



SmartAgriChain: A Blockchain Based Solution for Agri-food Certification and Supply Chain Management

Rocha T.¹, Costa P.¹, Sousa V.², Coelho P.², Sousa F.¹, Cardoso N.¹

¹Fraunhofer Portugal Research, Research and Development, Porto, Portugal

²Zenithwings, Aveiro, Portugal

Received: 11 May 2021; Received in revised form: 09 Jun 2021; Accepted: 20 Jun 2021; Available online: 30 Jun 2021

©2021 The Author(s). Published by Infogain Publication. This is an open access article under the CC BY license

(<https://creativecommons.org/licenses/by/4.0/>).

Abstract— Management of certification issuance and product counterfeit verifications in the Agri-food supply chain are very serious and reaching problems nowadays. The currently existing management systems for this process are either outdated or have significant issues when it comes to security, trust, traceability, management or product certification. The introduction of Blockchain technology, due to its intrinsic properties, has the potential to solve identity, ownership, data temper, traceability and certification issues. This is possible due to the unique identity of each actor and signing verification at each transaction/action. The decentralized nature and constant verification of the chain state also contribute to this security and trust in the system. The proposed solution does not compromise currently existing features, but it will, however, allow all the actors to take part in the Agri-food supply chain system and constantly monitor its actions. The SmartAgriChain project intends to implement a supply chain and certification system based on Hyperledger Sawtooth that will be capable of identity management, hierarchical users/organizations, significant scalability, low costs, low energy consumption and compatibility with legacy systems. In this paper, we will explore and explain the system design and architecture in detail as well as a cost projection based on the number of nodes of the distributed system.

Keywords— Agri-Food, Certification, Food Traceability, Sawtooth, SupplyChain

I. INTRODUCTION

Nowadays, supply chain systems are expanding in volume while becoming increasingly complex and global (Unno et al. 2020; Gonczol et al. 2020). However, little information about the product life cycle is available to the end-user. More importantly, actors in the middle of the supply chain may not have detailed information regarding the product's certification and origins of the raw material they use. This may allow greedy companies or individuals to hide some less correct behaviours for the sake of profit. These behaviours can range from low-quality control, product contamination, false quality certificates, unskilled or illegal workforce, among several other possibilities (Koegh et al. 2020; Tse et al. 2017). Another crucial point here is the product(s) certification reflected in the price and brand image/status. For these reasons, the need for a more transparent and traceable system in supply chain businesses is of the most importance (Unno et al. 2020; Koegh et al. 2020; Tse et al. 2017). Typically, supply chain systems are managed by a single central organization, having complete

control over what and how information is shared. Not only does this raises trust issues, as it also provides leverage to big organizations due to possession and authority over valuable information (Koegh et al. 2020). Nonetheless, this approach also represents a single point of failure for the organization, leaving the whole system vulnerable if compromised (Gonczol et al. 2020).

The field of agri-food represents one of many where supply chain management is of utmost importance. From the production of an alimentary good to its processing, distribution, retail, and finally, the end consumer, a lot can happen. It is challenging to assure the end-user of the process's full validity to ensure all quality control and guidelines were followed. This issue is crucial for premium products such as, for example, biologically grown products certification (Casado-Vara et al. 2018; Kamilaris et al. 2019). However, traceability is essential not only to the end-user but to all participating actors within the supply chain. This happens because, to produce a good product, the producer will need to make sure its raw materials are

consistent with the expected quality patterns and certification guidelines. Otherwise, the final product may have lower quality and should not be considered verified (Unno et al. 2020; Tse et al. 2017). All these issues may be traced to some complicated yet straightforward features of the system. Features such as transparency, security, trust, and accountability should not rely on a single entity.

SmartAgriChain is a project that intends to address the short falling mentioned above and focus on certification by adopting Distributed Ledger Technology (DLT), more specifically, blockchain. Blockchain is, in a very shallow description, a decentralized database for storing transaction information. It provides a complete, immutable history of transactions where entities do not have to trust each other to exchange information securely. Since it operates on a decentralized network, there is no single entity that controls the flow of information in the network, nor does it have a single point of failure. Every actor can have a stake in the network process, improving system decentralization, and requiring most of the network to validate transactions. Since each transaction must be digitally signed by the sender and verified by the network trust, traceability and transparency are naturally increased (Unno et al. 2020; Casado-Vara et al. 2018; Ghode et al. 2020). It becomes evident that blockchain technology usage in the supply chain could favour both producers and end-users theoretically. This article will lay the groundwork for SmartAgriChain with a study of the technologies available to understand the available solutions' vantages and disadvantages. We will also conduct a comprehensive cost estimation for the best candidates and proposes a solution addressing SmartAgriChain requirements.

