Horizon 2020 – The EU Framework Programme for Research and Innovation
Project Co-funded by the European Commission
Contract number: 761145
Call identifier: NMBP-22-2017
Project Start Date: 1st January 2018

MANUfacturing eco**S**ystem of **QUA**lified **R**esources **E**xchange

D3.1
Connecting IoT devices to blockchain services

Dissemination Level	Public
Partners	IBM
Authors	IBM
Planned date of delivery	31/12/2018
Date of issue	31/12/2018
Document version	V0.7

DOCUMENT HISTORY			
Version	Issue date	Content and changes	Author
0.1	12/11/2018	First version (ToC)	IBM
0.2	26/11/2018	Content highlights for all sections	IBM
0.3	05/12/2018	First draft – full content	IBM
0.4	06/12/2018	Second draft – initial comments from SUPSI	IBM
0.5	16/12/2018	Address elaborate comments from SUPSI and internal IBM review	IBM
0.6	18/12/2018	Address comments from INESC	IBM
0.7	31/12/2018	Final review	SUPSI

Role	Partner	Person
Reviewer 1	SUPSI	Andrea Barni, Giuseppe Landolfi
Reviewer 2	INESC	Henrique Diogo Silva, Antonio Soares
Quality assurance	SUPSI	Andrea Bettoni

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Table 1: List of Abbreviations

Acronym	Description
ACL	Access Control List
API	Application Programming Interface
B2B	Business to Business
CA	Certificate Authority
GUI	Graphical User Interface
IIoT/ Industrie 4.0	Industrial Internet of Things
IoT	Internet of Things
LDAP	Lightweight Directory Access Protocol
MANU-SQUARE	MANUfacturing ecoSystem of QUAlified Resource Exchange
MVCC	Multiversion concurrency control
REST	Representational State Transfer
RFI	Request for Information
RFID	Radio Frequency Identification
RFQ	Request for Quotation
RPi	Raspberry Pi
SDK	Software Development Kit
SME	Small and Medium Enterprises
VM	Virtual Machine
WP	Work Package

GLOSSARY

Table 2 List of Terms

Term	Meaning
Certificate Authority	An entity that issues digital certificates to be used for authentication of the communicating party holding the certificate.
Chaincode	The implementation of a smart contract within Hyperledger Fabric. A programmatic manner which governs the interaction between a blockchain client and the blockchain network itself.
Endorsing peer	A member of the blockchain network who may be called to endorse transactions for a particular chaincode. The process of endorsing a transaction includes a speculative execution of the chaincode function included in the transaction proposal responding to the caller with the read and write sets (including versions) corresponding to the chaincode execution. In addition it indicates whether it supports the transaction.
MVCC	A multi-version concurrency control employed by Hyperledger Fabric to enable speculative concurrent execution of transactions without corrupting the underlying data storage.
Ordering service	A Fabric entity that provides total order among incoming transactions and their inclusion into blocks.

Peer	A central entity in the Fabric network which is in charge of validating incoming blocks (and associated transactions) and committing the blocks to its own copy of the shared ledger.
REST	A common manner of interaction among different processes.
RFID	Mainly used to automatically identify and track tags attached to objects.

1 EXECUTIVE SUMMARY

The main goal of this document is to present the blockchain based platform to be used within MANU-SQUARE for supply chain scenarios support. The document starts from a description of the underlying blockchain platform as a building block for supply chain relationships, proceeds to describe the integration of IoT devices within a blockchain network along with performance characteristics, and finally presents relevant use cases, such as tracking & tracing.

The outcome of this deliverable provides the means in which a blockchain infrastructure shall support supply chain use cases within the project, along with the manner in which it shall be realized. A recorded demonstration of the described capabilities shall accompany this deliverable.

The main concrete capabilities within MANU-SQUARE which are to be supported by the blockchain component are:

1. Traceability of exchanged goods
2. Manufacturing systems' data certification
3. Traceability of innovative ideas
4. RFQ management
5. Reputation management

As a report of the activities carried out within T3.1, this document emphasizes the work carried out in order to enable IoT devices to be first class citizens in a blockchain network. Following activities will further enhance and demonstrate security and privacy related capabilities. Finally, all blockchain related work performed shall feed into the infrastructure of MANU-SQUARE , supporting the various use cases.

The description of the work is organized in the following sections:

- Section 2 briefly introduces the MANU-SQUARE project and the manner in which the activities done in WP3 relate to other components within the project. It further provides an introduction to the blockchain technology in general and its relevance to the supply chain scenarios in particular (including relevant use cases);
- Section 3 clarifies more specifically the role of the blockchain within MANU-SQUARE and interactions with other platform components;
- Section 4 details the work performed on the integration IoT devices into a blockchain network;
- Section 5 reports on the performance benchmarking performed to evaluate the viability of connecting IoT devices to a blockchain network as data providers;
- Section 6 describes the blockchain based negotiation support provided;
- Section 7 describes the design of dealing with flaky network connectivity of IoT devices;
- Finally, section 8 provides conclusions and next steps.

2 INTRODUCTION

The MANU-SQUARE project aims at fostering an ecosystem that acts as a virtual marketplace in which surplus of industrial resources can easily meet corresponding shortage, thus bringing the available capacity (such as production capacity), as well as other virtual and physical assets, closer to the demand to obtain the optimal match (See Figure 1). This scheme has two main advantages:

- The rapid and efficient creation of local distributed value networks for innovative providers of product services;
- The reintroduction and optimization in the loop of unused capacity and potential that would otherwise be lost.

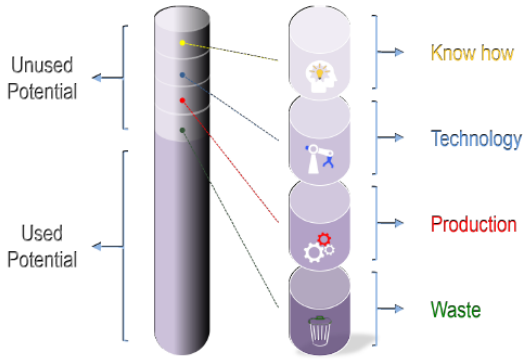


Figure 1: Composition of the unused potential

MANU-SQUARE establishes an ecosystem that is organized to match the needs of buyers with the availability of sellers in terms of know-how, technology, manufacturing capacity and waste. The blockchain technology introduced in this deliverable provides trust, transparency and security to the MANU-SQUARE platform, thereby serving as a single source of truth and distributed trust in a trustless environment amongst the different stakeholders of the platform.

Transactions through the platform are recorded in a final and immutable manner by the blockchain layer, providing all counter-parties with an identical real-time view of the process.

The main objectives of the project are to match shortage with surplus on a wide spectrum of areas. The main reason for integration blockchain technology at the basis of the platform is to provide security, privacy, and trust in the process.

T3.1, which is summarized in this deliverable, serves as the basic blockchain platform for supply chain scenarios. Closely following tasks shall construct additional layers on top of the base platform. T3.2 shall add security and privacy constructs such as to enable differentials views of data residing in the blockchain based on certificates and capabilities agreed upon between different members of a blockchain network. T3.3 shall employ specific supply chain constructs to ease the integration of use case required capabilities into the base platform.

2.1 State of the Art

2.1.1 Blockchain

A blockchain[1] revolves around the concept of a shared ledger, representing the system of record and a single source of truth for business interactions. The shared ledger is maintained by a cluster of peer processes, providing an append only transactions log, while guaranteeing the immutability of inserted and validated transactions. There are various mechanisms for the validation of transactions and blocks creation, chief among these is the proof-of-stake used by bitcoin. Nevertheless, in our setting we create a permissioned blockchain, which is more suited for enterprises; one of the outcomes of a permissioned network is the possibility to use more efficient algorithms to establish total order among transactions and blocks creation. It enables a network of business partners to perform transactions across organizations without resorting to a single unified trusted authority. A blockchain transaction represents a state change or asset transfer in the ledger; transactions are governed by smart contracts, which may be interpreted as the digital incarnations of legal contracts, containing the rules for transactions to be invoked. Blockchain provides a shared, replicated, permissioned ledger ensuring trust, provenance, immutability and finality, to replace inefficient, expensive, and vulnerable processes.

Finality refers to the property in which once an indication is received that a transaction has entered a block it cannot be taken out at a later point in time. Immutability refers to the property in which a transaction in the blockchain cannot be changed at a later point in time. These properties are guaranteed by elaborated cryptographic measures. These properties combined contribute to the possibility of extracting provenance trail of each item (asset) in the blockchain, and monitor the exact path taken by the asset in question from birth to this very day. These measures together provide a level of trust among partners which is difficult to achieve otherwise in an inherently trust-less environment. Most importantly, the trust is not due to a single actor within the network, but rather it is an outcome of the collective nature and properties of the underlying technology.

As can be seen in Figure 2 the shared ledger provides a real-time shared and replicated view of the state of the transactions among all members of a blockchain network. This reality stands in contrast to the pre-blockchain era in which each organization held its own ledger, opening the door to inconsistencies and disputes.

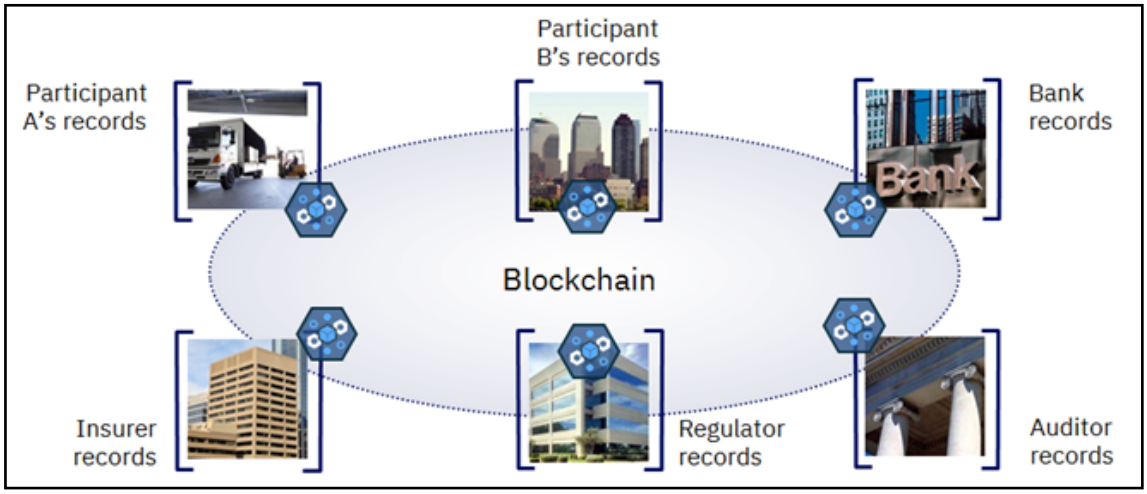


Figure 2: Blockchain's heart - the shared ledger

The four cornerstones comprising the blockchain structure are (as depicted in Figure 3):

1. Shared, replicated, (permissioned) ledger - an append-only distributed system of records serving as the single source of truth. It records all transactions across a business network, and is shared between participants who have their own copy through replication.

[1] We mainly refer to the permissioned flavor of a blockchain which is more suitable for business scenarios. We mainly focus on the Hyperledger Fabric implementation.

2. Trust – by which all parties agree to network verified transactions
3. Privacy and security - ensuring appropriate visibility and authentication, such that only authorized parties are exposed and have access to blockchain activities. There are various levels of privacy supported, governed by the agreements upon partners in the network. These vary from having all transactions and associated data exchange visible to all members of the network (and to no one outside the network), through the exposure that there are transactions among partners without all associated data to be revealed, to the complete isolation of a business relationship such that only the involved partners can deduce that such a relationship exists. Cryptography is central to these processes.
4. Smart Contract - Business terms embedded in transactions. It is verifiable, signed, and encoded in a programming language. This logic becomes an integral part of the blockchain network upon partners and thus ensures the automatic execution of agreed upon actions based on data triggers flowing into the network in the form of endorsed transactions.

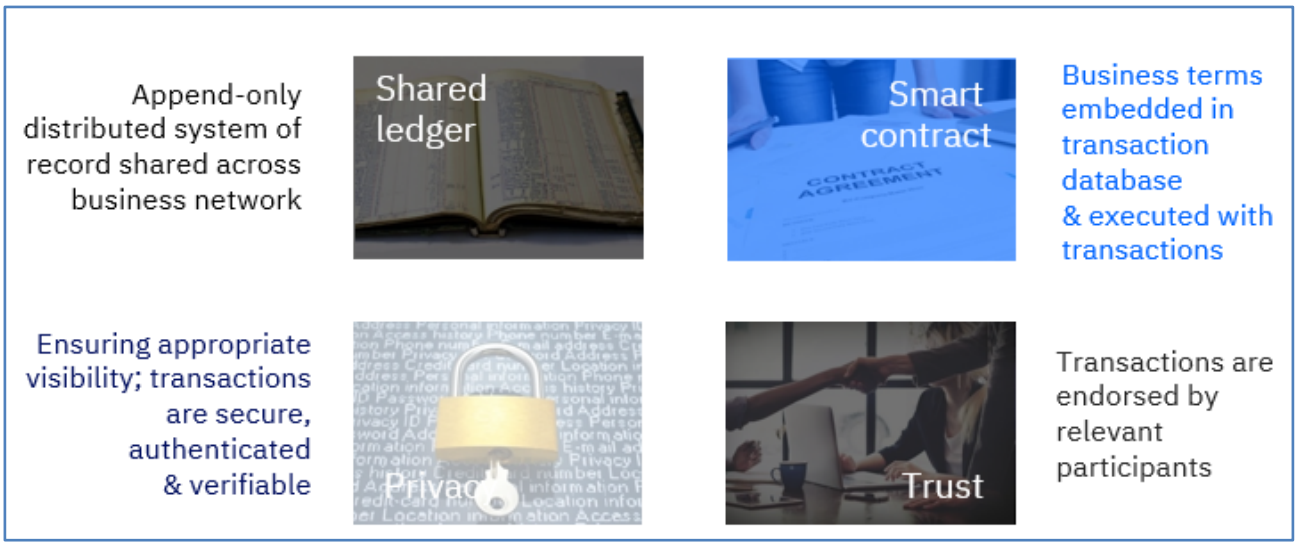


Figure 3: BlockChain essentials

All these building blocks combined together provide assurance for provenance, immutability, and finality, by assuring that only mutually agreed upon transactions become part of the ledger, and once a transaction is inserted to the shared ledger it cannot be erased or modified in any way. It is always possible to track back through the blockchain structure all previous states of a specific item. Thus, being cryptographically secure, the shared ledger is updated by consensus and becomes an immutable and indelible record of all transactions, exhibiting the entire life cycle of assets registered in the blockchain.

