New ICT infrastructure and reference architecture to support Operations in future PI Logistics NETworks

D2.9 Blockchain Transactional Ledgers and Smart Contracts as PI Enablers

Document Summary Information

Grant Agreement No	769119	**Acronym**	ICONET
Full Title	New **IC**T infrastructure and reference architecture to support **O**perations in future PI Logistics **NET**works		
Start Date	01/09/2018	**Duration**	30 months
Project URL	https://www.iconetproject.eu/		
Deliverable	D2.9 Blockchain Transactional Ledgers and Smart Contracts as PI Enablers		
Work Package	WP2		
Contractual due date	31/07/2019	**Actual submission date**	24/07/2019
Nature	Other	**Dissemination Level**	PU
Lead Beneficiary	ILS Group		
Responsible Author	Alexander Papageorgiou (ILS Group)		
Contributions from	CNIT, IBM, NGS, VLTN, eBOS		

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No 769119.

Disclaimer

Copyright message

Table of Contents

List of Figures

Glossary of terms and abbreviations used

Abbreviation / Term	Description
ABCI	Application BlockChain Interface – An interface which enables programmers to code blockchain transactional logic in any programming language.
API	Application Programming Interface – A set of clearly defining communication methods for interacting with an application in an intuitive way.
BFT	Byzantine Fault Tolerance – A trait via which a technology, usually operating via a network of nodes, is able to operate consistently even if a significant portion of it is compromised.
Blockchain	A blockchain is a particular type of DLT that establishes an immutable ledger of state alterations that are secured via cryptographic and algorithmic methods.
DAG	Directed Acyclic Graph – A type of graph that is finite, possesses no circular references and has a direction.
DD	Decimal Degrees – A mathematical system of representing latitude and longitude coordinates as decimal fractions used by most positional systems to date such as GPS.
DLT	DLT, which stands for Distributed Ledger Technology, is a type of technology that allows one to retain a database synchronous and geographically distributed among an array of devices.
DoA	Description of Action
GPS	Global Positioning System
Hyperledger Fabric	Hyperledger Fabric is a project under the Hyperledger umbrella project that is spearheaded by IBM and has developed an enterprise-grade blockchain targeted towards business networks.
ID	Identifier
JSON	JavaScript Object Notation – A representational format of structured data that is commonly used in web technologies.
KPI	Key Performance Indicator
LADL	Logic As a Distributive Law – A process via which spatial-behavioural type systems can be created.
LL	Living Lab
OLI	Open Logistics Interconnection model – A model derived from the Open Systems Interconnection (OSI) model that can be applied to PI and logistics in general.
PI	Physical Internet
PI Packet	A PI Packet refers to a package in the PI network that is being transported from one place to another as part of a commercial transaction.
PoC	Proof-of-Concept

REST	Representational State Transfer – A software architectural style usually based on to create web services and APIs.
Ricardian Contracts	A document that describes a software process in legal terms that is legally-binding if agreed upon and fully enforceable in court.
RPC	Remote Procedure Call – Another type of software architectural style that is usually used in legacy and purpose-built systems offering a single uniform endpoint for interactions.
SLA	SLA stands for Service Level Agreement, a type of agreement between two parties that specifies certain conditions that are to be met by the purveyor of a service. If these conditions are broken, the recipient of the service can in most cases be reimbursed in one form or another.
Smart Contracts	Smart Contracts are functional blocks of code that are executed in a fully deterministic manner, meaning that for each combination of inputs a single output corresponds. This allows one to store the inputs and outputs of a smart contract in a database immutably.
SPoF	Single Point of Failure
SSE	Server-Side Events – A web technology for uni-directional communication from a server to a client.
Tendermint Core	Tendermint Core is a distributed state-replication machine that allows programmers to code advanced logic on it that take advantage of this state-replication to build a blockchain.
Transactional Ledger	A database that stores all types of state alterations of another database, sometimes even of itself.
WP	Work Package
WS	WebSockets – A web technology for real-time bidirectional communication between two parties.

1 Executive Summary

The objective of this deliverable was to identify PI use cases, derived from the ICONET Living Labs, that blockchain could be applied on and describe the reasoning behind the research and programmatic steps taken during the solution's development. These programmatic steps are meant to describe the formation of smart contracts that also interact with PI-specific components such as "PI routing" and "PI packets". All objectives of the first version of this report have been fulfilled, with the forthcoming versions to expand upon the findings listed within.

Within this report we present the analysis and comparison of selected candidates from the state-of-the-art blockchain technological landscape with the purpose of defining the prime candidate for ICONET and the PI as a whole. The results of the analysis point towards the application of a purpose-built blockchain created upon the Tendermint consensus mechanism instead of a readily-available enterprise grade blockchain solution, such as the one referenced heavily during this project's conception, Hyperledger Fabric. The rationale behind this choice is that the Tendermint consensus engine provides a lot of freedom in the way the blockchain operates down to the block acceptance protocol, enabling us to create faster, more efficient mechanisms for the PI in comparison to the unnecessary computational overhead that exists in Hyperledger Fabric.

The potential application of blockchain technology on multiple use cases is researched with the goal of focusing on a single potential application within the report while simultaneously examining the available options. After correlating the various use cases identified, it became apparent that the dynamic allocation of space in PI Hubs for PI Packets was the most interesting and relevant use case to expand upon. A notable output of this section is that the PI, as it is slowly but steadily evolving, already has and will open up more areas of interest for blockchain technology, as it is a heavily misunderstood technology, with actual benefits provided they are exploited in a correct manner.

To fully capitalize on the benefits of blockchain technology, the automatic generation of Ricardian contracts will be inspected within the report. This inspection will be on both a theoretical level, validating the assumption that it is legally possible to generate enforceable Ricardian contracts in a mechanical manner, and a practical level, showcasing the programmatic automation of a traditional PI contract translated to the vocabulary of a Ricardian contract. This point is to be further investigated.

Finally, the avail of blockchain as a traceable and immutable record of all actions occurring with regards to the Ricardian contracts and the relevant PI components will be exploited for resolving disputes. The resulting blockchain-aided mechanism for dispute resolution will be realized in theory and in practice. This point is to be further investigated.

The related effort for the above is outcome of the ICONET WP2, Task 2.4 "Blockchain mechanisms for secure and privacy-preserving distributed transactional ledgers", the subtasks concerning the realization of smart contracts & the blockchain specifications, and the related versioned deliverables 2.9 (v1), 2.10 (v2) and 2.11 (v3).

2 Introduction

The focus of this deliverable is to define how a PI-oriented blockchain solution could be appropriately applied to a specific use case in the PI realm while breaking down in detail the abstract steps taken in properly establishing what needs the blockchain is set to solve before developing the actual reference implementation of the said steps. This deliverable serves as the outcome of WP2 and particularly task 2.4 (Blockchain mechanisms for secure and privacy-preserving distributed transactional ledgers) led by the INLECOM Group (ILS Group). In this WP, as the report indicates, the viability of blockchain technology as a PI enhancement is investigated under the scope of a specific use case, with tangible outputs showcasing the conclusions of the aforementioned investigation.

The PI domain is moderately but firmly progressing along with the domain of blockchain and distributed ledgers in general, raising questions as to how the outputs of this deliverable will be relevant in the coming years. To tackle this obstacle, the fundamental theoretical principles that have been applied to the brainstorming process of this deliverable attempt to distance themselves from restrictive named components and instead focus on the traits of blockchain technology, prevalent issues of the PI domain and the intersections between them leading to a pertinent theoretical analysis and appreciable programmatical implementation.

This deliverable at its core tries to formalize the following key statements:

- The idealistic blockchain solution for the PI.
- The promising use cases within PI where blockchain would be useful.
- The practical approach to applying blockchain technology to such a use case.

The Tendermint Blockchain Consensus Engine (https://www.tendermint.com/) has been chosen as the blockchain technology of choice for the reference implementation of this deliverable within ICONET as it possesses all the preferable traits of an ideal blockchain solution as discussed within this report.

2.1 Deliverable Overview and Report Structure

In accordance to the ICONET workplan, this report describes task 2.4 and all subtasks that pertain to it with impartiality and transparency in mind.

The primary purpose of this deliverable is to set the precedence for the following version 2 and version 3 blockchain ICONET deliverables by identifying the blockchain solution to use in the project, selecting the living lab use case the blockchain prototype should be geared towards and investigating how the blockchain would be applicable to the other living lab use cases. The primary tangible output of this report is a software prototype that will initiate the blockchain instance and record some form of alpha interaction between a PI packet and the blockchain itself. Secondary outputs of this report include some analysis performed on what traits the optimal blockchain solution would possess to be considered a fruitful candidate for a PI application, what the current state of blockchain and PI looks like and what generic use cases seem to have potential in the PI space.