The rest of this paper is organized as follows. Section II presents an overview of agri-food supply chains, a description of the SmartAgrichain objectives, how blockchain can be used, and DLT platforms available. The related work is specified in Section III. Next, Section IV describes the SmartAgrichain solution and architecture. Section V draws the main conclusions and points out future work.

II. OVERVIEW

1.1. Agri-food supply chain

To better understand how DLT can solve the currently existing problems in the agri-food supply chain, it is necessary to describe how several actors should ideally interact and how the system works. Figure 1 allows us to have a high-level overview of the generic process, where several actors, processes, and interactions take part in the agri-food supply chain.

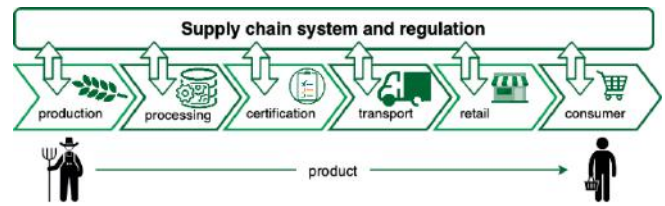


Fig.1: Generic representation of the Agri-food supply chain

Fig.1 represents a fairly complete product life cycle, however, most products and goods can have only a part of this cycle. We can briefly identify the most common actors in a complete supply chain and understand how they interact with each other and with the supply chain system and regulation. Ideally, the entities participating in the supply chain would have access to the relevant history of the products they process. This would allow them to be sure of the product origin and lifecycle. Nevertheless, most of the time, this is not the case. The actors receiving products from producers or other actors down the chain trust that the quality and type of the product given is what they expect. The supply chain information systems are not interconnected and do not work with the full process in mind. In the rare cases where we have a full-fledged supply chain system from the start of the process to the end, it will, in most cases, depend on a centralized infrastructure with all associated downsides of this architecture.

To summarize, this process needs an infrastructure where all the actors can trust the system, perform identity management, trace actions, and verify that records' immutability is assured.

1.2. SmartAgriChain proposed solution

Farmers, producers, and sellers are challenged to identify the best solution to safely and reliably interconnect all product management stages. The main goal is to identify opportunities for improvement in its operation and obtain certifications and quality guarantees in a more comfortable and faster way. SmartAgriChain aims to research and develop a solution to support the various entities related to the production and sale of agri-food products (farmers, sellers, certifying entities, etc.) in improving their supply chain processes and the respective certification of their products. The main objective of the SmartAgriChain is to investigate and develop a web-based technological solution that will combine blockchain technology with the entire process necessary for the certification of agri-food products (for example, certification of organic products), thus eliminating the excessive bureaucracy generally associated with this step and making the process faster, more transparent, safer and easily verifiable. Product certification is also an instrument that allows producers to 1) demonstrate impartially and credibly the quality, reliability,

and performance of its products, insofar as it reinforces customer confidence; 2) differentiate from the competitors; 3) increase competitiveness by reducing non-quality costs; 4) reinforce the company's image; and 5) facilitate access to new markets, showing compliance with regulatory requirements. Digital certification through blockchain technology (traceable, decentralized, and reliable) has the potential not only to facilitate everyday processes in the agri-food value chain but also to include small producers in the value chain, stimulated by the reduction of process bureaucracy, ease of access and language, in addition to reducing the transaction cost related to the other certification models currently in force. With all the mistrust in the agri-food sector regarding the origin, quality, and veracity of products, this digital tool can benefit all parties involved, valuing certified goods/foods. SmartAgriChain should include and act on all phases associated with supply chain management to transform the current chain into a more modern one, with guarantees of security, transparency, traceability, and tracking of products in all its phases. In this way, the consumer will be able to check all the information related to the product he is consuming, such as certifications and licenses acquired, information on the purchase of the seed, production methods, transport, sale, among other aspects. Therefore, the implementation of the SmartAgriChain project is based on meeting the following objectives:

- Simplify and improve the certification process for agri-food products, making it easier, faster, automated, and less paper-based
- Democratize access to certification services by small producers in the agri-food sector
- Provide a meeting point between producers and sellers seeking certification services, certifying entities, their experts, and consultancy entities in the area of certification
- Scan the certification processes
- Connect, safely and reliably, all stages of product management to identify opportunities for improvement in its operation
- Supporting producers and sellers of agri-food products in improving their supply chain management processes
- Provide the final consumer with information on certified products, giving greater confidence at the time of purchase.

