclave**: responsible for private key management and encryption and decryption of private transaction data.
- **QuorumChain**: voting-based, BFT-hardened consensus mechanism that utilises core Ethereum features to verify and propagate votes through the network.
- **Network Manager**: controls access to the network, enabling a permissioned network to be created.

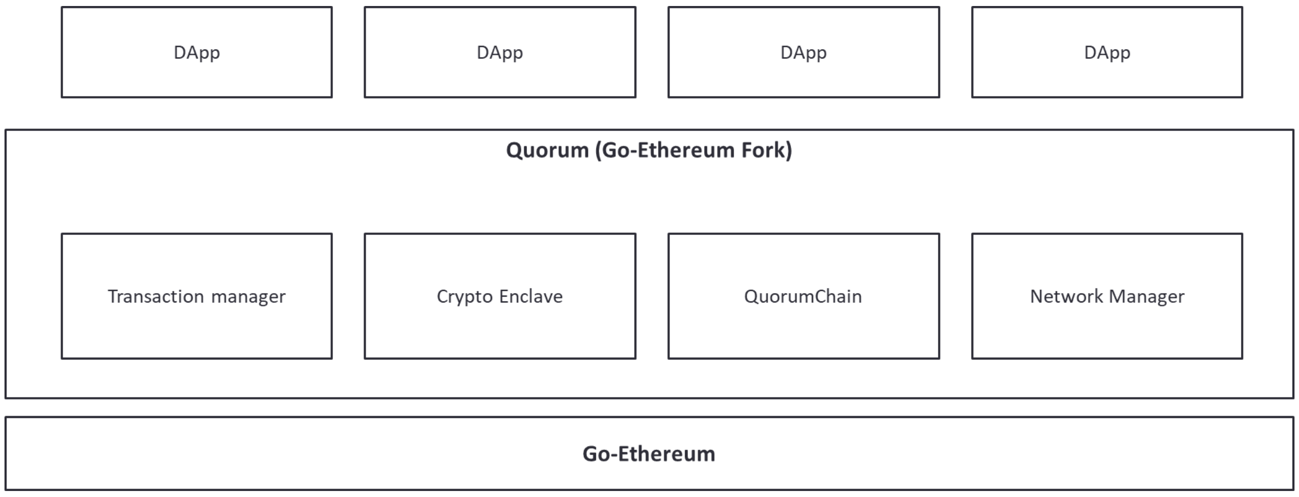


Figure 3. Quorum Framework Architecture

Permissionless blockchains reach consensus via a decentralised protocol applied across an unlimited set of nodes (or participants). These protocols typically make minimal assumptions regarding the trustworthiness of these nodes and do not require them to reveal their identities beyond a pseudonymous identifier. As a result, permissionless networks do not assume any level of trust in the participants as the identities of these participants are not known, and participants can control multiple identities and freely acquire new identities and dispose of old ones. Permissioned blockchains, in contrast, restrict the set of validator nodes that may update the blockchain, restrict participation of nodes even as users, and approved nodes are often required to provide full transparency regarding their identities.

In permissioned blockchains, where the validators can be trusted, consensus can be established with high performance and efficiency, and the transparency regarding identities, enabling trust and the accountability that is required in certain environments. The key distinction between permissioned and permissionless blockchains is that in the former the identities of the permissioned nodes can be established outside of the blockchain, and thus these nodes can accumulate reputation and be subject to contractual enforcement outside the blockchain ecosystem, agreeing, for instance, that participation in an attack would incur a certain penalty or forfeit a bond posted outside the blockchain.

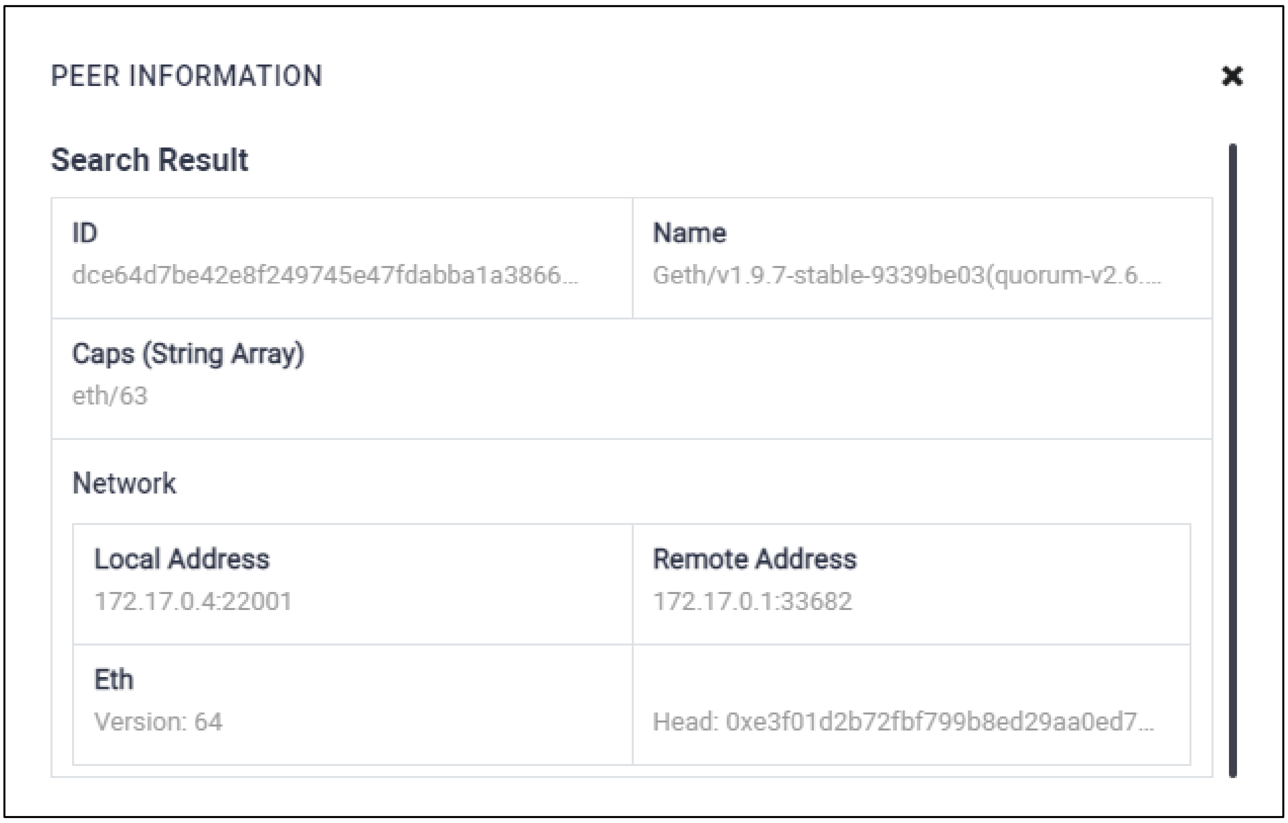
Validators may also have established reputations outside the blockchain, which could be negatively affected if they are discovered to participate in an attack. On the other hand, the fact that validators have known

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

identities enables them to enter into side agreements in the form of either formal or relational contracts that could facilitate the coordination required for multiple validators to collude and carry out an attack. While nodes in permissionless blockchains may also contract based on their blockchain identities and mechanisms like smart contracts, the ability to identify the validators in permissioned blockchains outside of the blockchain allows punishment via negative reputation feedback as well as institutional enforcement mechanisms such as the courts. Permissioned blockchains thus enable a wider set of means to incentivise desired behaviours from their validators.

The implementation of nodes in this framework is a lightweight fork of Geth and takes the same advantage of the regular Ethereum nodes, with the difference that there is no need to have many nodes as the Ethereum network (4437 (Ether Nodes 2021) at the current time) because, in private blockchain environments, any node is able to download the content of the blockchain, while, writing capabilities are strictly related to enabled participants.

Within the Critical-Chains Consortium, we implemented 3 replicative nodes, avoiding the need of extra resources at this point of the project, to enable data redundancy. In fact, in the case of a data loss from one of our nodes, the other nodes will replicate data available in the corrupted node acting as backup. The node details are presented in Figure 4, 5 and 6.


PEER INFORMATION

Search Result

ID
dce64d7be42e8f249745e47fdabba1a3866...

Name
Geth/v1.9.7-stable-9339be03(quorum-v2.6....

Caps (String Array)
eth/63

Network

Local Address
172.17.0.4:22001

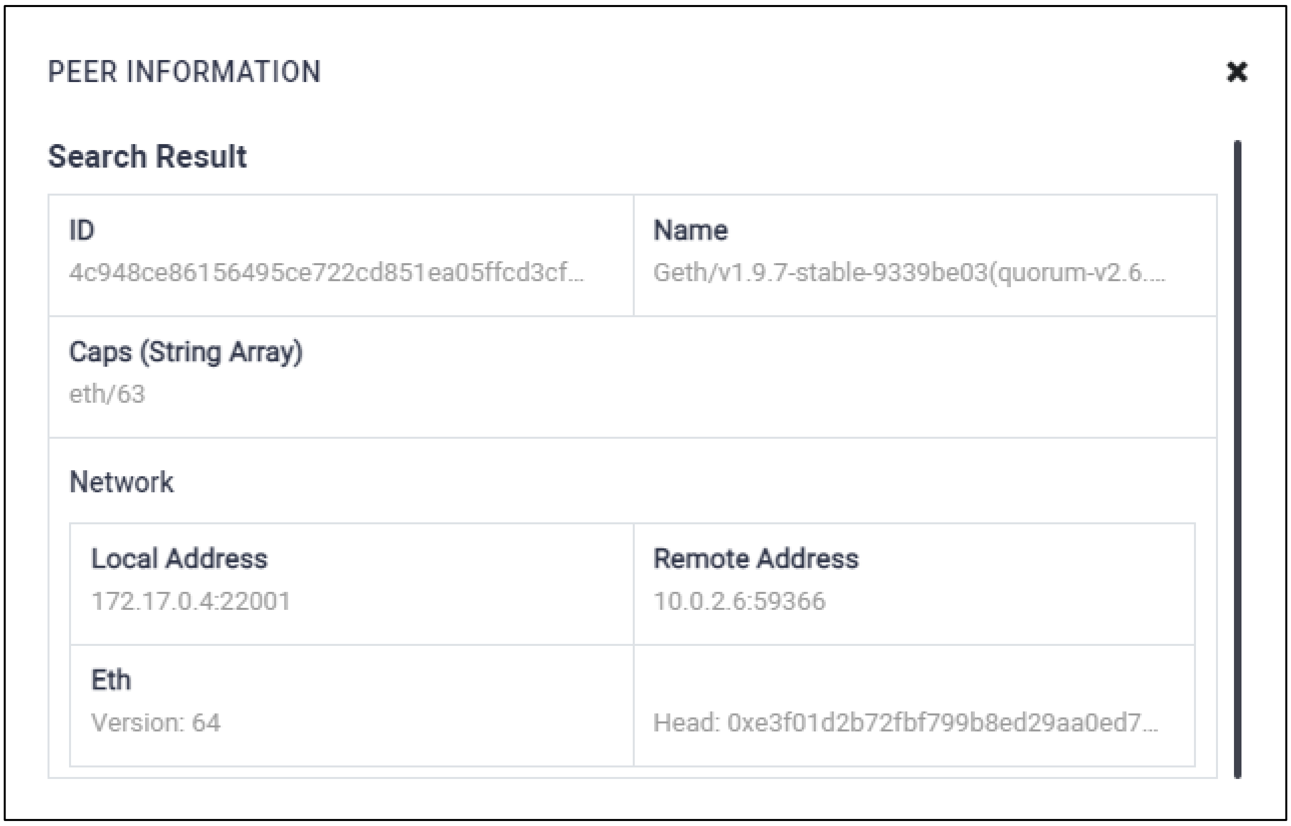
Remote Address
172.17.0.1:33682

Eth
Version: 64

Head: 0xe3f01d2b72fbf799b8ed29aa0ed7...

Figure 4. CC Node 1 information

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018



PEER INFORMATION

Search Result

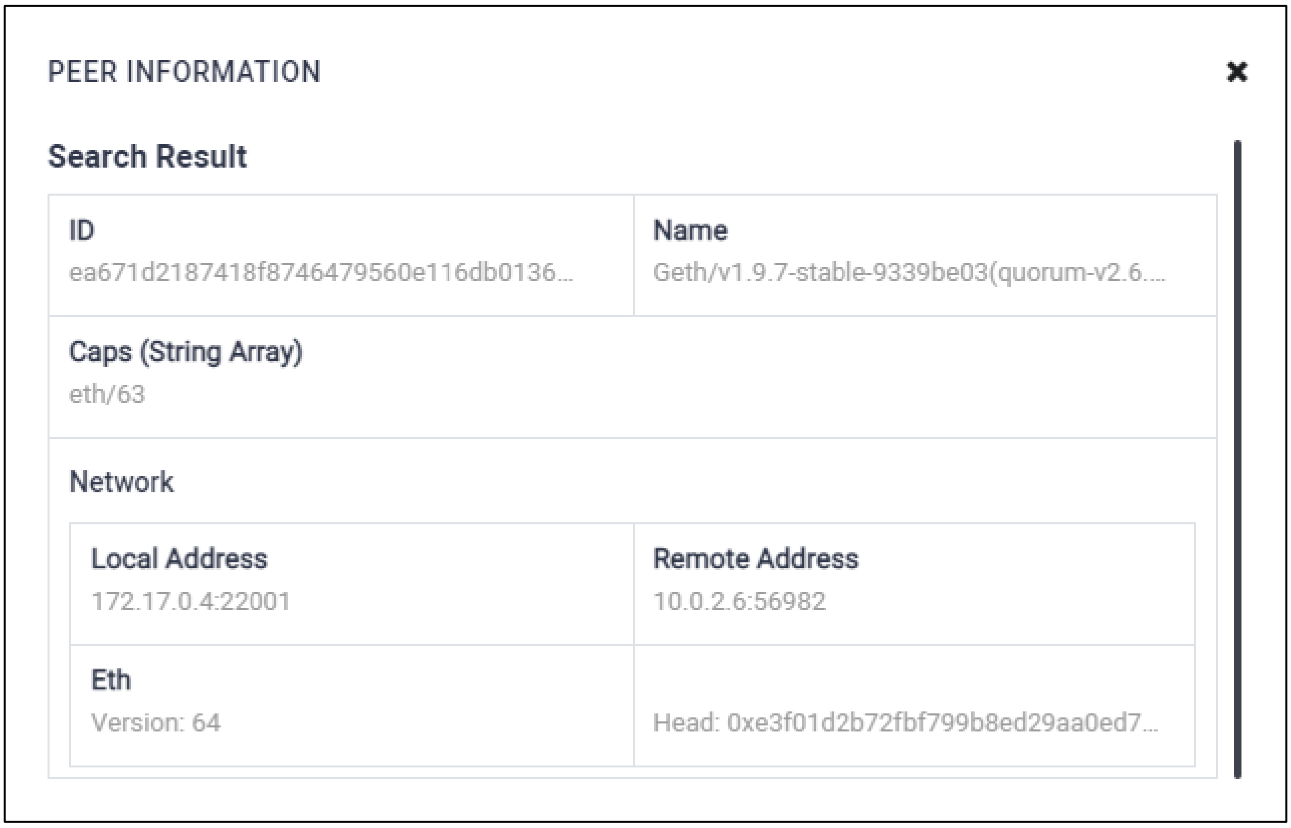
ID	Name
4c948ce86156495ce722cd851ea05ffcd3cf...	Geth/v1.9.7-stable-9339be03(quorum-v2.6....

Caps (String Array)
eth/63

Network

Local Address	Remote Address
172.17.0.4:22001	10.0.2.6:59366
Eth Version: 64	Head: 0xe3f01d2b72fbf799b8ed29aa0ed7...

Figure 5. CC Node 2 information



PEER INFORMATION

Search Result

ID	Name
ea671d2187418f8746479560e116db0136...	Geth/v1.9.7-stable-9339be03(quorum-v2.6....

Caps (String Array)
eth/63

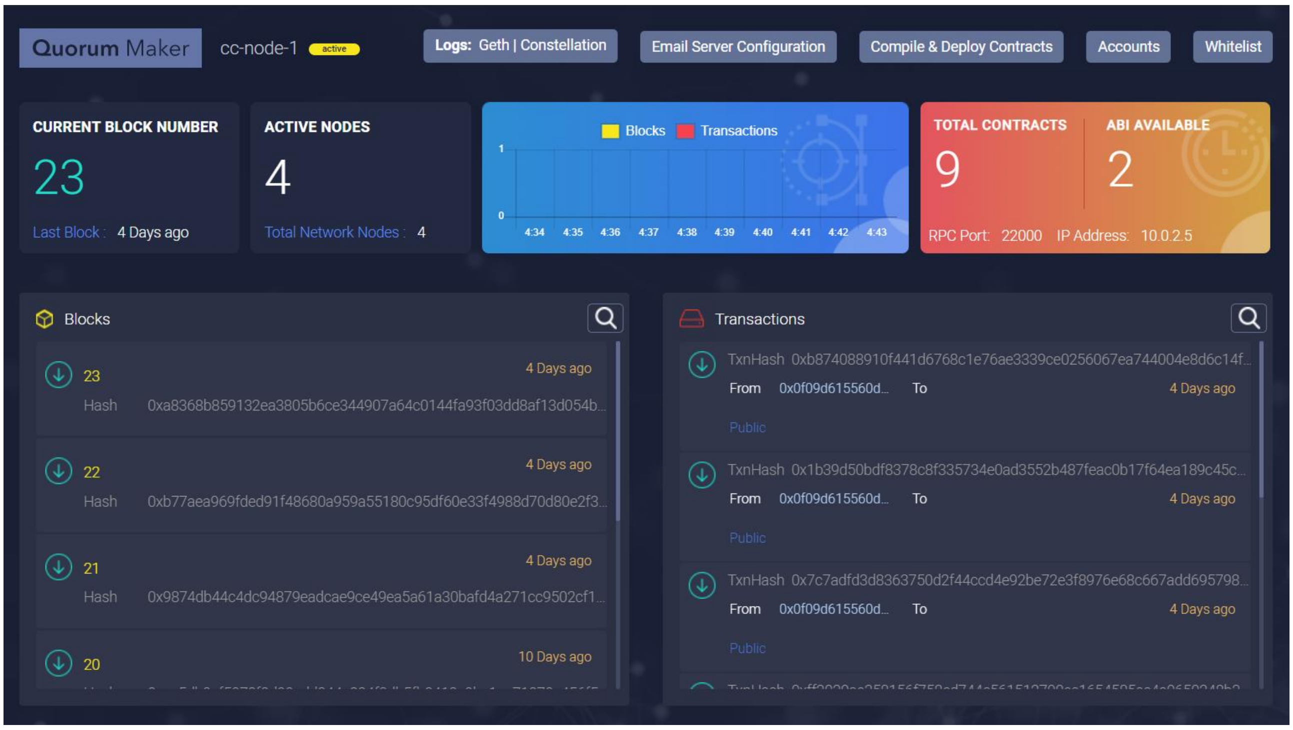
Network

Local Address	Remote Address
172.17.0.4:22001	10.0.2.6:56982
Eth Version: 64	Head: 0xe3f01d2b72fbf799b8ed29aa0ed7...

Figure 6. CC Node 3 information

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

The Critical-Chains Quorum network is managed through Quorum Maker, a tool that allows to create Quorum nodes by inputting basic information and configuring the backend automatically. It is possible to create any number of Quorum nodes which can be distributed on separate Linux based machines or cloud instances for a production environment. In addition to its easy-to-use wizard noninteractive setup interface, it also provides a web-based interface for managing and monitoring the Quorum network. As Figure 7 shows, the interface reports an individual node monitor, a block explorer, and an online log watcher.



CURRENT BLOCK NUMBER 23
Last Block : 4 Days ago

ACTIVE NODES 4
Total Network Nodes : 4

TOTAL CONTRACTS 9
ABI AVAILABLE 2
RPC Port: 22000 IP Address: 10.0.2.5

Blocks

```
23                                                      4 Days ago
Hash  0xa8368b859132ea3805b6ce344907a64c0144fa93f03dd8af13d054b...
22                                                      4 Days ago
Hash  0xb77aea969fded91f48680a959a55180c95df60e33f4988d70d80e2f3...
21                                                      4 Days ago
Hash  0x9874db44c4dc94879eadcae9ce49ea5a61a30bafd4a271cc9502cf1...
20                                                     10 Days ago
```

Transactions

```
TxnHash 0xb874088910f441d6768c1e76ae3339ce0256067ea744004e8d6c14f...
From 0x0f09d615560d...  To                          4 Days ago
Public
TxnHash 0x1b39d50bdf8378c8f335734e0ad3552b487feac0b17f64ea189c45c...
From 0x0f09d615560d...  To                          4 Days ago
Public
TxnHash 0x7c7adfd3d8363750d2f44ccd4e92be72e3f8976e68c667add695798...
From 0x0f09d615560d...  To                          4 Days ago
Public
```

Figure 7. Quorum Maker interface

The Critical-Chains nodes are maintained through 2 Azure virtual machines for avoiding the use of Azure Blockchain Services to achieve more flexibility and security with the following specifications:

Table 1. Quorum Blockchain, Specification of the Nodes

Service type	Custom name	Region	Description
Virtual Machines	Quorum Blockchain	West Europe	2 B2S (2 vCPUs, 4 GB RAM) x 730 Hours; Linux – Ubuntu; 2 managed disks – S10, 100 transaction units; Inter Region transfer type, 5 GB outbound data transfer from West Europe to East Asia

3.2.2 Synchronisation of the Transactions on Quorum

Quorum introduces the idea of public and private transactions, where Quorum has extended the Ethereum transaction model to include an optional “privateFor” parameter, where this makes the transactions to be treated as private, and the transaction type method as “IsPrivate” to recognise the private transactions.

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Public transactions are transactions with the payload visible to all the participants within the same Quorum network. This type is the same as the Ethereum transactions.

Private transactions have the payload only visible to the network participants whose public keys are specified in the “privateFor” parameter of the transaction. This parameter can have multiple addresses separated by a comma. When the Quorum Node comes across a transaction whose “privateFor” value is non-null, it automatically sets the Transaction Signature to be set as 37 or 38, in other words, to ‘private,’ as opposed to the 27 or 28 values which are ‘public’ within Ethereum. Public transactions are executed in the standard Ethereum way, and so if a Public Transaction is sent to an account that holds a contract code, each participant will execute the same code and their underlying StateDBs will be updated accordingly.