The report begins with detailing the research activities that were conducted in order to meticulously select the quintessential blockchain solution for the ICONET project's intentions. ICONET aims to create a technology stack that advances PI by providing additional capabilities to the network via software and hardware revolving around the optimization of PI routes and the interconnection of PI nodes. In this context, the potential use case scenarios for blockchain technology in the realm of PI were laid out to properly gauge the requirements that would arise if one were to implement them. Afterwards, the blockchain selection process commenced.

To properly select a use case to focus on, various academic resources were considered, consulted and linked in the report to pinpoint in which areas current and future PI technology lacks in and attempt to rectify these issues with a software implementation that utilizes blockchain at its core. While other use cases were identified, they did not prevail as blockchain technology would be superfluous if applicated to them.

During this process, several key performance indicators were taken into account as formed by the requirements of the ICONET proof-of-concept blockchain scenario such as the transactional throughput of the blockchain and its scalability. While the DoA specifically mentions the usage of Hyperledger Fabric, the findings of the analysis in this report deviate from it, as a superior blockchain implementation was identified, and will be used within the project. The rationale behind this development path is also detailed within.

Once the blockchain solution was finalized and agreed upon by the ICONET consortium, developmental steps were initiated to realize the concept that was formed by the preceding chapters. In parallel, further research activities were conducted to investigate the potential integration of SELIS outputs and specifically its publish-subscribe infrastructure, affecting the course of development as well as the blockchain's core logical dogmas. Test suites specific to this implementation were also produced as part of this report to properly illustrate how it performs under heavy load. To conclude, the work ahead that will be included in the upcoming version 2 and version 3 deliverables is elaborated to provide insight to the course of this Task's fulfillment.

Chapter 3 distinguishes and evaluates the potential use case scenarios where blockchain technology could intelligently be applied to to produce tangible value. The proper evaluation of these use case scenarios is critical to laying out the theoretical groundwork for the practical utilization of blockchain technology to take place. The results of this Chapter are exploited by the ensuing Chapter 4.

Chapter 4 focuses on examining the current technological state of the blockchain space to pursue and pinpoint the optimum blockchain solution for ICONET. The desirable capabilities and relevant blockchain traits as necessitated by the PI infrastructure and reference use case scenario are consumed to shape blockchain prerequisites that coincide with the reality of PI. These prerequisites are compared to the capabilities of state-of-the-art blockchain solutions, narrowing down the list of available solutions to a single and educated selection. The output of this Chapter was and will be exploited by all versions of this deliverable as it establishes the foundation upon which all developmental activities will take place.

Finally, Chapter 5 concentrates on the practical activities that transpired to devise the alpha-version of the PI scenario implementation. This Chapter examines the steps taken from both a software angle, providing snippets to actual code with accompanying descriptions, and an academic angle, showcasing the abstract programming principles that were taken into account. A test suite was also developed for this chapter to ensure that the code that was produced is practical, negligently error-prone and demonstratable.

The report concludes with three (3) Annexes that include supportive material for the ICONET blockchain implementation, Annex I (blockchain source code), Annex II (ICONET Pub-Sub API) and Annex III (ICONET API Test Suite).

Overall the approach taken in this report attempts to be as abstractive as possible, since more efficient blockchain solutions are likely to emerge in the coming years and the PI requirements and use case scenarios are fluid.

3 Potential Use Cases

While practical technological initiatives in the PI have yet to be applied, it has been extensively studied by the academic community[1,2]. The common consensus is that the market is at an infancy stage for robust PI solutions to materialize[3], however it is expected that by 2040 the market will be at a decent stage owing to the rapid advancement of IoT sensors and their constantly lowering material cost[4].

As a result, analysts have already began speculating what practical use case scenarios could be realized with the help of PI[5]. Within these use cases, certain technological requirements are laid out that are to be satisfied by any experimental attempt at implementing solutions for them. These requirements, if gauged correctly, can be fulfilled and accomplished via a better method, and that method is a blockchain. After inspecting the available academic resources, we have differentiated three potential use cases[6,7,8] where blockchain technology could efficiently be applied to.

3.1 PI Space Allocation

In a typical PI scenario, as Figure 1 below shows, packets are constantly flowing through the physical network of hubs that provide them with the corresponding accommodation facilities required to reach their destination. If we look at the current state of things, lack of modernized communication channels and redundant manual labor lead to a slow and ineffective system. This precludes any form of dynamic travel paths for PI packets and as such does not allow corporations to fully exploit the capabilities of the PI.

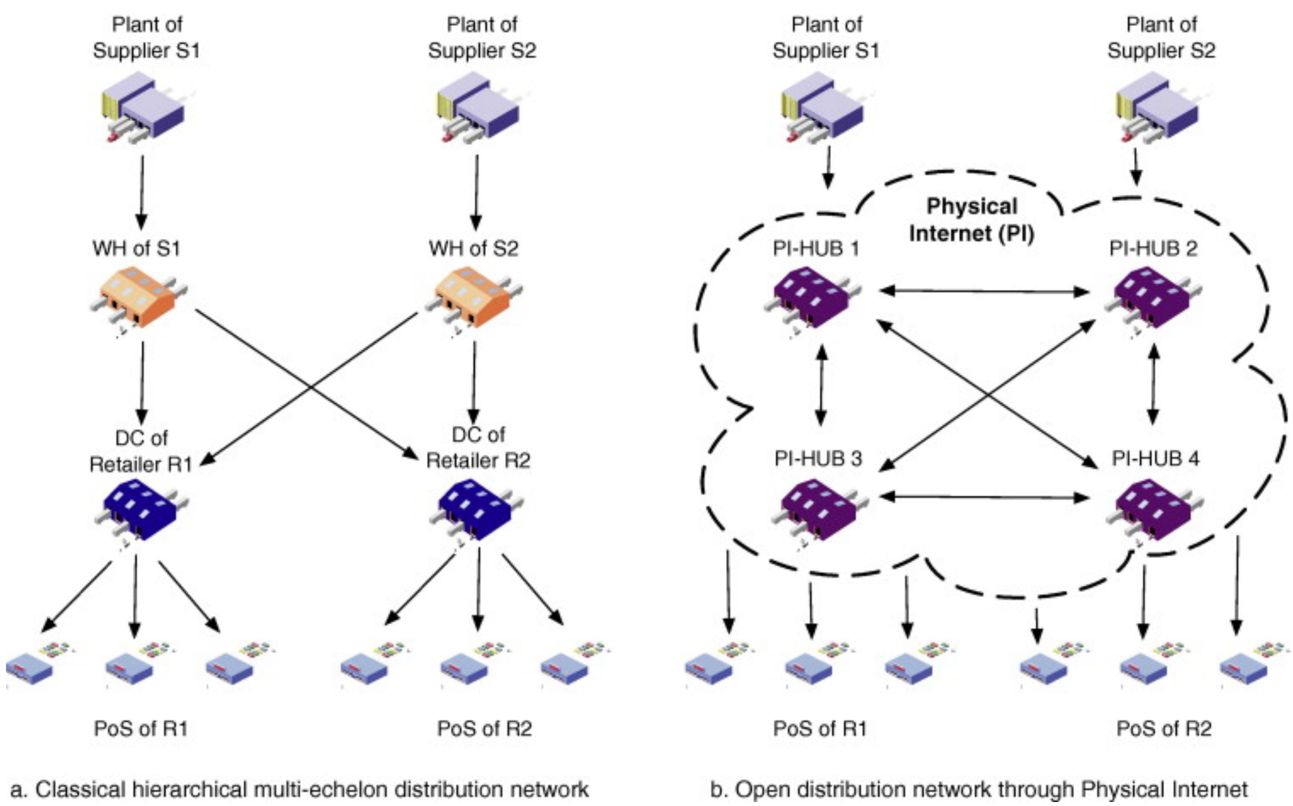

Figure 1 - Traditional Logistics Networks vs PI Networks[1]

If paths were to be arranged dynamically, PI hubs would benefit greatly as they would be able to conduct micro-leasings of space for small time periods and utilize their available space to its maximum potential. On the other end, transport operators in the network would also prosper as they would be offered competitive rates and

[1] Image copyright © 2014 Elsevier Ltd. All rights reserved, source: https://www.sciencedirect.com/science/article/abs/pii/S036083521400415X

faster lanes of transport, reducing order fulfillment times and increasing their operational throughput. The dynamic availability of PI hubs would also lead to fairer market rates, as PI hubs would need to dynamically adjust their offering rates based on the availability in their region and capacity.