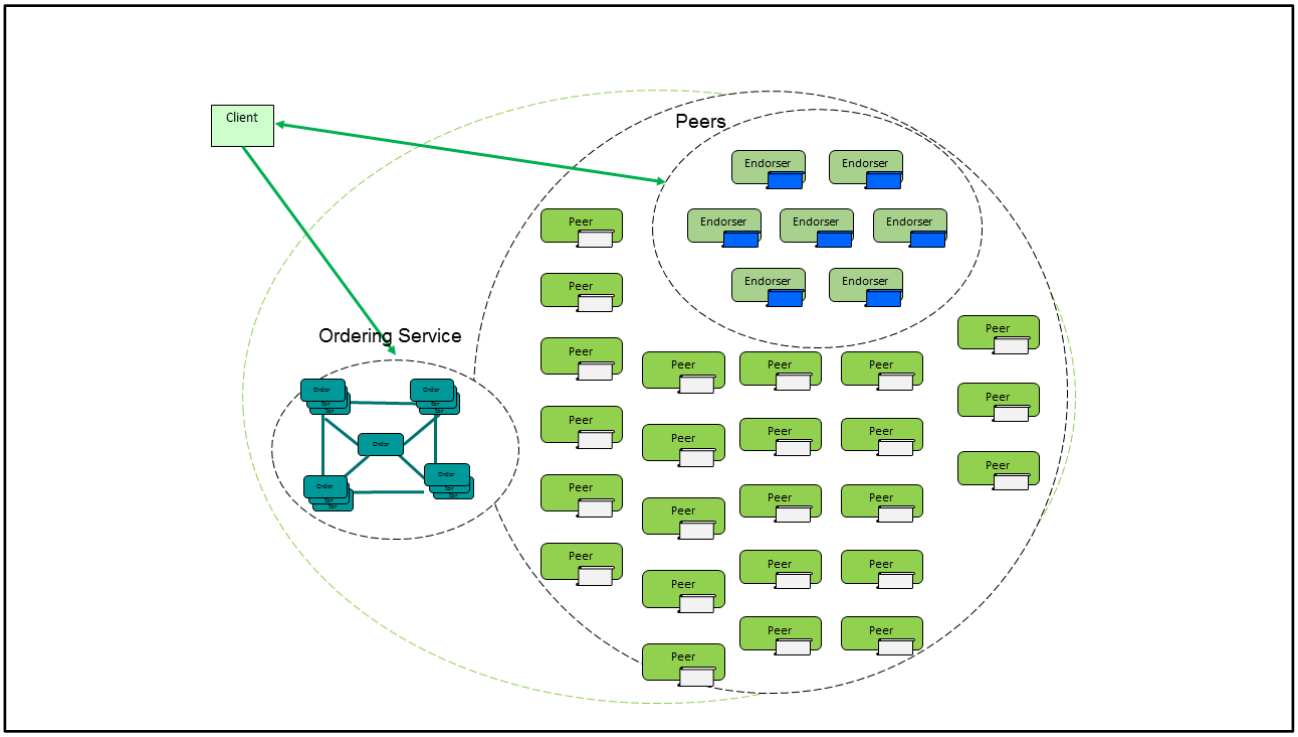


Figure 4: Hyperledger Fabric high level architecture

As depicted in Figure 4, a central part of the system contains an ordering service which determines the order of newly received transactions, creates the corresponding blocks containing the ordered transactions, and makes them available to all replicating peers. In addition, at the heart of the system exists a group of peers, representing consortium partners, that endorse (or not) each incoming proposed transaction. A potentially larger set of peers retrieve the new blocks from the ordering service and proceed to validate and store the blocks thus maintaining the full history of the shared ledger. Such a shared ledger enables carrying out business transactions among different entities in a trustless environment, in which there's no central authority that needs to be trusted by all participants for the participation in the system to be feasible, but rather the collective trust and the incentives to abide by it encourage and enable entities to use such a technology.

In addition, the blockchains space is divided into public vs. private (or permissioned) environments. In a public environment, such as bitcoin, every entity can participate in the system in every role, and no one knows who the person or organization behind each entity are. A permissioned environment, on the other hand, calls for the participation only of approved entities both as blockchain peers and as clients. That is more suited for an enterprise environment in which parties need to know which other parties they are interacting with, and where regulatory requirements need to be upheld. For the purposes of this work we intend to make use of Hyperledger Fabric (see https://hyperledger-fabric.readthedocs.io/en/latest/), a permissioned ledger, which is suitable for the business environment we are targeting.

Blockchain technology usage is relatively new but interest in it is growing in many fields. The first such field is the financial services arena, but more areas are exploring the usage of this technology, supply chain being in the forefront. In various analysis reports it can be seen that Banking / Financial Services and Supply Chain remain top industries for blockchain activity[2]). A lot of attention and funds are being devoted to exploring blockchain contribution to supply chain scenarios[3], both by industrial partners, as well as large IT providers, such as IBM, Oracle, and Microsoft.

[2] https://www2.deloitte.com/content/dam/Deloitte/cz/Documents/financial-services/cz-2018-deloitte-global-blockchain-survey.pdf
[3] https://newsroom.ibm.com/2018-08-09-Maersk-and-IBM-Introduce-TradeLens-Blockchain-Shipping-Solution
https://www.coindesk.com/pwc-australia-port-of-brisbane-unveil-blockchain-supply-chain-pilot
https://www.zdnet.com/article/alibaba-pilots-blockchain-supply-chain-initiative-down-under/

2.1.2 Blockchain based supply chain

Blockchain can offer various benefits in a supply chain scenario. Chief among these:

1. **Saving time** – processes and transactions taking days in a traditional manner can be significantly reduced.
2. **Removing cost** – by offering extra automation and the removal of intermediaries
3. **Reducing risk** – of tampering and fraud. Increasing role-based transparency and visibility; helping dispute resolution.
4. **Automating trusted processes** and ensuring the trust in the record system

A manufacturer can create a blockchain-based system for holistically managing all its relationships with suppliers of parts and components, and having all necessary stakeholders share the exact same information. Moreover, manufacturers can use a blockchain-based system for managing the financial relationships and transactions connected to each step with all its suppliers. All stakeholders will be able to enjoy elevated levels of trust and accountability provided by the blockchain integration, by having the blockchain as the single source of truth supporting immutability and non-repudiation.

The vision calls for providing a Blockchain based platform infrastructure for supply chain scenarios. The low hanging fruit in this realm is to help track supply chain transactions reliably and transparently. Efficient track and trace solution brings clarity to the relationship among all partners in a supply chain relationship. Transactions on the blockchain create a traceable permanent history of the product or interaction, throughout their lifecycle, which is shared by all members of a collaboration. The end goal is to enhance trust in a trust-less environment without relying on a single entity to manage all the information. The distributed nature of information sharing provides an undeniable single source of truth reflecting reality as it took place.

As a part of the vision to incorporate a blockchain based supply chain platform, the first step is to incorporate partners data, putting a specific emphasis on IoT data. The data serves as the driving element to the agreed upon logic which resides in the blockchain level as well in the form of smart contracts. Examples as to the kind of data which shall participate includes negotiations process, tenders processes, and data coming from sensors which enables tracing the state of items in real-time to drive potential notification on out-of-bounds conditions affecting the agreement.

Along with the transparency brought in by the technology it is important in enterprises scenarios to ensure selective visibility of collaborations and data. Data needs to be available to the participating partners and only to the designated partners. Visibility and full transparency should be limited to only the members who are entitled to have access to the data. Thus, the technology provided will ensure that differentiated visibility is supported at multiple levels. A first kind of visibility is expected in cases of full collaboration and exposure to data among members of a consortium who share a blockchain channel. A second level may allow companies to be aware that a business relationship exists between other partners without being exposed to the internal details and data associated to that relationship. Finally, a complete separation in which a company is not able to know that other companies are partnering and exchanging business related information.

One of the advantages enabled by the blockchain as the underlying technology, is to provide accurate real-time distributed visibility fed by collaborators data. Thus, serving as a single source of truth, replicated on all interested and allowed entities. The view is shared by all entities in real-time, thus the exact same view is provided and can be seen by all partners. This capability enables a coherent and updated view of the status of the supply chain ecosystem including availability of production resources, flow of materials and components, and the associated state as can be reported by attached IoT devices; all according to the scope, rules, and conditions agreed upon among the network partners. This journey starts with providing a proof of ownership to a specific entity and the trusted chain of events that followed changing the state of the entity in question (such as passing ownership).

All in all, a blockchain infrastructure can be used to reduce the rate of disputes and errors in logistics and to enable real-time tracking of transactions in the supply chain providing elevated accuracy, security and speed.

2.2 Blockchain based scenarios suitable for supply chain

In general, several kinds of capabilities can be supported by a blockchain as a basis for the MANU-SQUARE platform.

- Tracking and Tracing: the first and foremost examples of blockchain based capabilities supporting the MANU-SQUARE platform, and blockchain based supply chain engagements at large lies within the realm of tracking and tracing. With the help of technology that can easily and cheaply provide product identification (such as QR codes) enable the individual products to be traced throughout their lifetime; including change of ownership or inclusion into larger shipments. This capability enables the platform to keep track of the state of a product and play back the entire history of the specific product from the moment it was introduced into the blockchain until the present day.

  There are numerous examples of such capabilities being introduced by large and small companies (Track and Trace – Deloitte's blockchain-powered supply chain solution[4], IBM: Tracking an item through a supply chain with blockchain[5]; BlockChain is the key to modern Track & Trace Systems[6], and many more such examples exist).
- Tie in fragmented and siloed systems: the current practice is that each organization maintains its own system of records. Information sharing is done by the exchange of written (preferably digital) documents. This leads to fragmented systems in which each party holds on to their own records and no reconciliation is performed among all parties, thus it very well may happen that different entities hold differing views on what has happened. This manner of conducting business is prone to inconsistencies and disputes[7]. The shared ledger, with all the capabilities described above is positioned to remedy this situation by providing a unified view to all participants at the same time.
- Enhance trust and visibility by providing confidence in available data and in the fact that counterparts will have access to the exact same data. One of the important consequences of this capability is to provide clear picture for making decisions which may open the door for new and more agile business models.
- Minimize costly disputes – Current systems in which every partner holds on to their own system of records, thus potentially observing a different version of reality, leads to disputes which are often numerous and the resolution process may be costly and time-consuming. By virtue of having a single source of truth which is verifiable and leaves a provenance trail through the system, the number of disputes can be reduced. Remaining disputes resolution time is supposed to be reduced due to the data is being readily available and verifiable for all parties to access.
- IoT data integration: the inclusion of IoT data in blockchain transactions may bring a lot of value to the MANU-SQUARE platform in various ways. First, it is a source of data which can be fed into the platform and serve as a digital window into the physical world. The inclusion of IoT data leads to greater transparency within and across organizations. Nevertheless, not only data influx is of importance but also the automatic manner in which actions can be taken based on incoming data. These actions can reside in smart contracts which may contain logic that takes specific actions based on incoming data. Actions, are programmed into the blockchain, and can perform a multitude of operations, starting from sending alerts, all the way to performing payments.
- Digital twin data – the inclusion of IoT data in a trusted form enhances the digital twin capability, which aims at constantly creating and maintaining a digital structure which mirrors a corresponding entity or structure in the physical world. The full trail available from the physical world helps to conceptualize the lifecycle of a physical construct from birth to death, including all the transformations along the way.

The possible blockchain based supply chain functionalities that Blockchain can support can be eventually grouped in one of the scenarios that will be introduced below. Each scenario would make use of built-in Blockchain first principles, with some specific scenario related information, customization, and code.

[4] https://www.youtube.com/watch?v=aFGOcIgyJdQ
[5] https://www.youtube.com/watch?v=WML6YtYvBT4
[6] https://www.linkedin.com/pulse/blockchain-key-modern-track-trace-systems-mike-bradley-sr-/
[7] https://medium.com/@lyaffe/using-the-blockchain-to-bridge-data-silos-852aa82d84eb

2.2.1 Automating contracts and processes

Terms of a contractual agreement between two parties can be manifested as a smart contract running in the blockchain. For example, a buyer wants an efficient way of converting a purchase order into validated, self-executing contract updated to reflect the status of the supply. The agreement and resulting state is shared among all stakeholders. In this case the blockchain provides a shared record of the contract status which is updated as the contract progresses. Benefits of such an approach include the increased efficiency and transparency across the supply chain, as well as risk management improvement through the near real time update of all contracts. Number of disputes and the resolution time thereof is expected to decline significantly. A representative manifestation of this capability in the MANU-SQUARE context may be the establishment of an RFQ process between different parties.

2.2.2 Traceability and provenance – IoT Integration

Provenance of each component part in a complex system is hard to track, but is of great value. Such information may include manufacturer identification, production date, batch and even the manufacturing machine program. In addition, producers require transparency on where and how their raw materials and sub-contracted products and supplies are made. In addition, governmental entities in some areas require more information about corporate supply chains, with penalties for non-compliance. In such a case blockchain enables the safe digital transfer of material and goods end to end, across the supply chain. That information includes which party had ownership to what part at what time, and what changes were performed. This can be aided for example by attaching RFID or QR codes to specific material / shipments. Expected benefits include a verifiable, traceable transaction log preventing any party from altering the agreed upon transaction. In such a case the blockchain holds complete provenance details of each component part, making it accessible to the sub-set of stakeholders that require access to that information based on the contractual agreements between the parties.

In MANU-SQUARE , Such a scheme can be used for example for initiating automatic payment of an agreed upon part of the global sum based on the current advancement of the process as can be deduced by RFID readings.

The blockchain can provide a complete audit trail in the supply chain (asset tracking capabilities that demonstrate provenance information across the supply-chain), by tracking provenance of parts from birth to death. Blockchain based supply chain relationships will become a validated, trusted, self-executing process, supporting non-repudiation. The immutable nature of the blockchain can be used to maintain the complete provenance details of each component. The distributed and replicated nature of the ledger enables visibility to all allowed parties.

In such a case, even if a problem is discovered later it would be possible to track back the history of the problematic component, and deduce from that possible related problematic components.

2.2.3 Adding state tracking – via IoT integration

Provide the capability to track, monitor, and report the location and status, of shipments, goods, or supplies with the integration of IoT devices. This may help optimizing logistics and provide early warning when things are going wrong, and a larger chance for a plan B (potentially automatic) to take place due to early detection of foreseen problems and delays. Even if a plan B is not in place for a certain unexpected occurrence, if disputes arise, the resolution is made easier because the immutable ledger holds records of when and where the event occurred and who the responsible party was at the time.

In MANU-SQUARE , such a capability may help the coordination of movement of goods among multiple stakeholders. Location and state of exchange goods are known and shared in real-time among the relevant parties,

Location tracking is the first and easiest to implement and see value of the introduction of IoT devices interacting with the blockchain and smart contracts. Such capabilities may go a long way to enhance and enforce key quality criteria. For example, for perishable food delivery we can track not only the location but also the environmental conditions such as temperature of the shipment, thus ensuring that indeed the shipment abides by the delivery requirements. In addition, it may provide early warning, for example having the temperature reaching too close to the allowed maximum and have some time to save the shipment from being thrown away completely.

For example, a company depending on various suppliers can optimize its operations by having a clear updated picture of the location and availability of items it requires for its own operation. Blockchain can help tie in and optimize the timing of incoming supplies and reduce the quantity of storage required. Having an updated view based on data supplied by IoT devices can provide valuable information to optimise operations.

3 BLOCKCHAIN USE IN THE MANU-SQUARE PROJECT

3.1 Blockchain roles in the overall platform architecture

As can be seen in Figure 5: Architecture - high level view, the blockchain platform plays a role both in the underlying data layer as well as in the tools layer. It is located at the lower infrastructure layer of the platform, exposing interfaces leading to services that the different higher level components of the platform can use. At the data layer it does serve as a unique kind of data store in the form of a shared ledger exhibiting the capabilities detailed in 2.1. At the same time, it does play a role at the tools layer as well, since a part of the overall logic does reside in the blockchain internals in the form of a smart contract[8] which is a programmatic manner to declare and enforce the rules that govern specific interactions via the blockchain. Thus, different platform components shall use interfaces exposed by the blockchain component in order to take advantage of the capabilities and promises of a shared ledger. The blockchain component shall expose a REST interface to the rest of the platform components. There shall be two main kinds of activities supported by the interface:

1. Invoke smart contact transactions – intended for MANU-SQUARE modules to be able to invoke transactions residing in smart contracts
2. Query - expose query capabilities to retrieve data which was previously stored at the blockchain.