Private transactions, however, are not executed per standard Ethereum: prior to the sender’s Quorum node propagating the transaction to the rest of the network, it replaces the original transaction payload with a hash of the encrypted payload that it receives from Constellation/Tessera - Haskell and Java implementations of a general-purpose system for submitting information securely. Participants that are party to the transaction will be able to replace the hash with the actual payload via their Constellation/Tessera instance, whilst those Participants that are not a party to the transaction will only see the hash. The result is that if a private transaction is sent to an account that holds contract code, those participants who are not a party to the Transaction will simply end up skipping the transaction, and therefore not execute the contract code. However, those participants that are party to the transaction will replace the hash with the original Payload before calling the EVM for execution, and their StateDB will be updated accordingly. In the absence of making corresponding changes to the geth client, these two sets of participants would therefore end up with different StateDBs and not be able to reach a consensus. In order to support this bifurcation of contract state, Quorum stores the state of public contracts in a Public State Trie that is globally synchronised, and it stores the state of Private contracts in a Private State Trie that is not synchronised globally.

In Figure 8 we describe how private transactions are executed within the Quorum network:

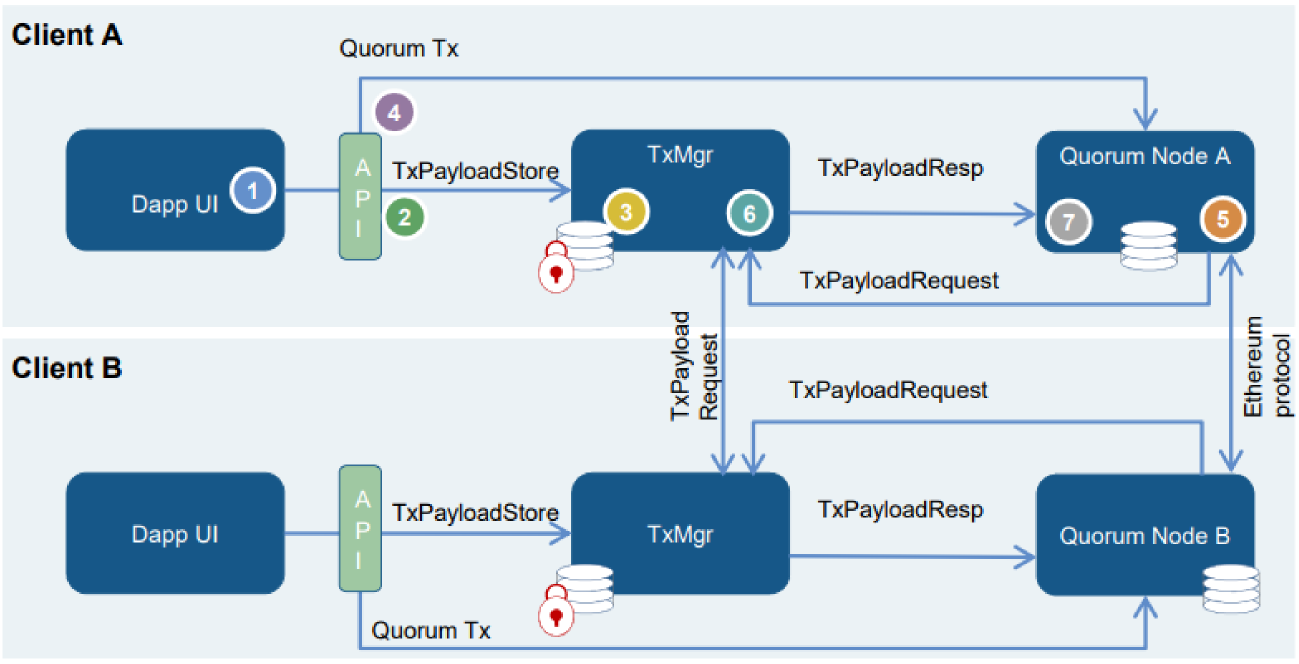


Figure 8. Quorum private transactions process

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

1. Dapp sends a transaction to the Quorum Node, specifying recipient and transaction payload.
2. Prepare Tx Payload Record by generating a symmetric key, encrypt the payload with the symmetric key, hash of the encrypted payload, encrypt the symmetric key with the public keys of the parties to the Tx, then send to the TxMgr for storage.
3. TxMgr validates the sending signature and stores the TxPayload message.
4. Tx sent to the Quorum node containing only the hash of the encrypted payload generated in step 2.
5. Quorum Node receives a new block for validation containing the private Tx. It requests the payload data from the TxMgr (passing its Pubkey, TxHash, Sig).
6. TxMgr validates the signature, looks up the TxHash and if the requester is party to the Tx, return the encrypted payload and encrypted Symmetric key.
7. Quorum Node decrypts the symmetric key, decrypts the Tx Payload and sends this to the EVM for contract code execution.

TxPayload includes:

- Hash of encrypted Tx payload (TxHash)
- Party 1 Public Key encrypted Symmetric Key
- Party 2 Public Key encrypted Symmetric Key
- Party n Public Key encrypted Symmetric Key

3.3 3rd Iteration: Positioning and Integration of the KSI with Quorum Blockchain

3.3.1 Linking and Mapping of the KSI over Quorum Blockchain

The main components of the Quorum Integrity solution and their relations are shown in the component diagram in Figure 9. The solution consists of two applications – Quorum Signer and Quorum Verifier. The first one is a long-running daemon process that listens for new block events from the Quorum network, creates and links dockets and stores dockets in the docket repository implemented as a PostgreSQL database. The second, Quorum Verifier, is an interactive command line application that verifies the integrity and consistency of a running Quorum network with the block hashes and timestamps stored in dockets.

Both of these applications, implemented in Java, are based on a common software library that provides application specific shared functionality like database access, handling of docket structures and KSI signatures, and interfacing with Quorum nodes. The functionality provided by the common library is, in turn, based on existing software libraries, including PostgreSQL JDBC library, Docket SDK and web3j-quorum library.

The Quorum Integrity solution depends on the following external components. First, it requires access to a repository for storing dockets. In the Critical-Chains setup, a PostgreSQL database is used as a docket repository. Second, access to one or more Quorum nodes is required in order to read block index values, block hashes and timestamps. The Quorum Integrity solution does not store Quorum transactions or complete block headers in dockets. Based on the information available in the docket repository, it is possible to infer information about the number of blocks created by the Quorum network and timestamps of Quorum blocks, but no information about the contents of private or even public transactions is leaked. Finally, as a third external dependency, the solution needs access to a KSI gateway for creating, extending and verifying the unextended KSI signatures.

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

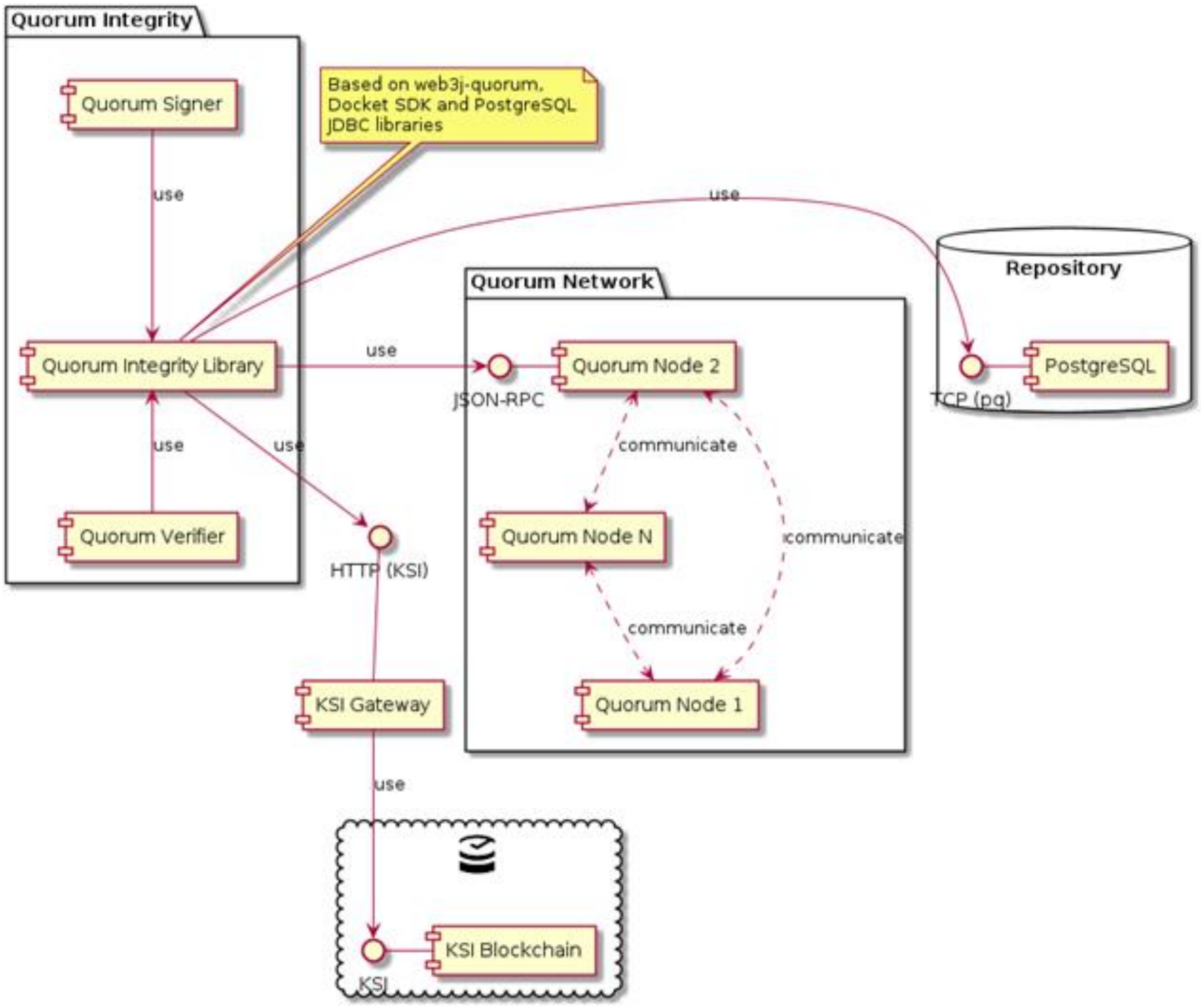


Figure 9. Quorum Integrity component diagram

More details about the internal functioning of Quorum Signer and Quorum Verifier applications are provided in Section 3.3.3.

3.3.2 Synchronisation of Quorum Blockchain blocks with KSI Signatures

The Quorum Integrity solution requires read-only access to the Quorum network. It can be configured to communicate with one or more Quorum nodes. Communicating with just one Quorum node is sufficient for the basic service, but connecting to more than one node is recommended, as it provides additional availability guarantees and assurance that all Quorum nodes provide a consistent view of the ledger. The means provided by the web3j-quorum library are used for extracting new block events and block header data from the Quorum network. Internally, the library uses the RPC interface exposed by Quorum nodes exposed over TCP. For communicating with the database, Quorum Integrity uses the standard PostgreSQL JDBC library which also uses TCP and PostgreSQL frontend/backend protocol. Standard authentication mechanisms, including TLS certificates and passwords are supported.

The Quorum Integrity solution uses Docket SDK (and embedded KSI library) for communicating with the KSI gateway over TCP. As dockets do not contain any sensitive information, the encryption of dockets or the docket repository is not required. Moreover, due to the nature of KSI signatures, an attacker with access to

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

the KSI gateway cannot compromise the security of the Quorum Integrity solution because KSI signatures cannot be backdated.

For integrating the Quorum Integrity deployment with a monitoring solution, the monitoring solution can monitor the process table to check the existence of a Quorum Signer process. Also, logs can be monitored for liveness and potential errors. Finally, the increasing count of dockets in the database can be used as an indicator of normal functioning of the Quorum Signer process. As the Quorum Verifier is run interactively (directly or via some interface), it does not require additional monitoring, because the report generated by the Quorum Verifier is itself an indication about the success or failure of the process. Another viable approach is to run the verification process regularly and store the output in a log file. The monitoring solution can then check the logs for verification status.

3.3.3 Signing, Verification and Overall Data Integrity with KSI

The sequence diagram shown in Figure 10 describes the main steps performed by the Quorum Signer process. When the application is started, it begins with an initialisation phase to figure out the point at which to pick up the processing. Then it enters a waiting loop that reacts to new block events from the Quorum network by creating new dockets, linking them to already existing dockets and securing these structures with KSI signatures.

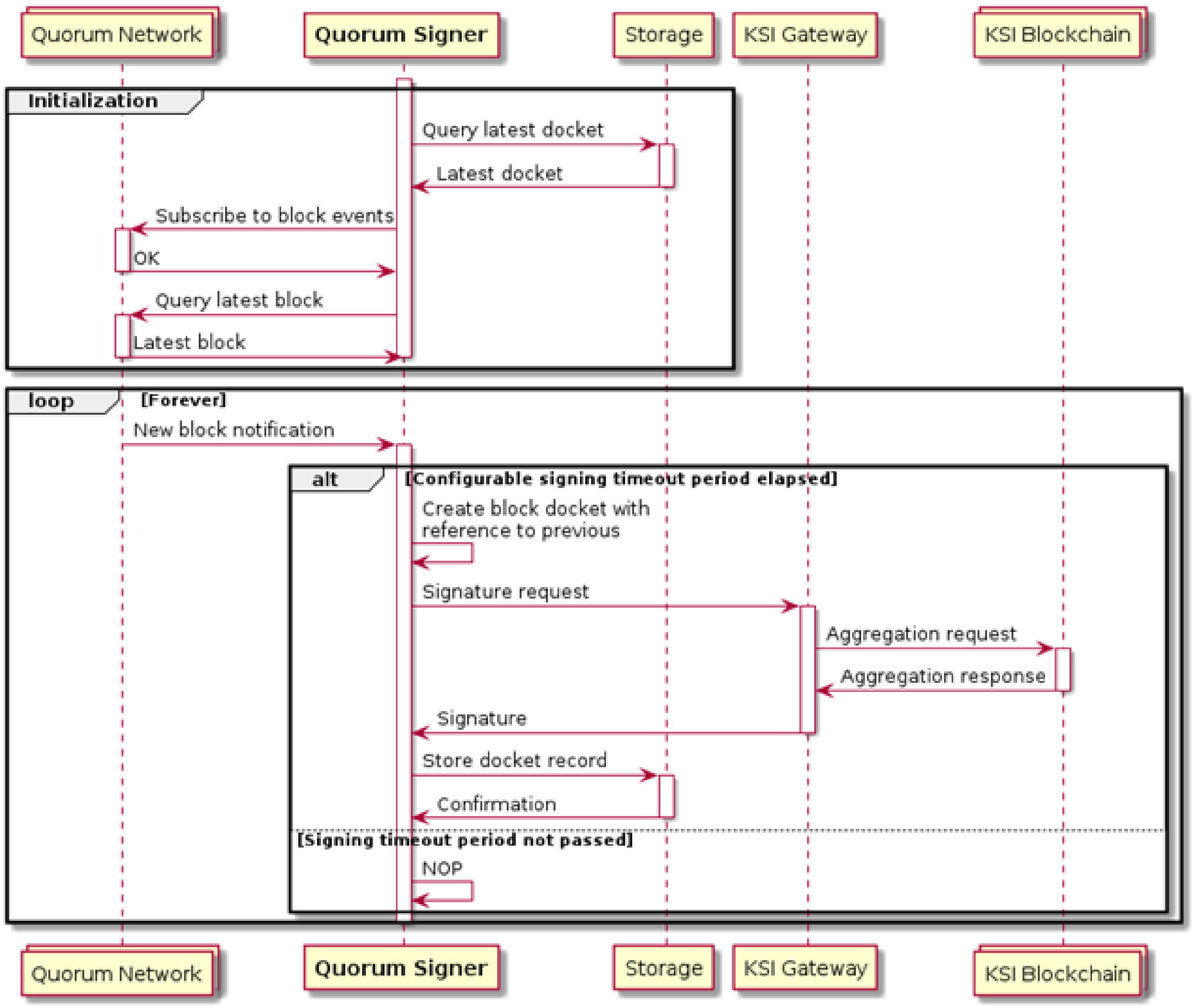


Figure 10. Quorum Signer sequence diagram

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

The Quorum Integrity solution provides better assurance of the data integrity when it is running continuously. When the Quorum Signer process is not running, the time interval between dockets increases and the quality of integrity proofs hence decreases. Short periods of time (in minutes) without a running Quorum Signer process do not compromise the overall security or the value of the proofs already stored in the docket repository, so the service or servers can be restarted without additional measures. Only one instance of Quorum Signer can be running at the same time in one environment, as multiple processes running at the same time may generate conflicting proofs due to race conditions. It is safe to run multiple processes in separate environments (the docket repositories of Quorum Signer processes must be isolated).

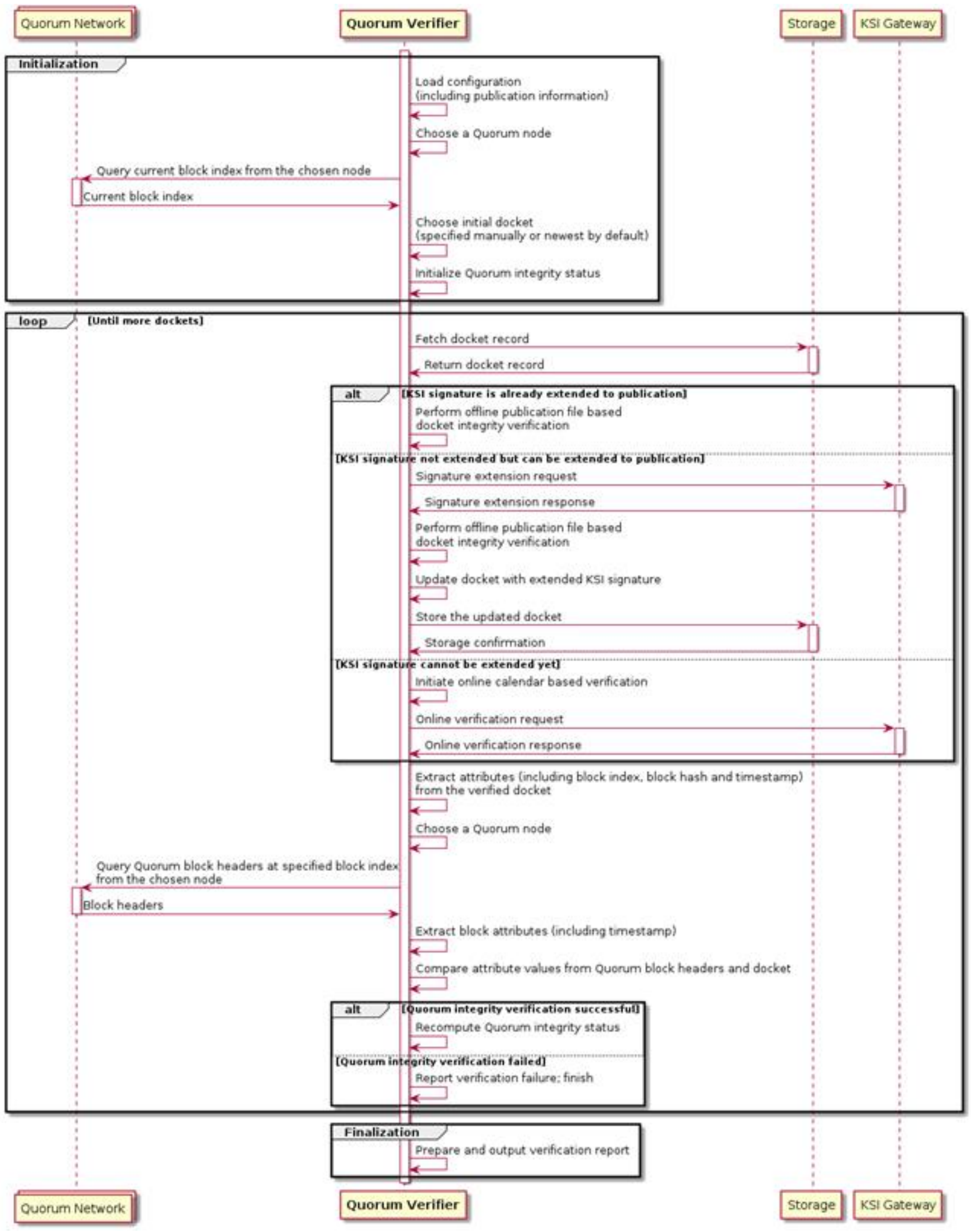

Quorum Network
Quorum Verifier
Storage
KSI Gateway
Initialization
Load configuration (including publication information)
Choose a Quorum node
Query current block index from the chosen node
Current block index
Choose initial docket (specified manually or newest by default)
Initialize Quorum integrity status
loop
[Until more dockets]
Fetch docket record
Return docket record
alt
[KSI signature is already extended to publication]
Perform offline publication file based docket integrity verification
[KSI signature not extended but can be extended to publication]
Signature extension request
Signature extension response
Perform offline publication file based docket integrity verification
Update docket with extended KSI signature
Store the updated docket
Storage confirmation
[KSI signature cannot be extended yet]
Initiate online calendar based verification
Online verification request
Online verification response
Extract attributes (including block index, block hash and timestamp) from the verified docket
Choose a Quorum node
Query Quorum block headers at specified block index from the chosen node
Block headers
Extract block attributes (including timestamp)
Compare attribute values from Quorum block headers and docket
alt
[Quorum integrity verification successful]
Recompute Quorum integrity status
[Quorum integrity verification failed]
Report verification failure; finish
Finalization
Prepare and output verification report
Quorum Network
Quorum Verifier
Storage
KSI Gateway

Figure 11. Quorum Verifier sequence diagram

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

A sequence diagram of a Quorum Verifier process is shown in Figure 11. It begins with an initialisation phase that evaluates the status of the Quorum network (current block height) and available integrity evidence in the docket repository. Then it loops through all the evidence and compares the integrity of block hashes and timestamps between the values read from dockets and the values fetched from the live Quorum network. Finally, an integrity report is generated.