The dynamic allocation of space would also have multiple aftereffects. It would provide incentives for PI Hubs to reinforce their infrastructure while simultaneously increasing transport activity uniformly in the geographical area the PI network is located. This increase in activity would lead the local economies to gain greater traction while also dispersing emissions from centralized roads to the distributed road networks of PI hubs. In addition to emissions being reduced, as travel paths will in most cases be shortened, they would also be spread over a larger geographical area, thus leaving a smaller fingerprint on the surrounding environment and ecosystem.

In the end, a dynamic system of allocation PI space based on machine learning applied to availability, transport paths and market rates can form the basis for the flourishment of PI and empower other initiatives to enhance or supersede it. However, for such a system to work, its security would need to be heavily fortified. The more value a technology brings into the space, the higher the yield of compromising it.

If the underlying technological infrastructure supporting the dynamic allocation of PI hubs were to be compromised, it would lead to disastrous side effects. As prime examples, PI packets could be misdirected to sabotage a transport agency's operation, market rates could be manipulated to fabricate economic instability, sensitive market data could be leaked, and inventory availabilities could be misrepresented to freeze transport lanes or activity for a specific PI hub. As the stakes of such a dynamic PI space allocation system are high, it makes sense to develop a secure-by-design solution that operates in a fault-tolerant manner to prevent centralized points of failure.

This is where blockchain comes into play. By nature, blockchain implements cryptographic principles for verifying the identities of the operators of the network as well as its users (see Figure 2). A blockchain is also distributed and Byzantine Fault Tolerant[9], nullifying any SPoF as compromisation of, in most times, over 50% of the network is required to control it. Even so, the network's state could be reverted to the latest healthy one in the unlikely event of a successful attack by re-initializing all network peers to the said state thanks to the immutable record history retained by blockchains. Furthermore, the efficient distributed communication protocol of blockchains enable them to operate in optimal capacity under heavy load even if the network is fragmented over a large geographical area.

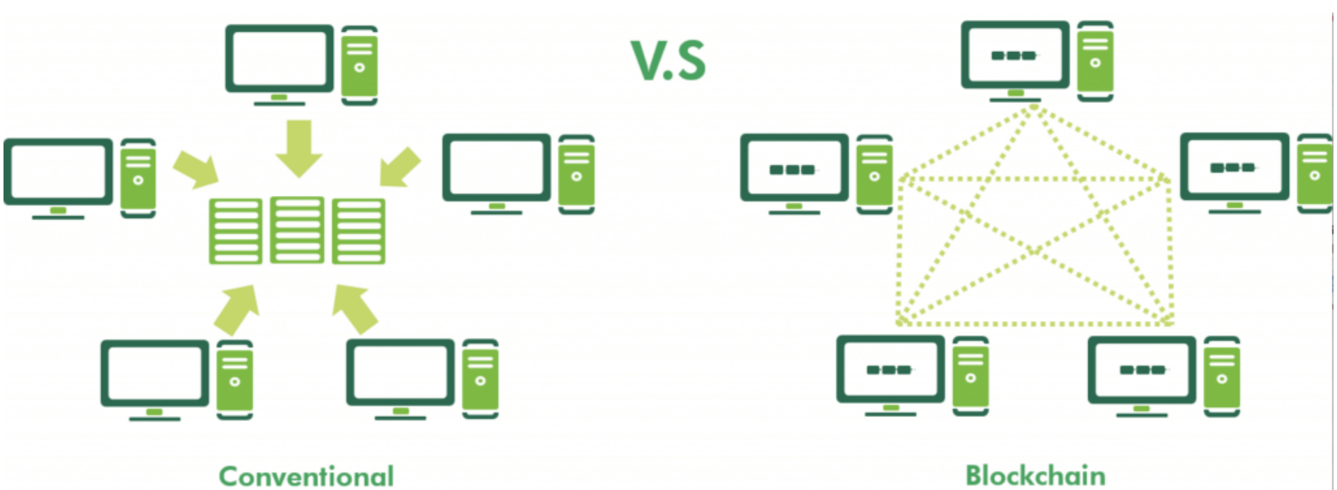

Figure 2 - Conventional Networks in contrast to Blockchain Networks

These traits are all desirable by a dynamic PI space allocation system and as such, it makes sense to implement such a system within a blockchain infrastructure. This particular scenario ties in with the second LL of ICONET, the Routing Optimization, as both this and the LL's use case pertain the usage of IoT & advanced software to optimize the shipment paths of PI packets travelling in the PI network.

3.2 Enforcement of SLAs

Another aspect of PI that could be beneficial for current operations is the autonomous enforcement of Service Level Agreements. SLAs basically establish a baseline as to what service a customer is expected to receive. In the current state of things, it is practically impossible to ensure SLAs are met in the transport industry as the infrastructure is not competently equipped to properly produce the necessary data to confirm these SLAs. PI assumes the availability of multiple sensors on all components of the network, thus allowing us to establish a theoretical methodology for applying SLAs in the transport industry.

The enforcement of SLAs with the help of IoT sensors would increase customer's satisfaction as they would be able to always see where their packages are currently located and lead to a healthier market as service providers would have to follow stricter standards. Operators are incentivized to adopt SLAs as it would lead to greater match of expectations for their customer base and customers would be more likely to select a service offering SLAs than one that does not, as indicated by the market's response to other services traditionally offered with SLAs such as server hosting[10].

In order to prohibit service providers from tampering with the data feeds that are acquired by the various IoT sensors on each component of the PI to verify the SLAs, the data feeds need to be securely generated, stored and validated. Once again, the multiple enticing traits of data integrity that blockchain technology possesses are fit for this purpose. Should the data feeds be stored in an immutable and distributed ledger, game theory would prevent singular components of the network from falsifying data as they would be shunned by the greater network and the irregularity would be detected in a swift manner.

This use case came to fruition based on the first LL, the Hub Network, as autonomous provision and enforcement of SLAs can shorten the time required to establish business collaborations between the various entities of the said network.

3.3 Market Rate Transparency

The final aspect of PI where blockchain technology could come into play is market rate transparency which will be analyzed in this report. The current archaic communication systems utilized by the various underdeveloped PI actors render it cumbersome to obtain a clear picture to the median rates offered by the service market. As such, logistics users tend to simply select the first service that satisfies their budget without looking for an economical solution.

This would change if a system that properly portrays the current rates, availability and general market state of the PI were to be developed. The timely fashion data from the market could be fetched at would indirectly force participants of the PI to offer competitive, informed and fair rates for their products. The transparency offered by such a platform would positively impact the greater PI ecosystem as it would lead to healthy competitiveness, rendering services more efficient and inexpensive.

A prerequisite for the above to be realized is a stable and secure platform for PI actors to upload their data to. As market rates essentially relate to sensitive data manipulation of which for even a miniscule timeframe can affect the income flow of a business greatly, it is reasonable for businesses to be hesitant in trusting an online and not on-premise platform. Such an issue can be resolved if a network between the businesses was established that respects the privacy rules of each business channel as set by the businesses themselves. This exact purpose is one blockchain can aptly fulfill.

If blockchain was used as the distributed database for storing market data such as the rates of each business and each business was given the option to either store their data on an on-premise blockchain node or a cloud-provisioned one, the network would become trustless, meaning that each participant would be responsible for his own share of market data. This would greatly enhance the security and stability of the overall system as a nefarious take-over of a business's blockchain node would only affect that business's rates while the rest of the system remains operational. In other words, businesses would achieve the same security they have today while making their data more accessible.

Most enterprise-grade blockchains, which are the ones considered for applications such as the one of ICONET, offer some system of privacy-control over the data that is transmitted throughout the network. This capability is especially useful in our context, as it enables businesses to label their data as they see fit and correctly provide access on a need-to-know basis rather than an allow-all basis, preserving the privacy and value of their data.

Market rate transparency is basically a requirement for any type of efficient dynamic situational routing to take place and as such, it became apparent that the third living lab could utilize blockchain as a transparent record for achieving this level of transparency.

3.4 Common Blockchain Requirements

The use cases analyzed in the preceding sub-chapters have been examined in detail and common key performance indicators of an ideal blockchain solution for them have been extracted. These KPIs have been consolidated in the following list:

- Adequate transactional throughput to satisfy the multi-faceted needs of PI.
- Highly scalable foundation, expanding in tandem with the lineally evolving PI infrastructure.
- Rule-based access controls to increase business confidence.
- Sufficient ratio of fault tolerance, providing a resilient network foundation.

These characteristics rule out any form of public blockchain implementation, such as Ethereum[17], as they are incapable of imposing rule-based access controls on participants of the network. The options that remain are enterprise-grade blockchains. These options will be discussed assiduously in Chapter 4.