[8] In Hyperledger Fabric smart contracts and implemented in the form of chaincode (https://hyperledger-fabric.readthedocs.io/en/release-1.3/chaincode.html)

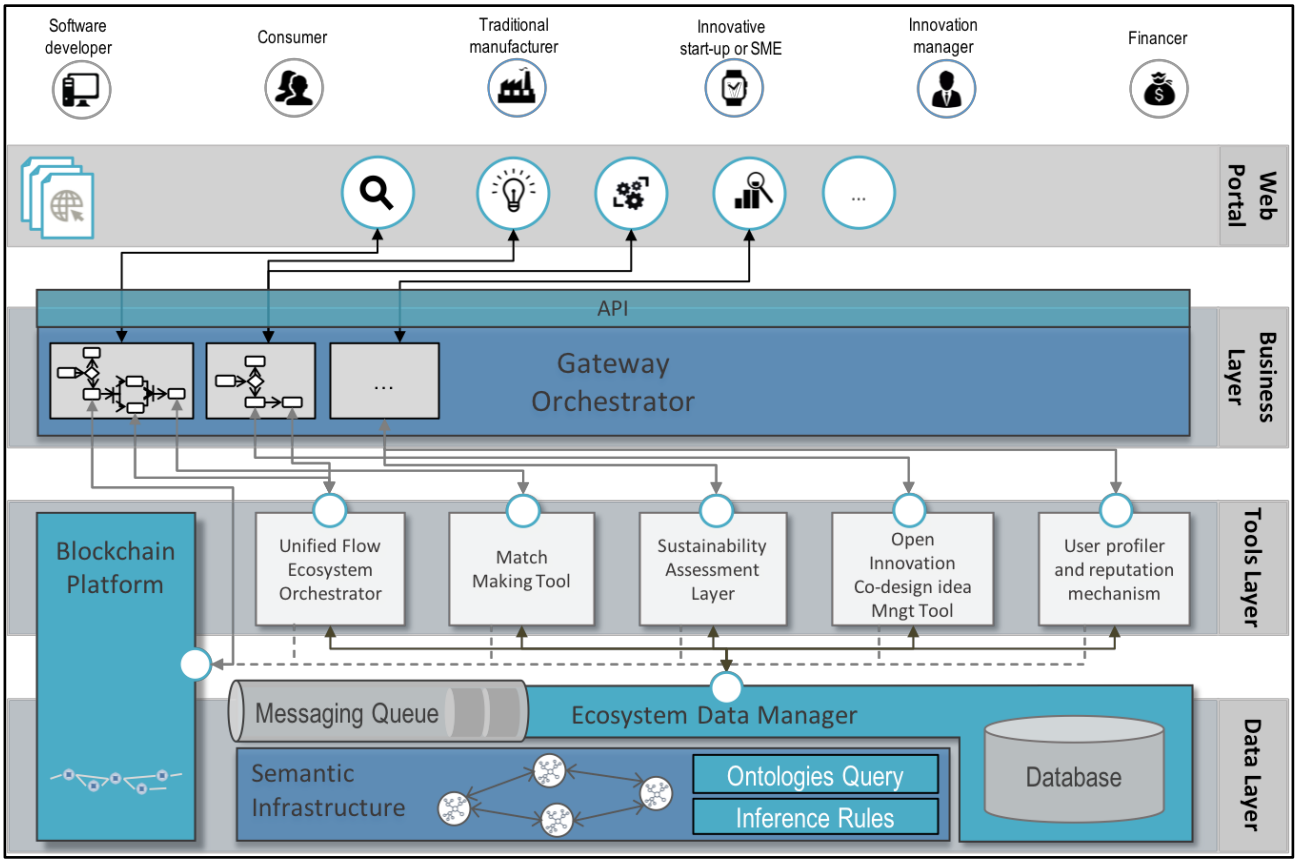


Figure 5: Architecture - high level view

This section is intended to dive deeper into the definition of the specific use cases foreseen within the integration of a blockchain infrastructure within the MANU-SQUARE project. Taking as a starting point the DoA and the evolution of the project use cases as the project advances as main result of the WP1, the Blockchain integration shall focus on five representative scenarios, namely ***traceability of exchanged goods***, ***manufacturing systems' data certification, traceability of innovative ideas***, ***RFQ management*** and ***reputation management***

At a high level, the use cases can be divided into two broad categories. First, matching between surplus and need (for example of production capacity or by-products) to support capacity sharing. Second, innovation management of collaborative design. For both categories the MANU-SQUARE platform shall take advantage of a blockchain based infrastructure as the providers of a trusted (in a trust-less environment) single source of truth.

3.2 Capabilities supported in the MANU-SQUARE platform

The functionalities that a blockchain infrastructure can support in supply chain scenarios will be translated into 5 specific use cases to be deployed in the MANU-SQUARE environment. Hereinafter, a short description of each use case is provided.

3.2.1 Traceability of exchanged goods

MANU-SQUARE supports the exchange of goods among suppliers of manufacturing capacity and providers of machines/technologies able to fulfil their needs. The offering of such a service is therefore connected to the manufacturing and delivery of items whose supply history could be of relevance in guaranteeing an adequate level of trust among customers and the MANU-SQUARE platform. The blockchain infrastructure shall therefore support the traceability of the supplied items, tracking one or more levels of the involved supply chain (i.e. raw material supplier to company manufacturing the requested item to MANU-SQUARE platform to final customer). This should involve the tracing of every step of the supported supply chain with IoT devices as data providers, and will enable tracking the most relevant information connected to the supplied items, in each phase of their lifecycle.

3.2.2 Manufacturing system data tracking and delivery

IoT in the manufacturing system provides real time view of manufacturing data. The integration of IoT devices in the manufacturing system is instrumental for the real time acquisition of manufacturing data to be used at different services throughout the MANU-SQUARE platform. The extraction, storage, and access to such data for the calculation of sustainability related performance indicators in the manufacturing system shall be supported by the blockchain infrastructure. In this case, the adoption of a blockchain to support data integration is meaningful as the traceability of the data is relevant for the calculation and provisioning of assessment data, both internally to the company and externally through the MANU-SQUARE platform. The ability to provide traceable data allows the MANU-SQUARE ecosystem to create a trusted layer related to the calculation of sustainability impacts that can be therefore exploited to generate reliable sustainability performance related rankings.

3.2.3 Traceability of innovative ideas

In this use case Blockchain can be applied to the tracking of contribution of innovative ideas within the MANU-SQUARE ecosystem. Considering that the platform is intended to support the evolution of ideas form basic concepts to fully set-up projects in a cooperative and open manner, the Block Chain should be involved to keep track of the contributions of each participant and register the ownership of every single contribution. This capability should support the ability to reward respectively the participants of ideas creation at a later stage in the product development.

One possibility is to have a blockchain based incentives program in place rewarding participants for contributing ideas and information to the project. Incentives can take many shapes and could encourage people to participate in the research effort.

3.2.4 RFQ management

RFQ is a structured and often complicated process which may involve multiple hops and interactions between the entities involved (from the initial offer through a negotiation process, culminating in a signed deal). The blockchain will help structure the process and safe guard all the interactions and advancements of the process throughout the lifecycle of the process. The process is initiated by a prospective customer and is targeted towards a potential supplier, and may consist of various items to be agreed upon (such as price and time). Several rounds of negotiation may be required for the positive (or negative) finalization of the process. All information exchanged digitally shall be a part of a permanent record kept and made available by the blockchain.

3.2.5 Reputation management

Reputation management is of crucial importance to the adoption of the platform. Thus, the trust that can be associated with this component is of great importance as well. The blockchain infrastructure shall support the traceability of the entire history related to the reputation of all involved entities at different points in time.

3.3 Contextualising blockchain use in the capacity sharing scenario

Following a first version of the envisioned business process flow for a capacity sharing scenario[9]. The process in general calls for the following steps:

1. Customer to Invoke the *match making capability* of the platform to receive a ranked list of candidates
2. Embark on an *RFQ process* between the customer and the prospective supplier
3. For the chosen supplier, *start the project* upon input material delivery
4. *Periodic monitoring of the project status* and a corresponding *update of the project status* and advancements
5. *Project closure* – delivery of the goods
6. *Update reputation management* for both the customer and the supplier

[9] A comprehensive explanation can be found in D1.3 (Business processes and early validation scenarios)

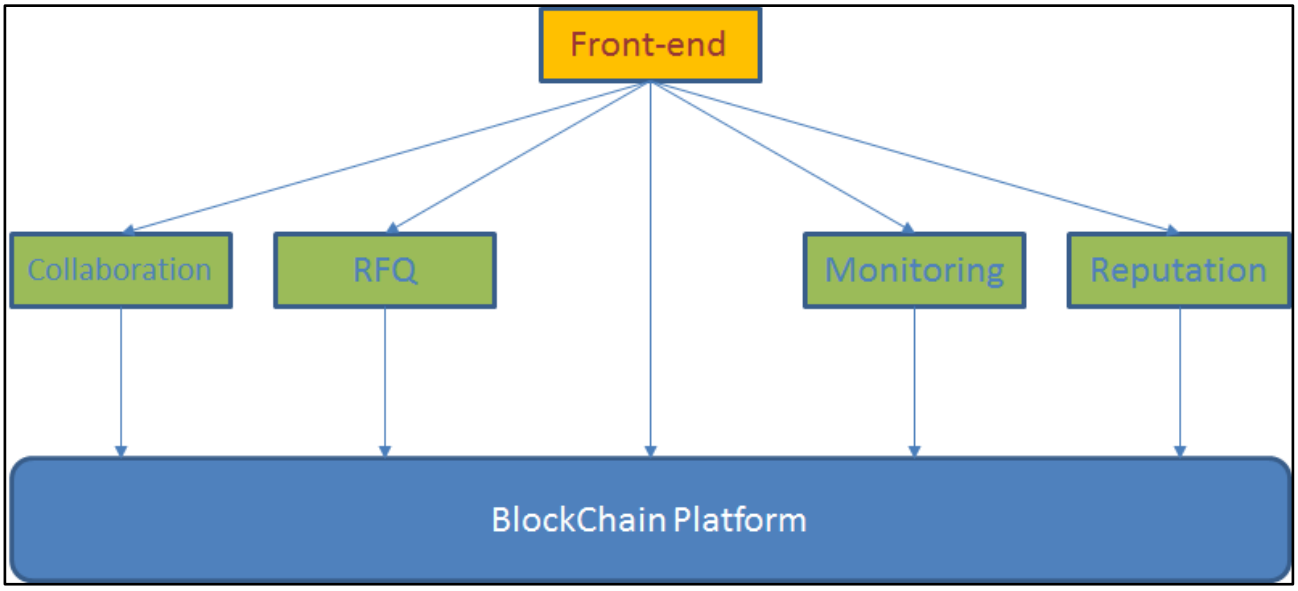


Figure 6: positioning blockchain in MANU-SQUARE

In the following we can see the main corresponding interactions of the blockchain component with additional MANU-SQUARE platform components (for a visual reference please turn to Figure 6). For all such interactions there are two broad categories of actions that are taken and supported by a REST interface exposed by the blockchain platform, namely invoking a transaction on the blockchain, mainly intended to record data, and querying the blockchain platform for data previously inserted (latest state or historical data). A possible third mode of interaction will be discussed and developed based upon need, in which the blockchain platform invokes callbacks on registered entities based on events in the blockchain activity.

1. ***Collaboration*** – As explained above the lifecycle of a capacity sharing MANU-SQUARE interaction goes through a set of states in which interaction is expected between parties in the platform (mostly a customer, a supplier, and the platform). At a high level the blockchain shall serve as the reference point which keeps track of the current location in the process view for each interaction, along with all the history that led to this point. The information shall be querriable for all allowed parties (such as the platform manager, supplier, and customer), via the platform front-end. Potentially an exposed programmatic interface shall be provided as well.
2. ***RFQ management*** – the blockchain component shall provide the underlying mechanism for keeping track and advancing the RFQ process among the customer and the suppliers. Naturally, the blockchain shall keep track of the history of the interaction from beginning to end and can serve as the reference point for the agreed upon terms and the evolution of the process. An example tentative basis for this component is described later in section 6.
3. ***Project monitoring at the production site*** making use of IoT devices potentially in connected machinery (or specific RFID gates) for tracking in real-time the state of the supply chain to verify the advancement of the production at each moment in time, and provide feedback to the planning workflow to be able to evaluate if there are deviations from the original plan. This process terminates at the end of the production cycle with the delivery of the goods from the supplier to the customer.
4. ***Reputation management*** – the reputation management process shall be handled by the MANU-SQUARE platform, for all involved entities, both suppliers and customers. This process is deemed an important one for attracting entities to use the platform, and for the long term sustainability of the platform. Having a blockchain based infrastructure to keep track of reputation management is essential for the trust associated with the entire platform by current and prospective customers to join and use the platform.

Similar capabilities and interactions are foreseen as well for the collaborative design and ideation management scenarios. The content and internal structure of tracked items shall differ, but at the core, similar blockchain related processes and

interactions shall take place. Thus, for innovation management and ideas tracking the blockchain infrastructure shall provide a basis for the management of the entire process from a design need all the way to delivery of a product. Including along the way the contribution of each party to the final products

4 BLOCKCHAIN FROM THE EDGE

As the focus of T3,1 is about connecting IoT devices to a blockchain infrastructure, the rest of the deliverable is devoted to describing the work performed on this specific topic.

One of the main intentions and challenges in fulfilling the promise of a blockchain based supply chain platform is to ensure that IoT devices are treated as first class citizens in a Blockchain network. In that respect the intention is to be able to incorporate data from IoT devices directly into blockchain transactions as a provenance preserving data store and as a possible trigger to actions. The intention is for the devices to be communicating directly with the blockchain network without resorting to intermediate applications and services which will carry the work on behalf of the edge device.

4.1 Overview of work done

The effort in T3.1 started by deploying a Hyperledger Fabric peer on a Raspberry Pi (RPi). The RPi was chosen as a representative IoT device, which is very popular and heavily used in the field. That mostly entailed creating a new target CPU architecture to the build process, since ARM is not an officially supported architecture. This work was mainly performed in order to demonstrate basic feasibility of the approach. Namely, it is possible to create and deploy an entire Fabric network on a set of RPi devices. In addition, we demonstrated the deployment of a simple Fabric Ordering Service[10] on a RPi as well. With these steps we demonstrated the possibility to run a full-fledged network (including peers and an ordering service) on IoT devices, but this path was not the main one taken since it is not certain there is a business need for this approach. Thus, we mainly concentrated on the second stage which is to deploy a Hyperledger Fabric client on a RPi.

With this we did implement the vision of devices acting as first class citizens in a fabric network. We implemented and demonstrated the complete lifecycle of a client within a Fabric network, from enrolment with a certificate authority (CA) and obtaining certificates, all the way through bi-directional interaction with the Fabric network by submitting transactions and register for callback events. With this we demonstrated the integration of IoT devices, such as sensors, as providers of data, which in turn, trigger Blockchain transactions.

Once the full client functionality was in place, we turned to devise and run performance benchmarking of Fabric client running on a RPi. This step was important to determine the feasibility of this approach, namely to determine whether the approach can sustain a real world scenario. Benchmarking was centred around understanding characteristics such as sustained throughput in the form of number of transactions per second that can be invoked by a client on a pi. In addition resources consumption on the RPi while running the experiments was measured and diagnosed. Finally, latency in the form of the elapsed time from transaction initiation to obtaining confirmation of the transaction inclusion in a block was tracked.

Once performance characteristics were gathered and analysed, we turned to creating an end-to-end demonstrator centring around sensor readings from a Raspberry triggering blockchain transactions, and with this same RPi acting as a gateway for additional (typically smaller) IoT devices, enabling these smaller devices to make use of their data to invoke blockchain transactions as well. In addition, a query mechanism was demonstrated by which data stored in the blockchain can be displayed. Overall the depicted scenario followed a shipment of goods that need to be kept at certain conditions in terms of temperature, humidity, and pressure. The demonstrator enables all involved parties to see the route taken by the

[10] The Fabric component in charge of receiving transactions, presenting them in total order, cutting transaction blocks, and making them available to the Fabric committing peers.

shipment and display the maximum and minimum reading from each of the sensors throughout the shipment journey, thus all parties can determine easily whether shipment abided by the agreed upon conditions.