4 Linking, Mapping and Synchronisation of Authentication Service with Blockchain Service

4.1 Integration of the Authentication Service and Blockchain Service

4.1.1 Linking and Mapping of the Authentication Service and Blockchain Public - Private Keys

The two fundamental components of the application are Keycloak, which acts as an identity-authentication service, and Vault, a service that creates an encrypted storage and secure access to a series of data called secrets, each associated with an identifier. In the context of Critical-Chains, Vault acts as a glue to the various components of the service since it maintains within itself the private keys of the wallets, i.e., an alphanumeric string used to sign transactions within the blockchain thanks to the property of asymmetric encryption, generated on the spot by a node of the quorum network, which will start from a pseudorandom base.

The service is also a safe space to save this kind of data, it is in fact completely encrypted and uses its own secure protocol to manage access to the secrets stored inside. Each private key will be tied to a Keycloak ID, an alphanumeric string that uniquely distinguishes the user within the Keycloak instance. The service enables anyone who is registered, to perform daily operations on the Quorum instance, regardless of other factors related to the Ethereum blockchain such as gas fees, using a username and password, without relying on services such as Metamask and managing policies at the same time (i.e., 100 users limitation per day).

Access to the service is managed by Keycloak that, in addition to managing various information about the user, controls access through the OpenID protocol. To maintain all other information that are not included in Keycloak or Vault a Postgres SQL database will be used also chosen for its ability to work with JSON documents. Finally, each call to the chain will be handled by the Ether.js library that abstracts the GRPC model, used by the Ethereum Base to receive the tasks to be performed, passing through HTTP and WebSocket, the two protocols most suitable for this kind of operation.

The service is entirely written in Node.js, server-side runtime for applications written in JavaScript and Typescript and can be used through the Https protocol. However, the service is modelled to work as a REST API enabling it to be used directly by front end but also by other server applications. In Figure 12 and 13 we describe sequence/ response relations:

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

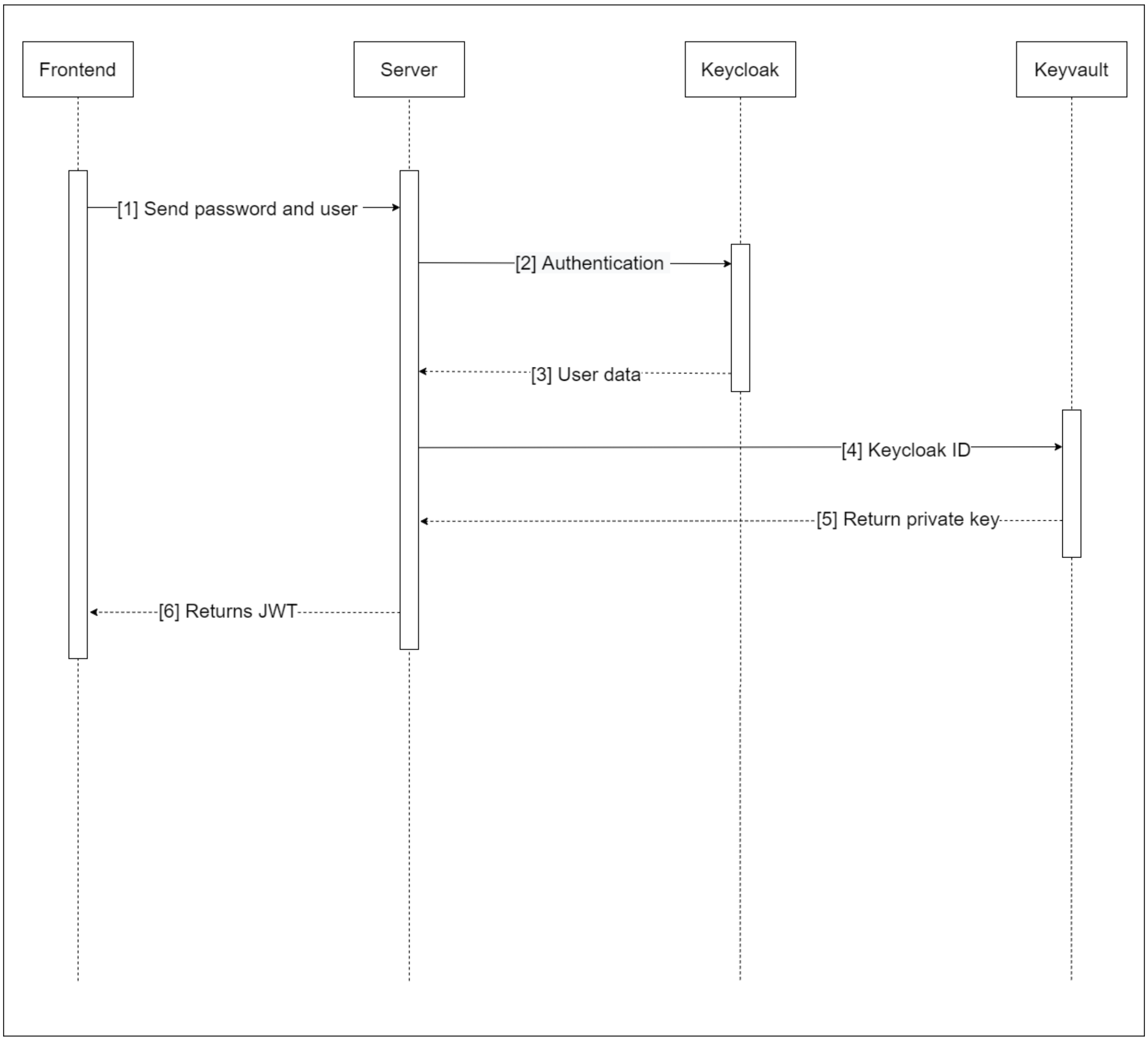


Figure 12. Authentication through Keycloak API

The flow of the process is given below.

1. The user sends username and password from the frontend to the server.
2. The server forwards the credentials to the Keycloak instance in order to proceed with the authentication.
3. Keycloak returns a set of data where the Keycloak ID has been extracted.
4. Keyvault receives the Keycloak ID.
5. Keyvault returns private key
6. Server returns JWT.

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

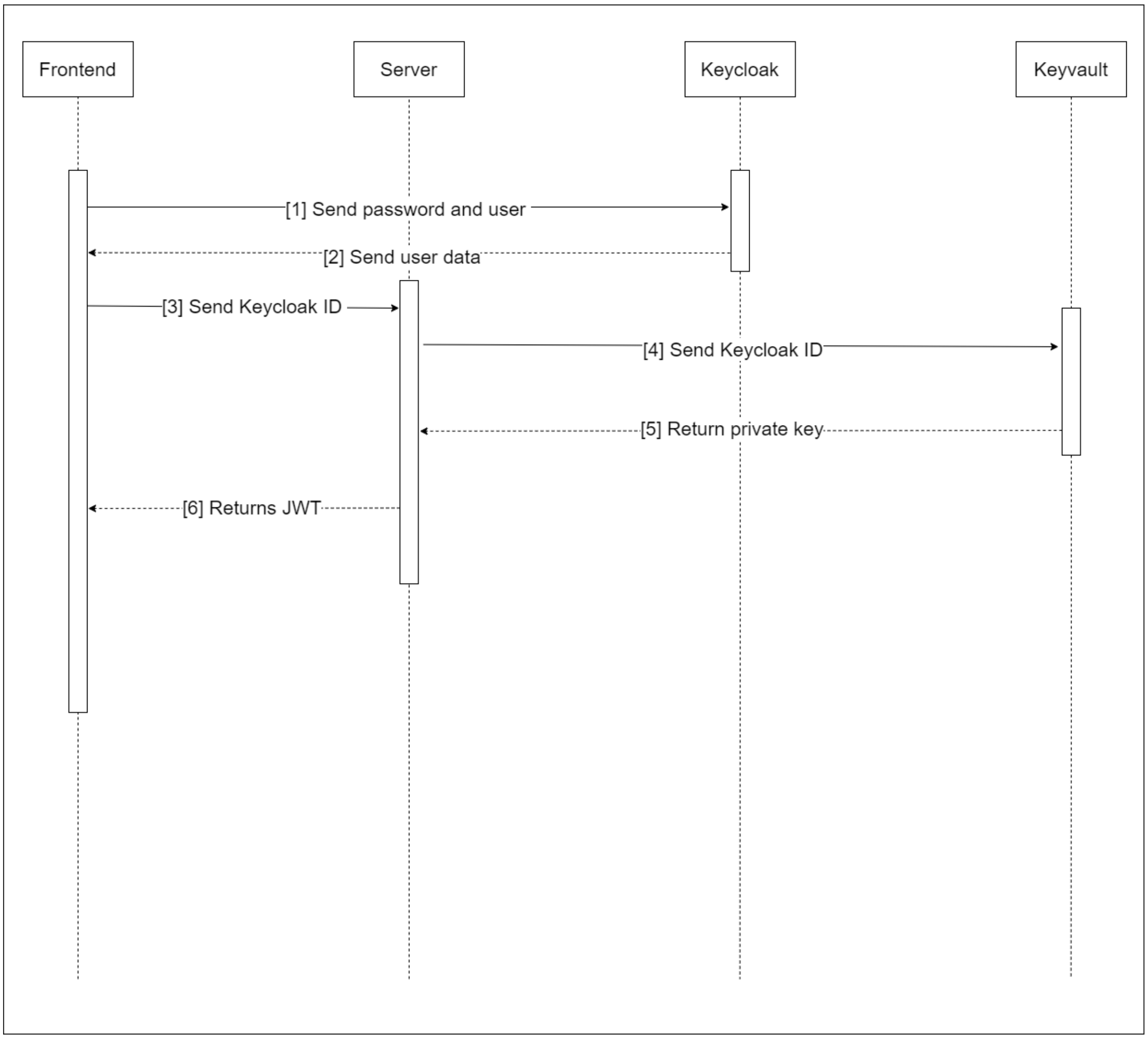


Figure 13. OAUTH2 Authentication

The flow of the process is given below.

1. The Frontend redirect the user to the OAUTH 2 page of Keycloak.
2. Keycloak returns a set of data where the Keycloak ID has been extracted.
3. The server receives the Keycloak ID from the frontend.
4. The server sends the Keycloak ID to Keyvault.
5. Keyvault returns a private key to the server.
6. The server returns a JWT to the front end.

The overall architectural positioning of the service is represented in Figure 14:

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

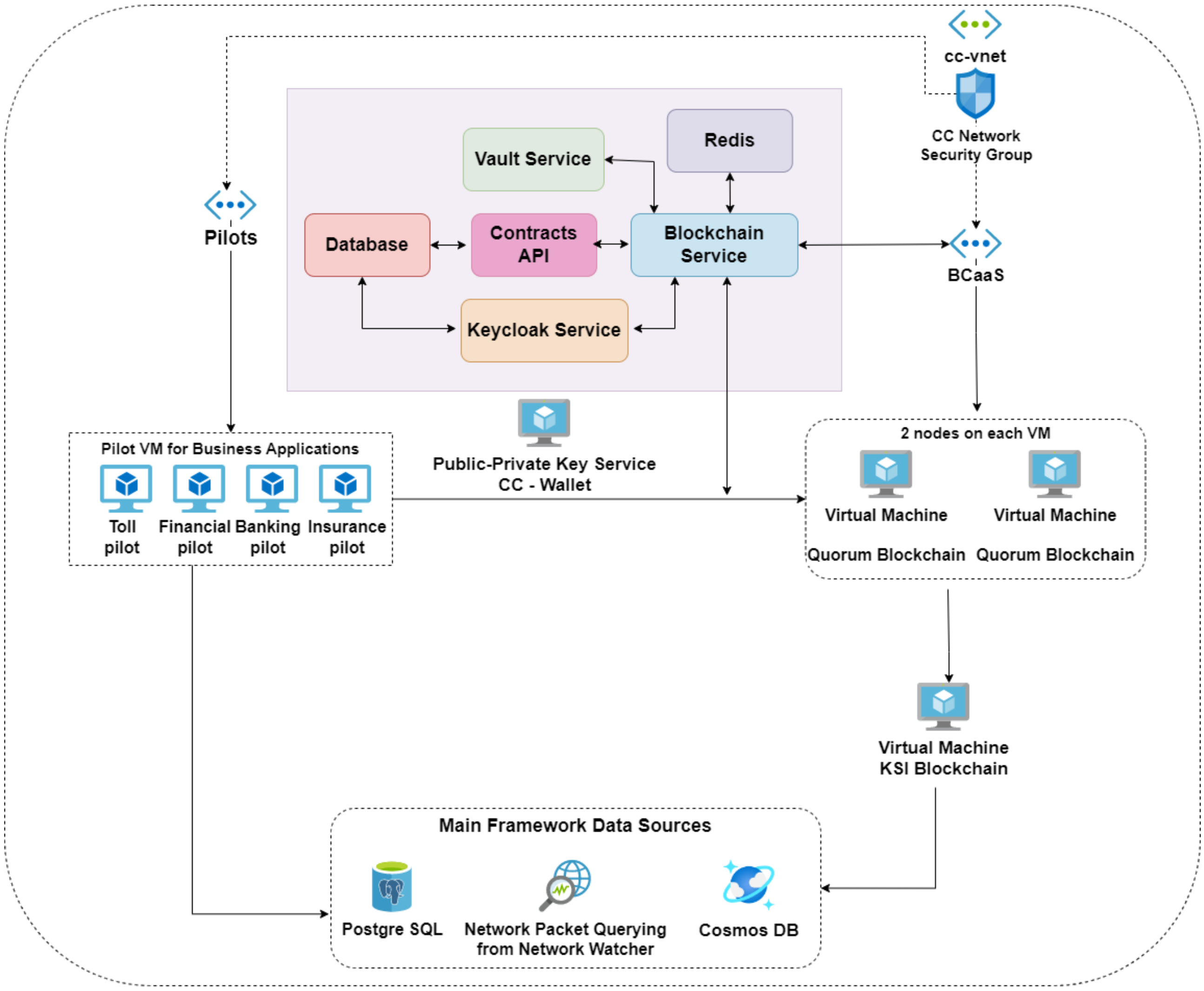


Figure 14. Public – Private Key service positioning in the architecture

4.1.2 Synchronisation of the Authentication Service and Blockchain Keys

To allow the synchronisation of the authentication service and blockchain keys the technologies adopted are:

- NodeJS
- Quorum network
- Vault
- Keycloak
- Postgre SQL database
- APIs
    - https: to use the service.
    - OAuth 2: different from OpenID.
    - integrated Vault APIs.
    - GRPC: used from Web3.

The keypairs registered for each user are registered on the Vault, through the association of KeycloakID and private key, keeping on user data and password on the database. At the same time, for each new registration a Keycloak user and a wallet are created and associated. Security and integrity of the keys are ensured by the Vault through encryption.

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Figure 15 shows the Vault high-level overview.

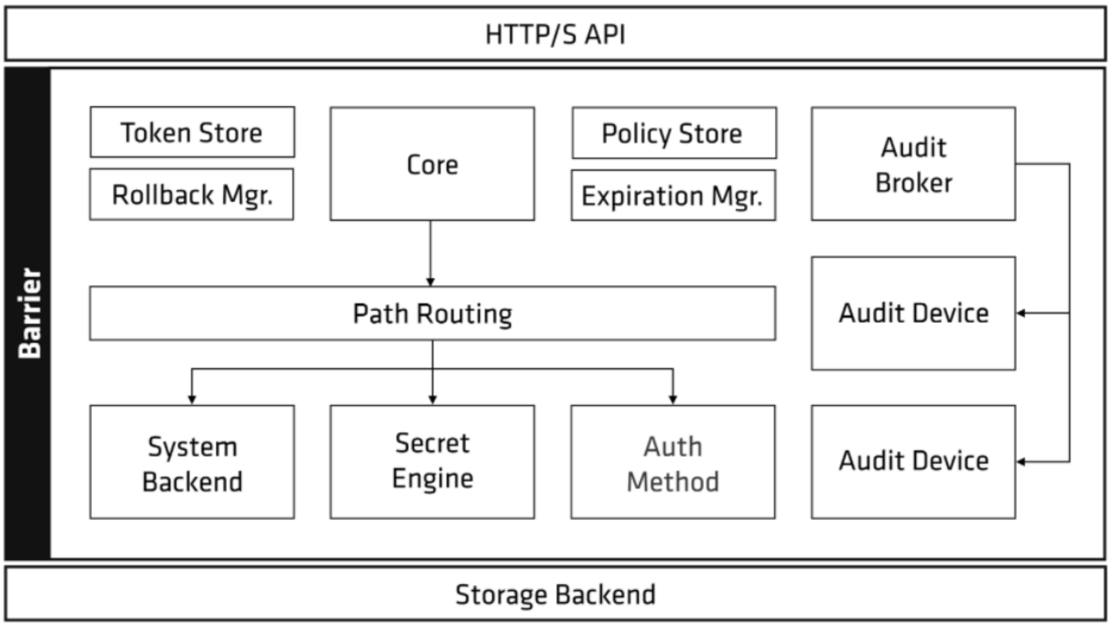


Figure 15. Vault high-level overview

The storage backend is untrusted and is used to durably store encrypted data. When the Vault server is started, it must be provided with a storage backend so that data is available across restarts. The HTTP API similarly must be started by the Vault server on start so that clients can interact with it.

Once started, the Vault is in a sealed state. Before any operation can be performed on the Vault it must be unsealed. This is done by providing the unseal keys. When the Vault is initialised, it generates an encryption key which is used to protect all the data. That key is protected by a master key. By default, Vault uses a technique known as Shamir's secret sharing algorithm to split the master key into 5 shares, any 3 of which are required to reconstruct the master key.

5 Conformance Testing

5.1 Web Application Functional Tests

Web application functional testing is one of the common testing methods used for applications. It is mostly used to test every function and every component of the software application. As such this testing method has been used to test the correct working of each function of each pilot. There are different types of software testing methods in the market. Functional tests provide an efficient means of assessing the quality of the systems, mainly focusing on any possible system errors. In this deliverable, there are two testing approaches followed. These are Graphical User Interface (GUI) and Regression testing. GUI testing aims for the validation of the visual components and their functionalities such as buttons, text boxes, etc working as expected. Regression testing aims for the validation of the applications, whether they performed in high standards as previous versions.

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

5.1.1 Graphical User Interface Testing

Graphical User Interface (GUI) testing process of testing the visual elements of an application and its functionalities such as checkbox buttons, textbox etc. The reason why GUI testing is important is that applications are tested from the user's perspective. Efficiently the test reduces the number of risks towards the end of the development life cycle. It offers developers and testers ease of use and learning. It helps to validate the compliance of various icons and elements with their functionality and even design specifications.

In this section, visual design and functionalities based on each pilot in the form of web applications are tested. Standard table of the visual design provided and according to standard table, the evaluation of the system has been held. Functionality side of the GUI testing based on manual testing in which checkbox, buttons, and other component functionalities are tested.

5.1.1.1 Banking Pilot

In this section the Banking Pilot application has been tested according to Pilot visual design and visual element's functionalities. The table below represents the GUI testing of Banking application stated with Field, Field Type, Description, and status.

Table 2. Banking Pilot GUI Testing

#	Field	Field Type	Description	Status
1	Customer private key	Textbox	Customer enters private key information	Pass
2	Login	Button	Customer clicks on login button	Pass
3	Search transaction	Textbox	Customer searches the transaction to score	Pass
4	Search	Searchbar	Customer presses the search button	Pass
5	Rate merchant	Button	Customer presses rate merchant button	Pass
6	Positive feedback	Togglebutton On	Customer gives positive evaluation	Pass
7	Negative feedback	Togglebutton Off	Customer gives negative evaluation	Pass
8	Submit feedback	Button	Customer submits the feedback	Pass
9	Feedback submitted	Banner	After feedback submission a banner will be displayed	Pass
10	Logout	Button	Customer logs out	Pass
11	Merchant private key	Textbox	Merchant enters private key information	Pass
12	Login	Button	Merchant clicks on login button	Pass
13	Merchant feedbacks	Sidebar Menu Link	Merchant clicks on “your feedback” sidebar menu	Pass
14	Feedback's pagination	Table	Merchant clicks on desired page number to change table page using pagination	Pass
15	Merchant scoring	Sidebar Menu Link	Merchant clicks on “merchant scoring” sidebar menu	Pass
16	Logout	Button	Merchant logs out	Pass
17	Acquirer private key	Textbox	Acquirer enters private key information	Pass
18	Login	Button	Acquirer clicks on login button	Pass
19	Merchants	Sidebar Menu Link	Acquirer clicks on “Merchants” sidebar menu	Pass
20	Select merchant	Button	Acquirer selects the merchant	Pass

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

#	Field	Field Type	Description	Status
21	Merchant feedback	Button	Acquirer clicks to see merchant's feedback	Pass
22	Feedback's pagination	Table	Acquirer clicks on desired page number to change table page using pagination	Pass
23	Previous page	Button	Acquirer clicks to move on previous page	Pass
24	Merchant scoring	Sidebar Menu Link	Acquire clicks to see the score of the selected merchant	Pass
25	Dashboard	Sidebar Menu Link	Acquirer clicks on "dashboard" sidebar menu	Pass
26	Logout	Button	Acquirer logout	Pass

5.1.1.2 Insurance Pilot

In this section the Insurance Pilot application has been tested in accordance with the Pilot visual design and visual element's functionalities. Below table represents the GUI testing of an insurance application stated with Field, Field Type, Description, and status.