3.5 Use Case Verdict

While all use cases that were mentioned have great commercial potential, it would be impractical and time-consuming to focus on all of them. Instead, it became apparent that one particular use case fits the ICONET narrative greatly in comparison to the rest, namely the PI space allocation scenario in conjunction with SLAs. Market rate transparency, while provably beneficial, is not a candidate ICONET should focus on as it mostly relates to economy-dynamics whereas ICONET is mostly a software and hardware solution that provides tangible benefits rather than solely long term benefits.

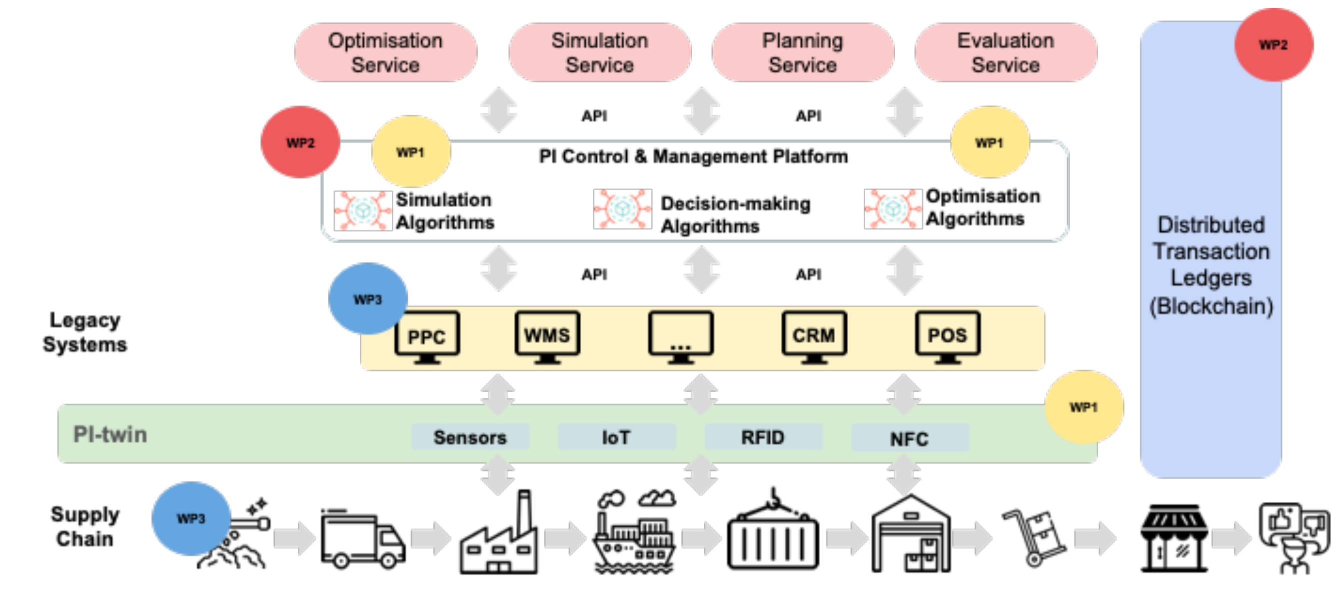

Figure 3 - Diagram from D2.1 that showcases blockchain integration in PI architecture

Creating blockchain-eccentric logic around dynamic PI space allocation would enable the blockchain component that will be developed to conceivably be integrated to the overall ICONET architecture as both a demonstrative example and a functional component of the ICONET system (see Figure 3 above).

The automatic enforcement of SLAs will be studied and applied to our blockchain implementation as a focal point. Regardless of the use case, the blockchain principles that will be applied and the KPIs that will be taken into consideration when choosing the correct blockchain implementation will remain the same across the use cases as all of them reflect the requirements of a cog in the grand PI system.

Within the purview of Task 2.4 we will precisely construe a blockchain-based system of recording the location of a PI packet flowing in the PI network along with metadata describing the physical conditions surrounding the packet, such as the temperature, to be able to establish and verify SLAs depending on the nature of the packet. Packet flows will be verified by multiple parties depending on the context, for instance when the delivery service drops the packet off to a PI hub and the PI hub confirms the receival of the package. Every actuation of the blockchain will re-affirm that the SLAs are upheld and, if broken, the relevant parties will be notified and the SLA severance will either be automatically resolved based on the data residing in the blockchain or require manual review.

3.6 ICONET Living Labs Association

Within the ICONET's project lifetime, four distinct living lab environments are to be realized to validate that the vision of ICONET provides tangible results and holds true in real use-case scenarios. These LLs represent the idealistic conditions under which the ICONET system of services is meant to distinguish itself. While the blockchain component of the ICONET system is somewhat experimental and independently ventured, it must conform to and be developed with these LL scenarios in mind to guarantee that the blockchain component stays within the scope of ICONET.

3.6.1 LL1: Hub Network

The first LL is that of the port of Antwerp, where the establishment of a PI network between the terminals and hubs of the port will be conducted and its aftereffects will be studied in relation to the optimizations such a network would yield. This particular LL exploits the high influx of shipments the port of Antwerp handles and opts to minimize the time required for collaboration processes to take place between the actors that reside in the port i.e. the terminals and hubs.

This case is quintessential for the blockchain component as the component itself is designed to eliminate human interaction from being required to procure and create the underlying contracts that support these collaboration processes. With the aid of the ICONET blockchain, it would be possible to enact certain pre-determined contractual templates in a fully automated way, shortening the time required for such contracts to be formed greatly. The resulting upturn in operational efficiency would assist the port of Antwerp in enlarging its serving capacity.

3.6.2 LL2: Routing Optimization

The second LL pertains the connectivity of IoT sensors located within containers flowing as PI packets in the PI network, and specifically in PI corridors, to PI services that exploit the accumulated data to provide insight into alternative and potentially optimum routings for the containers (see Figure 4). The goal of this LL is to highlight that it is possible to reduce the cost of PI transport, reduce its emissions and benefit both the transporter and the recipient in the process as current container routings are unoptimized and are based on a high-level path finding approach, leading to lengthier routes being chosen that increase the time a PI packet emits greenhouse gases and also incur an avoidable expense for the transporter and a delivery mark-up for the recipient.

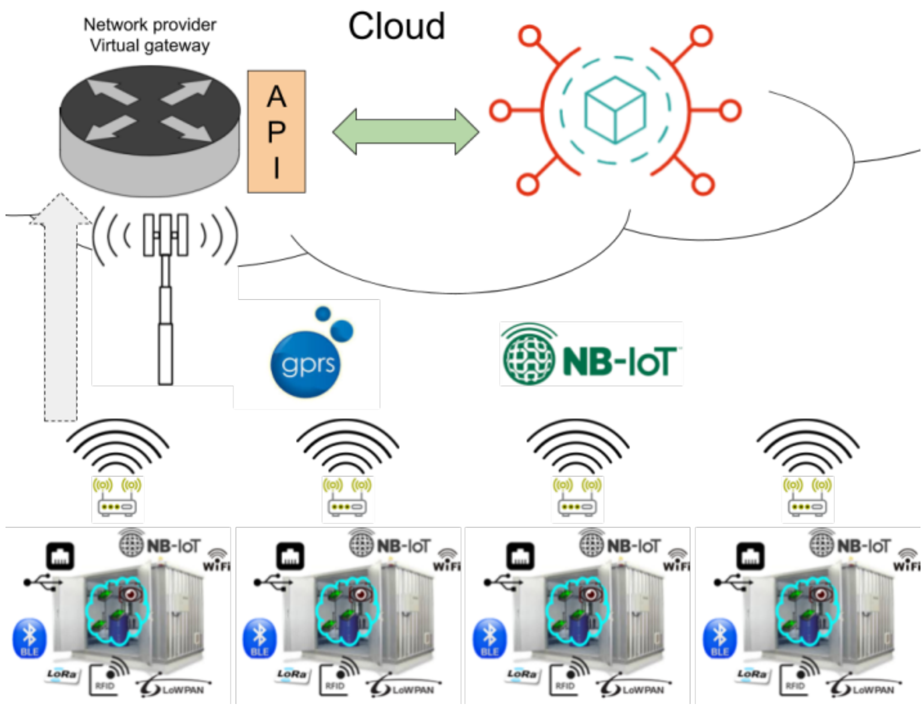

Figure 4 - Figure from D2.6 showcasing how LL2 will integrate with ICONET

Within this concept, the ICONET blockchain can fit into the provision of automated SLA guarantees for the parties involved. To elaborate in detail, an SLA can automatically be formed between the transporter and the recipient of a PI packet that dictates the packet to be transported at a certain range of temperatures, a specific container turbulence margin and a mutually-agreed transport timeframe. These conditions could in the future be consumed by the ICONET service that provides the alternative routings to avoid roads that may contain beaten installations that would result in heightened turbulence for the transporter’s vehicle and consequently for the container. Additionally, the ICONET blockchain would continuously gather data from the sensors located on the container and would automatically severe the SLA contract in case of a breach f.e. when the temperature rises above 20 degrees.