As a prequel to the demonstrator described above we explored and demonstrated blockchain support for negotiation. This support enables a party to create and send an offer, including conditions[11] as mentioned above, to another party in the MANU-SQUARE ecosystem. Both parties can proceed to submit counter-offers, changing any aspect they see fit. At the end of the process we either reach an agreement or get a rejection. Regardless of the end result, the blockchain based support enables both parties to have access to the exact same information regarding the negotiation process. This capability was integrated within the demonstrator as well.

4.2 Blockchain Identity

The first action which needs to take place in a permissioned blockchain before an entity can participate in the network is for the entity to register and obtain the proper security certificates which enable the identification of the device by blockchain network entities. Obtaining an identity for the Blockchain entails first for an administrator to register the device in the proper CA, the device can in turn enrol with that same CA, finally obtaining the certificates which enable it to transact directly and securely with the blockchain network. An IoT device carrying its own blockchain identity in the form of the proper certificates becomes a full-fledged member in the blockchain network.

Once a blockchain identity has been established the IoT device can produce, sign, and invoke transactions using an embedded Hyperledger Fabric client running on the device itself. The flow will be identical to a client running on "standard" machines, namely produce a transaction proposal, send it to the appointed Fabric endorsing peers, collect back the endorsements results and verify that the endorsement policy is respected; if so, send the transaction along with the endorsement results to the ordering service for the transaction to be included in a forming block.

Our client running on the edge acts and is viewed as a standard client, in particular it can establish bi-directional interaction with the Blockchain network, by invoking transactions as described above, and registering to receive call-back events from the blockchain network on items which are of interest to it, such as the inclusion of transactions in a block, which was received and committed by a Fabric peer.

4.3 Expected flow: Trigger smart contracts from the edge

There are two main envisioned flows for the proper inclusion of edge generated data in blockchain transactions. The first is for every new relevant measurement taken on the edge to produce a corresponding transaction which will store the information in the blockchain network. Logic to handle incoming data may reside in blockchain smart contracts, invoked based on call-back events from the blockchain network, or process in batch at later certain points in time.

A second option is for edge rules to report on interesting events. This option entails having some sort of processing to take place on the edge, normally to follow simple rules, and invoke blockchain transactions only when the local logic determines that a worthy situation has taken place. Whenever such an event materializes it shall trigger a corresponding blockchain transaction. For example, whenever a temperature is within expected bounds no external calls shall be made, and only upon crossing a threshold a corresponding transaction shall be invoked from the edge device.

4.4 Advantages

There are various advantages to triggering smart contracts from the edge, mainly in the form of added trust and security due to the direct interaction of an edge device transacting using its own established identity. This leads immediately to the possibility of supporting edge scenarios, such as supply chain, without mediators, which in turn adds an extra hop that needs to be verified along the way, in order not to lose security and trust related blockchain properties. Moreover, making use of the design laid out above we can have a lightweight mechanism and deployment which can run on edge devices and reduce the amount of cloud related interactions required by the edge device.

[11] Mostly physical condition under which the item in question needs to be kept, such as temperature and humidity.

4.5 Challenges

The main technical challenge faced by any attempt to involve IoT devices in an IT environment is the resource constrained nature of such devices. Careful design needs to be carried out to ensure that the device in question is capable of fulfilling its intended role using its available resources. Accordingly, we embarked on a detailed process to evaluate the performance and resources consumption of our approach to be able to respond to feasibility questions about this approach.

A second challenge faced by this approach is that IoT devices are not always connected and there is a need to support unstable connectivity, including fluctuations in available bandwidth all the way to sustaining periods of no connectivity at all. We did elaborate on a design using a "store-and-forward" approach, but the implementation of this aspect shall be covered at a later stage (outside the intended scope for this project).

Finally, the fabric network itself needs to be able to support a large number of devices transacting directly with it. The support of a large number of enrolled identities is provided by the possibility of using LDAP as the user registry. Performance in terms of throughput to the fabric network is being handled constantly in the development of the code base itself (and is out of the scope for this project).

4.6 Current embodiment

Based on the design and desired capabilities detailed above we have implemented the system which is depicted in Figure 7. To implement an entire blockchain network to run our scenarios and experiments, we have created a basic blockchain network consisting of two organizations with a couple of peers belonging to one organization, and one peer belonging to a second organisation. In addition an ordering service has been set up and connected to the peers, and a CA has been established to handle enrolment requests for entities belonging to each of the organizations.

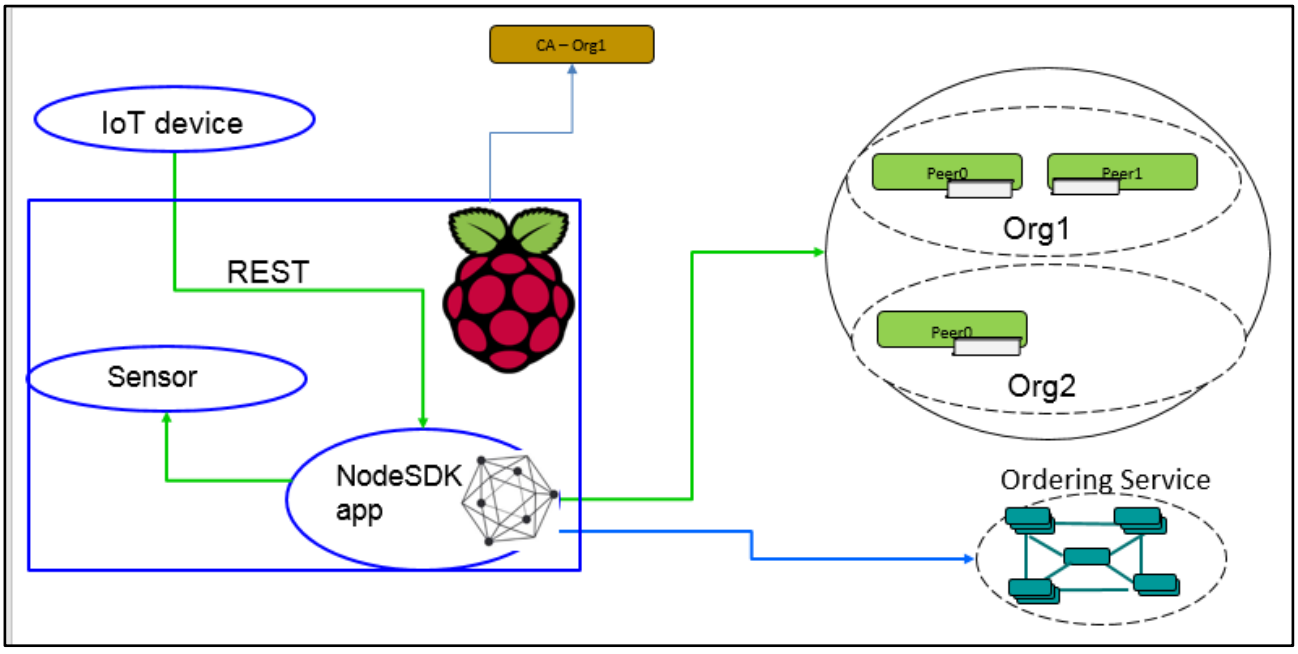


Figure 7: IoT in Blockchain – embodiment

In addition to the basic blockchain network the bulk of the effort was to set up an edge device to interact with the blockchain network. The base structure we have set up and experimented with includes an application, running inside a RPi, which embeds internally a Hyperledger fabric client. The application interfaces with the RPi itself to retrieve up-to date values of internal sensors installed, which are then used to initiate corresponding blockchain transactions using the embedded Fabric client. In addition the client application exposes a REST interface which enables it to serve as a gateway to additional, usually smaller and weaker IoT devices, which post to the client application their updated values, which are once again used to invoke corresponding blockchain transactions. This set up demonstrates the manner in which IoT edge

devices can become first class blockchain citizens, and in addition serve as a gateway for less powerful devices to participate in the network as well.

4.7 Blockchain from the edge, and end-to-end scenario

4.7.1 Creation and deployment of Channels and chaincode

At the complete flow we are providing, the first steps consist of creating a channel to be used for a business interaction along with an associated chaincode (refer to Figure 8). All these steps are performed from a standard admin server, and do not exhibit any changes due to the future participation of edge devices as client. A Fabric channel is analogous to a separate ledger which only the organizations which are members of the corresponding consortium are aware of. Thus, the first stage in the creation of every business interaction through the blockchain consists of the creation of a new channel.

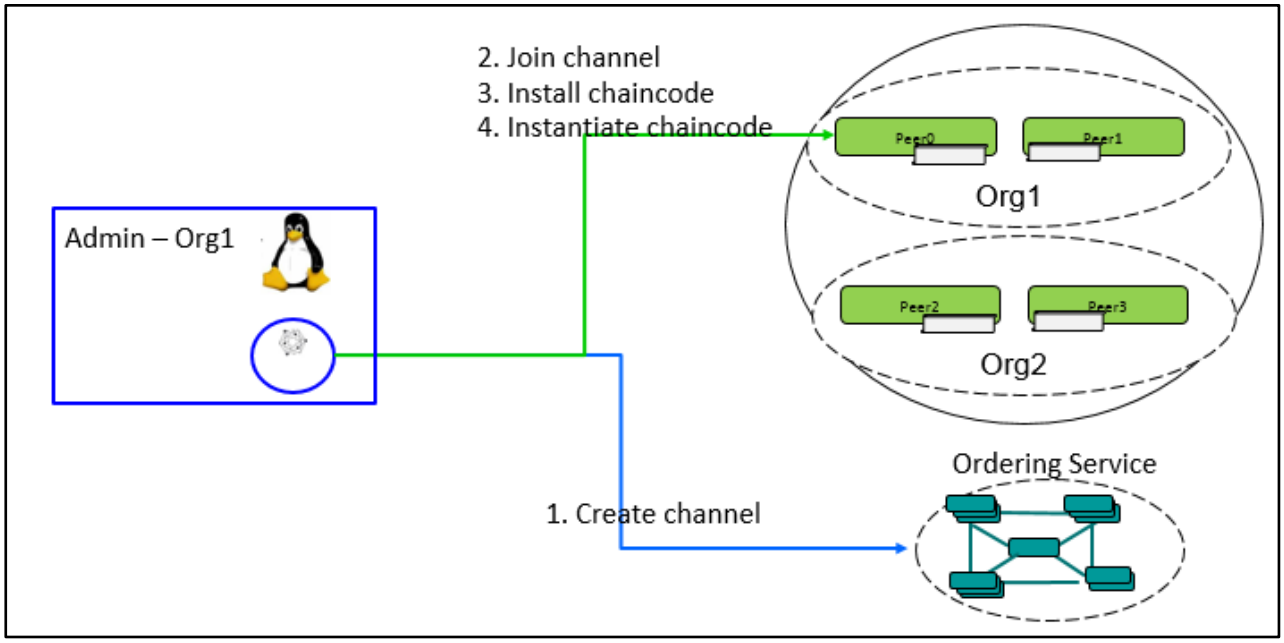


Figure 8: Channels and chaincodes

Once a new channel has been created the administrator proceeds to instruct the corresponding peers from the organizations who belong to the channel that they should join the channel. This step establishes the relationship between the corresponding peers and the new channel. The next step consists of installing the chaincode to be used in this business interaction on a subset of the peers that have joined the channel. These peers will be eligible to serve as endorsers of the specific chaincode related transactions. Endorsing peers are the only ones that speculatively execute the proposed transaction, and return to the caller a corresponding read and write sets which will be used at the last stage by the committing peers to determine whether a transaction is valid. In addition, endorsing peers indicate whether they support the inclusion of the proposed transaction. A more detailed description can be found in[12].

4.7.2 Enrolling an edge device into the blockchain network

The following stage in the process consists of enrolling the edge device such that it carries its own identity when interacting with the blockchain (refer to Figure 9). This is a two stages process, in which the first stage is carried out by a standard admin and consists of registering the device with the organization's CA. In return the CA returns a secret which the admin passes back to the device, in an encrypted manner, making use of the REST interface exposed by the application installed on the edge device.

[12] https://hyperledger-fabric.readthedocs.io/en/release-1.3/peers/peers.html

Equipped with the decrypted secret, the edge device proceeds to enrol itself with the organization's CA, in return obtaining certificates. These certificates are later used when the edge client wishes to interact with the fabric network.

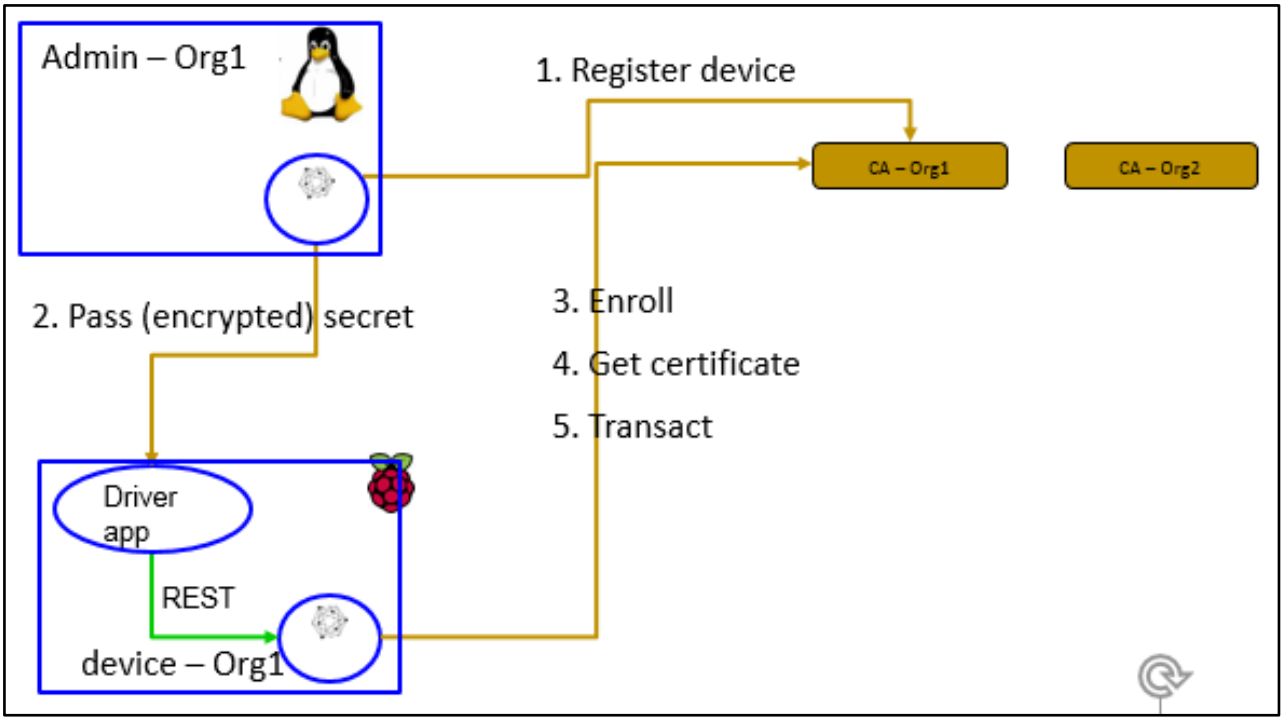


Figure 9: registration

4.7.3 Transaction invocation from the edge

At this point, the stage is set for starting to use the blockchain network (refer to Figure 10). The goal set for this stage was that a Fabric network should not be possible to distinguish between a traditional fabric client and an edge based fabric client. The accompanying demonstration shows that we have managed to achieve this major goal.