To allow all existing features in legacy systems while reaching these objectives we must select a blockchain platform that provides the following requirements: (1) High transaction throughput and scalability - The network capacity to cope with an increasing demand for transactions

and interactions. The supply chain management system should deal with products at least at the lot/stack level. Given this, the Supply chain management system based on blockchain technology needs to cope with many transactions, least close to the thousands per day; (2) Open-access - For the SCM ecosystem participants, this point is not relevant, but we want to allow partial access to the end-user/consumer. By doing so, we can provide any consumer with verifiable information regarding the products they buy; (3) Secure and traceable - We do not want the system to be exposed or compromised. This relates to up-time-keep, system stability, external attacks, nefarious information temper, and role-controlled access; (4) Decentralized - Not only geographical decentralization is needed. The system should also offer subjective decentralization, meaning that each actor interested should deploy a node and be part of the network's validation process. By having several copies of the system, we can be sure that there is no single point of failure; (5) Competitive cost - The cost of the infrastructure must be competitive. If the system cost is not competitive, there is no practical advantage, so the system must be cost-competitive with existing SCM solutions. Also, the cost evolution over time must be predictable and ideally constant with the system volume.

1.3. Why Blockchain

This section will dive into a high-level overview of the technical details that allow blockchain to offer a significant set of advantages needed for scenarios such as this one. First, it is essential to address the differences and similarities between DLT and Blockchain. It is often common for these two terms to be confused or regarded as one. Despite their similarities and relation, blockchain and distributed ledger are different things. A blockchain can be considered a DLT, but there are several DLTs other than blockchain.

So, what is a Distributed Ledger Technology? A DLT is a decentralized database operated and managed by multiple actors across multiple validation nodes. On the other hand, a Blockchain is a type of DLT capable of recording transactions with an immutable cryptography signature called a hash. All "transactions" (meaning changes on the database) are then grouped into blocks. Each block contains a hash of the previous block, creating a connection between them, hence the term "blockchain".

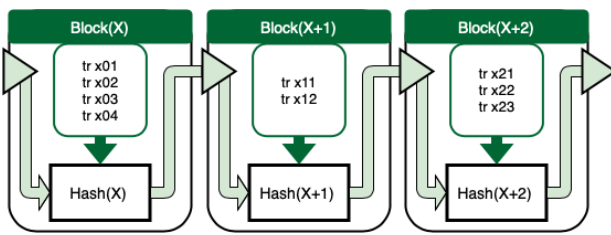


Fig.2: How blockchain achieves data immutability and integrity

For the cryptography concept details on how a blockchain works, we will analyze Fig.2. As the name indicated, a blockchain is defined by a group of blocks connected, resulting in a chain. These blocks are connected in a specific order, and once connected, they can no longer be disconnected or changed. This is the basic concept that allows for security and immutability in the blockchain. As we can see in Fig.2, for each block, two core elements exist, the transactions also referred to as payload and the hash of the block. The basic concept to assure the immutability of the chain is directly related to how each block's hash is generated:

$$\text{hash}(x) = \text{hash}(\text{hash}(x - 1) + \text{payload}(x))$$

Each block hash depends on its content and the hash of the previous block. The by-product of these designs is the immutability of record, precisely what we wanted to achieve. The logic dictates that it is impossible to keep the cryptography puzzle intact for every change of record of any block. This will break the connection between the tempered block and the next one, resulting in a compromised chain and invalid records if any malicious attempt to alter past data takes place (Unno et al. 2020; Kamilaris et al. 2019; Ferrag et al. 2018; Wang et al. 2018). However, the system's malicious actor could somehow alter the target block and all the forward block on the chain to make it valid. This takes us to the second security layer of the blockchain model, decentralization. There is no central entity in a Blockchain to process the network and validate the transactions, yet every transaction within the system is considered secured and verified. This can only be possible with the usage of a consensus protocol, a core part of any blockchain network. The consensus protocol acts as a mechanism through which all peers of the network or "nodes" reach a common agreement about the current state of the distributed ledger. By doing so, the protocol can achieve reliability and trust between unknown actors within the network. At its core, the algorithm assures that every new block added to the chain is the truth and agreed by all participants. However, according to the use-case and application field, several implementations are currently available with pros and cons.