Table 3. Insurance Pilot GUI Testing

#	Field	Field Type	Description	Status
1	Broker Wallet ID	Textbox	Broker enters Wallet ID information	Pass
2	Password	Textbox	Broker enters Password information	Pass
3	Login	Button	Broker clicks on login button	Pass
4	Dashboard	Link	Broker clicks on Dashboard sidebar menu	Pass
5	Dashboard Contract View	Button	Broker clicks on contract view button in order to display contract details	Pass
6	Dashboard Contract Detail Modal Confirm	Button	Broker clicks on Confirm button in order to approve contracts coming from Users	Pass
7	Dashboard Contract Detail Modal Close	Button	Broker clicks on Close button to close modal	Pass
8	Dashboard Pagination	Table	Broker clicks on desired page number to change table page using pagination	Pass
9	Send New Proposal	Sidebar Menu Link	Broker clicks on Send New Proposal sidebar menu	Pass
10	Client Name	Textbox	Broker enters Client Name information	Pass
11	Email Address	Textbox	Broker enters Email Address information	Pass
12	Wallet ID	Textbox	Broker enters Wallet ID information	Pass
13	Phone Number	Textbox	Broker enters Phone Number information	Pass
14	Type of Insurance	Checkbox	Broker selects Type of Insurance	Pass
15	Address	Textbox	Broker enters address information	Pass
16	Total Insurance Value	Textbox	Broker enters Total Insurance Value information	Pass
17	Expiration Date	Calendar	Broker selects Expiration Date	Pass
18	Type of Covered Adversities	Checkbox	Broker selects Type of Covered Adversities	Pass
19	Premium Payment	Textbox	Broker Premium Payment value	Pass
20	Send New Proposal Button	Button	Broker clicks on Send New Proposal button to send proposal	Pass
21	Send New Proposal BlockUI	BlockUI	After broker clicks on Send New Proposal button BlockUI blocks all the page	Pass
22	Sidebar Menu - Proposal	Link	Broker clicks on Proposal sidebar menu	Pass

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

#	Field	Field Type	Description	Status
23	Proposal Contract Detail Modal View	Button	Broker clicks on View button to display Contract detail	Pass
24	Proposal Contract Detail Modal Close	Button	Broker clicks on Close button to close opened Modal	Pass
25	Proposal Pagination	Table	Broker clicks on desired page number to change table page using pagination	Pass
26	Sidebar Menu – Claim	Link	Broker clicks on Claim sidebar menu	Pass
27	Claim Contract View	Button	Broker clicks on View button to see claim contract detail	Pass
28	Claim Contract Modal Approve	Button	Broker clicks on Approve button to approve claim status coming from Users	Pass
29	Claim Contract Notification	Toaster Message	After Metamask transaction is done, toaster message will be displayed to broker	Pass
30	Claim Contract Modal Close	Button	Broker clicks on Close button to close opened Modal	Pass
31	Notification	Link	Broker clicks on Notification link to see contract notifications	Pass
32	Logout	Button	Broker clicks on Logout button	Pass
33	User Wallet ID	Textbox	User enters Wallet ID information	Pass
34	Password	Textbox	User enters Password information	Pass
35	Login	Button	User clicks on login button	Pass
36	Dashboard	Link	User clicks on Dashboard Sidebar Menu link	Pass
37	Dashboard Pagination	Table	User clicks on desired page number to change table page using pagination	Pass
38	Approve Contract	Button	User clicks on approve button to approve contract coming from Broker	Pass
39	Approve User Notification	Toaster Message	After Metamask transaction is done, toaster message will be displayed to user	Pass
40	Dashboard View Contract Detail	Button	User clicks on contract view button in order to display contract details	Pass
41	My insurance	Link	User clicks on My Insurance Sidebar Menu link	Pass
42	My Insurance Pagination	Table	User clicks on desired page number to change table page using pagination	Pass
43	My Insurance View	Button	User clicks on contract view button in order to display contract details	Pass
44	My Insurance Contract Detail Modal Close	Button	User clicks on Close button in order to close opened modal	Pass
45	My Insurance Claim	Button	User clicks on Claim button in order to send claim request to broker	Pass
46	My Insurance Claim Submit	Button	User clicks on Notification link to see contract notifications	Pass
47	My Insurance Claim Modal Close	Button	User clicks on Close button to close opened Modal	Pass
48	My Insurance Claim Notification	Toaster Message	After Metamask transaction is done, toaster message will be displayed to user	Pass
49	Notification	Link	User clicks on Notification link to see contract notifications	Pass
50	Logout	Button	User clicks on Logout button	Pass

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

5.1.1.3 Financial Market Infrastructure Pilot

In this section Financial Market Infrastructure Pilot application has been tested according to Pilot visual design and visual element functionalities. Below table represents the GUI testing of Financial Market Infrastructure application stated with Field, Field Type, Description, and status.

Table 4. Financial Market Infrastructure Pilot GUI Testing

#	Field	Field Type	Description	Status
1	Username	Textbox	User enters username information	Pass
2	Password	Textbox	User enters Password information	Pass
3	Login	Button	User clicks on login button	Pass
4	SPID authentication	Button	User authenticates through the Italian SPID	Pass
5	SPID confirmation	Button	User confirms its data available on SPID and access to the application	Pass
6	User profile	Sidebar Menu Link	User accesses his personal profile	Pass
7	User wallet	Sidebar Menu Link	User clicks on the wallet link	Pass
8	Available funds	Sidebar Menu Link	User accesses to the available funds in the application	Pass
9	Funds pagination	Table	User clicks on desired page number to change table page using pagination	Pass
10	Subscribe	Button	User decides to subscribe to one of the pension funds available in the pagination	Pass
11	Subscription form	Textbox	User inserts his information to subscribe to the funds (many data fields are prefilled)	Pass
12	Submit	Button	User submits the subscription	Pass
13	User wallet	Sidebar Menu Link	User clicks on the wallet link	Pass
14	Subscribed fund deposit	Button	User chooses to deposit cryptos to the subscribed fund	Pass
15	Value to deposit	Textbox	User inserts the amount to deposit	Pass
16	Confirm deposit	Button	User confirms the deposit	Pass

5.1.1.4 Toll Collection Pilot

In this section Toll Collection Pilot application has been tested according to Pilot visual design and visual element's functionalities. The table below represents the GUI testing of Toll Collection application stated with Field, Field Type, Description, and status.

Table 5. Toll Collection Pilot GUI Testing

#	Field	Field Type	Description	Status
1	Username	Textbox	User enters Wallet ID information	Pass
2	Password	Textbox	User enters Password information	Pass
3	Login	Button	User clicks on login button	Pass
4	Transaction Search	Link	User clicks on Transaction Search	Pass
5	Transaction Search Pagination	Table	User clicks on desired page number to change table page using pagination	Pass

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

#	Field	Field Type	Description	Status
6	Transaction ID Transaction Type Agency Date Operator Payment Method	Filter	User filters the table according to desired field	Pass
7	Operation Search	Link	User clicks on Operation Search	Pass
	Operation Search Pagination	Table	User clicks on desired page number to change table page using pagination	Pass
8	Operation ID Operation Type Transaction Date Location Agency Travel Document Type Payment Method	Filter	User filters the table according to desired field	Pass
9	Accounting Journal Search	Link	User clicks on Account Journal Search	Pass
10	Accounting Journal Pagination	Table	User clicks on desired page number to change table page using pagination	Pass
11	Summary ID Summary Name Account Journal Agency Account Journal Date Account Operation ID	Filter	User filters the table according to desired field	Pass
12	Logout	Button	User clicks on logout button	Pass

5.1.2 Regression Testing

Regression Testing is a type of testing that is carried out to verify that development and enhancements in the software do not impact the existing functionality of the product. Regression testing is to make sure the pilots work fine with functionalities, bug fixes, or any change in the existing features. Previously executed test cases are re-executed in order to verify the impact of change. Regression testing ensures that the unchanged parts of the pilots work well as well as the previous versions. The reason why regression testing is so important for the pilots is that there too many dependencies inside the pilots and existing functionalities must be performed with high standards. In addition, upcoming features must be built into the successful applications. Regression testing ensures that the existing applications are well performed, and new features can be developed continuously. In this deliverable context, the following tests ensures that the existing applications are well performed. Also, the linking, mapping, and the synchronisation of the blockchain components as well as with the other integrating components such as Authentication-as-a-Service and Backend system are working in the way expected.

In the following subsections the Regression Testing is done according to pilot functionalities. The standard table of the Regression Testing is presented according to the related pilot. The Table 6, 7, 8 and 9 presents the Test Case Name, Test Steps, Process Time of the Test Cases, Covered Requirement, and their statuses.

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

5.1.2.1 Banking Pilot

Table 6. Banking Pilot Regression Testing

Test Case Number	Test Case Name	Test Steps	Process Time	Covered Requirement	Status
TC1	Customer login	1. Go to https://cc-pilots-vm2.westeurope.cloudapp.azure.com/banking/ website 2. Click on customer 3. Enter private key 4. Click on login button 5. System will direct to the search transactions bar	462ms	Critical-Chains main framework Users' authenticity and authentication (REQ-L3-010)	Pass
TC2	Rate merchant	1. Find the transaction 2. See not scored status 3. Rate the merchant with positive/ negative feedbacks 4. Submit rating 5. See confirmation message	796ms	Smart contract trigger (REQ-L3-039) Scoring definition (REQ-L3-041)	Pass
TC3	Merchant login	1. Go to https://cc-pilots-vm2.westeurope.cloudapp.azure.com/banking/ website 2. Click on merchant 3. Enter private key 4. Click on login button 5. System will direct to the merchant dashboard	469ms	Critical-Chains main framework Users' authenticity and authentication (REQ-L3-010)	Pass
TC4	Merchant feedback/ scoring	1. Merchant clicks on feedbacks dashboard 2. See feedbacks status and valuation 3. Merchant clicks on scoring dashboard 4. See scoring updates	543ms	Check Analytics about Customer's Financial Behaviour (REQ-L0-029)	Pass
TC5	Acquirer login	1. Go to https://cc-pilots-vm2.westeurope.cloudapp.azure.com/banking/ website 2. Click on acquirer 3. Enter private key 4. Click on login button 5. System will direct to the acquirer dashboard	316ms	Critical-Chains main framework Users' authenticity and authentication (REQ-L3-010)	Pass
TC6	Acquirer monitoring of merchant feedbacks/ scoring	1. Select merchant 2. See merchant feedbacks 3. See merchant scoring 4. Continuous monitoring of merchant scoring	542ms	Check Analytics about Customer's Financial Behaviour (REQ-L0-029) Continuous scoring (REQ-L3-040)	Pass

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

5.1.2.2 Insurance Pilot

Table 7. Insurance Pilot Regression Testing

Test Case Number	Test Case Name	Test Steps	Process Time	Covered Requirement	Status
TC1	Broker Login	1. Go to https://criticalchains-test.vercel.app/ website 2. Click on Insurance 3. Click on Broker 4. Enter Wallet ID 5. Enter password 6. Click on Login button 7. System will direct to Insurance Broker Dashboard	335ms	Critical-Chains main framework Users' authenticity and authentication (REQ-L3-010)	Pass
TC2	Broker Dashboard	1. On dashboard page, contracts will be shown according to their Expiration Date, Policy Contract ID and Status 2. Only Active and Expired contract are shown in dashboard page	111ms	Critical-Chains main framework Users' authenticity and authentication (REQ-L3-010)	Pass
TC3	Send New Proposal	1. Click on Send New Proposal tab 2. Enter Client Name 3. Enter E-mail Address 4. Enter Wallet ID 5. Enter Phone Number 6. Select Type of Insurance (Crop Insurance pre-selected) 7. Enter Address 8. Enter Total Insured Value 9. Select Expiration Date from calendar 10. Select Type of Covered Adversities 11. Enter Premium Payment value 12. Click on Send New Proposal 13. Metamask will be open for the transaction fee 14. Send New Proposal transaction via Metamask	3320ms	Third-parties actions authorisation related to smart contracts (REQ-L3-015)	Pass
TC4	Broker Proposal	1. Click on Proposal tab 2. Contracts will be displayed on a table with columns; Expiration Date, Policy Contract ID, and Status (Approved, Expired) 3. Approved and Expired contract are read only 4. Click on View button to display contract details	186ms	Third-parties actions authorisation related to smart contracts (REQ-L3-015)	Pass

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Test Case Number	Test Case Name	Test Steps	Process Time	Covered Requirement	Status
TC5	Broker Claim	1. Click on Claim tab 2. Contracts will be displayed on a table with columns; Expiration Date, Policy Contact ID and Status (Claim, Claim Paid, Expired) 3. Claim Paid and Expired contract can be read only. In order to display details, click on View 4. Contracts which status are Claim can be approved. 5. Click on View which status is Claim 6. Click on Approve button 7. Metamask will be open for the transaction fee 8. Send "Claim Approve" transaction via Metamask	1010ms	Smart Contracts status changes (REQ-L3-035) Smart Contracts status changes (REQ-L3-035)	Pass
TC6	User Login	1. Go to https://criticalchains-test.vercel.app/ website 2. Click on Insurance 3. Click on User 4. Enter Wallet ID 5. Enter password 6. Click on Login button 7. System will direct to Insurance User Dashboard	372ms	Third-parties actions authorisation related to smart contracts (REQ-L3-015)	Pass
TC7	User Dashboard	1. On dashboard page contract will be displayed with columns; Expiration Date, Contract Number and Status (Active, Expired) 2. In order to approve contract, Click on View 3. A page will be displayed with the details of the contract 4. Click on Approve button 5. Metamask will be open for the transaction fee 6. Send "Approve" transaction via Metamask 7. System will direct user to My Insurance page	Approve Contract: 1473ms	Third-parties actions authorisation related to smart contracts (REQ-L3-015) Smart Contracts status changes (REQ-L3-035)	Pass
TC8	My Insurance	1. Click on My Insurance page 2. Contract details can be displayed, Claim operation can be done via My Insurance page. 3. To display details of the approved contract, click on View button	Claim: 3543ms	Dispute resolution (REQ-L3-016) Oracle verification for insurance (REQ-L3-034)	Pass

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Test Case Number	Test Case Name	Test Steps	Process Time	Covered Requirement	Status
		4. Detail mode will be opened responsively 5. To make a claim request click on Claim button 6. Select Type of Covered Adversities and click on Submit button. 7. Metamask will be open for the transaction fee 8. Send "Claim" transaction via Metamask			
TC9	User Notification	1. Notification is placed on the navigation bar near by the logout button 2. Under the notification menu Active and Claim contracts count will be displayed. Users can track their Active and Claim status contracts.	90ms	Notification system (REQ-L3-035)	Pass

5.1.2.3 *Financial Market Infrastructure Pilot*

Table 8. Financial Market Infrastructure Pilot Regression Testing

Test Case Number	Test Case Name	Test Steps	Process Time	Covered Requirement	Status
TC1	User login	1. Go to financial market infrastructure pilot website 2. Enter username and password or authenticate with SPID 3. Enter to user wallet	1206 ms	Critical-Chains main framework Users' authenticity and authentication (REQ-L3-010) Verify Individual's Identity (REQ-L0-009)	Pass
TC2	Acquisition of a Financial Digital Asset	1. Click on available funds 2. Select fund to subscribe 3. Click on subscribe button 4. Fill user data if not auto-filled 5. Submit acquisition	1264 ms	Subscription to a Fund by an individual (REQ-L0-002)	Pass
TC3	Deposit amount	1. Select the fund subscribed 2. Click deposit 3. Enter the amount to deposit 4. Confirm deposit	1432 ms	Subscription to a Fund by an individual (REQ-L0-002)	Pass
TC4	View amount invested	1. Click on wallet 2. See transaction history 3. See total amount invested	464 ms	Subscription to a Fund by an individual (REQ-L0-002)	Pass

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

5.1.2.4 Toll Collection Pilot

Table 9. Toll Collection Pilot Regression Testing

Test Case Number	Test Case Name	Test Steps	Process Time	Covered Requirement	Status
TC1	Login	1. Go to https://criticalchains-test.vercel.app/ website 2. Click on Toll 3. Click on Employee Online Portal 4. Enter username 5. Enter password 6. Click on Login button 7. System will direct to Toll Collection Dashboard 8. Transaction Search, Operation Search, Account Journal Search shown according to role-based access. User's only can see their data via login operation.	311ms	REQ-L3-031 - Backoffice system to transmit to other operator debts REQ-L3-032 - Search for accounting entry REQ-L3-033 - Search for a business transaction	Pass
TC2	Transaction Search	1. Click on Transaction Search menu located in sidebar 2. Transaction Search table will be displayed 3. The Transaction Search table consist of Transaction ID, Transaction Type, Agency, Source, Target, Date, Operator, Product, Product Profile, Amount, Travel Document, Travel Document Agency, Payment Method, Eco.Transaction Ref columns. 4. Transaction ID, Transaction Type, Agency, Date, Operator, Payment Method columns can be sorted according to preferred criteria.	5079ms	REQ-L3-004 - Remittance Send of TAG Transactions Detected in the Toll Highway REQ-L3-024 - A customer/driver drives should be identified by a travel document (a tag or the licence plate) being unique	Pass
TC3	Transaction Search Filter	1. Enter desirable Traction ID 2. System will sort the table according to entered Transaction ID 3. Select Transaction Type (RECH, SALE, TRIP, LEG)´ 4. System will sort the table according to Transaction Type selection 5. Enter Agency click on search 6. System will sort the table according to entered Agency. 7. Select date from calendar 8. System will sort the table according to date selection. 9. Enter desirable Operator, click on Search	538ms	REQ-L3-004 - Remittance Send of TAG Transactions Detected in the Toll Highway REQ-L3-024 - A customer/driver drives should be identified by a travel document (a tag or the licence plate) being unique	Pass

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Test Case Number	Test Case Name	Test Steps	Process Time	Covered Requirement	Status
		10. System will sort the table according to entered Operator information. 11. Select Payment Method (CASH, CREDIT CARD, BANK, PAYPAL) 12. System will sort the table according to Payment Method selection.			
TC4	Transaction Search Multiple Filters	1. Enter Transaction ID 2. Select Payment Method 3. System will sort the table according to selected criteria	538ms	REQ-L3-004 - Remittance Send of TAG Transactions Detected in the Toll Highway REQ-L3-024 - A customer/driver drives should be identified by a travel document (a tag or the licence plate) being unique	Fail
TC5	Operation Search	1. Click on Operation Search menu located in sidebar 2. Operation Search table will be displayed 3. The Operation Search table consist of Operation ID, Transaction ID, Operation Type, Transaction Date, Location Agency, Location, Travel Document Type, Travel Document, Travel Document Agency, Product, Product Agency, Payment Method, Amount columns. 4. Operation ID, Transaction ID, Operation Type, Transaction Date, Location Agency, Travel Document Type and Payment Method columns can be sorted according to preferred criteria.	2006ms	REQ-L3-029 - Authority should have access to the system to consult TAG transactions for audit purpose	Pass
TC6	Operation Search Filter	1. Enter desirable Operation ID 2. System will sort the table according to entered Operation ID 3. Select Transaction Type (RECH, SALE, TRIP, LEG) ´ 4. System will sort the table according to Transaction Type selection 5. Select Transaction Date from calendar 6. System will sort the table according to Transaction Date selection. 7. Enter Location Agency click on search	2006ms	REQ-L3-029 - Authority should have access to the system to consult TAG transactions for audit purpose	Pass

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Test Case Number	Test Case Name	Test Steps	Process Time	Covered Requirement	Status
		8. System will sort the table according to entered Location Agency information. 9. Select Travel Document Type (TAG, CONTACTLESS, CARD, TICKET), click on search 10. System will sort the table according to selected Travel Document Type. 11. Select Payment Method (CASH, CREDIT CARD, BANK, PAYPAL) 12. System will sort the table according to Payment Method selection.			
TC7	Operation Search Multiple Filters	1. Enter Operation ID 2. Select Payment Method 3. System will sort the table according to selected criteria	2006ms	REQ-L3-029 - Authority should have access to the system to consult TAG transactions for audit purpose	Fail
TC8	Account Journal Search	1. Click on Account Journal Search menu located in sidebar 2. The Account Journal Search table consist of Summary ID, Summary Name, Summary Description, Summary Start Date, Summary End Date, Account Journal Agency, Account Journal Date, Account, Concept, Operation ID, Operation Date, Amount columns. 3. Summary ID, Summary Name, Account Journal Agency, Account Journal Date, Account, Operation ID columns can be sorted according to preferred criteria.	2111ms	REQ-L3-031 - Backoffice system to transmit to other operator debts REQ-L3-032 - Search for accounting entry REQ-L3-033 - Search for a business transaction	Pass
TC9	Account Journal Search Filter	1. Enter desirable Summary ID 2. System will sort the table according to entered Summary ID. 3. Enter desirable Account Journal Agency, click on Search 4. System will sort the table according to entered Account Journal Agency. 5. Enter desirable Account, click on Search 6. System will sort the table according to entered Account information. 7. Enter desirable Operation ID, click on Search 8. System will sort the table according to entered Operation ID information.	2787ms	REQ-L3-031 - Backoffice system to transmit to other operator debts REQ-L3-032 - Search for accounting entry REQ-L3-033 - Search for a business transaction	Pass

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Test Case Number	Test Case Name	Test Steps	Process Time	Covered Requirement	Status
		9. For clear filter, click on Reset button. All filters have the Reset feature.			
TC10	Account Journal Search Multiple Filters	1. Enter Summary ID 2. Enter Account Journal ID 3. System will sort the table according to selected criteria	2787ms	REQ-L3-031 - Backoffice system to transmit to other operator debts REQ-L3-032 - Search for accounting entry REQ-L3-033 - Search for a business transaction	Fail

5.2 Web Application Non-functional Tests

Web application non-functional testing is one of the common testing methods used to check aspects such as performance, usability, reliability and so on. It is highly important for testing different pilots and pilot performance, web services, their installation and pilot integration with each other. There are different types of software non-functional testing methods followed by different developer groups. Non-Functional tests play an important role in ensuring that any usability limitations are addressed. In this deliverable, web services design compliance Testing, Installation Testing, Core Blockchain-as-a Service Standard Testing and Smart Contract Test approaches have been followed. Web service testing is to verify that the offered services follow the REST standard, and that their response time has also been measured. Installation testing is to ensure the integrated operation of all four pilots. Core Blockchain-as-a Service testing is to verify the availability and Smart Contract Tests is used to validate that the smart contracts meet the predefined standards.

5.2.1 Web Services Design Compliance Testing

REST APIs are the primary form of web service, used widely to power websites, mobile applications, and most enterprise integrations. Representational State Transfer (REST) is an architectural style that provides constraints which guide API design.

Commonly adopted principles of REST include:

- Client-server separation: the client determines how responses are displayed to the user, enabling the server to parse the request and produce the response.
- Stateless requests: the server does not have to store any context information between requests—everything needed is within each request.
- Resource identifiers: the interface is designed around resources that are identified by URLs.

The key abstraction of information in REST is a resource. Any information that can be named can be a resource: a document or image, a temporal service, a collection of other resources, a non-virtual object (e.g., a person), and so on. REST uses a resource identifier to identify the resource involved in an interaction between components.

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

The state of the resource at any particular timestamp is known as a resource representation. A representation consists of data, metadata describing the data and hypermedia links which can help the clients in transition to the next desired state.