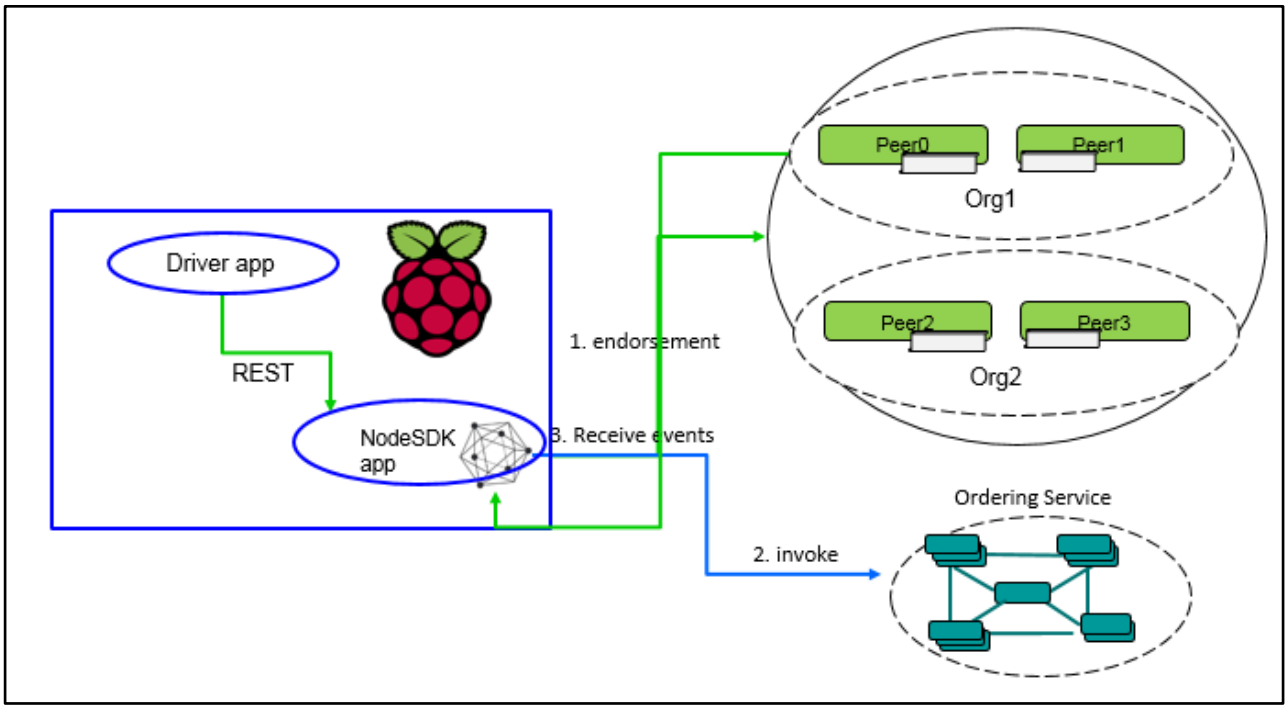


Figure 10: transact from the edge

At this stage, the edge based client can interact with the blockchain just like any other client. The application running at the edge prepares, signs, and sends transactions just like any other client. Namely, it shall prepare a transaction and send it to the relevant endorsing peers. At a second stage it shall receive responses from the endorsing peers and parse the results to see whether the transaction should go through (ensure deterministic execution of the transaction, and that the endorsement policy established for the specific chaincode is upheld). Transactions that have passed this stage are sent, along with the endorsement information, to the ordering service. The ordering service determines the order in which transactions will be inserted into blocks, cuts a block when it is ready, and makes the blocks available to the peers to grab and commit.

Finally, the client may register itself as a listener to call-backs produced by events taking place at the peers. These events mostly relate to specific transactions or blocks.

4.8 IoT integration test

Based on the constructs laid out above we have created a demonstrator for the IoT integration scenario. A similar set up served for the implementation deployment and execution of related performance benchmarking detailed in section 5. The individual ingredients comprising the demo are depicted in Figure 11. The setup is comprised of a base fabric network including an ordering service and several peers belonging to several organizations. In addition, each organization has its own root CA. The corresponding cryptographic material has been generated and distributed among all the different players. In addition, two major players have been set up, namely an organization admin running on a standard server, and a RPi which serves as the edge client.

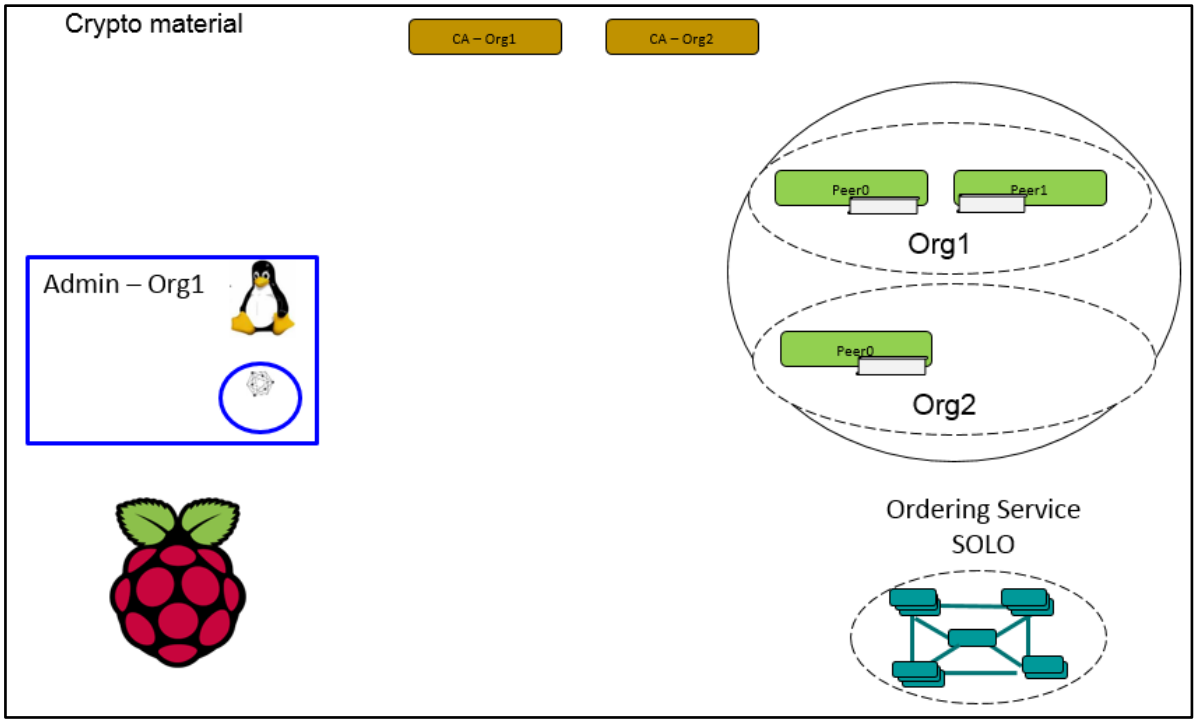


Figure 11: IoT integration demo

Figure 12 depicts the manner in which the different entities described above are deployed. The setup includes 4 VMs running on the IBM cloud, in which the different Fabric entities are deployed (CA, peers, orderer). In addition, we have 4 RPi devices and a standalone server interacting with the cloud based blockchain network.

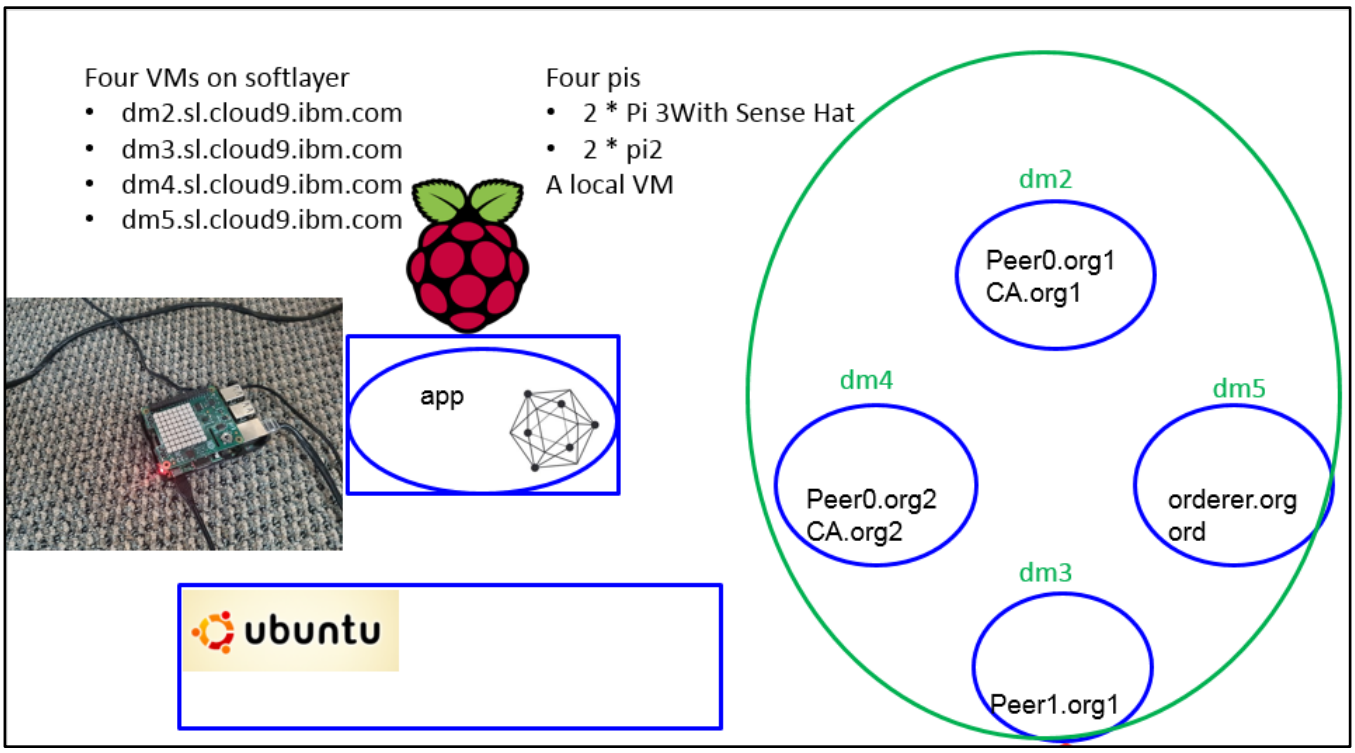


Figure 12: IoT demo deployment

Figure 13 depicts the complete architecture of the IoT integration demo. On the right hand side, we can see the blockchain network. The left hand side contains a VM which acts as a host to the demonstrator frontend, and the edge device, in which an application is deployed embedding a fabric client; local sensors are made available programmatically; and a

REST interface is exposed to external entities such that the edge device may serve as a blockchain client gateway or proxy.

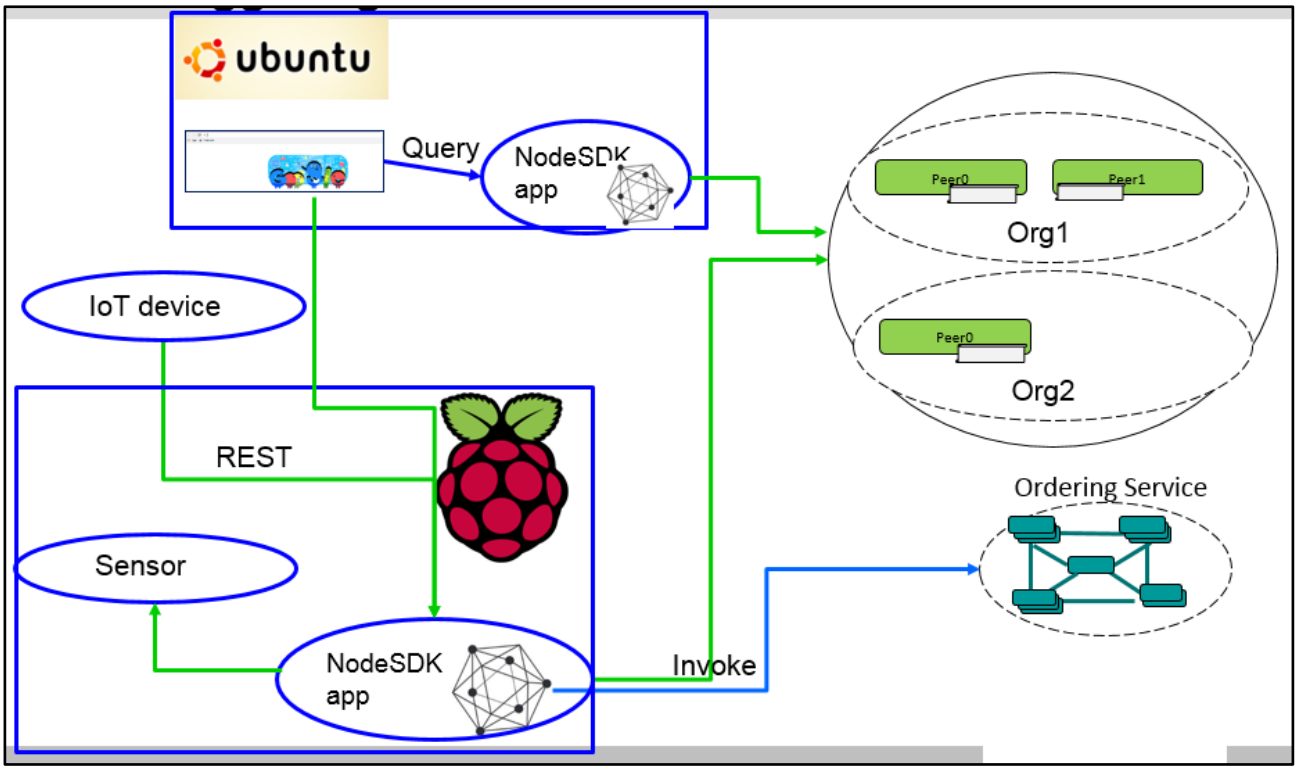
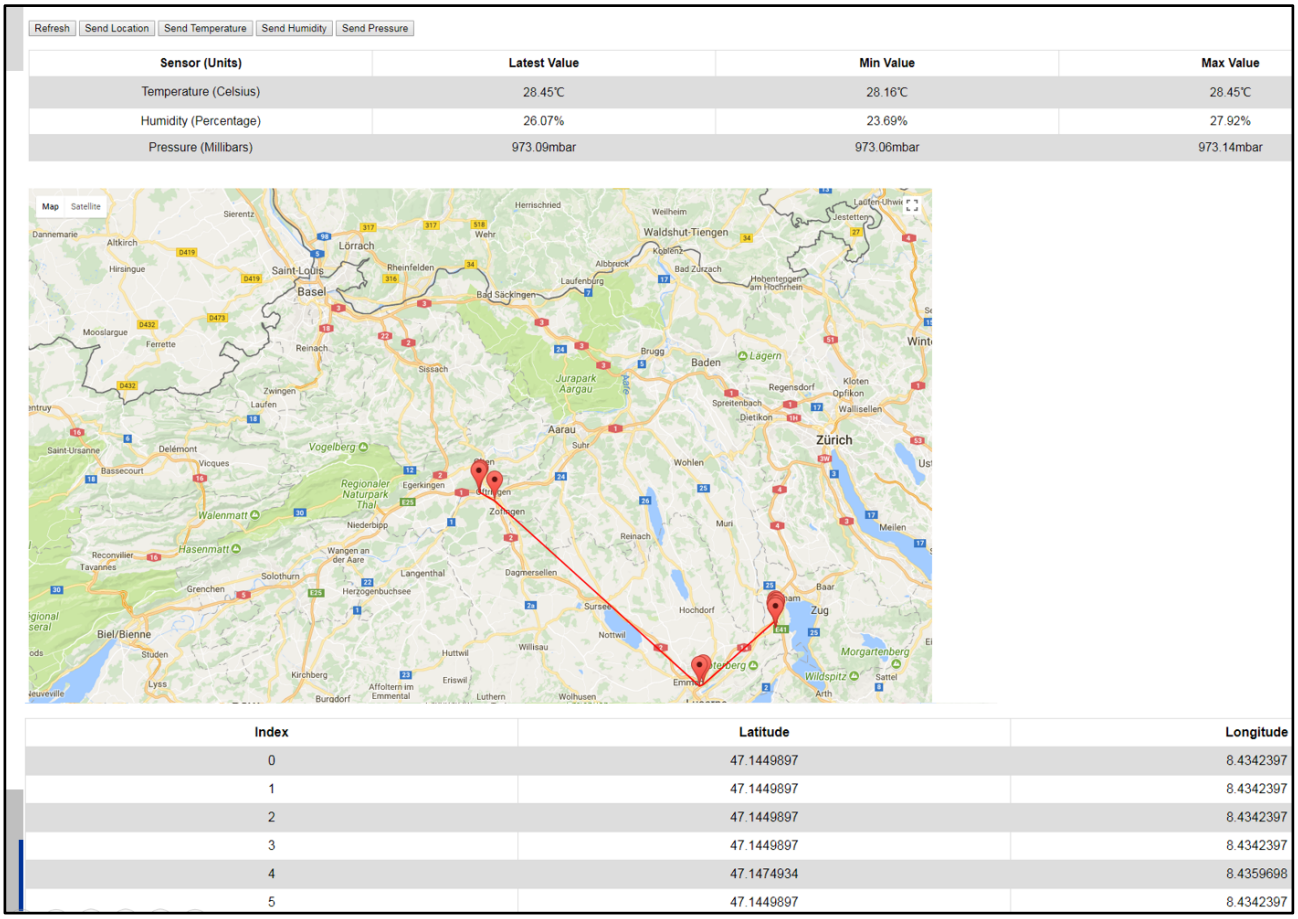