3.6.3 LL3: Dynamic Situational Routing

The third LL, while similar to the second, focuses on providing advanced mathematical algorithms that are able to deal with dynamic conditions surrounding the area of transport of a PI packet such as traffic congestions, accidents, road blockages, natural disasters and more. This type of contextual awareness is more geared towards eCommerce scenarios where the PI packets flow all the way towards the end-customer and would potentially deal with more complex situations and road-routes than highways, flight paths and other generally-stable route types.

Here, blockchain technology fits in as the SLA intermediary and allows even the end-customer to observe the PI packet's conformity to the SLAs that have been mutually agreed upon. The real-time aspect of the IoT sensor data feeds and the blockchain ledger's immutability both contribute to the authenticity and truthfulness of the SLA data reported by the blockchain service, thus offering peace of mind for the customer regardless of which actor of the PI network is involved in the PI packet's transport.

3.6.4 LL4: Warehouse Capacity Allocation

Finally, the fourth LL attempts to rectify the issues faced by the current pen-and-paper approach of PI warehouses to the allocation of storage units that inhibits the packet throughput of these warehouses and prohibits external parties from having an on-demand inspection of storage availability. This LL in itself is a difficult task to undertake as the warehousing industry has operated with analog human-operated processes since its conception and it would take radical changes for a digital alternative to emerge and be commercially viable. However, should this be achieved, the benefits would be immense as eliminating the human factor and introducing smart collaborative services between the warehouses which can enable them to dynamically allocate packets between them and optimize their storage spaces to a very large extent.

The ICONET blockchain can help with and potentially facilitate these smart collaborative processes by waiving their legal overhead through automatization. In an ideal future scenario, we could also permit these warehouses to operate their own ICONET blockchain node, thus enabling them via intelligent sharding and channeling mechanisms to retain the agreements they form with different entities confidential.

3.7 OLI Classification

In terms of the OLI model[11], the technology that has been described thus far sits on top of the seventh (7^{th}) layer, the Logistics Web Layer, whilst consuming data that is produced by the first (1^{st}) layer, the Physical Layer. A prerequisite of these types of smart contracts' existence is the capability of PI packets to produce the necessary metrics and data that can be exploited to automate SLA processes and the paperwork involved in logistics in general.

As such, even though the data is produced at the first layer of the OLI model, the smart contracts that reside on the seventh layer take advantage of all the intermittent layers as the data flows through them and is sometimes even altered by them f.e. in the instance of data format transformation. The ICONET blockchain solution being a 7^{th} layer technology enables it to be attachable, meaning that it is easily append-able to pre-existent OLI model networks without requiring changes to the core architecture making the technology easily adoptable.

It would be unreasonable to expand the ICONET blockchain technology's domain to the other OLI layers as, apart from being unsuited for this purpose, they are taken care of by other components of the ICONET solution. The ICONET blockchain is meant to be a complementary technology rather than a pivoting point for the ICONET project as it is exploratory and yet unproved.

4 Blockchain Solution Analysis

The blockchain of choice for ICONET needs to satisfy certain operational requirements to be capable of hosting the blockchain functionality that is desired by the use case that will be implemented in the project's lifetime. Nonetheless, any blockchain selected will be incapable of remaining relevant until PI is realized in practice, as blockchain is a technology that rapidly evolves and so is PI. Instead, the blockchain that will be selected will have a focus on being practical and relevant now and in the coming years, to provably display a proof of concept of the idea behind the implementation.

When the idea of ICONET was first conceived, the blockchain landscape indicated that Hyperledger Fabric would be the ideal blockchain for our implementation. The findings of this report dictate otherwise as the blockchain space has advanced and, after properly gauging the requirements of the ICONET implementation, another blockchain candidate arose. This revelation will be described in the sub-chapters that follow.

4.1 Available Blockchain Solutions

Before properly drafting a list of blockchain solutions that will be compared according to our KPIs, we must first narrow down the list by establishing a few functional requirements rather than operational ones. As inferred by sub-chapter 3.4, these operational requirements mostly pertain the privacy and general business aspect of the implementation. This immediately invalidates any and all blockchain implementations that operate in a public context as they would be unfit for the enterprise requirements of PI. As such, the list is narrowed down to enterprise blockchain implementations that can be utilized.

4.1.1 Hyperledger Fabric

The first well-known candidate is the child of the Hyperledger umbrella project, Hyperledger Fabric[13]. Fabric is an enterprise-grade blockchain that operates in a permissioned environment, with built-in business logic enabling a multi-channel network where multiple businesses can participate either independently or cooperatively. This segregation of the network allows its participants to separate their internal private logic and assets from the public facing ones, allowing the blockchain to simultaneously run on-premise and make data accessible to other participants of the network in a seamless manner (see Figure 5).

In the context of PI, Hyperledger Fabric is exceptionally useful in that it allows businesses to retain full authority over their data while permissively sharing it with other parties as necessary. Additionally, the fault tolerance of the network and high scalability allow it to be geographically dispersed over a large array of devices, which is the exact need of PI. Another benefit of this particular solution is the language agnostic smart contract system, although arguably that is a common trait among enterprise blockchain implementations.

A major drawback of Hyperledger Fabric is the need for multiple nodes to be operative in parallel. The complex architecture of the network requires each business to operate a set of Orderer nodes and a set of Peer nodes, each of which are ideally in separate devices. A technician responsible for maintaining the overall system operational is thus requisite along with the physical devices to operate it.

Additionally, the large dispersion of the system due to the requirement of at least two types of nodes per business entity would render it less scalable than other candidates as more time would be required for data to be propagated to all nodes of the network. The blockchain also comes with a set of pre-determined smart contracts that enable the creation of transactions, storage and more. While this might pose a benefit to most developers, for the purposes of PI the less the complexity of the system the more performant it will be. As such, a purpose-built blockchain with specific types of transactions rather than a generic blockchain with smart contracts would be more performant.

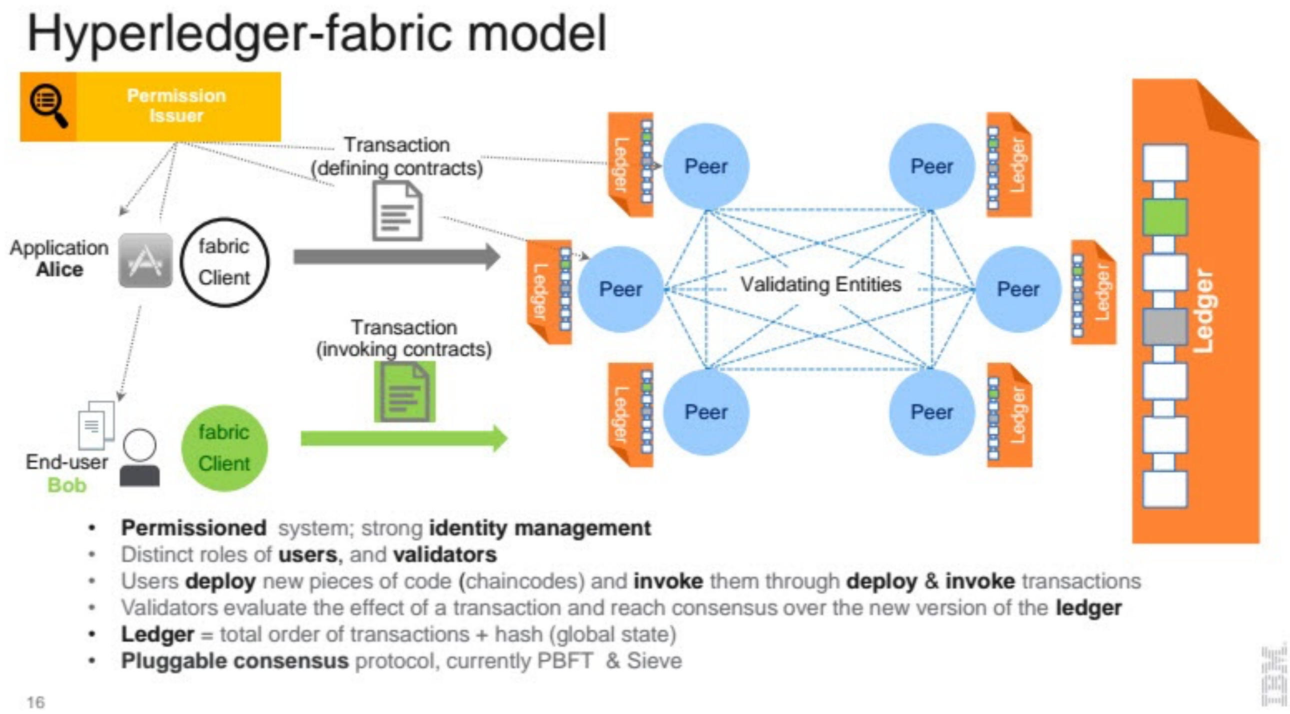

Figure 5 - Hyperledger Fabric model[2]

4.1.2 R Chain Cooperative

The R Chain Cooperative is another enterprise blockchain candidate that attempts to solve some of the issues of other similar blockchain implementations while operating a public network as well. It is developed by an independent consortium and has seen exposure mostly owing to its concurrent transaction execution, proprietary strict type checking algorithm and resulting high scalability.