The data format of a representation is known as a media type. The media type identifies a specification that defines how a representation is to be processed. A truly RESTful API looks like hypertext. Every addressable unit of information carries an address, either explicitly (e.g., link and id attributes) or implicitly (e.g., derived from the media type definition and representation structure). The resource representations shall be self-descriptive: the client does not need to know if a resource is employee or device. It should act based on the media-type associated with the resource.

A REST API should be entered with no prior knowledge beyond the initial URI (bookmark) and set of standardised media types that are appropriate for the intended audience (i.e., expected to be understood by any client that might use the API). From that point on, all application state transitions must be driven by client selection of server-provided choices that are present in the received representations or implied by the user's manipulation of those representations. The transitions may be determined (or limited by) the client's knowledge of media types and resource communication mechanisms, both of which may be improved on-the-fly (e.g., code-on-demand).

In the REST architectural style, data and functionality are considered resources and are accessed using Uniform Resource Identifiers (URIs). The resources are acted upon by using a set of simple, well-defined operations. The clients and servers exchange representations of resources by using a standardised interface and protocol – typically HTTP. Resources are decoupled from their representation so that their content can be accessed in a variety of formats, such as HTML, XML, plain text, PDF, JPEG, JSON, and others. Metadata about the resource is available and used, for example, to control caching, detect transmission errors, negotiate the appropriate representation format, and perform authentication or access control. Every interaction with a resource is stateless.

Most of the modern APIs driving data to web and mobile applications use HTTP [RFC7230] as their protocol. The health and uptime of these APIs determine availability of the applications themselves. In distributed systems built with a number of APIs, understanding the health status of the APIs and making corresponding decisions, for caching, failover or circuit-breaking, are essential to the ability to provide readily available solutions.

There exists a wide variety of operational software that relies on the ability to read the health check response of APIs. However, there is currently no standard for the health check output response, so most applications either rely on the basic level of information included in HTTP status codes [RFC7231] or use task-specific formats. In order to verify the web service health following response and object checks need to be done:

- API Health Response (status, version, releaseId, notes, checks, links, description)
- The Check Object (observedValue, observedUnit, affectedEndpoints, output, links, additional)

5.2.1.1 Critical-Chains RESTful Web Services Testing

In this section, the Critical-Chains web services were evaluated according to RESTful Web Service standards identified in Section 5.2.1 and the results are presented in Table 10, 11 and 12.

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Table 10. Toll Collection RESTful Web Service Testing

Toll Collection Application – 1		
API Health Response	Status	Pass
	Version	V 1.0
	Release ID	Set transaction tx
	Notes	Request Time Out 20000
	Checks	Response Time: 1189ms
	Link	https://rinkeby.etherscan.io/tx/0xcaeb2af6b14a7aa4734dde6f947b7158c6f9e206e989908e19aab341bd0a3088
	Description	Sets the transaction information
The Check Objects	Observed Value	250
	Observed Unit	Ms
	Affected End Points	id, transactionId, transactionTypeId, agencyId, sourceId, destId, destDateUtc, Legs, transitId, transDate, placeId, travelDoc, categoryDet, operatorId, operatorId, productId, productCode, productAgencyId, productProfile, amount, travelDocId, travelDocAgencyId, paymentMethod, economicTransactionRef

	Output	{
		  "method": "post",
		  "url": "https://jboss.germanywestcentral.cloudapp.azure.com/setTransactionmethod",
		  "headers": {
		    "Content-Type": "application/json",
		    "Request-Timeout": "20000"
		  },
		  "body": {
		    "id": 0,
		    "transactionId": "string",
		    "transactionTypeId": "string",
		    "agencyId": "string",
		    "sourceId": "string",
		    "destId": "string",
		    "destDateUtc": "string",
		    "legs": 0,
		    "stretches": [
		      {
		        "transitId": "string",
		        "transDate": "string",
		        "placeId": "string",
		        "travelDoc": "string",
		        "categoryDet": 0,
		        "operatorId": "string"
		      }
		    ],
		    "operatorId": "string",
		    "productId": "string",
		    "productCode": "string",
		    "productAgencyId": "string",
		    "productProfile": "string",
		    "amount": 0,
		    "travelDocId": "string",
		    "travelDocAgencyId": "string",
		    "paymentMethod": "string",
		    "economicTransactionRef": "string"
		  }
		}

	Links	https://rinkeby.etherscan.io/tx/0xcaeb2af6b14a7aa4734dde6f947b7158c6f9e206e989908e19aab341bd0a3088

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Toll Collection Application – 2		
API Health Response	Status	Pass
	Version	V 1.0
	Release ID	Set operation tx
	Notes	Request Time Out 20000
	Checks	Response Time: 710 ms
	Link	"https://rinkeby.etherscan.io/tx/0x9f0222d9777ac64c424b959a2e000d4d847c52c3f4e1efa612abcc84c0390723"
	Description	Sets the operation information
The Check Objects	Observed Value	710
	Observed Unit	ms
	Affected End Points	id, travelDocumentType, travelDocumentId", travelDocumentAgencyId, locationAgencyId, locationId, transactionId, transactionDate, productId, productAgencyId, operationTypeId, classificationMatrixId, paymentMethodId, amount

	Output	{
		 "method": "post",
		 "url": "https://jboss.germanywestcentral.cloudapp.azure.com/setOperation",
		 "headers": {
		  "Content-Type": "application/json"
		 },
		 "body": {
		  "id": 0,
		  "travelDocumentType": "string",
		  "travelDocumentId": "string",
		  "travelDocumentAgencyId": "string",
		  "locationAgencyId": "string",
		  "locationId": "string",
		  "transactionId": "string",
		  "transactionDate": "string",
		  "productId": "string",
		  "productAgencyId": "string",
		  "operationTypeId": "string",
		  "classificationMatrixId": 0,
		  "paymentMethodId": "string",
		  "amount": 0
		 }
		}

	Links	"https://rinkeby.etherscan.io/tx/0x9f0222d9777ac64c424b959a2e000d4d847c52c3f4e1efa612abcc84c0390723"

Toll Collection Application – 3		
API Health Response	Status	Pass
	Version	V 1.0
	Release ID	Set account tx
	Notes	Request Time Out 200000
	Checks	Response Time : 625 ms
	Link	"https://rinkeby.etherscan.io/tx/0xcbd73cd08d334e6e34b3f392eb46017fd05c14cc84366565177a0d6a2b26e86a"
	Description	Sets the account information
The Check Objects	Observed Value	625
	Observed Unit	ms
	Affected End Points	summaryId, summaryName, summaryDescription, sumaryStartDate, sumaryEndDate, accountJournalAgencyId, accountJournalDate, accountId, concept, operationId, entryDate, amount

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

	Output	{ "method": "post",
		 "url": "https://jboss.germanywestcentral.cloudapp.azure.com/setaccount",
		 "headers": { "Content-Type": "application/json" },
		 "body": { "summaryId": 0,
		   "summaryName": "string",
		   "summaryDescription": "string",
		  "sumaryStartDate": "string",
		   "sumaryEndDate": "string",
		   "accountJournalAgencyId": "string",
		   "accountJournalDate": "string",
		   "accountId": "string",
		   "concept": "string",
		   "operationId": 0,
		"entryDate": "string",
		   "amount": 0
		 }}

	Links	"https://rinkeby.etherscan.io/tx/0xcbd73cd08d334e6e34b3f392eb46017fd05c14cc84366565177a0d6a2b26e86a"
Toll Collection Application – 4		
API Health Response	Status	Pass
	Version	V 1.0
	Release ID	Get operation tx
	Notes	Request Time Out 200000
	Checks	Response Time: 2276ms
	Link	https://jboss.germanywestcentral.cloudapp.azure.com/get
	Description	Gets all operation information
The Check Objects	Observed Value	2276
	Observed Unit	ms
	Affected End Points	-

	Output	{
		 "raw": [
		  [
		   "11",
		   [
		    "traveldoctype",
		    "traveldocagencyid",
		    "dsadsa",
		    "locationagencyid",
		    "locationid"
		   ],
		   [
		    "transactionid",
		    "transactiondate",
		    "productdata",
		    "productagencyid"
		   ],
		   "operationtypeid",
		   "123",
		   "paymentmethodid",
		   "12313.32"
		  ]

	Links	https://jboss.germanywestcentral.cloudapp.azure.com/get
Toll Collection Application – 5		
	Status	Pass

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

API Health Response	Version	V 1.0
	Release ID	Get transaction tx
	Notes	Request Time Out 200000
	Checks	Response Time: 5684ms
	Link	https://jboss.germanywestcentral.cloudapp.azure.com/getTransactionmethod
	Description	Gets all transaction information
The Check Objects	Observed Value	5684
	Observed Unit	ms
	Affected End Points	-
	Output	{ "raw": [[["2", "123456789", "TRIP", "CRTM", "1234"], ["4321", "2019-12-04T05:58:33.454Z", "2", [["asd", "asd", "asd", "asd", "asd", "asd"], ["asd", "asd", "asd", "asd", "asd", "asd"]], "CRTM"], ["TAG", "TAGPOS", "CRTM", "COMMON", "123.55"], ["12", "1", "CASH", "-"]], [
	Links	https://jboss.germanywestcentral.cloudapp.azure.com/getTransactionmethod
Toll Collection Application – 6		
API Health Response	Status	Pass
	Version	V 1.0
	Release ID	Get account tx
	Notes	Request Time Out 200000
	Checks	Response Time: 1628ms
	Link	https://jboss.germanywestcentral.cloudapp.azure.com/getaccount
	Description	Gets all account journal information
The Check Objects	Observed Value	1628
	Observed Unit	ms
	Affected End Points	-

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

	Output	
		```
		{
		 "raw": [
		  [
		   "23",
		   [
		    "asdsa",
		    "dasdsa",
		    "dsadsa",
		    "dsadsa"
		   ],
		   [
		    "dsasda",
		    "dsadsa",
		    "dsadas",
		    "dsadsa"
		   ],
		   "dasdsa",
		   "opdate",
		   "321321.2"
		  ]
		```

	Links	https://jboss.germanywestcentral.cloudapp.azure.com/getaccount

Table 11. Insurance Application RESTful Web Services Testing

Insurance Application – 1		
API Health Response	Status	Pass
	Version	V 1.0
	Release ID	Create a Deploy New Policy
	Notes	Request Time Out 200000
	Checks	Response Time: 463ms
	Link	https://jboss.germanywestcentral.cloudapp.azure.com/deployNewPolicy
	Description	Deletes all subscriptions
The Check Objects	Observed Value	463
	Observed Unit	ms
	Affected End Points	"TypeOfCoveredAdverties, status, issuer, client, fullname, email, phone, Expiration Date, TotalInsuredValue, PremiumPayment, requestedAmount, ReimbersumenetValue

	Output	
		```
		{
		 "TypeOfCoveredAdverties": [
		  0
		 ],
		 "status": 0,
		 "issuer": "0x0f09D615560Dd3d47d23Cf243C010b783D11B243",
		 "client": "0xBc536ed0C97D24e6D48209c238d0806be98012df ",
		 "fullname": "tes test",
		 "email": "test@test.com",
		 "phone": "009099999999",
		 "ExpirationDate": 1111221,
		 "TotalInsuredValue": 10000,
		 "PremiumPayment": 1000,
		 "requestedAmount": 0,
		 "ReimbursementValue": 0
		}
		```

	Link	https://jboss.germanywestcentral.cloudapp.azure.com/deployNewPolicy

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Table 12. Financial Market Infrastructure Application RESTful Web Services Testing

Financial Market Infrastructure Application - 1		
API Health Response	Status	Pass
	Version	V 1.0
	Release ID	Get All Assets
	Notes	-
	Checks	Response Time: 315 ms
	Link	https://main-ff-bc-ctp651i8qlrdcfmh-gtw.qovery.io/api/assets
	Description	Gets the list of Financial Assets
The Check Objects	Observed Value	315
	Observed Unit	ms
	Affected End Points	Id, title, description, provider, address_provider, address_deposit_contract, createdAt, UpdatedAt
	Output	-
	Links	https://main-ff-bc-ctp651i8qlrdcfmh-gtw.qovery.io/api/assets
Financial Market Infrastructure Application - 2		
API Health Response	Status	Pass
	Version	V 1.0
	Release ID	Delete All Assets
	Notes	-
	Checks	Response Time: 215 ms
	Link	https://main-ff-bc-ctp651i8qlrdcfmh-gtw.qovery.io/api/assets
	Description	Gets the list of Financial Assets
The Check Objects	Observed Value	215
	Observed Unit	ms
	Affected End Points	All assets
	Output	-
	Links	https://main-ff-bc-ctp651i8qlrdcfmh-gtw.qovery.io/api/assets
Financial Market Infrastructure Application - 3		
API Health Response	Status	Pass
	Version	V 1.0
	Release ID	Create Assets
	Notes	-
	Checks	Response Time: 963 ms
	Link	https://main-ff-bc-ctp651i8qlrdcfmh-gtw.qovery.io/api/assets
	Description	Creates asset
The Check Objects	Observed Value	963
	Observed Unit	ms
	Affected End Points	title, description, terms, provider name, provider address, deposit contract address

	Output	{
		"_id": "60c3004ea7924b001938d45e",
		"title": "string",
		"description": "string",
		"terms": "string",
		"provider": "string",
		"address_provider": "string",
		"address_deposit_contract": "string",
		"createdAt": "2021-06-11T06:18:54.256Z",
		"updatedAt": "2021-06-11T06:18:54.256Z",
		"__v": 0
		}

	Link	https://main-ff-bc-ctp651i8qlrdcfmh-gtw.qovery.io/api/assets

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Financial Market Infrastructure Application - 4		
API Health Response	Status	Pass
	Version	V 1.0
	Release ID	Find one Asset
	Notes	-
	Checks	Response Time: 602 ms
	Link	https://main-ff-bc-ctp651i8qlrdcfmh-gtw.qovery.io/api/assets
	Description	Creates asset
The Check Objects	Observed Value	602
	Observed Unit	ms
	Affected End Points	title, description, terms, provider name, provider address, deposit contract address
	Output	[{ "_id": "60cb3bfc6587d36fe0711a14", "title": "Test Alternative Pension Fund", "description": "Pension funds are financial intermediaries which offer social insurance by providing income to the insured persons following their retirement. Often, they also provide death and disability benefits. Pension schemes are important cornerstones of European households' income during retirement. Pension funds also play a role in financial markets as institutional investors. Pension funds typically aggregate large sums of money to be invested into the capital markets, such as stock and bond markets, to generate profit (returns). A pension fund represents an institutional investor and invests large pools of money into private and public companies. Pension funds are typically managed by companies (employers). The main goal of a pension fund is to ensure there will be enough money to cover the pensions of employees after their retirement in the future. Pension funds were created to provide employees with a supplementary pension, in addition to the pension paid by public social security institutions. State pensions usually operate through the principle of redistribution (workers' social security contributions fund the pensions of already retired workers). Through pension funds, workers accrue a portion of the income they receive during their working life, according to the principle of capitalisation: regular payments by the subscribers contribute to a fund managed along financial lines, according to the clients' risk tolerance. Clients can choose among various types of managed funds: shares, bonds or balanced funds. At the end of the agreement, typically a long-term contract, ideally lasting the full working life of the insured, a monthly annuity is paid. This varies according to the contributions made, the duration of the policy and the return on the investment. Pension funds can be open or closed (also known as negotiated funds). Open funds are available to any person in employment, while closed funds are reserved for specific professional categories, as the funds are established based on agreements between trade unions and business organisations for specific industries. Further individual supplementary pension plans are available to anyone wishing to create a supplementary pension (including, for example, people not in employment or students). These are personal pension plans, which provide more investment flexibility and allow people to stop, and later resume, payments into the pension without penalties and without having to break the contract. Payment of benefits to the policyholder are ruled by the same constraints are for collective funds (e.g. full payment into a lump sum is not allowed).", "terms": "You'll open your Pension by completing the online application. When you do this, we will set up an account for you and you will become a member of the scheme. The scheme provides pension accounts for many customers, of whom you are one. The money that you pay in builds up to create your pension pot. The Legal & General Personal Pension is suitable for eligible customers who want to save towards their retirement and are happy to make their own investment decisions. You will have access to a range of investment options to meet your attitude to risk as well as online support. The scheme was established by a declaration of trust and is governed by a set of scheme rules which are required for it to be a registered pension scheme. The investments and money in the scheme are held by the trustee in the trustee's name. Benefits under the scheme are payable by the trustee on our instruction. We will automatically claim basic rate tax relief from the government, based on the value of your payment, and we will add it to your Pension by no later than the end of business the following working day. If you pay one of the higher rates of income tax, you may be entitled to receive tax relief at the higher rate, but you will need to claim the additional relief through your annual self-assessment tax return. You will select how your

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

		Pension is invested according to your preferred level of risk and the Funds available. Your investment will be administered by us, and assets will be held by the trustee. We may use other firms to support us in the performance of our administration duties. From time to time, we may add or remove assets. Where regulation and/or changes to legislation materially increase the cost and/or the complexity of the administration to us of holding a particular asset. We cannot accept any responsibility for losses, costs and/or legal fees that may be incurred as a result of your investment choices. The value of your Pension savings can go down as well as up. We may change the Funds available for you to invest in your Pension from time to time. If we remove a Fund from the range available in your Pension it may mean that you need to choose an alternative Fund. If the Fund you are invested in is removed from the available Funds, we may either ask you to choose a new Fund or we may automatically move, you to an alternative Fund as we deem appropriate. We will notify you by email if the Funds available change and if the Fund you are invested in is affected, giving you at least 30 days' notice before any such change takes effect. Any cash you hold in your Pension that is not invested will be held in a bank account in the name of the trustee. You will be able to see the value of the cash held in your Pension at any time, by logging onto My Account. The bank account your cash is held in does not attract interest and therefore you will not receive any return on any cash balance you hold in your Pension. ", "provider": "Financial Provider", "address_provider": "0xf215b88ba88d486ade72736b9378bf4880699a", "address_deposit_contract": "0xf215b88ba88d486ade72736b9378bf4880699a", "createdAt": "2021-06-17T12:11:40.593Z", "updatedAt": "2021-06-17T12:11:40.593Z", "__v": 0 }]
	Link	https://main-ff-bc-ctp651i8qlrdcfmh-gtw.qovery.io/api/assets
Financial Market Infrastructure Application – 5		
API Health Response	Status	Pass
	Version	V 1.0
	Release ID	Get All Subscriptions
	Notes	-
	Checks	Response Time : 651 ms
	Link	https://main-ff-bc-ctp651i8qlrdcfmh-gtw.qovery.io/api/customers/published
	Description	Gets all published customers
The Check Objects	Observed Value	651
	Observed Unit	ms
	Affected End Points	given_name, family_name, fiscal_number, country_of_birth, place_of_birth, gender, mobile_phone, asset_title, asset_description, asset_terms, asset_provider, asset_address_deposit_contract

	Output	{
		}

	Link	https://main-ff-bc-ctp651i8qlrdcfmh-gtw.qovery.io/api/customers/published
Financial Market Infrastructure Application – 6		
API Health Response	Status	Pass
	Version	V 1.0
	Release ID	Delete All Subscriptions
	Notes	-
	Checks	Response Time: 417 ms
	Link	"https://main-ff-bc-ctp651i8qlrdcfmh-gtw.qovery.io/api/transactions
	Description	Deletes all subscriptions
The Check Objects	Observed Value	417
	Observed Unit	Ms
	Affected End Points	-

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

	Output	{
		}

	Link	"https://main-ff-bc-ctp651i8qlrdcfmh-gtw.qovery.io/api/transactions
Financial Market Infrastructure Application – 7		
API Health Response	Status	Pass
	Version	V 1.0
	Release ID	Create a Subscription
	Notes	-
	Checks	Response Time : 725 ms
	Link	https://main-ff-bc-ctp651i8qlrdcfmh-gtw.qovery.io/api/transactions
	Description	Deletes all subscriptions
The Check Objects	Observed Value	725
	Observed Unit	ms
	Affected End Points	given_name, family_name, fiscal_number,country_of_birth, place_of_birth, gender,mobile_phone, asset_title, asset_description, asset_terms, asset_provider, asset_address_deposit_contract
	Output	"_id": "60c35ee9a7924b001938d461", "from_address": "string", "from_name": "string", "to_address": "string", "to_name": "string", "description": "string", "date": "string", "amount": "string", "createdAt": "2021-06-11T13:02:33.489Z", "updatedAt": "2021-06-11T13:02:33.489Z", "__v": 0
	Link	"https://main-ff-bc-ctp651i8qlrdcfmh-gtw.qovery.io/api/transactions
Financial Market Infrastructure Application – 8		
API Health Response	Status	Pass
	Version	V 1.0
	Release ID	Find one Subscription
	Notes	-
	Checks	Response Time : 603 ms
	Link	https://main-ff-bc-ctp651i8qlrdcfmh-gtw.qovery.io/api/customers/id
	Description	Deletes all subscriptions
The Check Objects	Observed Value	603
	Observed Unit	Ms
	Affected End Points	given_name, family_name, fiscal_number, country_of_birth,place_of_birth, gender,mobile_phone,asset_title, asset_description, asset_terms, asset_provider, asset_address_deposit_contract
	Output	[{ "_id": "60cb51626587d36fe0711a15", "given_name": "Lucrezia", "family_name": "Borgia", "fiscal_number": "BRGLRZ80D58H501Q", "contry_of_birth": "FE", "place_of_birth": "Ferrara", "gender": "F", "mobile_phone": "36482716632", "asset_title": "Test Alternative Pension Fund", "asset_description": "Pension funds are financial intermediaries which offer social insurance by providing income to the insured persons following their retirement. Often, they also provide death and disability benefits. Pension schemes are important cornerstones of European households' income during retirement. Pension funds also play a role in financial markets as