Figure 13: IoT demo in action

The demonstrator itself follows a traditional supply chain script of a tracking and tracing scenario. The storyline is that of an asset produced by one organization and is being bought by a second organization. The asset needs to be kept within a controlled and bounded physical environment which is being monitored by associated IoT devices. We pick up the story when a track carrying the asset leaves the premises of organization A on its way to organization B. Along the way an external location tracking device reports the current location to the RPi using the REST interface exposed by the edge application. The edge application in turn conveys the information to the blockchain by applying an associated smart contract transaction. In addition, local sensors are available to monitor temperature, humidity, and pressure. Periodically these sensors are being read and their current value is being updated in the blockchain via a transaction triggered by the edge application. In the current version of the demonstrator, for the purpose of clarity of the presentation, the invocation of sensors reading and their associated blockchain transactions are initiated from the frontend by clicking the corresponding buttons.

The frontend is hosted on a regular VM and serves both for triggering sensors reading and for querying the state of the blockchain and interpreting the results. The frontend triggers the blockchain transactions by using the edge application exposed REST interface. For querying the state of the blockchain it uses and embedded Fabric client which queries the blockchain peers. As depicted in Figure 14 the frontend displays the most recent value that has been inserted to the blockchain for each of the monitored items, and in addition displays the minimum and maximum value of each monitored item. The route taken by the shipment is being tracked and exhibited on a map as well. Thus, at each point in time users belonging to both organizations can look at the frontend and immediately gather whether the agreed upon environmental boundaries have been respected, and determine the current location of the shipment.



Sensor (Units)	Latest Value	Min Value	Max Value
Temperature (Celsius)	28.45℃	28.16℃	28.45℃
Humidity (Percentage)	26.07%	23.69%	27.92%
Pressure (Millibars)	973.09mbar	973.06mbar	973.14mbar

Index	Latitude	Longitude
0	47.1449897	8.4342397
1	47.1449897	8.4342397
2	47.1449897	8.4342397
3	47.1449897	8.4342397
4	47.1474934	8.4359698
5	47.1449897	8.4342397

Figure 14: IoT demo front-end

Figure 15 depicts the last part of the frontend which enables taking a closer look at the internals of the blockchain itself, by displaying in a (sort of) human readable form the information stored in different blocks within the chain.

Block Index	Data Hash	Transaction IDs	Write Sets
2	581ea71dc9d78cfd88b8c25d4b8a1d305fef1d2fe73c64d653317cceffa90c81	038562fa1dae66c77de17e0f18fa56f33c3217b593adb6bab67edc16f88139ba	{"key":"latest_location","is_delete":false,"value":"47.1449897:8.4342397"} {"key":"location_0","is_delete":false,"value":"47.1449897:8.4342397"} {"key":"location_index","is_delete":false,"value":"1"}
3	cef8be6450918768878d89eacad4551cf7684288aa4eb165bb6c6531524970754	00e2b20c432972244fbdeacb77b499869435fb2bef462af16c2ecaf0c3741af883d0b5c54eb124e2593099a82efc98b472cdd11c7ffc30fcbd4026ac83338d4bfb895a57905eb035182bc55e0f039b6607270fcecdb734c72bb387cdd7dac744	{"key":"latest_temperature","is_delete":false,"value":"28.17"} {"key":"max_temperature","is_delete":false,"value":"28.17"} {"key":"min_temperature","is_delete":false,"value":"28.17"} {"key":"latest_humidity","is_delete":false,"value":"24.99"} {"key":"max_humidity","is_delete":false,"value":"24.99"} {"key":"min_humidity","is_delete":false,"value":"24.99"} {"key":"latest_pressure","is_delete":false,"value":"970.62"} {"key":"max_pressure","is_delete":false,"value":"970.62"} {"key":"min_pressure","is_delete":false,"value":"970.62"}
4	3accb035a6ad052fe1199f6ca0f499bca7f7668f64671e0e4f7ea1f8ecaebba0	a533361222d440b4503c6e508254619ab22f79f486e8e32983a01385cd8376c3	{"key":"latest_location","is_delete":false,"value":"47.1449897:8.4342397"} {"key":"location_1","is_delete":false,"value":"47.1449897:8.4342397"} {"key":"location_index","is_delete":false,"value":"2"}
5	20c476e140b3526e54cfc834c3fbb60793aa239c5a93f2659cd23626f926a470	9890cedf842f37ef91a9e03702cbe5bac14162a9b9dabd3831808b086e3c08eb	{"key":"latest_location","is_delete":false,"value":"47.1449897:8.4342397"} {"key":"location_2","is_delete":false,"value":"47.1449897:8.4342397"} {"key":"location_index","is_delete":false,"value":"3"}
6	8501f16b964e8ec64626f13545a86e442d804d658c3cc819441ca9e59168c80a	08363fce56070d700a90eb5b4d3c47d483df3eda90b4eaed1b43c0e5c4231f08	{"key":"latest_location","is_delete":false,"value":"47.1474934:8.4359698"} {"key":"location_3","is_delete":false,"value":"47.1474934:8.4359698"} {"key":"location_index","is_delete":false,"value":"4"}
7	3726850c7848dc53575733c4b0da5299a38cab6a3159909d9a59c27b4e280399	9efec885a3e2c4afb8ebec65c8322e05b563b90187d830d5bae18c088abbf082	{"key":"latest_location","is_delete":false,"value":"47.1482953:8.4340348"} {"key":"location_4","is_delete":false,"value":"47.1482953:8.4340348"} {"key":"location_index","is_delete":false,"value":"5"}
8	ed50eeccf25b9e3e01b40a17ccdc1d8481fc4d071cbc29c1951827aa4c98d179	79f74d65ea5bb0f9ac99509e112cf248518d267fc6723a124bd1baf2f6aa3ca8	{"key":"latest_location","is_delete":false,"value":"47.1516423:8.4366857"} {"key":"location_5","is_delete":false,"value":"47.1516423:8.4366857"} {"key":"location_index","is_delete":false,"value":"6"}

Figure 15: IoT demo - blocks information

5 PERFORMANCE BENCHMARKING

Once the entire setup of integrating edge devices with a blockchain network we turned to perform thorough performance benchmarking. The intention was to understand whether this is a feasible approach from a business point of view. This work was meant to identify performance characteristics, thresholds, and limitations of running Fabric clients on edge devices. Thus, the intention was not to evaluate the performance of the blockchain network itself, but rather concentrate on the edge devices as clients aspects of the entire spectrum.

The main aspect we set out to understand is the expected sustained throughput of a blockchain client on an edge device. The corresponding measurement was the number of transactions per second supported by the client. The second interesting measure related to the consumption of resources of the edge device while performing the experiments. We mainly focused on the usage level of CPU, memory, and network. Last, we measured the latency of transactions initiated from the edge, and diagnosed the complete breakdown of the overall latency in different stages of a blockchain transaction.

During the experiments we varied the amount of transaction invocations per client, and in addition we varied the number of clients participating concurrently. In addition the exact workload distribution was varied as well, in terms of the read-write ratio in the transactions. We created a synthetic workload via a configurable simulation generation application, that simulates the effects of an IoT devices reading being recorded on the blockchain. The program itself simulated updates on a key – value store with a wide variety of keys.

The generic set up we have deployed to execute the performance benchmarking is depicted in Figure 16. An instance of Apache JMeter[13], serving as the workload generator, was installed on a local VM. JMeter would produce workloads based on configuration files. The JMeter instance made use of an exposed REST interface of an application residing on a RPi to indicate when transactions need to be invoked, and indicate the appropriate content of the transactions,

The application running on the edge device interpreted the received calls from the JMeter instance and proceeded to invoke corresponding blockchain transactions. The Fabric client application registered to obtain call-back events from the blockchain network, to complete the measurements of a transactions latency breakdown.

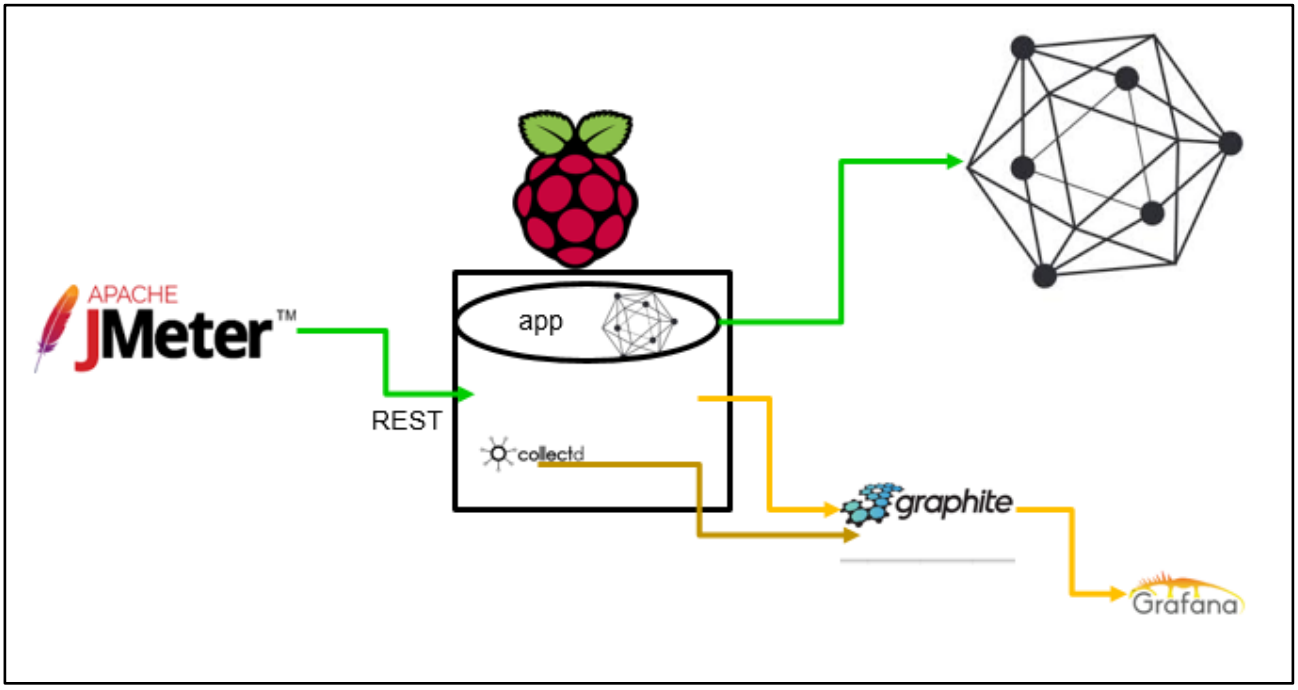


Figure 16: benchmarking setup

[13] https://jmeter.apache.org/

Within the edge device we installed an instance of collected[14], which is in charge of monitoring the level of resource consumption on the RPi. We used collected to monitor and report on the level of consumption of CPU, memory, and bandwidth while executing the experiments.

All recorded metrics are directed to an instance of graphite[15] serving as the monitoring hub. An instance of Grafana[16] in turn was used to grab the data from graphite and display it in easy to grasp graphs, easing the task of analysing the results of numerous test runs.

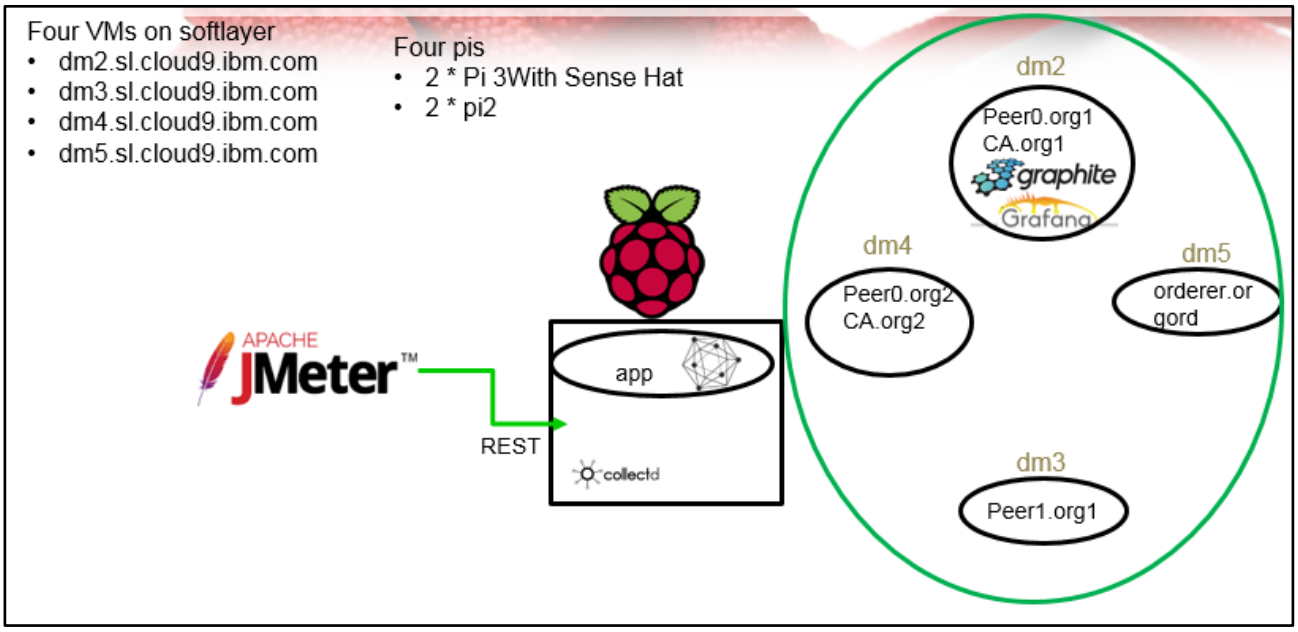


Figure 17: Benchmarking deployment

Figure 17 depicts the physical deployment of the benchmarking network. For the sake of the experiments we made use of 4 VMs deployed on IBM's cloud, on which all the blockchain components were deployed along with graphite and Grafana. Four RPis were used as the blockchain clients. Finally, a local VM hosted an instance of JMeter used as the benchmarking driver.