The concurrency of transactions enables R Chain[14] to remain performant in a system where multiple transactions are expected to be executed simultaneously, such as in PI. This allows the network to remain performant under heavy load regardless of the distribution of the network as subsets of it would be able to execute transactions autonomously. Smart contracts also support the said concurrency as long as they are built around it, meaning that even complex operations can be parallelized. The strict type checking of the LADL algorithm of R Chain also eliminates common pitfalls in smart contract development such as race conditions, ensuring contracts are secure via a certain subset of attacks by design.

While at first glance it seems like an interesting candidate, certain drawbacks render it in an unfavourable position. First and foremost, it has a proprietary programming language for its smart contracts which puts off a large section of the developer community and renders maintenance of smart contracts in the network expensive. The network has also faced a few developmental setbacks and exploits during its development, raising questions as to how mature it is for a real use-case deployment. Overall, it is a blockchain candidate that while unidealistic at the moment is worth keeping an eye on for future developments.

4.1.3 Tendermint Consensus ABCI

Lastly, Tendermint[15] is a blockchain foundation rather than full solution, specifically offering the software necessary for the consensus aspect of the network. ABCI stands for Application BlockChain Interface, which is essentially a protobuf-powered[12] communication protocol for interacting with the Tendermint BFT consensus

[2] Unknown source, attributed to IBM

engine and exchanging messages between the peers of the Tendermint network. It describes itself as a “fault tolerant state replication machine”, discerning itself against full blockchain solutions. Tendermint’s logic is highly extensible and configurable, enabling developers to dictate the blockchain’s logic down to how blocks are accepted, and what types of transactions exist. This in turn allows developers to create purpose-built blockchains that follow a minimal approach and are highly performant.

Tendermint has been identified an exceptional candidate for PI as it allows the blockchain to be stripped of unnecessary functionality and instead only contain the logic necessary for the PI functionality desired to operate. Tendermint’s highly configurable nature allows the voting weight of each of the business participants of the network to be altered dynamically, fortifying the fault tolerance of the network in case a proof-of-importance protocol is adopted, where the higher the activity of a business the greater its voting power. A proof-of-importance protocol essentially awards network participants that have greater incentive to keep the network’s operational integrity intact while discouraging new participants of the network to act maliciously. Tendermint’s ABCI allows blockchain applications to be completely language-agnostic whereas in Hyperledger Fabric an SDK and schematic must first be developed by the consortium before a language is usable.

Although the benefits of Tendermint are numerous and outweigh the negatives greatly, that does not nullify them. Developers of blockchain applications on Tendermint need to be especially careful when handling its functionality as there are little to no safety checks for the applications and full responsibility befalls on the author of any application developed on Tendermint. This level of freedom can be compared to the powerful C language in comparison to a strict type-checked language such as Golang. An additional drawback of this freedom is that privacy and cryptography regarding data stored on the blockchain is not built-in on the implementation. The author of the application is supposed to implement these capabilities although many components are actually readily available online as open-source initiatives.

4.2 Blockchain Selection Justification

As evident by the preceding Chapters, Hyperledger Fabric is a powerful blockchain solution but for a performance- and time-sensitive environment such as PI it is not the best candidate. Summarizing the preceding sub-chapters:

Hyperledger Fabric – Powerful technology stack allowing complex ecosystems and business networks to form with privacy in mind. Perplexed nature leads to potentially unnecessary operational cost and high network participant network dispersion ratio.

R Chain Cooperative – Interesting technology with innovation in operational concurrency and smart contract type safety. Premature development stage and proprietary contract programming language mark it as a currently weak but potentially strong future candidate.

Tendermint Consensus ABCI – Highly configurable and barebones nature enabling purpose-built blockchain implementations achieving the highest performance capable with current tools. The great degree of freedom does come with certain responsibilities, opening up applications to certain types of attacks otherwise nullified in other implementations and forcing developers to re-write the wheel as not much blockchain functionality is built-in.

Taking the above summaries into account, a purpose-built blockchain based on Tendermint is better than a generic blockchain implementation based on Hyperledger Fabric as the latter would result in unnecessary transaction overhead, higher dispersion of the network and lower transactional throughput than Tendermint.

To ensure anyone can build on and interact with the blockchain ecosystem, an accompanying API gateway will be developed that allow services to communicate with the blockchain layer. Once again, Hyperledger Fabric actually requires each party that interacts with the network to be enrolled with an authority before interacting whereas Tendermint allows the developer to create an authentication system as he/she wishes. For the purposes of ICONET, an authentication system based on digital signature algorithms will be developed allowing users who interact with the network, to easily integrate it in their internal operations.

5 Dynamic PI Space Allocation

As mentioned in the previous paragraphs, this deliverable will focus on describing the efforts that were spent on coding the functionality behind the recording of PI packet movements in a PI network and their allocation to PI hubs along their transport path. Development was executed on the premise that the algorithms that dynamically allocate PI space have already been developed by other components of the ICONET network and the blockchain is mainly responsible for being capable of storing this type of data in a sensible format.

The developmental steps of the solution will be described while test suite cases will be developed in tandem to stress-test the blockchain network and ensure that it behaves performantly under the heavy load expected by future PI networks. As Tendermint's Consensus Engine will form the basis of our blockchain implementation, we will have to define the blockchain's logic down to the block acceptance protocol. For this purpose, we will attempt to utilize a minimalistic approach in these processes so as to incur the least possible operational overhead for the actual valuable operations to take place.

5.1 Use Case Details

This particular use case refers to geolocation sensors reporting the physical coordinates of a PI packet in the PI network back to a service which then propagates this change to the blockchain network for it to be stored in the distributed blockchain ledger and validated by all peers in the network. Additional metadata is also attached to the geolocation payloads for them to be associated with the correct PI packet and stored alongside them. If we assume that the physical equipment is in place in a PI packet and has already established communication with a service that will then propagate the data to our blockchain, we simply need to code and document an external facing endpoint that users of the blockchain will use to communicate with it.

For this purpose, we will define a RESTful interface that will be exposed directly by the Tendermint nodes. While general blockchain implementations expose their proprietary APIs, we are able to define our own in Tendermint as we are fully responsible for the node's logic. Alongside the RESTful interface, a JSON-RPC 2.0 interface will also be defined that will allow access to standard transactional operations in comparison to query operations. With JSON-RPC, we ensure that we follow the current standard of blockchain implementations while not inhibiting the operational capacity of our blockchain.

5.2 SELIS Pub-Sub Infrastructure Concept Integration

The low-level operational components of Tendermint follow a similar approach to the StreamMine3G framework, which is the foundation of the SELIS Pub-Sub infrastructure. StreamMine3G processes event streams through a set of stages which from a Directed Acyclic Graph (DAG). Tendermint's blockchain structure and internal transaction relations form a DAG to allow asynchronous commitance of transactions. While StreamMine3G can follow a unicast, anycast or broadcast stage path during the filtering process, Tendermint's DAG uses a multi-cast approach.

While the similarity between the two technologies ends here, it is apparent that both of them act as the ground-level of the technologies being developed in both projects. As SELIS has finalized its development cycle, we are able to discern the concepts that would be useful for ICONET's implementation and import those into this task's development. The SELIS Pub-Sub Infrastructure fundamentally enables external parties to subscribe to events processed by the StreamMine3G engine based on a set of pre-conditions. This is possible due to the fact that StreamMine3G inherently deals with streams of events.

Tendermint, on the other hand, has no concept of an Event unless it is defined by the developer. To that end, we will create events in the form of <key, value> pairs in uniformity with how SELIS events are depicted. These Events will be emitted whenever a sensitive state alteration is made on the blockchain, such as when the position of a PI packet is updated. While this development will be conducted on the blockchain level, an additional layer above the blockchain needs to be coded that will support the Publish-Subscribe concept.

For this purpose, we will define an API that will support users to subscribe to certain endpoints by specifying an asset ID, such as the ID of a PI packet, and the type of change that the subscriber is interested in.