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

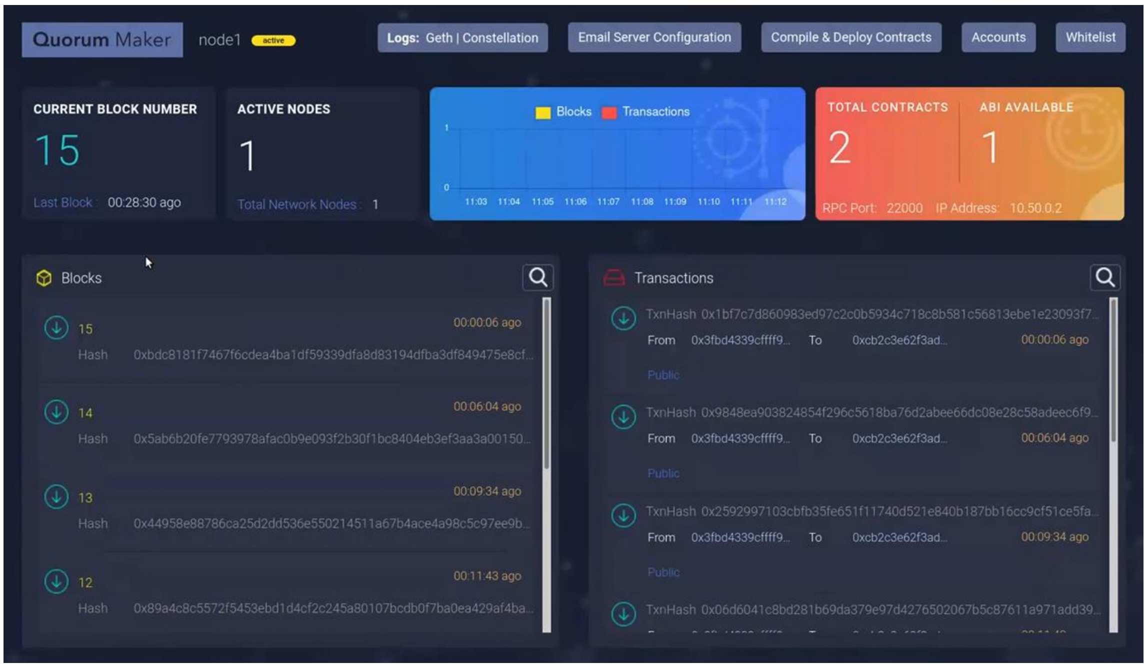
institutional investors. Pension funds typically aggregate large sums of money to be invested into the capital markets, such as stock and bond markets, to generate profit (returns). A pension fund represents an institutional investor and invests large pools of money into private and public companies. Pension funds are typically managed by companies (employers). The main goal of a pension fund is to ensure there will be enough money to cover the pensions of employees after their retirement in the future. Pension funds were created to provide employees with a supplementary pension, in addition to the pension paid by public social security institutions. State pensions usually operate through the principle of redistribution (workers' social security contributions fund the pensions of already retired workers). Through pension funds, workers accrue a portion of the income they receive during their working life, according to the principle of capitalisation: regular payments by the subscribers contribute to a fund managed along financial lines, according to the clients' risk tolerance. Clients can choose among various types of managed funds: shares, bonds, or balanced funds. At the end of the agreement, typically a long-term contract, ideally lasting the full working life of the insured, a monthly annuity is paid. This varies according to the contributions made, the duration of the policy and the return on the investment. Pension funds can be open or closed (also known as negotiated funds). Open funds are available to any person in employment, while closed funds are reserved for specific professional categories, as the funds are established based on agreements between trade unions and business organisations for specific industries. Further individual supplementary pension plans are available to anyone wishing to create a supplementary pension (including, for example, people not in employment or students). These are personal pension plans, which provide more investment flexibility and allow people to stop, and later resume, payments into the pension without penalties and without having to break the contract. Payment of benefits to the policyholder are ruled by the same constraints are for collective funds (e.g., full payment into a lump sum is not allowed).",
    "asset_terms": "You'll open your Pension by completing the online application. When you do this, we will set up an account for you and you will become a member of the scheme. The scheme provides pension accounts for many customers, of whom you are one. The money that you pay in builds up to create your pension pot. The Legal & General Personal Pension is suitable for eligible customers who want to save towards their retirement and are happy to make their own investment decisions. You will have access to a range of investment options to meet your attitude to risk as well as online support. The scheme was established by a declaration of trust and is governed by a set of scheme rules which are required for it to be a registered pension scheme. The investments and money in the scheme are held by the trustee in the trustee's name. Benefits under the scheme are payable by the trustee on our instruction. We will automatically claim basic rate tax relief from the government, based on the value of your payment, and we will add it to your Pension by no later than the end of business the following working day. If you pay one of the higher rates of income tax, you may be entitled to receive tax relief at the higher rate, but you will need to claim the additional relief through your annual self-assessment tax return. You will select how your Pension is invested according to your preferred level of risk and the Funds available. Your investment will be administered by us and assets will be held by the trustee. We may use other firms to support us in the performance of our administration duties. From time to time, we may add or remove assets. Where regulation and/or changes to legislation materially increase the cost and/or the complexity of the administration to us of holding a particular asset. We cannot accept any responsibility for losses, costs and/or legal fees that may be incurred as a result of your investment choices. The value of your Pension savings can go down as well as up. We may change the Funds available for you to invest in your Pension from time to time. If we remove a Fund from the range available in your Pension it may mean that you need to choose an alternative Fund. If the Fund you are invested in is removed from the available Funds, we may either ask you to choose a new Fund or we may automatically move, you to an alternative Fund as we deem appropriate. We will notify you by email if the Funds available change and if the Fund you are invested in is affected, giving you at least 30 days' notice before any such change takes effect. Any cash you hold in your Pension that is not invested will be held in a bank account in the name of the trustee. You will be able to see the value of the cash held in your Pension at any time, by logging onto My Account. The bank account your cash is held in does not attract interest and therefore you will not receive any return on any cash balance you hold in your Pension. ",
    "asset_provider": "Financial Provider",
    "asset_address_deposit_contract": "0xf215b88ba88d486ade72736b9378bf4880699a",
    "createdAt": "2021-06-17T13:42:58.487Z",
    "updatedAt": "2021-06-17T13:42:58.487Z",
    "__v": 0
  }

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

		]
	Link	https://main-ff-bc-ctp651i8qlrdcfmh-gtw.qovery.io/api/customers/id

5.2.2 Installation Testing

Installation testing is performed to check if the software has been correctly installed with all the inherent features and that the product is working as per expectations. As presented in Regression Testing Section, the pilots have examined as a whole, and outcomes evaluated accordingly. Therefore, the Figure 16 presented the successful installation of the Quorum Blockchain network to the Critical-Chains infrastructure. Following section will cover the tests of the Blockchain Core Components in more detail.



CURRENT BLOCK NUMBER 15
Last Block : 00:28:30 ago

ACTIVE NODES 1
Total Network Nodes : 1

TOTAL CONTRACTS 2
ABI AVAILABLE 1
RPC Port: 22000 IP Address: 10.50.0.2

Blocks

```
15                                                        00:00:06 ago
Hash    0xbdc8181f7467f6cdea4ba1df59339dfa8d83194dfba3df849475e8cf...
14                                                        00:06:04 ago
Hash    0x5ab6b20fe7793978afac0b9e093f2b30f1bc8404eb3ef3aa3a00150...
13                                                        00:09:34 ago
Hash    0x44958e88786ca25d2dd536e550214511a67b4ace4a98c5c97ee9b...
12                                                        00:11:43 ago
Hash    0x89a4c8c5572f5453ebd1d4cf2c245a80107bcdb0f7ba0ea429af4ba...
```

Transactions

```
TxnHash 0x1bf7c7d860983ed97c2c0b5934c718c8b581c56813ebe1e23093f7...
From  0x3fbd4339cffff9...  To  0xcb2c3e62f3ad...        00:00:06 ago
Public
TxnHash 0x9848ea903824854f296c5618ba76d2abee66dc08e28c58adeec6f9...
From  0x3fbd4339cffff9...  To  0xcb2c3e62f3ad...        00:06:04 ago
Public
TxnHash 0x2592997103cbfb35fe651f11740d521e840b187bb16cc9cf51ce5fa...
From  0x3fbd4339cffff9...  To  0xcb2c3e62f3ad...        00:09:34 ago
Public
TxnHash 0x06d6041c8bd281b69da379e97d4276502067b5c87611a971add39...
```

Figure 16. Quorum interface, Blocks are on the left and Transactions are on the right.

5.2.3 Core Blockchain-as-a-Service Standards Testing

The Blockchain-as-a-Service solution of the Critical-Chains project depends on the functioning of three independent components – the Quorum blockchain network dedicated to the project, the KSI Blockchain service, and a PostgreSQL database instance.

As all these components are already mature and tested, we perform only basic tests to verify that the services provided by these components are indeed available from the production environment. When the basic validation request receives an expected response, we assume these components are working properly and focus on the validation of the Quorum Integrity functionality.

To validate the availability of the baseline functionality, three sets of tests are performed. First, the availability of the Quorum network is tested with the following steps:

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Table 13. Availability of the Quorum network

Check Item	Status
Connect to a Quorum node using the geth command line utility.	The connection should be successful.
Request the headers of the latest block using the geth tool.	The request should return the headers of the last Quorum block.

Second, the basic availability of KSI Gateway is tested with the following steps:

Table 14. Availability of KSI Gateway

Check Item	Status
The command line tool ksi is used to request a KSI signature from the KSI Gateway dedicated to the Critical-Chains network.	The signature request should return a signature within one second.
An unextended KSI signature that is created before the latest KSI publication is extended to the latest publication using the ksi tool and the KSI Gateway dedicated to the Critical-Chains project.	The request should successfully extend the signature.
The extended signature from the previous step is verified offline against the latest publication file using the KSI tool.	The signature verification should succeed.

Third, the PostgreSQL database should be available. To validate it is the case, the following test is performed:

Table 15. Availability of the PostgreSQL Database

Check Item	Status
Connect to the PostgreSQL database using the psql command line tool with the credentials used by the Quorum Integrity solution.	The connection should be successful.

When the tests described above succeed, we will assume the preconditions of the Quorum Integrity solution are fulfilled and the testing of the Quorum Integrity solution can proceed. The code implementing the Quorum Integrity solution is covered with unit tests which are executed during the development and CI/CD process. It is not necessary to re-run these tests in the production environment.

5.2.3.1 KSI Standards and Testing

Before starting the tests of Quorum Integrity, it is recommended to perform the test steps described in Section 5.2.3 to ensure the dependencies are available and to simplify the troubleshooting process in case of potential issues.

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

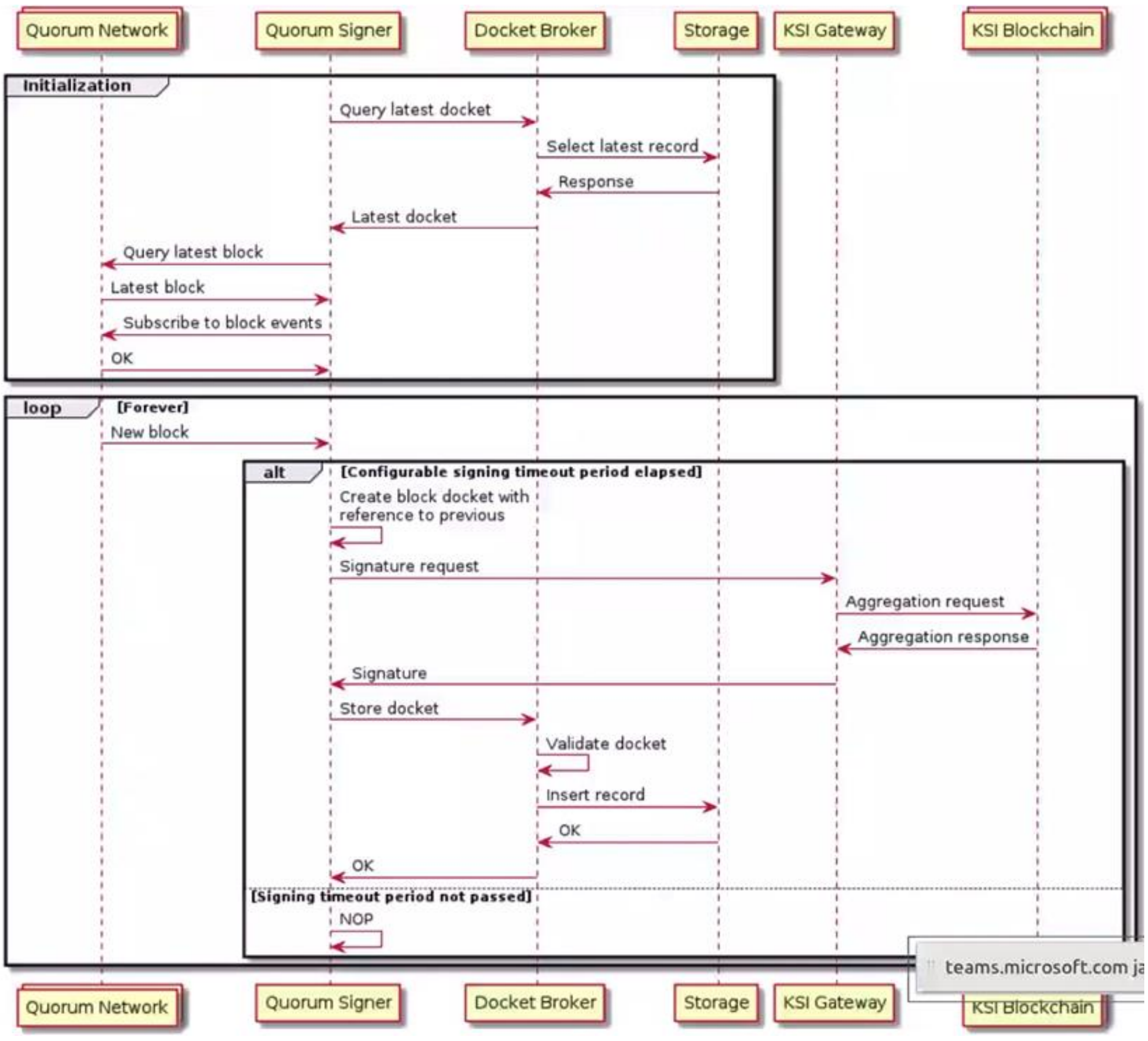


Figure 17. Flow of events, followed by tests

As the Quorum Integrity solution consists of two parts – quorum-signer and quorum-verifier, these parts are tested separately. The functioning of these applications is described in the sequence diagrams shown in Figures 10 and 11, respectively.

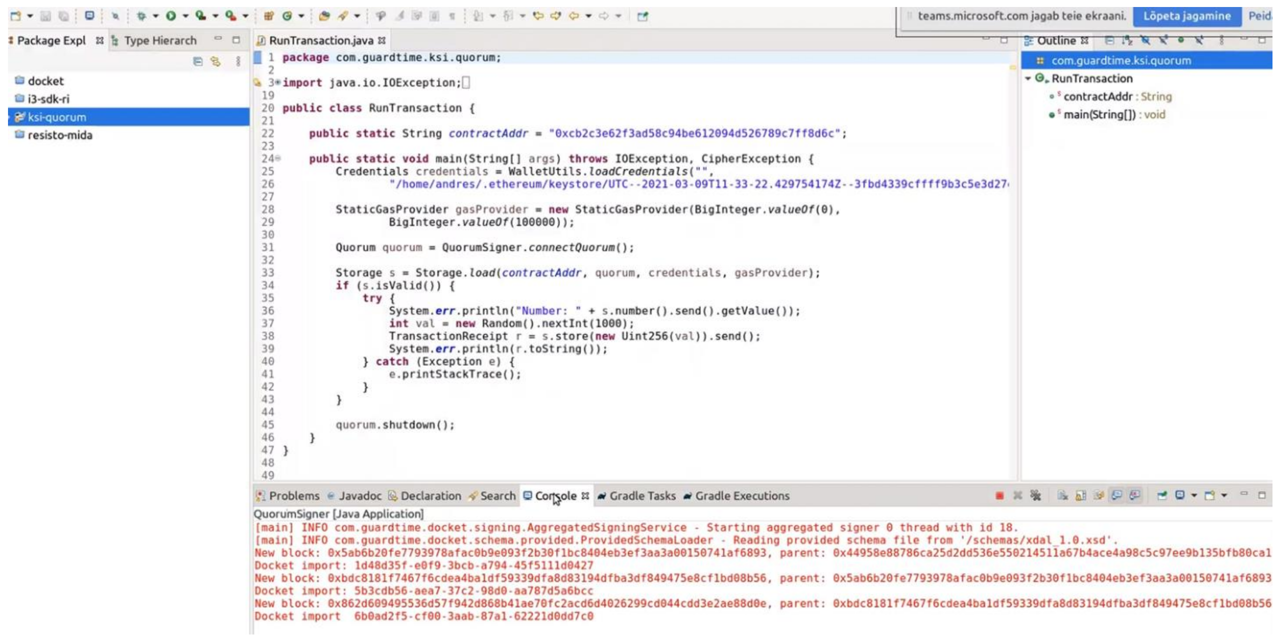
Table 16. Quorum-signer Test

Check Item	Status
The quorum-signer is intended to be running continuously as a system daemon managed by systemd.	Pass
The systemd unit file is provided as part of the installation package.	Pass
The quorum-signer process connects to the configured Quorum node(s) and listens to new block events.	Pass
According to the configured time interval the quorum-signer process creates docket objects (containing Quorum block hashes, KSI signatures and references to previous dockets) and stores the dockets into the PostgreSQL database.	Pass
To ensure the process is working the following steps are performed: The process is enabled: systemctl enable quorum-signer	Pass

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

Check Item	Status
The process is started: systemctl start quorum-signer	Pass
The status of the process is checked: systemctl status quorum-signer.	The status should be “active”.
A transaction is submitted (by the tester or some other process) to the Quorum network in order to trigger the creation of a new block.	After the configured signing timeout, the PostgreSQL database is queried to verify the number of dockets has increased.
The log file of the quorum-signer is checked for any error or warning messages.	In the case of the presence of error or warning messages, the issues must be resolved according to the information provided in the messages.
The functioning of quorum-verifier is tested. As this application can be run as an interactive command line tool, the testing consists of just running the tool and inspecting the output.	If the tool reports a successful run, the system is set up properly. In case of any errors, the errors are reported to the console and must be resolved before the system can be considered fully functional.

Accordingly, the Figures 18, 19 and 20 show the steps of the test. The tests start with new block generation in the Quorum Blockchain. On the Figure 18 the docket import script can be seen. Red log shows new block and parent code. Also, Docket import code is next. The Log shows three new block generations.



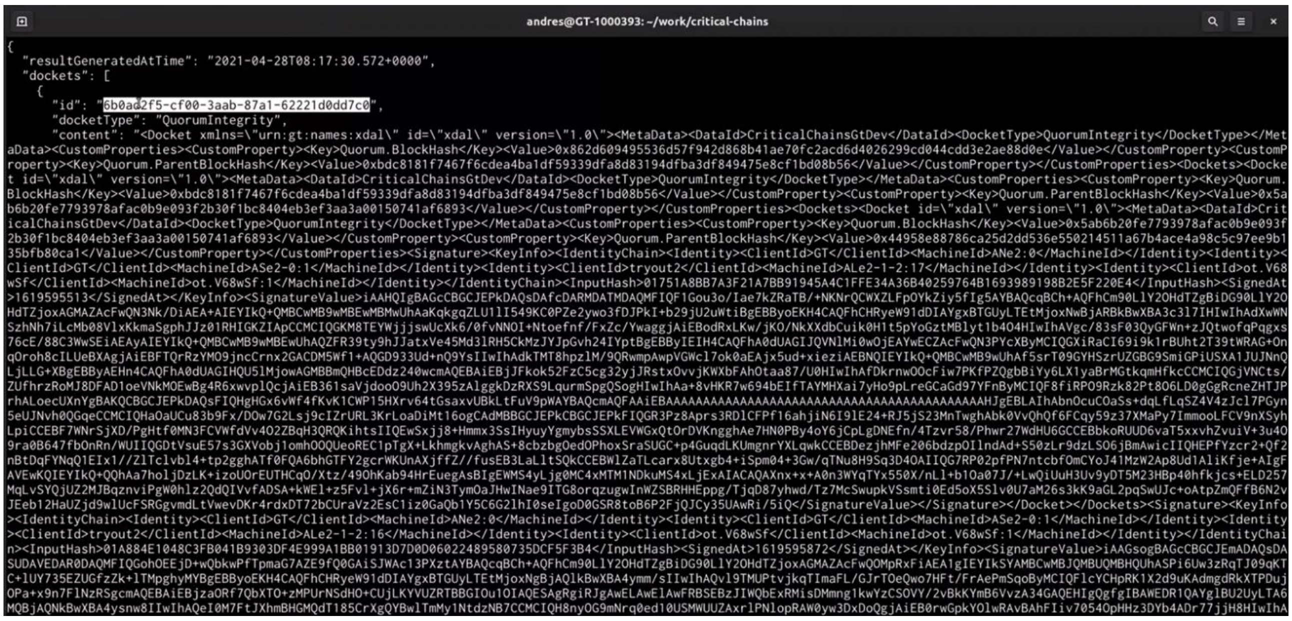
```java
package com.guardtime.ksi.quorum;

import java.io.IOException;

public class RunTransaction {

    public static String contractAddr = "0xcb2c3e62f3ad58c94be612094d526789c7ff8d6c";

    public static void main(String[] args) throws IOException, CipherException {
        Credentials credentials = WalletUtils.loadCredentials("",
                "/home/andres/.ethereum/keystore/UTC--2021-03-09T11-33-22.429754174Z--3fbd4339cfff9b3c5e3d27

        StaticGasProvider gasProvider = new StaticGasProvider(BigInteger.valueOf(0),
                BigInteger.valueOf(100000));

        Quorum quorum = QuorumSigner.connectQuorum();

        Storage s = Storage.load(contractAddr, quorum, credentials, gasProvider);
        if (s.isValid()) {
            try {
                System.err.println("Number: " + s.number().send().getValue());
                int val = new Random().nextInt(1000);
                TransactionReceipt r = s.store(new Uint256(val)).send();
                System.err.println(r.toString());
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

        quorum.shutdown();
    }
}
```

```
QuorumSigner [Java Application]
[main] INFO com.guardtime.docket.signing.AggregatedSigningService - Starting aggregated signer 0 thread with id 18.
[main] INFO com.guardtime.docket.schema.provided.ProvidedSchemaLoader - Reading provided schema file from '/schemas/xdal_1.0.xsd'.
New block: 0x5ab6b20fe7793978afac0b9e093f2b30f1bc8404eb3ef3aa3a00150741af6893, parent: 0x44958e88786ca25d2dd536e550214511a67b4ace4a98c5c97ee9b135bfb80ca1
Docket import: 1d48d35f-e0f9-3bcb-a794-45f5111d0427
New block: 0xbdc8181f7467f6cdea4ba1df59339dfa8d83194dfba3df849475e8cf1bd08b56, parent: 0x5ab6b20fe7793978afac0b9e093f2b30f1bc8404eb3ef3aa3a00150741af6893
Docket import: 5b3cdb56-aea7-37c2-98d0-aa787d5a6bcc
New block: 0x862d609495536d57f942d868b41ae70fc2acd6d4026299cd044cdd3e2ae88d0e, parent: 0xbdc8181f7467f6cdea4ba1df59339dfa8d83194dfba3df849475e8cf1bd08b56
Docket import  6b0ad2f5-cf00-3aab-87a1-62221d0dd7c0
```

Figure 18. Transaction script test (new block generated, docket imported).