5.1 Benchmark matrix

The first set of tests were run on a single RPi and repeated the exact same scenario and setup on a single VM. In order to establish a baseline, the first test consisted of determining the maximum throughput sustained by a single core. In all tests we kept track of performance (transactions per second), resources consumption (CPU, memory, network bandwidth), and latency, keeping track of the end-to-end lifetime of a transaction.

Once a baseline was established following tests started performing several variations. The first test including varying the number of cores used by the test application. The baseline was the use of a single core, and consecutive tests added an additional core to each run until all the cores (four in the case of the RPi) were in use. In addition tests were configured to use thread affinity and varying the number and identity of threads participating in the tests.

The following variation on top of the configurations above was to vary the number of endorsers each transaction requires in order to be admitted as a valid transaction to the blockchain. This variation was added after analysing resources consumption and latency breakdown of the previous tests.

[14] https://collectd.org/
[15] https://graphiteapp.org/
[16] https://grafana.com/

The next set of tests started varying the number of devices (pis and VMs) executing the tests simultaneously. This test was designed to verify that there are no interdependencies among test running individually, and provide as indication as to possible scale implications on a fabric network.

Following set of tests concentrated on the possible effects of specific configurations of a blockchain network on performance indications. First the maximum block creation time was varied. This parameter determines the maximum time elapsed between the creation of two consecutive blocks by the ordering service. A second configuration parameter variation examined was the maximum number of transactions per block. This parameter also influences the rate in which blocks will be created.

5.2 Benchmark findings

The significant findings (see Table 3) are that a RPi 3, running a single process can sustain 11 transactions per second[17]. As the number of cores used increases, throughput is almost multiplied by the number of cores (measured 30 transactions per second using 3 cores on a RPi), but there is a need to leave one core to handle network interrupts.

It was further found that throughput decreases as the number of endorsers increases. It can be seen in the resources consumption portion, when focusing on the CPU, that the larger the number of endorsers per transaction, the larger the CPU consumption is. The main reason for that is that for each endorsement received from a peer, the client needs to invoke cryptographic function that are CPU bound. These functions are required to verify the authenticity and identity of the peer and its response. It was measured that the time required for the endorsement response verification phase on a RPi is equal to 0.05 times the number of endorsers. An additional contention point is the network handling of interrupts that led to the recommendation to have a dedicated thread for this job.

Table 3: Tests findings

Test	Result / explanation
Execute test on a single core	Throughput: 11 transactions per second as a baseline
Execute test on a multiple cores	Almost reach throughput of a single core multiplied by number of cores
Vary number of endorsers	Big degradation as the number of endorsers increases
Run tests on multiple RPis simultaneously	Reach aggregated throughput of all individual devices
Thread affinity	Best performance when leaving a system thread to perform network related activities
Vary maximum block creation time	No apparent difference
Vary maximum number of transactions per block	Observe a decrease in latency (blocks closing faster) with smaller numbers

Table 4: performance - RPi

pi	Single process	Multi-process
1 endorser	Throughput: 11 tps Latency: ~2 seconds	Throughput: 3*10 tps Latency: ~1 sec
2 endorsers	Throughput: 7 tps Latency: ~2.5 sec	Throughput: 3*6 tps Latency: ~2 sec

[17] Note that all measurements were performed on an early version of fabric release 1.1. Since than some work has been performed in the fabric core in areas related to performance.

3 endorsers	Throughput: 4 tps	

Table 5: Performance – VM

VM	Single process	Multi-process
1 endorser	Throughput: 65 tps Latency: ~0.5 seconds	Throughput: 3*50 tps Latency: ~0.4 seconds
2 endorsers	Throughput: ~45 Latency: ~0.7 sec	Throughput: ~3*35 Latency: ~0.5

Latency at these rates is governed by block closing parameters mentioned above as they are the dominant factor. In the latency breakdown shown in Figure 18 we can see that the only stage which takes place locally on the client, and causes the CPU consumption patterns seen on the same figure, is the endorsement responses verification. All the other stages represent remote calls which invoke operation on the fabric network itself, thus most of the time is spent on the operations to be completed on the blockchain side and for the actual overhead of passing information back and forth between the client and the remote entities.

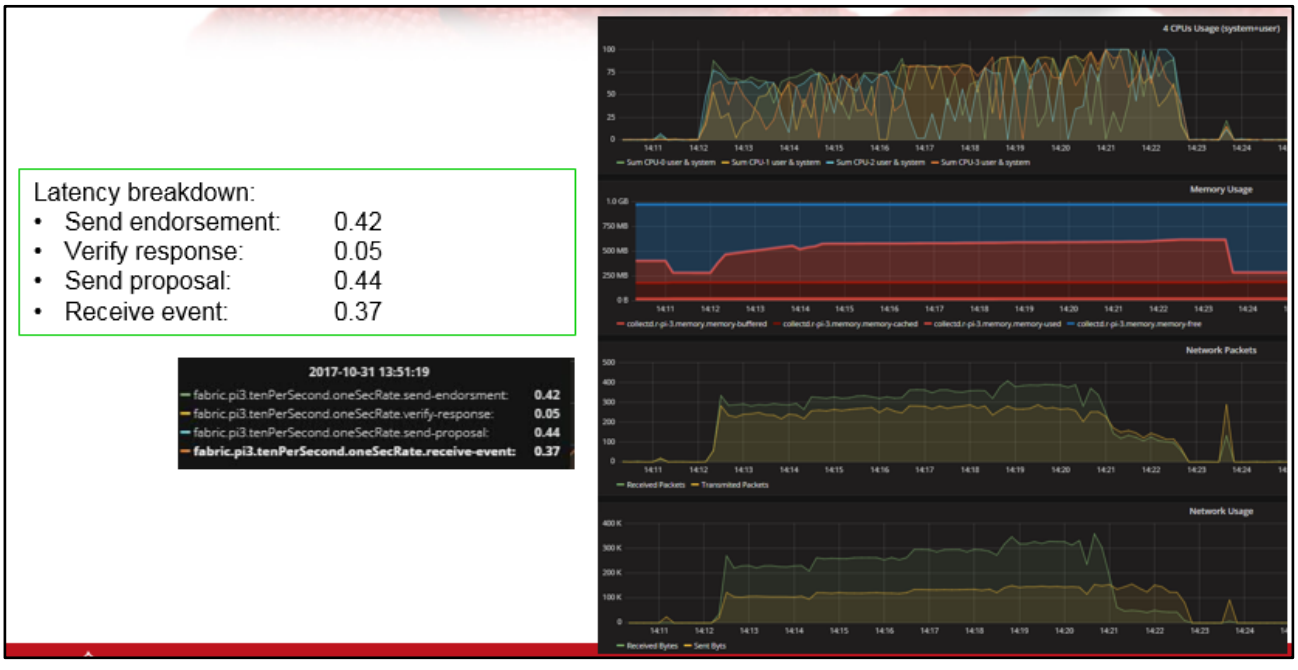


Figure 18: Performance summary

In Figure 19 we can further see details of the manner in which the latency breakdown is determined. The breakdown follows naturally the different stages of a transaction lifecycle throughout the network. The first stage is for the proposed transaction to be sent for endorsement to the relevant peers. Once responses are back they need to be processed at the client side and if approved the proposed transaction along with the endorsement information is packaged and sent to the ordering service. Next, a response is received from the ordering service, as an indication that the request has been

received by the service, not an indication that it has been included in a block. Finally, the client registers to the blockchain network event hub to be notified when the transaction in question has been committed by a peer.

The latency breakdown is calculated based on the flow described above. Time is recorded at each one of these stages, and time differences between two consecutive stages are reported.

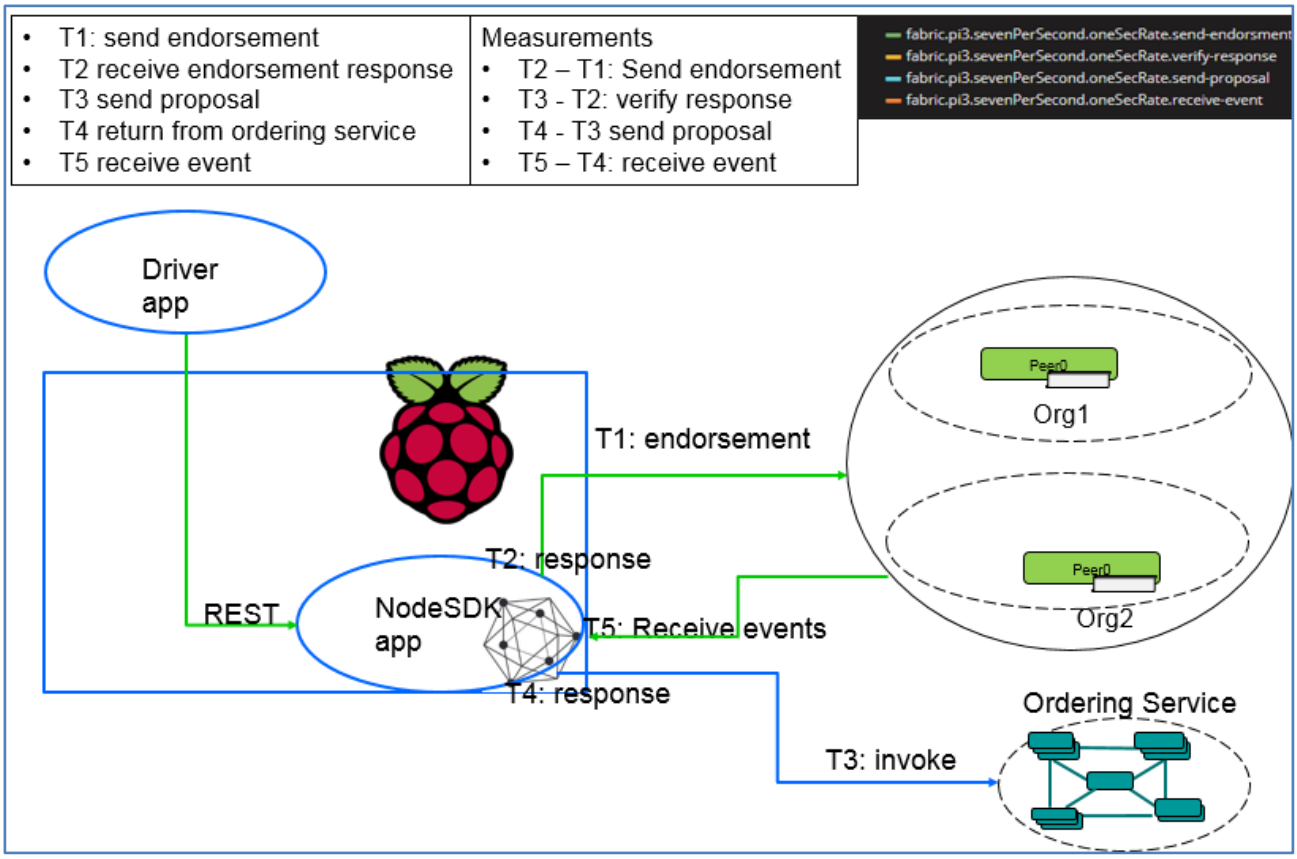
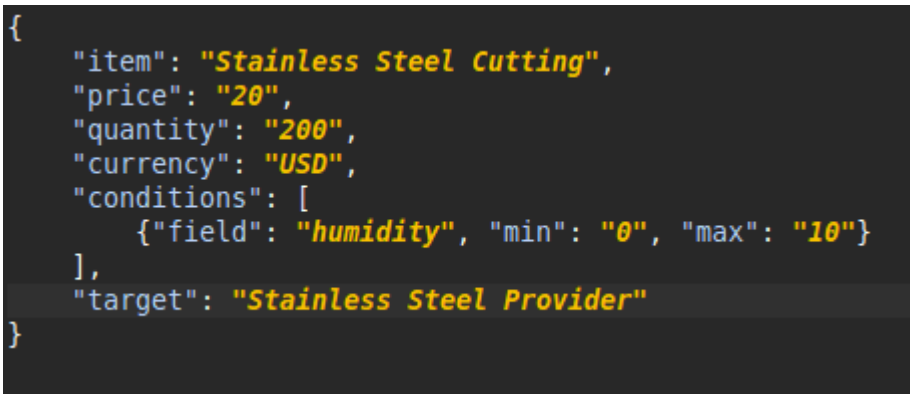


Figure 19: Latency Measurements

6 NEGOTIATION SUPPORT – RFQ MANAGEMENT

One of the example supply chain constructs that can be supported by a blockchain backbone is the negotiation process. We have created a simple template which demonstrates the utility of the blockchain as the underlying negotiations back-end. In this case the blockchain serves both as the single source of trust and truth, and as the driver of the mechanism used to communicate between different parties. Naturally, variations of this negotiation process can easily be constructed, based on the foundations we detail here, supporting both structured and less structured manners of interaction between the counter-parts.

The proposed and demonstrated negotiation process contains three primitives, namely: offer, counter-offer, and accept or reject. In the background entities which are interested in participating in such negotiations process register themselves as interested in receiving events on negotiations related transactions performed on the blockchain.



```
{
    "item": "Stainless Steel Cutting",
    "price": "20",
    "quantity": "200",
    "currency": "USD",
    "conditions": [
        {"field": "humidity", "min": "0", "max": "10"}
    ],
    "target": "Stainless Steel Provider"
}
```


Figure 20: Negotiation Offer

The process is initiated by an entity producing an offer and submitting it to the blockchain as a transaction invoking the "offer" function of a negotiation smart contract (please refer to Figure 20 for an example). In the offer structure conveyed we can see that we have both information in the items requested (such as item name and price), and in addition there's an optional "conditions" section in which characteristics governing the transport of goods are detailed. In the running example we use in this section the item the first entity is interested to buy is a capacity of a service for stainless steel cutting, and the conditions dictate the humidity in which the produced artefact should be kept throughout the process.

```
{
    "id": "875a22a4-fe1f-425b-af80-3b481225e9bc",
    "changes": {
        "fields": [
            { "field": "price", "newValue": "30" }
        ],
        "conditions": [

        ],
        "removedConditions": [

        ]
    },
    "note": "this price is low for me. how about this deal?"
}
```


Figure 21: Negotiation counter offer

In Figure 21 we see an example of a counter-offer put forward by the prospective seller. In a counter-offer structure each party can propose changes both to the general entries section and to the accompanying conditions. In this example we

see that the conditions were accepted as-is, but there is a request to change the price. The blockchain based process governing the counter-offers can contain as many iterations as both sides require and is culminated by the invocation of an accept or reject transaction. Once again, the counter-offer is submitted to the blockchain as a transaction, is recorded and committed into the ledger, and both parties to the negotiation process are notified as to the existence of a new counter-offer by using the blockchain callback events mechanism.