1.4. Viable blockchain platforms

Once we established the advantages of Blockchain, the following step is to study the existing platforms and solutions that will allow us to implement a solution tailored to our needs. This section studies some of the existing platforms and verifies their pros and cons to implement our proposed solution.

1.4.1. Ethereum

One of the most well-known platforms when it comes to programable blockchain solutions is Ethereum, arguably the most widespread option. Ethereum has a virtual machine, called the Ethereum Virtual Machine (EVM). The EVM allows code to be verified and executed on the blockchain, providing guarantees it will be run the same way on everyone's machine. This code is contained in Smart Contracts, chunks of logic that can have autonomous behaviour according to its programmable logic. They can interact with other Smart Contracts or with other actors in the blockchain. The state of the network, memory, and logic of the Smart Contracts is processed and confirmed in a decentralized manner. The consensus in the network is achieved currently with PoW based algorithm called Ethash. Ethereum is a proven solution from a technical standpoint with several working decentralized applications. However, at the time of writing, the currently available Ethereum network has several scaling problems and a high cost. Nevertheless, it is also the biggest public blockchain in existence.

1.4.2. Hyperledger Sawtooth and Fabric

Hyperledger is an umbrella project of open-source frameworks, tools, and libraries for enterprise-grade blockchains. It was started by the Linux Foundation and has received contributions from Intel, IBM, J.P. Morgan, Airbus, and many others. The main objective is to develop reliable and efficient blockchains to support global businesses. The several subprojects integrate different protocol and consensus algorithms as well as different ways of managing smart contracts. For this case, we will check Hyperledger Sawtooth and Hyperledger Fabric.

Sawtooth is designed for versatility and scalability. It supports both permissioned and permissionless deployments, meaning the access or participation of all users to the blockchain is optional. With its distribution, a novel consensus algorithm is included, Proof of Elapsed Time (PoET). However, in this implementation, the consensus algorithm is interchangeable. PoET, however, targets mostly distributed validator populations with minimal resources. Transactions business logic is decoupled from the consensus layer and abstracted into Transaction Families. Due to this architectural characteristic it currently offers various programming

languages for development such as Python, Go and JavaScript, Java, C++, and Swift. Currently, Hyperledger Burrow integration allows for Ethereum Smart Contract support. Sawtooth currently has several deployed enterprise applications with a wide range of fields.

On the other hand, also from Hyperledger, we have Fabric. The main differences between Sawtooth and Fabric are that Fabric only supports permissioned blockchains and that there is no concept of transaction families. Smart Contracts are quite similar to Ethereum ones. Fabric presents a new architecture in smart contracts execution. Most existing smart-contract blockchain platforms follow an order-execute architecture: each transaction is validated, ordered, and sent to all nodes; only after does each node execute the transaction. This means that smart contracts must be deterministic. Otherwise, a consensus might not be reached. Usually, this ends up being forced with a DSL.

1.4.3. Other considered platforms

Other platforms could be used for the proposed scenario. However, some of the other considered candidates are either in early development phases and are not yet ready to be used, such as Cardano, or offer no significant advantages over the Hyperledger solutions as the case of R3 Corda.

III. RELATED WORK

Before explaining our solution, it is essential to explore and understand previous studies and the work conducted in the same field. This section will study and evaluate the similarities and differences that SmartAgriChain is meant to have compared to some other implementations in the same field, or at least with a similar purpose and the same technology.

The first case study (Shahid et al, 2020) is relatively recent and involves a solution with the Ethereum blockchain. This solution uses the InterPlanetary File System (IPFS) as a base for raw data storage. However, the Ethereum blockchain is used as a confirmation method for the IPFS content using data hashes from the stored content. By doing so, the validity of the data that is off-chain can be confirmed. Since Ethereum is a public blockchain, to control user access, a smart contract was developed to manage new user registrations and allow these new users to take part in the process according to their role. The 3 SmartContracts employed in this solution will assure user identity when interacting with the system, store the hashes of each action in the process, and interact with the IPFS. This system works as intended and can scale to a degree. It seems to be an excellent approach to offload work from the main chain. However, it is necessary to address the fact that even though the transactions and interaction with the Ethereum

blockchain are minimal and the cost estimations in this case study are realistic, the price of Ethereum GAS and the price of the Ethereum token are largely outdated. If the same study were conducted at the date of writing this article, the costs would not be practical.