In Figure 19, a search with the ID code finds the hash that shows the docket is available.

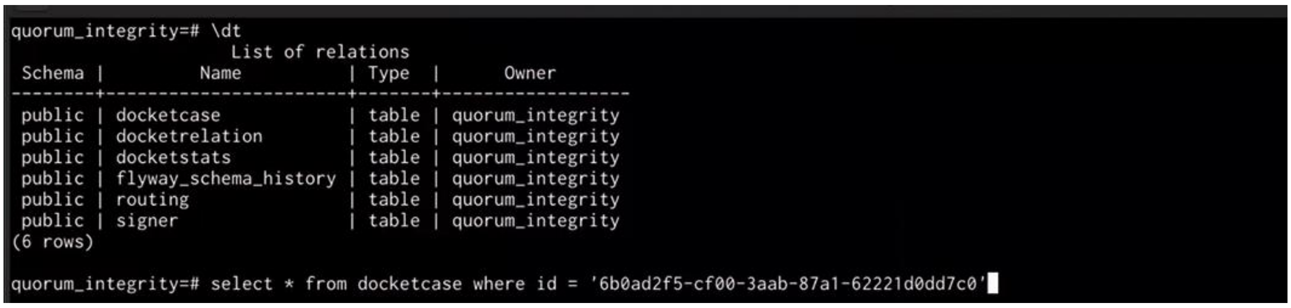
This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018



```
{
  "resultGeneratedAtTime": "2021-04-28T08:17:30.572+0000",
  "dockets": [
    {
      "id": "6b0ad2f5-cf00-3aab-87a1-62221d0dd7c0",
      "docketType": "QuorumIntegrity",
      "content": "<Docket xmlns=\"urn:gt:names:xdal\" id=\"xdal\" version=\"1.0\"><MetaData><DataId>CriticalChainsGtDev</DataId><DocketType>QuorumIntegrity</DocketType></Met
aData><CustomProperties><CustomProperty><Key>Quorum.BlockHash</Key><Value>0x862d609495536d57f942d868b41ae70fc2acd6d4026299cd044cdd3e2ae88d0e</Value></CustomProperty><CustomP
roperty><Key>Quorum.ParentBlockHash</Key><Value>0xbdc8181f7467f6cdea4ba1df59339dfa8d83194dfba3df849475e8cf1bd08b56</Value></CustomProperty></CustomProperties><Dockets><Docke
t id=\"xdal\" version=\"1.0\"><MetaData><DataId>CriticalChainsGtDev</DataId><DocketType>QuorumIntegrity</DocketType></MetaData><CustomProperties><CustomProperty><Key>Quorum.
BlockHash</Key><Value>0xbdc8181f7467f6cdea4ba1df59339dfa8d83194dfba3df849475e8cf1bd08b56</Value></CustomProperty><CustomProperty><Key>Quorum.ParentBlockHash</Key><Value>0x5a
b6b20fe7793978afac0b9e093f2b30f1bc8404eb3ef3aa3a00150741af6893</Value></CustomProperty></CustomProperties><Dockets><Docket id=\"xdal\" version=\"1.0\"><MetaData><DataId>Crit
icalChainsGtDev</DataId><DocketType>QuorumIntegrity</DocketType></MetaData><CustomProperties><CustomProperty><Key>Quorum.BlockHash</Key><Value>0x5ab6b20fe7793978afac0b9e093f
2b30f1bc8404eb3ef3aa3a00150741af6893</Value></CustomProperty><CustomProperty><Key>Quorum.ParentBlockHash</Key><Value>0x44958e88786ca25d2dd536e550214511a67b4ace4a98c5c97ee9b1
35bfb80ca1</Value></CustomProperty></CustomProperties><Signature><KeyInfo><IdentityChain><Identity><ClientId>GT</ClientId><MachineId>ANe2:0</MachineId></Identity><Identity><
ClientId>GT</ClientId><MachineId>ASe2-0:1</MachineId></Identity><Identity><ClientId>tryout2</ClientId><MachineId>ALe2-1-2:17</MachineId></Identity><Identity><ClientId>ot.V68
wSf</ClientId><MachineId>ot.V68wSf:1</MachineId></Identity></IdentityChain><InputHash>01751A8BB7A3F21A7BB91945A4C1FFE34A36B40259764B1693989198B2E5F220E4</InputHash><SignedAt
>1619595513</SignedAt></KeyInfo><SignatureValue>iAAHQIgBAGcCBGCJEPkDAQsDAfcDARMDATMDAQMFIQF1Gou3o/Iae7kZRaTB/+NKNrQCWXZLFpOYkZiy5fIg5AYBAQcqBCh+AQFhCm90LlY2OHdTZgBiDG90LlY2O
HdTZjoxAGMAZAcFwQN3Nk/DiAEA+AIEYIkQ+QMBCwMB9wMBEwMBMwUhAaKqkgqZLU1lI549KC0PZe2ywo3fDJPkI+b29jU2uWtiBgEBByoEKH4CAQFhCHRyeW91dDIAYgxBTGUyLTEtMjoxNwBjARBkBwXBA3c3l7IHIwIhAdXwWN
SzhNh7iLcMb08VlxKkmaSgphJJz01RHIGKZIApCCMCIQGKM8TEYWjjjswUcXk6/0fvNNOI+Ntoefnf/FxZc/YwaggjAiEBodRxLKw/jKO/NkXXdbCuik0H1t5pYoGztMBlyt1b4O4HIwIhAVgc/83sF03QyGFWn+zJQtwofqPqgxs
76cE/88C3WwSEiAEAyAIEYIkQ+QMBCwMB9wMBEwUhAQZFR39ty9hJJatxVe45Md3lRH5CkMzJYJpGvh24IYptBgEBByIEIH4CAQFhA0dUAGIJQVNlMi0wOjEAYwECZAcFwQN3PYcXByMCIQGXiRaCI69i9k1rBUht2T39tWRAG+On
qOroh8cILUeBXAgjAiEBFTQrRzYMO9jncCrnx2GACDM5Wf1+AQGD933Ud+nQ9YsIIwIhAdkTMT8hpzlM/9QRwmpAwpVGWcl7ok0aEAjx5ud+xieziAEBNQIEYIkQ+QMBCwMB9wUhAf5srT09GYHSzrUZGBG9SmiGPiUSXA1JUJNnQ
LjLLG+XBgEBByAEHn4CAQFhA0dUAGIHQU5lMjowAGMBBmQHBcEDdz240wcmAQEBAiEBjJFkok52FzC5cg32yjJRstxOvvjKWXbFAhOtaa87/U0HIwIhAfDkrnwOOcFiw7PKfPZQgbBiYy6LX1yaBrMGtkqmHfkcCCMCIQGjVNCts/
ZUfhrzRoMJ8DFAD1oeVNkMOEwBg4R6xwvplQcjAiEB361saVjdooO9Uh2X395zAlggkDzRXS9LqurmSpgQSogHIwIhAa+8vHKR7w694bEIfTAYMHXai7yHo9pLreGCaGd97YFnByMCIQF8fiRPO9Rzk82Pt8O6LD0gGgRcneZHTJP
rhALoecUXnYgBAKQCBGCJEPkDAQsFIQHgHGx6vWf4fKvK1CWP15HXrv64tGsaxvUBkLtFuV9pWAYBAQcmAQFAAiEBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAHJgEBLAIhAbnOcuCOaSs+dqLfLqSZ4V4zJcl7PGyn
5eUJNvh0QGqeCCMCIQHaOaUCu83b9Fx/DOw7G2Lsj9cIZrURL3KrLoaDiMt16ogCAdMBBGCJEPkCBGCJEPkFIQGR3Pz8Aprs3RDlCFPf16ahjiN6I9lE24+RJ5jS23MnTwghAbk0VvQhQf6FCqy59z37XMaPy7ImmooLFCV9nXSyh
LpiCCEBF7WNrSjXD/PgHtf0MN3FCVWfdVv4O2ZBqH3QRQKihtsIIQEwSxjj8+Hmmx3SsIHyuyYgmybsSSXLEVWGxQtOrDVKngghAe7HN0PBy4oY6jCpLgDNEfn/4Tzvr58/Phwr27WdHU6GCCEBbkoRUUD6vaT5xxvhZvuiV+3u4O
9ra0B647fbOnRn/WUIIQGDtVsuE57s3GXVobj1omhOOQUeoREC1pTgX+LkhmgkvAghAS+8cbzbgOedOPhoxSraSUGC+p4GuqdLKUmgnrYXLqwkCCEBDezjhMFe206bdzpOIlndAd+S50zLr9dzLSO6jBmAwicIIQHEPfYzcr2+Qf2
nBtDqFYNqQ1EIx1//ZlTclvbl4+tp2gghATf0FQA6bhGTFY2gcrWKUnAXjffZ//fusEB3LaLltSQkCCEBWlZaTLcarx8Utxgb4+iSpm04+3Gw/qTNu8H9Sq3D4OAIIQG7RP02pfPN7ntcbfOmCYoJ41MzW2Ap8Ud1AliKfje+AIgF
AVEwKQIEYIkQ+QQhAa7holjDzLK+izoUOrEUTHCqO/Xtz/49OhKab94HrEuegAsBIgEWMS4yLjg0MC4xMTM1NDkuMS4xLjExAIACAQAXnx+x+A0n3WYqTYx550X/nLl+b1Oa07J/+LwQiUuH3Uv9yDT5M23HBp40hfkjcs+ELD257
MqLvSYQjUZ2MJBqznviPgW0hlz2QdQIVvfADSA+kWEl+z5Fvl+jX6r+mZiN3TymOaJHwINae9ITG8orqzugwInWZSBRHHEppg/TjqD87yhwd/Tz7McSwupkVSsmti0Ed5oX5Slv0U7aM26s3kK9aGL2pqSwUJc+oAtpZmQFfB6N2v
JEeb12HaUZjd9wlUcFSRGgvmdLtVwevDKr4rdxDT72bCUraVz2EsC1iz0GaQb1Y5C6G2lhI0seIgoD0GSR8toB6P2FjQJCy35UAwRi/5iQ</SignatureValue></Signature></Docket></Dockets><Signature><KeyInfo
><IdentityChain><Identity><ClientId>GT</ClientId><MachineId>ANe2:0</MachineId></Identity><Identity><ClientId>GT</ClientId><MachineId>ASe2-0:1</MachineId></Identity><Identity
><ClientId>tryout2</ClientId><MachineId>ALe2-1-2:16</MachineId></Identity><Identity><ClientId>ot.V68wSf</ClientId><MachineId>ot.V68wSf:1</MachineId></Identity></IdentityChai
n><InputHash>01A884E1048C3FB041B9303DF4E999A1BB01913D7D0D06022489580735DCF5F3B4</InputHash><SignedAt>1619595872</SignedAt></KeyInfo><SignatureValue>iAAGsogBAGcCBGCJEmADAQsDA
SUDAVEDAR0DAQMFIQGohOEEjD+wQbkwPfTpmaG7AZE9fQ0GAiSJWAc13PXztAYBAQcqBCh+AQFhCm90LlY2OHdTZgBiDG90LlY2OHdTZjoxAGMAZAcFwQOMpRxFiAEA1gIEYIkSYAMBCwMBJQMBUQMBHQUhASPi6Uw3zRqTJ09qKT
C+lUY735EZUGfzZk+lTMpghyMYBgEBByoEKH4CAQFhCHRyeW91dDIAYgxBTGUyLTEtMjoxNgBjAQlkBwXBA4ymm/sIIwIhAQvl9TMUPtvjkqTImaFL/GJrTOeQwo7HFt/FrAePmSqoByMCIQFlcYCHpRK1X2d9uKAdmgdRkXTPDuj
OPa+x9n7FlNzRSgcmAQEBAiEBjzaORf7QbXTO+zMPUrNSdHO+CUjLKYVUZRTBBGIOu1OIAQESAgRgiRJgAwELAwElAwFRBSEBzJIWQbExRMisDMmng1kwYzCSOVY/2vBkKYmB6VvzA34GAQEHIgQgfgIBAWEDR1QAYglBU2UyLTA6
MQBjAQNkBwXBA4ysnw8IIwIhAQeI0M7FtJXhmBHGMQdT185CrXgQYBwlTmMy1NtdzNB7CCMCIQH8nyOG9mNrq0ed10USMWUUZAxrlPNlopRAW0yw3DxDoQgjAiEB0rwGpkYOlwRAvBAhFIiv7054OpHHz3DYb4ADr77jjH8HIwIhA
```

Figure 19. Docket query positive results (hash).

In conclusion we must check that the docket is saved to the database, to the right table. As an instance the data tables created for docket case (docketcase, docketrelation, docketstats, flyway_schema_history, routing, and signer) can be found in Figure 20.



```
quorum_integrity=# \dt
                 List of relations
 Schema |         Name          | Type  |      Owner
--------+-----------------------+-------+------------------
 public | docketcase            | table | quorum_integrity
 public | docketrelation        | table | quorum_integrity
 public | docketstats           | table | quorum_integrity
 public | flyway_schema_history | table | quorum_integrity
 public | routing               | table | quorum_integrity
 public | signer                | table | quorum_integrity
(6 rows)

quorum_integrity=# select * from docketcase where id = '6b0ad2f5-cf00-3aab-87a1-62221d0dd7c0'
```

Figure 20. Database (docket case 6 tables, PostgreSQL), that Docket Broker is using.

5.2.3.2 Quorum Blockchain Standards and Testing

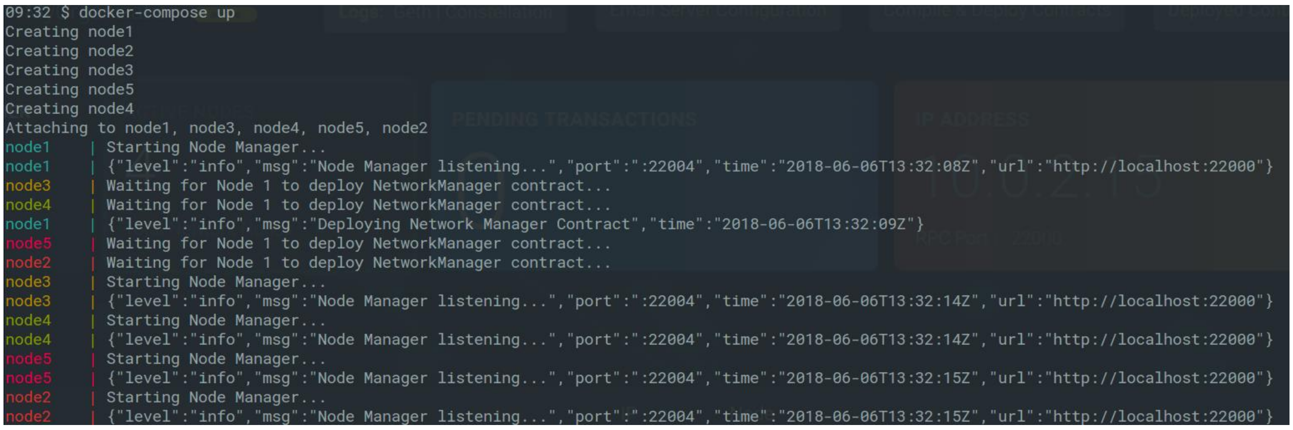
The testing setup of the Quorum blockchain environment is based on Quorum Maker, which is a tool used to manage the network and create any number of nodes of various configurations dynamically with reduced user input.

The tool provides the following benefits:

- An easy interface to create and manage the Quorum Network
- A modern UI to monitor and manage Quorum Network
- A Network Map Service to be used for identifying nodes and self-publishing roles.
- Block and Transaction Explorer
- Smart Contract Deployment
- Email Notifications
- Simplified testing

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

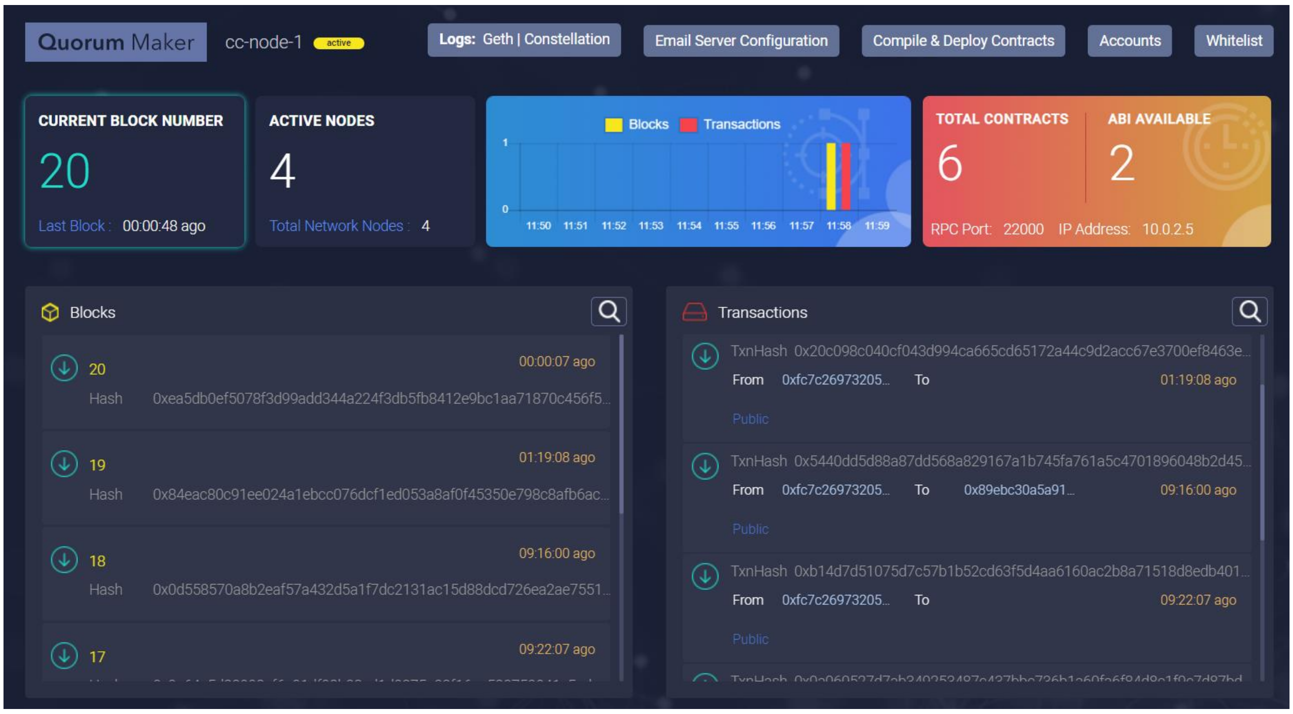
To see a full response of the network, after creating the project and setting up the nodes, we changed to the directory just created and ran "$ docker-compose up" to bring up the network. After this command, we saw each node coming up and node managers for each node getting started. Once the activities stopped, we had a confirmation that we had a fully functioning Quorum Network running. This process can be viewed in Figure 21.



```
09:32 $ docker-compose up
Creating node1
Creating node2
Creating node3
Creating node5
Creating node4
Attaching to node1, node3, node4, node5, node2
node1    | Starting Node Manager...
node1    | {"level":"info","msg":"Node Manager listening...","port":":22004","time":"2018-06-06T13:32:08Z","url":"http://localhost:22000"}
node3    | Waiting for Node 1 to deploy NetworkManager contract...
node4    | Waiting for Node 1 to deploy NetworkManager contract...
node1    | {"level":"info","msg":"Deploying Network Manager Contract","time":"2018-06-06T13:32:09Z"}
node5    | Waiting for Node 1 to deploy NetworkManager contract...
node2    | Waiting for Node 1 to deploy NetworkManager contract...
node3    | Starting Node Manager...
node3    | {"level":"info","msg":"Node Manager listening...","port":":22004","time":"2018-06-06T13:32:14Z","url":"http://localhost:22000"}
node4    | Starting Node Manager...
node4    | {"level":"info","msg":"Node Manager listening...","port":":22004","time":"2018-06-06T13:32:14Z","url":"http://localhost:22000"}
node5    | Starting Node Manager...
node5    | {"level":"info","msg":"Node Manager listening...","port":":22004","time":"2018-06-06T13:32:15Z","url":"http://localhost:22000"}
node2    | Starting Node Manager...
node2    | {"level":"info","msg":"Node Manager listening...","port":":22004","time":"2018-06-06T13:32:15Z","url":"http://localhost:22000"}
```

Figure 21. Quorum Nodes first activities

After the network setup and first response, we performed several transactions including smart contract deployment and tests (described in the following section) to confirm the reliability of the implemented network. This can be viewed in Figure 22.



CURRENT BLOCK NUMBER
20
Last Block : 00:00:48 ago

ACTIVE NODES
4
Total Network Nodes : 4

TOTAL CONTRACTS 6
ABI AVAILABLE 2
RPC Port: 22000 IP Address: 10.0.2.5

Blocks

20 00:00:07 ago
Hash 0xea5db0ef5078f3d99add344a224f3db5fb8412e9bc1aa71870c456f5...

19 01:19:08 ago
Hash 0x84eac80c91ee024a1ebcc076dcf1ed053a8af0f45350e798c8afb6ac...

18 09:16:00 ago
Hash 0x0d558570a8b2eaf57a432d5a1f7dc2131ac15d88dcd726ea2ae7551...