This infrastructure will be especially useful in the context of SLAs as verifiers will be able to subscribe to changes and be notified when the SLA is broken. From a technological angle, we will use Server-Side Events (SSE) rather than WebSockets (WS) as the latter is more useful in bi-directional communication scenarios whereas we are more interested in uni-directional communication past the first handshake.

5.3 Smart Contract & API Implementation Details

While the term “Smart Contract” is utilized in this sub-chapter’s title, we essentially define blockchain-level logic rather than smart contract functionality in Tendermint, and these terms while different if translated from a higher-level perspective refer to the same thing, programs that are executed directly on the blockchain itself. As such, the requirements of the blockchain logic will be listed on this sub-chapter.

The most common format of representing the position of an entity in a two-dimensional plane, such as the one of Earth’s maps, is to utilize coordinates. As one must take the curvature of the Earth into account when positioning an object on a map, it is reasonable to depict locations using latitude and longitude coordinates. These coordinates utilize Decimal Degrees (DD), the numerical precision of which defines the scale which an object can be unambiguously recognized. For instance, a numerical precision of 6 decimal places allows one to pinpoint the location of individual humans.

The precision of these DDs is dictated by the physical constraints of the sensors on the PI packet of our PI network and as such, we will permit the storage of up to 7 decimal places which is the practical limit of commercial surveying. By increasing the upper bound of decimal places, we will accommodate for future advances in GPS technology. As the latitude and longitude coordinates are practically restricted in a very limited range, ±90° and ±180° respectively, it makes sense to use an unsigned integer binary representation rather than a floating number representation that we transform before outputting it to a querior of the blockchain.

Specifically, we can use the 0° to 180° range for latitude and 0° to 360° range for longitude to ensure the smallest binary representation possible at rest. For assigning data to a specific ID on the blockchain, we will use a <key, value> pair for retrieving the value of a distinct ID and we will store this value as a Protobuf. Google’s Protobuf is essentially a binary representation that is more efficient than JSON and used natively in Tendermint for exchanging messages between nodes of the network.

Based on coordinates alone, one could also establish an SLA based on the permitted speed limit for a fragile package. If the deltas between two coordinate updates on the blockchain are computed factoring in the time difference between them which is always tagged on each transaction on the blockchain, the resulting calculations can be used to determine the speed the PI packet is travelling at, if we presume the coordinates are precise enough.

For the higher-level API which will offer the Pub-Sub infrastructure detailed in 5.2, we will use an API coded in JavaScript for the Node.JS environment utilizing the Fastify[16] framework. In comparison to its peers, Fastify is the fastest available framework for coding APIs in Node.JS by a substantial margin as showcased on its repository. It even exceeds the speed of http.Server, the native server implementation of Node.JS, as it patches a bug of the native implementation that arises when faced with high load.

With regards to how the API and related technologies fit into the ICONET Architecture, this is still in debate as the ICONET WP2 concerning the software activities was worked on in parallel to this report. As such, no definite and actionable outputs had been developed by the WP in time to create a direct linkage between the D2.1 Architecture deliverable and this blockchain deliverable. This link will be addressed in the next version of the deliverable.

5.4 Test Suite Showcase

Initially, we considered implementing the test suite as a simple load test by using a tool like autocannon. The results of this test would indicate how our solution operates under heavy load but would not showcase a practical scenario. As such, we have coded tests using Tap, a JavaScript testing framework, as well as custom tests that simulate coordinate fluctuations of a PI packet and test that the subscription to speed changes acts correctly. The code of these tests can be observed in Annexes III & IV.

6 Conclusions & Work Ahead

The experimentational nature of PI and blockchain technology both pose an issue to any pursuit at quantifying forthcoming needs of the space and deliberating solutions for them. Bearing this in mind, this report has sought a mostly conceptual train of thought in lieu of one wholly fixated on practicality. Consequently, the theoretical material that has been produced by this report will be relevant for years to come whereas the software that was coded as a Proof-of-Concept (PoC) will be deprecated in the imminent future inasmuch as new technological advances will supersede current ones. This, however, does not mean that the PoCs were not fruitful as they demonstrate that the concepts are feasible even with current tools.

The principles of the practical solution will remain the same, as the unique traits that underpin blockchain technology will only be enhanced rather than omitted in future implementations and PI will still depend on contextual data of the PI packets that flow within the network.

To maintain the theoretical material pertinent, potential use case scenarios were singled out and filtered based on the criteria that blockchain technology would be applicable and advantageous for them. Numerous academic sources were aggregated and consulted to insightfully select the quintessential use case scenario for ICONET's intents and purposes. This scenario was estimated to be the tracing and interaction recording of a PI Packet in the trackage of the PI network. The produced data would then be capitalized on to form beneficial services around PI such as the autonomous uphold of SLAs and the ability to subscribe to positional triggers of a PI packet.

The formation of such services necessitates specialized traits for the underlying technological infrastructure like the redundancy of data across the network and the heightened security of it. For this objective blockchain technology was identified as the peerless candidate. Therefore, contemporary blockchain solutions were inspected and evaluated resulting in a singleton preference among the choices at our disposal. This selection, Tendermint, was elected owing to its barebones and minimalistic approach providing greater freedom to the development course and empowering faster than traditionally conceived blockchain applications to be created.

In continuation to the theoretical groundwork that was laid out the developmental steps were detailed from both a pen-and-paper and hands-on approach. First, the rationale justifying the data structures and operational workflows was construed with the practical viewpoint listing the technologies that were used following. The output was code exhibiting the theories laid out in the predating chapters and tests that assess the developed system's functions based on the requirements outlined by the use cases.

The product of this report is to be consumed by the ensuing versions of this report, version 2 & version 3, while the code that was developed can be potentially consumed by other tasks of ICONET should an alpha integration of the blockchain system with the overall ICONET system be worthwhile. As indicated by the chapters pertaining the selection of the appropriate use case for the blockchain, Living Lab 2 and Living Lab 3 can and will make use of the blockchain software developed via the demonstrative examples that will be developed as the project progresses.

With regards to other deliverables of ICONET, the reference architecture laid out in D2.1 can be updated to better illustrate where the blockchain solution fits in the grand scheme as a clearer picture of what role the blockchain takes has been painted with this report.

To re-iterate in short, an alpha blockchain solution was developed that is able to function properly in a development environment and can be integrated in other services. The use case the blockchain will focus on will be dynamic PI space allocation coupled with automatic enforcement of SLA guarantees, a use case that greatly benefits the latter two living labs, LL1 and LL2. The blockchain that will be used is an implementation based on the Tendermint Consensus engine, with custom logic down to the block structure.

Amendments to the blockchain implementation attached to this report may occur as the project progresses to accommodate for any prospective requirements that may arise in the future.

7 References

[1] Ballot, E., Montreuil, B., & Meller, R. (2014). *The physical internet*. La Documentation Française.

[2] Montreuil, B. (2011). Toward a Physical Internet: meeting the global logistics sustainability grand challenge. *Logistics Research*, *3*(2-3), 71-87.

[3] Montreuil, B., Meller, R. D., & Ballot, E. (2010). Towards a Physical Internet: the impact on logistics facilities and material handling systems design and innovation.

[4] Maslarić, M., Nikoličić, S., & Mirčetić, D. (2016). Logistics response to the industry 4.0: the physical internet. *Open engineering*, *6*(1).

[5] Fazili, M., Venkatadri, U., Cyrus, P., & Tajbakhsh, M. (2017). Physical Internet, conventional and hybrid logistic systems: a routing optimisation-based comparison using the Eastern Canada road network case study. *International Journal of Production Research*, *55*(9), 2703-2730.

[6] Qiu, X., Luo, H., Xu, G., Zhong, R., & Huang, G. Q. (2015). Physical assets and service sharing for IoT-enabled Supply Hub in Industrial Park (SHIP). *International Journal of Production Economics*, *159*, 4-15.

[7] Kant, K., & Pal, A. (2017). Internet of perishable logistics. *IEEE Internet Computing*, *21*(1), 22-31.

[8] Ambra, T., Caris, A., & Macharis, C. (2019). Towards freight transport system unification: Reviewing and combining the advancements in the physical internet and synchromodal transport research. *International Journal of Production Research*, *57*(6), 1606-1623.

[9] Baliga, A. (2017). Understanding blockchain consensus models. In *Persistent*.

[10] Nakrani, S., & Tovey, C. (2003, December). On honey bees and dynamic allocation in an internet server colony. In *Proceedings of 2nd International Workshop on the Mathematics and Algorithms of Social Insects* (pp. 1-8).

[11] Montreuil, B., Ballot, E., & Fontane, F. (2012). An open logistics interconnection model for the Physical Internet. *IFAC Proceedings Volumes*, *45*(6), 327-332.