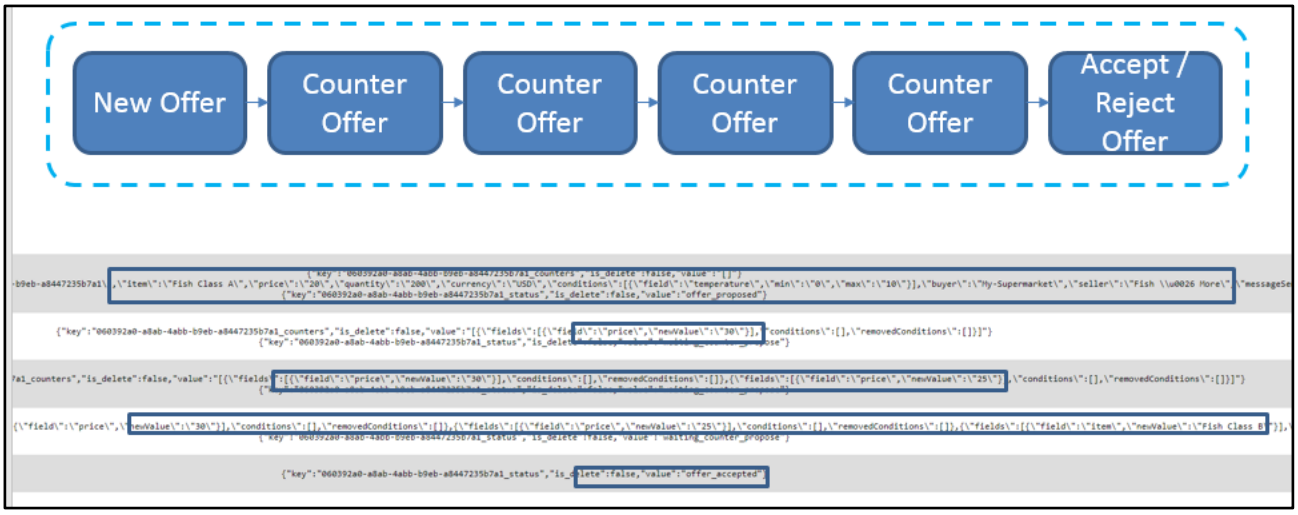


Figure 22: Blockchain based negotiation

In Figure 22 we can see the benefits of using a blockchain network at the back-end of a negotiation process. Being a shared ledger, all the steps of the negotiation process are clearly stored and can be made available to the parties involved. The information shown in the figure is taken directly from the transactions that made it to the blockchain, and due to the technical characteristics of the blockchain, this ledger can serve as the single source of truth which both parties can access and act upon.

7 HANDLING FLAKY NETWORKS - DESIGN

One of the prominent architectural challenges that need to be overcome when dealing with IoT devices is the unstable and sometimes unreliable network connectivity. Network bandwidth may be available at varying conditions, and network access may be lost for some periods of time, triggering connections, disconnections, and reconnections. The main method we chose to deal with this challenge is messages buffering, supporting a store and forward mode of operation.

To support correctness of operations under flaky network conditions we first need to analyse current flow through the system, identify the points which involve connectivity to the outside world, and design corresponding actions at these points. After performing the analysis of a flow of transactions from a client on an IoT device to a Fabric network the findings point to two main points in the flow which should be treated, first the point in which a transaction proposal is sent to the endorsing peers, and second when an endorsed transaction is sent to the ordering service. At the endorsement phase we can retry the request once connectivity is restored, and filter out outdated calls for endorsement. At the transaction invocation phase, we can replay transactions, counting on the Multiversion concurrency control[18] (MVCC[19]) mechanism of Fabric to ensure that only one of the duplicate transactions makes it to the blockchain as a valid transaction.

7.1 Standard transaction flow

A client SDK upon receiving a transaction execution request proceeds in two phases.

1. A corresponding transaction proposal is sent for endorsement to the relevant peers.
2. If the endorsement policy has been abided by the transaction is sent to the ordering service to be ordered and made available to the peers to be validated and committed.

To that end, the client SDK uses 2 modes of communication:

1. Stateless http (request / response) with the peers for sending transaction proposal for endorsement and with the ordering service (one or more orderers) for submitting a transaction.
   The corresponding “send_proposal” and “send_transaction” calls use this mechanism, thus when a network error occurs, the call will timeout, and an error will be reported to the application. Note that these methods may have a “retry” value which will cause this call to be attempted a “retry” number of times before declaring an error.
2. Persistent http connections with the peer serving as the event source for receiving events back from the Fabric network. Upon a network failure, the connection is broken, events may be lost, and registered listeners may not be triggered. The SDK should make attempts to re-establish the connection on behalf of the application. But if after “re-try” number of attempts the connection cannot be restored, it should notify the application of this condition with a reasonably high severity error.

7.2 Message buffering for dealing with flaky networks on devices

We need to distinguish between (network induced) failures at the endorsement and at the transaction submission phases. For endorsements, we can retry the request, filtering out outdated calls for endorsement. Extra care should be taken such that an incoming transaction execution request from the application does not get two different endorsements resulting in two different transactions being sent to be ordered, since at that case both transaction executions may succeed. For transaction invocation, we can replay transactions, counting on the built-in validation (MVCC) mechanism to ensure correctness (of course we want to minimize the waste of resources).

[18] https://en.wikipedia.org/wiki/Multiversion_concurrency_control
[19] https://hyperledger-fabric.readthedocs.io/en/release-1.3/arch-deep-dive.html

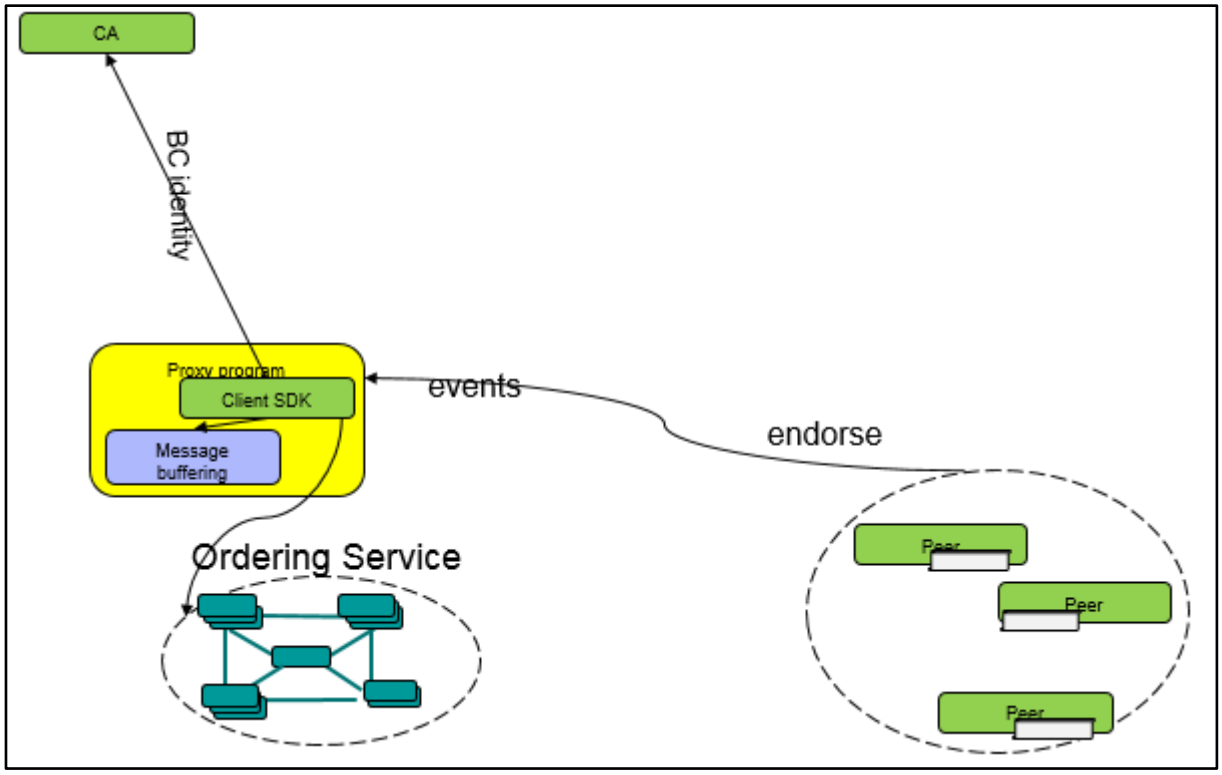


Figure 23: Dealing with flaky networks

The flow proposed is as follows (refer to Figure 23 for a view of participating entities): Upon reception of a transaction execution request by the client SDK it is inserted into the persistent messages buffer with a status of "pending endorsement".

At the first stage, the transaction request is sent to the relevant peers to be endorsed using the send_proposal() method. If the transaction endorsement response is negative, the corresponding entry is removed from the messages buffer and a corresponding response is conveyed to the requesting application. If an error due to timeout is received we leave the entry in the buffer and mark it as being in the "endorsement error" state (we may add an indication of how many times this has occurred). If, on the other hand the proposed transaction was endorsed, the endorsement information is added to the same entry in the message buffer (practically a copy of the transaction to be sent), and the state is updated to "pending ordering".

At this stage, the transaction is sent to the ordering service using the send_transaction() method. This method sends the transaction to the ordering service for ordering and later to be validated and committed by the peers. Note that this is an asynchronous call and the result of the transaction commitment will be made available via the callback events mechanism. The event "transaction submitted" is received when the transaction has been accepted by the ordering service, later the event "transaction complete" is received when the transaction has been successfully committed by the peer, and "error" will be reported if the call has timed out due to networking problems.

At this stage, receiving an error will lead to a change of state to "ordering error", while upon receiving a "transaction submitted" event the state will be switched to "pending commit". Note, that potentially we can unify the pending ordering and pending commit states, since in both cases a possible future remedy would be the same, namely resubmitting the transaction to the ordering service (we still keep it apart since it may help identify the cause of the failure, an error before the transaction submitted event indicates that the connection with the ordering service is experiencing problems, while an error before the transaction complete event indicates that the connection handling the events is experiencing problems).Finally, upon receiving a "transaction complete" event we can remove the entry from the message buffer, and upon receiving an error at this stage we will move to a "commit error" state.

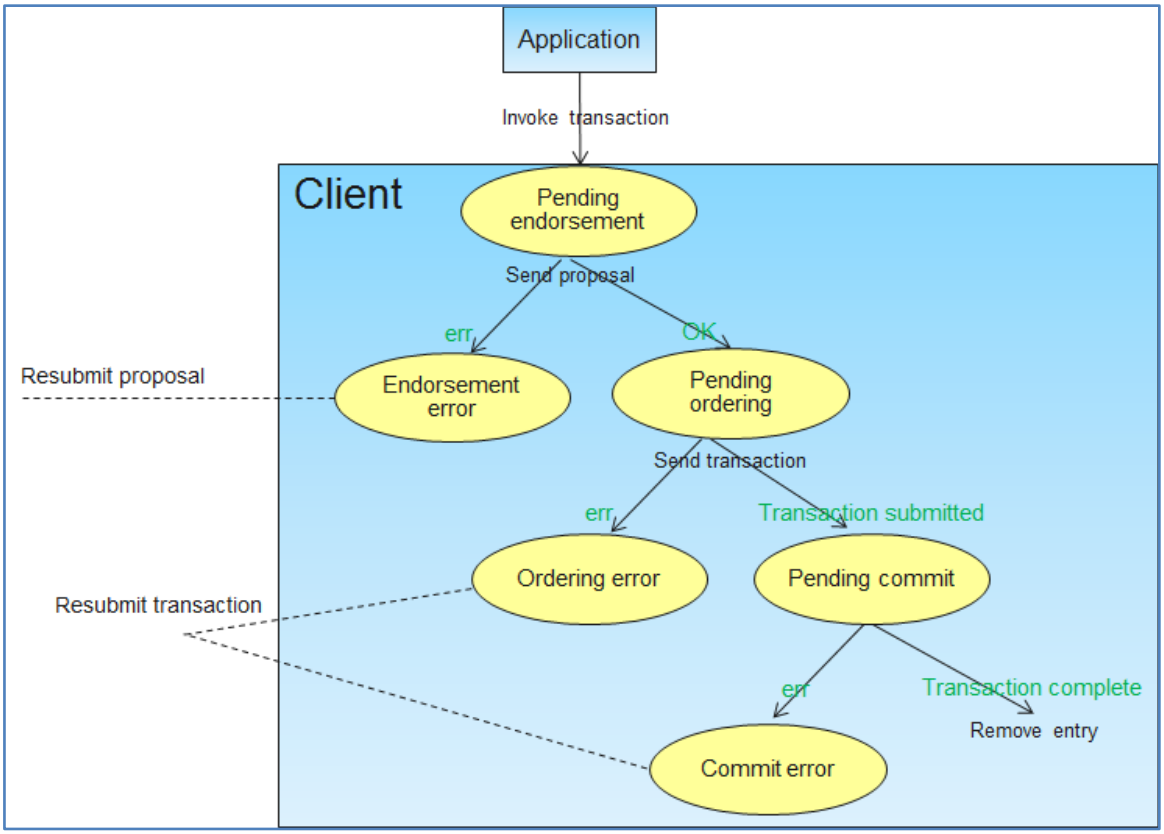


Figure 24: States in the messages buffer

7.3 Recovery actions

Recovery due to re-establishing network (or client coming back up after crashing – if decided that this is interesting, and will cause all updates to be persisted to disk). This recovery process can be initiated periodically (with exponential back-off), or based on an indication received.

This process will go through all entries in one of the “error” states and will resubmit the corresponding transaction to the corresponding phase, namely endorsement or ordering and commit. An entry in the messages buffer marked as being in the state “endorsement error” will be sent to endorsement, and its state will be changes to “pending endorsement”. Entries in one of the states “ordering error” or “commit error” will be sent again to the ordering service to be ordered and committed and its state will be updated to “pending ordering”.

In the case of recovering from a client failure we need to process all entries in the messages buffer and not only the ones marked in one of the “error” states. In addition to the description above, entries marked with “pending endorsement” will be sent again for endorsement, while entries in one of the states “pending ordering” or “pending commit” will be resent to the ordering service.

8 CONCLUSIONS AND NEXT STEPS

The main goal of this deliverable is to introduce the blockchain platform which shall be used as a cornerstone for the supply chain related activities within the MANU-SQUARE project. The bare technology itself is presented, along with relevance to the project and representative general and specific scenarios that can be supported, thus benefiting from the advantages the technology provides to users in a business setting.

A cornerstone of taking advantage of blockchain in supply chain relies on the possibility to have IoT devices as first class citizens in a blockchain network. This deliverable describes the work performed to successfully integrate IoT devices in a blockchain setting. Thorough performance benchmarking is presented as well, demonstrating the feasibility of the approach.

An exemplary use of blockchain to support a negotiation process is described as well. Putting it all together an end-to-end demonstrator has been developed, described and shown, from negotiation to the resulting monitoring (Tracking and Tracing) of the supply chain in action.

8.1 Next steps

Example support for tenders in the platform via the blockchain shall be demonstrated, based on existing artefacts providing support for negotiation aided by a blockchain network.

The outcomes of T3.1, which are described in this deliverable, shall be further elaborated in T3.2 and T3.3. In T3.2 a focus shall be placed on flexible security and privacy constructs, enabling various modes of interaction among partners in a blockchain network. A part of that effort shall include the use existing public / private key pair on the device for enrolment purposes, rather than obtaining these pieces of information from the CA. Such a capability shall strengthen trust and enhance individual device identity. It opens the door for potentially using information recorded on the device HW at manufacturing time to serve as the unique identifier of the device throughout its entire lifecycle. Moreover, differential views of information residing in the blockchain respecting different privacy policies shall be demonstrated.

T3.3 shall develop further supply chain constructs to be used in the project, with a goal to be able to present a dashboard including assets and associated state from creation to destruction, abiding by the privacy policies established on different parts of the network. Ultimately, developed services shall be incorporated as a part of the MANU-SQUARE platform, providing capabilities and support to the identified use cases within the project.