17 09:22:07 ago

Transactions

TxnHash 0x20c098c040cf043d994ca665cd65172a44c9d2acc67e3700ef8463e...
From 0xfc7c26973205... To 01:19:08 ago
Public

TxnHash 0x5440dd5d88a87dd568a829167a1b745fa761a5c4701896048b2d45...
From 0xfc7c26973205... To 0x89ebc30a5a91... 09:16:00 ago
Public

TxnHash 0xb14d7d51075d7c57b1b52cd63f5d4aa6160ac2b8a71518d8edb401...
From 0xfc7c26973205... To 09:22:07 ago
Public

Figure 22. Critical-Chains Quorum Maker interface

More tests and evaluation on Quorum high performance will be conducted in the future in order to confirm the significant advantage of this network. At the current time and with a basic implementation, we can say that Quorum can carry out more than 100 transactions per second, making Quorum the preferable choice for financial institutions.

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

5.2.4 Smart Contract Standards and Tests

Smart contracts are immutable, verifiable, and autonomous pieces of code that can be deployed and run on blockchain networks Ethereum-like. Due to the immutability nature of blockchain, no change is possible on a deployed smart contract or a verified transaction. On the other hand, at the current time there are $50.102B (Defi Pulse Index) carried out by smart contracts locked in DeFi, and hence, a faulty smart contract can lead to a huge monetary loss. Therefore, it is important for smart contract developers to fully test and check the correctness of their code before deploying it on the blockchain. In this sense, smart contract testing is important for the following reasons:

- Tests validate behaviours within the smart contract. They give confidence that your code performs in the ways intended and does not mis perform.
- When developing, tests provide assurance that newly added code does not have unintended side effects or regressions. As new logics increase, a passing test suite helps to confirm that any previous functionality is broken.
- A good test suite simplifies refactoring saving time in debugging. When an unexpected error appears, a test suite enables many potential causes. to be immediately ruled out.

There are different ways to test smart contracts and the main categories are:

- Unit tests are simple tests that verify a single behaviour or component within your code. A well written unit test is quick to execute and provides a clear picture of what went wrong when it fails.
- Integration tests are more complex tests that validate interactions between multiple components. For smart contract testing this can mean interactions between different components of a single contract, or across multiple contracts.

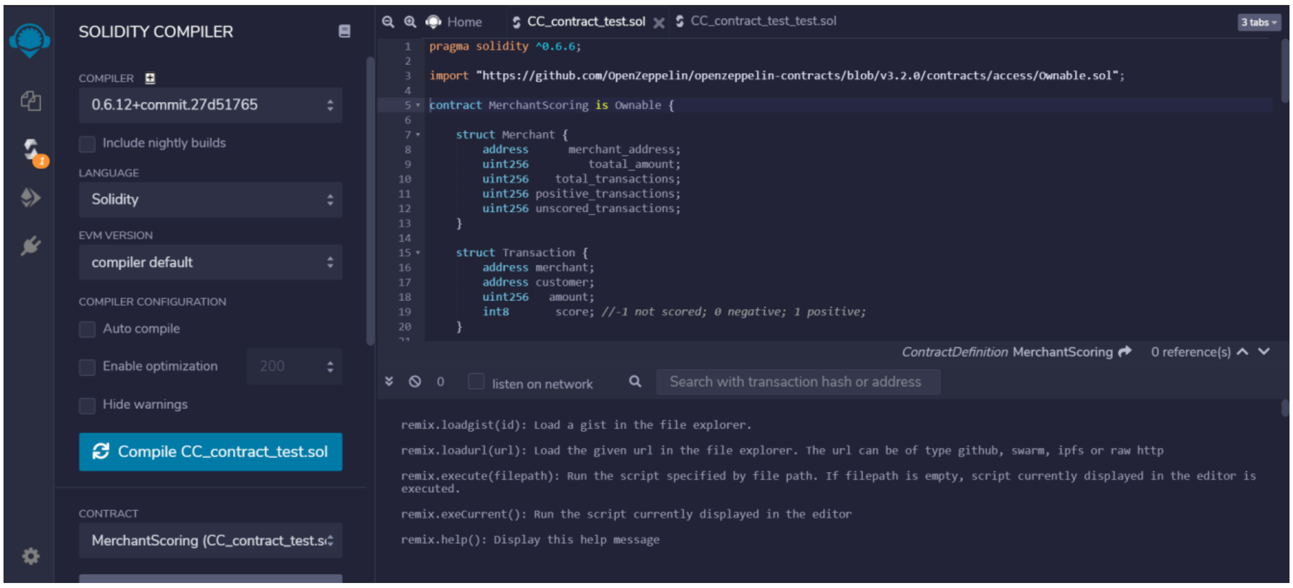
After the deployment of pilot smart contracts, we performed several unit tests using Remix IDE for the following reasons:

- Simple: Each unit test should evaluate a single behaviour. When a test fails it should be immediately obvious why. It is better to write many short tests with a single assertion, than one long test with many assertions.
- Repeatable: A unit test should always produce the same result, as long as the code being tested has not changed. Tests should never be influenced by some uncontrollable parameter. This is important so that when a test fails, it can be repeated, and the result observed to help discovered the issue.
- Isolated: Tests should not depend upon or affect other tests
- Fast: Unit tests should be written with speed in mind, also, running quick test suites ensure rapid feedback.

Remix IDE is an integrated development environment that supports the development, deployment, and testing of smart contracts for Ethereum-like blockchains. The IDE can be run online in a web browser or as a standalone cross-platform desktop application. The documentation for Remix IDE is available online. Remix IDE is targeting smart contract development in the Solidity programming language, and it is extendable via a plugin framework.

The main steps to test smart contracts with this tool are represented in Figures 23, 24, 25, 26 and 27:

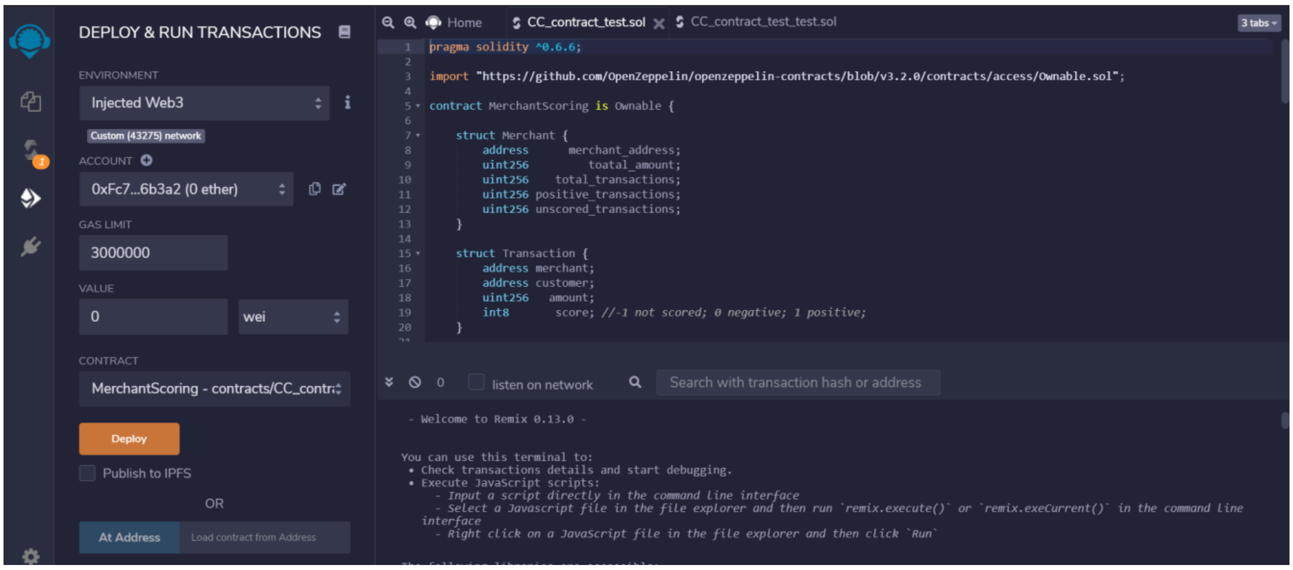
This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018



```
pragma solidity ^0.6.6;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.2.0/contracts/access/Ownable.sol";

contract MerchantScoring is Ownable {

    struct Merchant {
        address      merchant_address;
        uint256         toatal_amount;
        uint256    total_transactions;
        uint256 positive_transactions;
        uint256 unscored_transactions;
    }

    struct Transaction {
        address merchant;
        address customer;
        uint256   amount;
        int8       score; //-1 not scored; 0 negative; 1 positive;
    }
```

```
remix.loadgist(id): Load a gist in the file explorer.

remix.loadurl(url): Load the given url in the file explorer. The url can be of type github, swarm, ipfs or raw http

remix.execute(filepath): Run the script specified by file path. If filepath is empty, script currently displayed in the editor is
executed.

remix.exeCurrent(): Run the script currently displayed in the editor

remix.help(): Display this help message
```

Figure 23. Remix IDE Contract Compile

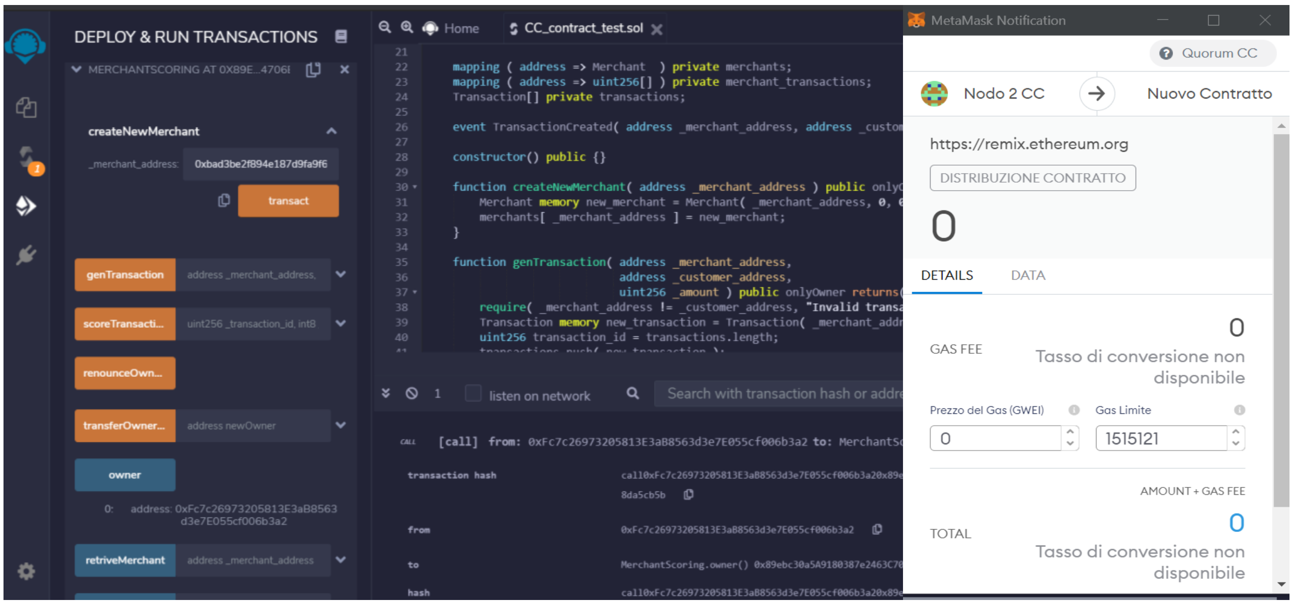


```
pragma solidity ^0.6.6;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.2.0/contracts/access/Ownable.sol";

contract MerchantScoring is Ownable {

    struct Merchant {
        address      merchant_address;
        uint256         toatal_amount;
        uint256    total_transactions;
        uint256 positive_transactions;
        uint256 unscored_transactions;
    }

    struct Transaction {
        address merchant;
        address customer;
        uint256   amount;
        int8       score; //-1 not scored; 0 negative; 1 positive;
    }
```

```
 - Welcome to Remix 0.13.0 -

You can use this terminal to:
 • Check transactions details and start debugging.
 • Execute JavaScript scripts:
    - Input a script directly in the command line interface
    - Select a Javascript file in the file explorer and then run `remix.execute()` or `remix.exeCurrent()` in the command line
 interface
    - Right click on a JavaScript file in the file explorer and then click `Run`
```

Figure 24. Remix IDE contract deploy.

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018



_merchant_address: 0xbad3be2f894e187d9fa9f6

genTransaction: address _merchant_address,

scoreTransacti...: uint256 _transaction_id, int8

transferOwner...: address newOwner

0: address: 0xFc7c26973205813E3aB8563d3e7E055cf006b3a2

retriveMerchant: address _merchant_address

```
    mapping ( address => Merchant  ) private merchants;
    mapping ( address => uint256[] ) private merchant_transactions;
    Transaction[] private transactions;

    event TransactionCreated( address _merchant_address, address _custo

    constructor() public {}

    function createNewMerchant( address _merchant_address ) public only
        Merchant memory new_merchant = Merchant( _merchant_address, 0, 
        merchants[ _merchant_address ] = new_merchant;
    }

    function genTransaction( address _merchant_address,
                             address _customer_address,
                             uint256 _amount ) public onlyOwner returns(
        require( _merchant_address != _customer_address, "Invalid transa
        Transaction memory new_transaction = Transaction( _merchant_addr
        uint256 transaction_id = transactions.length;
```

```
CALL  [call] from: 0xFc7c26973205813E3aB8563d3e7E055cf006b3a2 to: MerchantSc

transaction hash        call0xFc7c26973205813E3aB8563d3e7E055cf006b3a20x89e
                        8da5cb5b

from                    0xFc7c26973205813E3aB8563d3e7E055cf006b3a2

to                      MerchantScoring.owner() 0x89ebc30a5A9180387e2463C7

hash                    call0xFc7c26973205813E3aB8563d3e7E055cf006b3a20x89e
```

https://remix.ethereum.org

0

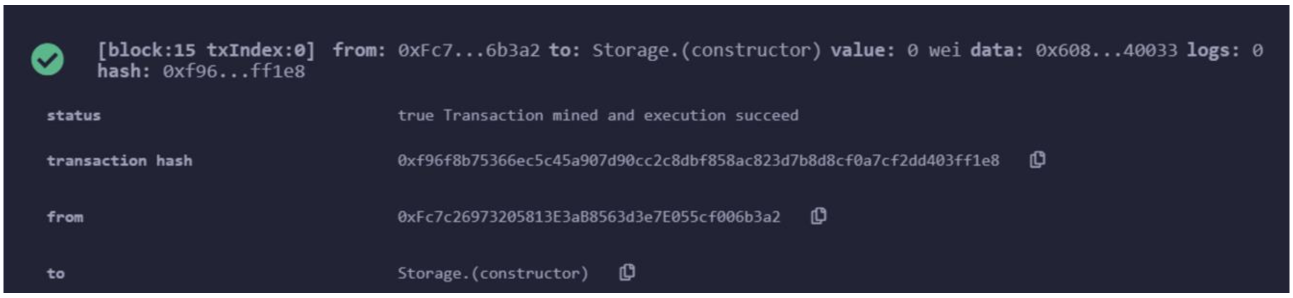
GAS FEE 0 Tasso di conversione non disponibile

Prezzo del Gas (GWEI): 0 Gas Limite: 1515121

AMOUNT + GAS FEE

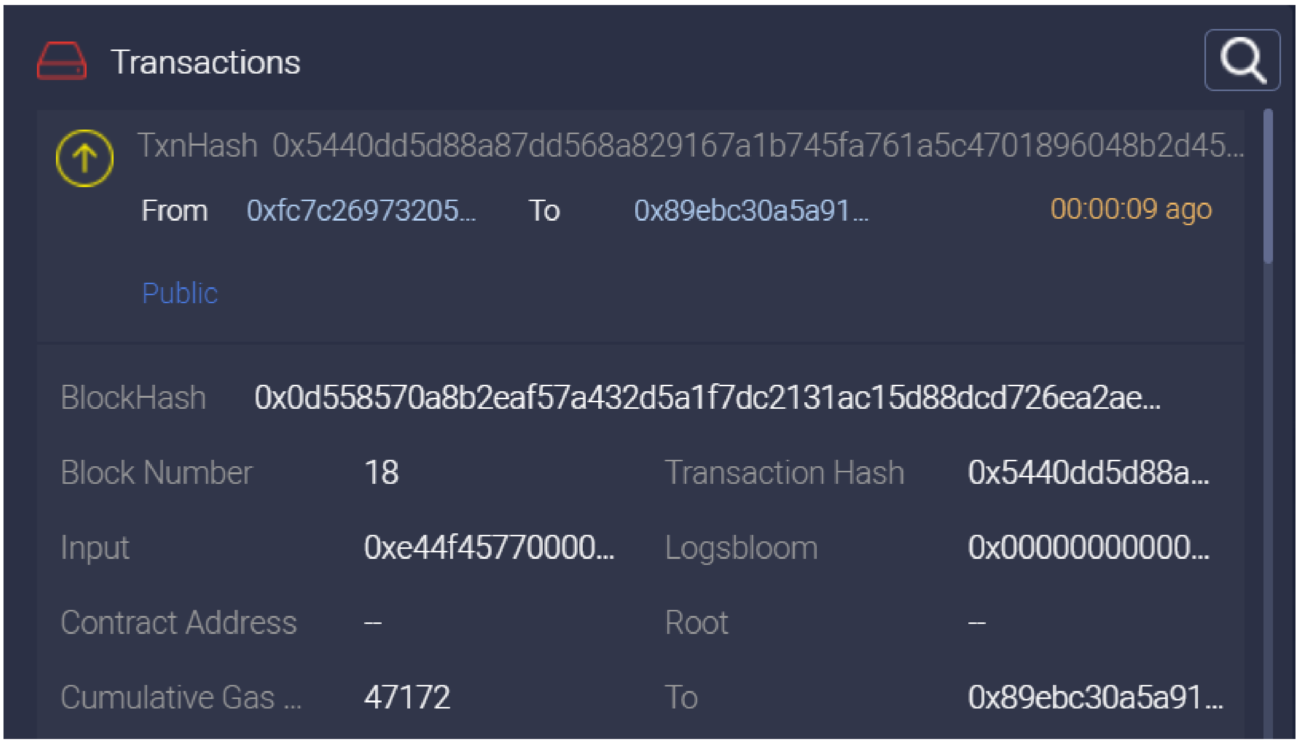
TOTAL 0 Tasso di conversione non disponibile

Figure 25. Remix IDE transaction



```
[block:15 txIndex:0] from: 0xFc7...6b3a2 to: Storage.(constructor) value: 0 wei data: 0x608...40033 logs: 0
hash: 0xf96...ff1e8

status              true Transaction mined and execution succeed

transaction hash    0xf96f8b75366ec5c45a907d90cc2c8dbf858ac823d7b8d8cf0a7cf2dd403ff1e8

from                0xFc7c26973205813E3aB8563d3e7E055cf006b3a2

to                  Storage.(constructor)
```

Figure 26. Remix IDE transaction approval



Transactions

TxnHash 0x5440dd5d88a87dd568a829167a1b745fa761a5c4701896048b2d45...

From 0xfc7c26973205... To 0x89ebc30a5a91... 00:00:09 ago

Public

BlockHash 0x0d558570a8b2eaf57a432d5a1f7dc2131ac15d88dcd726ea2ae...

Block Number	18	Transaction Hash	0x5440dd5d88a...
Input	0xe44f45770000...	Logsbloom	0x00000000000...
Contract Address	–	Root	–
Cumulative Gas ...	47172	To	0x89ebc30a5a91...

Figure 27. Transaction confirmation on Quorum Maker

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

5.3 Test Error Matrix

This section presents the errors that occur during Conformance Testing. The error matrix identified the relevant fixes that needed to be made for both Critical-Chains applications and/or web services.

Table 17. Test Error Matrix

Pilot	Test Case/Web Service Number	Error	Status	Priority	Comments
Toll Collection	TC1	Dashboard page is empty	Conditional Pass	Low	The dashboard for the Toll will provide business related insights for the 2nd phase use-cases.
Toll Collection	TC4	Transaction Search Multiple Filters	Fail	Low	The multiple filters will be developed for the 2nd phase use-cases.
Toll Collection	TC6	Operation Search Multiple Filters	Fail	Low	The multiple filters will be developed for the 2nd phase use-cases.
Toll Collection	TC10	Account Journal Search Multiple Filters	Fail	Low	The multiple filters will be developed for the 2nd phase use-cases.

6 Conclusions

This deliverable, D3.11, “Joint report on Conformance Analysis, Linking, Mapping, and Synchronisation”, has presented the design and development of a Blockchain-based infrastructure integrating components for the Critical-Chains solution stack and has validated the developed services, applications, and infrastructure and their compliance in the targeted deployment context.

The blockchain components for the distributed ledger mechanism, smart contracts, and integrity mechanism are presented within the linking, mapping, and synchronisation perspectives. Functionally, the deliverable has delivered the design and development aspects of a tailored blockchain key solution for enhanced access control, authentication, authorisation management, and privacy-preserving user input solution.

Finally, a series of test activities were conducted in order to determine the extent to which the developed services, applications, and infrastructure are compliant with the project requirements.

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018

7 References

Coindesk. 2020. *What Is Ethereum.* 3 December. https://www.coindesk.com/learn/ethereum-101/what-is-ethereum.

Coindesk. 2020. *How Do Ethereum Smart Contracts Work.* 20 December. https://www.coindesk.com/learn/ethereum-101/ethereum-smart-contracts-work.

Ethereum, Kevin Ziechmann. 2021. "Transactions." *Ethereum Org.* 30 March. https://ethereum.org/en/developers/docs/transactions/.

Nakamoto, Satoshi. 2009. "Bitcoin." *Bitcoin Org.* https://bitcoin.org/bitcoin.pdf.

Yiu, Neo C.K. 2021. "An Overview of Forks and Coordination in Blockchain Development." February.

Davies. 2020. *How do nodes work on a blockchain network.* https://www.daviescoin.io/blog/how-do-nodes-work-on-a-blockchain-network.

Hasan, Omar, L. Brunie, and E. Bertino. 2021. "Privacy Preserving Reputation Systems based on Blockchain and other Cryptographic Building Blocks: A Survey."

Parity Technologies. 2017. "Parity: Fast, light, robust Ethereum implementation."

Wood, Gavin. 2015. "Proof of Authority Private Chains."

JP Morgan. 2016. "A permissioned implementation of Ethereum supporting data privacy."

Ether Nodes. 2021. *Ethereum Mainnet Statistics.* 23 06. https://www.ethernodes.org.

****

This project has received funding from the European Union Horizon 2020 Programme for research, technological innovation and demonstration under Grant Agreement number 833326 H2020-SU-DS-2018