[12] Protocolbuffers. (2019, June 21). Protocolbuffers/protobuf. Retrieved June 21, 2019, from https://github.com/protocolbuffers/protobuf

[13] Hyperledger. (2019, June 21). Hyperledger/fabric. Retrieved June 21, 2019, from https://github.com/hyperledger/fabric

[14] Rchain. (2019, June 18). Rchain/rchain. Retrieved June 21, 2019, from https://github.com/rchain/rchain

[15] Tendermint. (2019, April 19). Tendermint/tendermint. Retrieved June 21, 2019, from https://github.com/tendermint/tendermint

[16] Fastify. (2019, June 12). Fastify/fastify. Retrieved June 21, 2019, from https://github.com/fastify/fastify

[17] Ethereum. (2019, July 24). "Ethereum/Wiki." GitHub, Ethereum Foundation, https://github.com/ethereum/wiki/wiki/White-Paper.

Annex I: Blockchain Source Code

The source code below depicts the functionality of a blockchain with purpose-built transactions that track the coordinates of a PI packet in a PI network and using the Haversine formula, track the speed a packet is moving. Should the speed surpass a pre-specified speed limit, the blockchain is altered to reflect that change.

```
const createABCIServer = require("abci");
const PORT = 26658;

// turn on debug logging
require("debug").enable("abci*");

const state = {};

const handlers = {
  info() {
    return {
      data: "ICONET Blockchain - PI Packet Tracking",
      version: "0.2.0",
      lastBlockHeight: 0,
      lastBlockAppHash: Buffer.alloc(0)
    };
  },
  query({ path }) {
    return { code: 0, log: JSON.stringify(state[path]) };
  },
  checkTx({ tx }) {
    const { valid, type, vars } = extractVariables(tx);

    if (!valid) {
      return { code: 1, log: "tx failure, parameters malformed" };
    } else if (type === 1 && state[vars[3]] === undefined) {
      return { code: 1, log: "tx failure, no SLA exists for specified ID" };
    }

    return { code: 0, log: "tx succeeded" };
  },
  deliverTx(request) {
    const { type, vars, valid } = extractVariables(request.tx);

    if (!valid) {
      return { code: 1, log: "tx failure, parameters malformed" };
    } else if (type === 1 && state[vars[3]] === undefined) {
      return { code: 1, log: "tx failure, no SLA exists for specified ID" };
    }
```

```
    switch (type) {
      // Lat, Long, Elapsed Time in Seconds, ID
      case 1: {
        vars[0] = (vars[0] / Math.pow(10, 7)).toFixed(7) - 90;
        vars[1] = (vars[1] / Math.pow(10, 7)).toFixed(7) - 180;

        if (state[vars[3]].lat !== undefined) {
          const distance = haversineDistance(
            state[vars[3]].lat,
            state[vars[3]].lon,
            vars[0],
            vars[1]
          );

          state[vars[3]].speed = distance / (vars[2] / 3600);

          if (state[vars[3]].speed > state[vars[3]].limit) {
            state[vars[3]].broken = true;
          }
        }

        state[vars[3]].lat = vars[0];
        state[vars[3]].lon = vars[1];

        break;
      }
      // Speed Limit (Km/h), ID
      case 2: {
        state[vars[1]] = { limit: vars[0], broken: false };

        break;
      }
    }

    return { code: 0, log: "tx succeeded" };
  }
};

function toRad(num) {
  return (num * Math.PI) / 180;
}
```

```javascript
// Based on the Haversine Formula
function haversineDistance(lat1, lon1, lat2, lon2) {
  const R = 6371; // Earth's mean radius in kilometers
  const la1Rad = toRad(lat1);
  const la2Rad = toRad(lat2);
  const laDiff = toRad(lat2 - lat1);
  const loDiff = toRad(lon2 - lon1);

  const a =
    Math.pow(Math.sin(laDiff / 2), 2) +
    Math.cos(la1Rad) * Math.cos(la2Rad) * Math.pow(Math.sin(loDiff / 2), 2);

  const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));

  return R * c;
}

function extractVariables(tx) {
  const vars = [];

  switch (tx[0]) {
    // Lat, Long, Elapsed Time in Seconds, ID
    case 1: {
      if (tx.length <= 13) return { valid: false };

      vars.push(tx.readUInt32BE(1));
      vars.push(tx.readUInt32BE(5));
      vars.push(tx.readUInt32BE(9));
      vars.push(tx.slice(13).toString());
      break;
    }
    // Speed Limit (Km/h), ID
    case 2: {
      if (tx.length <= 5) return { valid: false };

      vars.push(tx.readUInt32BE(1));
      vars.push(tx.slice(5).toString());
      break;
    }
  }

  return { type: tx[0], vars, valid: true };
}

createABCIServer(handlers).listen(PORT, () => {
  console.log(`listening on port ${PORT}`);
});
```

Annex II: ICONET Pub-Sub API

```
const fastify = require("fastify")();
const axios = require("axios").create({ baseURL: "http://localhost:26657" });
const timestamps = {};

fastify.register(require("sly-fastify-sse"));

fastify.post("/:id", async (request, reply) => {
  const { data } = await axios.post(
    `/broadcast_tx_commit?tx=0x02${(
      "0000000" + request.body.limit.toString(16).toUpperCase()
    ).slice(-8) +
      request.params.id
        .split("")
        .map(char =>
          (
            "0" +
            char
              .charCodeAt(0)
              .toString(16)
              .toUpperCase()
          ).slice(-2)
        )
        .join("")}`
  );

  return data;
});

fastify.patch("/:id", async (request, reply) => {
  let secondsElapsed = 0;

  if (timestamps[request.params.id] !== undefined) {
    secondsElapsed = Date.now() - timestamps[request.params.id];
  }

  timestamps[request.params.id] = Date.now();
```

```
  const { data } = await axios.post(
    `/broadcast_tx_commit?tx=0x01${(
      "0000000" +
      Math.floor((request.body.lat + 90) * Math.pow(10, 7))
        .toString(16)
        .toUpperCase()
    ).slice(-8) +
      (
        "0000000" +
        Math.floor((request.body.lon + 180) * Math.pow(10, 7))
          .toString(16)
          .toUpperCase()
      ).slice(-8) +
      ("0000000" + secondsElapsed.toString(16).toUpperCase()).slice(-8) +
      request.params.id
        .split("")
        .map(char =>
          (
            "0" +
            char
              .charCodeAt(0)
              .toString(16)
              .toUpperCase()
          ).slice(-2)
        )
        .join("")}`
  );

  return data;
});
```

```
fastify.get("/:id", (request, reply) => {
  reply.sse(`${request.params.id} Subscription Successful`, {});

  const interval = setInterval(async () => {
    const { data } = await axios.post("/", {
      method: "abci_query",
      jsonrpc: "2.0",
      params: [request.params.id, "", "-1", false]
    });

    if (JSON.stringify(data.result.response) === "{}") {
      reply.sse({ event: "SLA Validity", data: "No SLA exists" });
      reply.sse();
      return clearInterval(interval);
    }

    reply.sse({
      event: "SLA Validity",
      data: JSON.parse(data.result.response.log).broken ? "Broken" : "Valid"
    });

    if (JSON.parse(data.result.response.log).broken) {
      reply.sse();
      clearInterval(interval);
    }
  }, 1000);
});

fastify.listen(3000, "0.0.0.0", (err, address) => {
  if (err) throw err;
  console.log(`Server listening on ${address}`);
});

module.exports = fastify;
```

Annex III: ICONET API Test Suites

```
const tap = require("tap");

// Start the server
const fastify = require("../index");

const id = Math.random()
  .toString(36)
  .slice(-5);

tap.tearDown(() => fastify.close());
```

```
tap.test("POST `/:id` route", t => {
  t.plan(3);

  fastify.inject(
    {
      method: "POST",
      url: `/${id}`,
      body: {
        limit: 1
      }
    },
    (err, response) => {
      t.error(err);
      const { result } = JSON.parse(response.body);
      t.strictEqual(result.check_tx.log, "tx succeeded");
      t.strictEqual(result.deliver_tx.log, "tx succeeded");
    }
  );
});

tap.test("PATCH `/:id` route", t => {
  t.plan(3);

  fastify.inject(
    {
      method: "PATCH",
      url: `/${id}`,
      body: {
        lon: 37.9441471,
        lat: 23.7156973
      }
    },
    (err, response) => {
      t.error(err);
      const { result } = JSON.parse(response.body);
      t.strictEqual(result.check_tx.log, "tx succeeded");
      t.strictEqual(result.deliver_tx.log, "tx succeeded");
    }
  );
});
```