Kamilaris (Kamilaris et al. 2019) study is based on the same agri-food field as the one previously mentioned. However, it has a less practical approach and instead studied existing projects in agriculture, food, and supply chain to employ blockchain solutions. This study reached 49 different projects/initiatives where the used blockchain implementations ranged from Ethereum, Hyperledger Fabric, Hyperledger Sawtooth, and some proprietary implementations. They concluded that the usage of this type of solution has real advantages and increasing potential. The technology is still in its infancy, but with new developments, tools, and solutions, the implementation process will be improved. The most used technology in this case study was Ethereum.

Another related but slightly different scenarios are the ones with a focus on IoT devices (Caro et al. 2018; Ferrag et al. 2018). These studies, while still applied to agri-food goods' supply chain management, this scenario does not focus on the chain actors as entities. Instead, it mainly intends to deploy a solution that allows tracking product information with IoT devices along the supply chain. Information such as temperature, humidity and light. Caro et al specify a practical scenario where the authors deployed two proof of concept solutions using Hyperledger Sawtooth and Ethereum. In conclusion, the authors point out the different advantages and disadvantages of each implementation. For Ethereum, in some cases, it may be convenient to trade off the high latency of Ethereum with its scalability and reliability since it enables larger numbers of participants and the platform at the time was significantly more mature than Sawtooth. On the other hand, Sawtooth offered a significant range of development languages when compared to Ethereum. It also offers significantly faster transaction times, higher scalability and significantly lower costs to operate. It also does not require as much computational power since it offers a novel consensus algorithm, more suitable for low-end devices. Nevertheless, the level of decentralization is not even close to what Ethereum achieves.

Baralla (Baralla et al. 2019) also applied a case study to the food supply chain scenario developed with Hyperledger Sawtooth. This paper intends to create a solution from-farm-to-fork capable of being integrated into existing chains and legacy processes while allowing full traceability of goods. It points out the separation between the application level and the core system as a significant sawtooth advantage that focuses exclusively on defining the rules and

application logic. The paper itself is the initial development steps for an ongoing project, and as such, it still does not offer details regarding transaction throughput, scalability, and costs. Furthermore, it has a high level of system specification, architecture, workflow, and requirements for this use case and shows that Hyperledger Sawtooth is highly flexible to this end.

A very interesting article regarding sawtooth scalability and performance specifics by Ambel (Ampel et al. 2019) reveals very interesting and promising results. With a maximum of 2300 transactions per second achieved using a cost-effective setup composed of a Ryzen 1600x 6 core CPU and 16Gb Ram. Other interesting metrics are the memory usage versus the transaction numbers and transaction execution times averaging less than a second. Nevertheless, the results achieved are pretty encouraging and show that the Sawtooth platform is completely able to handle the throughput needed for our scenario.

IV. COST AND VIABILITY STUDY

Based on the existing platforms and previous case studies, we selected the solution that best fits our requirements and is currently mature enough to start the implementation. Due to its maturity and proven technical record we decided to use Ethereum as the public blockchain platform since it checks all the technical requirements and is perfectly capable of executing the use cases for the agri-food certification and supply chain scenarios. From the private/enterprise blockchain side, Fabric and Corda are not public. Given this, we will decide on Hyperledger Sawtooth.

Once established the technical capability for these 2 platforms it is extremely important to verify the business scalability and cost viability. Both overtime and by volume. For the purpose of this study, we will assume a base scenario and then linearly extrapolate the costs for the Ethereum and Sawtooth hypothesis.

Note that due to the differences in the proposed platforms Ethereum has a “pay to use” based policy where every non-read interaction with the blockchain is charged as a transaction fee. On the other hand, Sawtooth, as a deployable network does not have a token or transaction associated costs. However, it has the associated costs of deploying the network nodes by the interested participating agents. These costs range from hardware costs, network costs and maintenance.

The test scenario does not need to be very specific to have an idea of the costs for each platform. So, we will use a basic scenario with a simple smart contract for Ethereum with 1

interaction per user per day and a network of up to 100 nodes for Hyperledger sawtooth.

1.5. Ethereum

Since this is a best-case scenario, we will use a very simple Smart Contract deployment:

```
contract Storage {
    uint32 number;
    function store(uint32 num) public {
        number = num;
    }
    function get() public view returns (uint32){
        return number;
    }
}
```

To overview the process’s cost in the Ethereum network, we first need to know how it works. For the eth blockchain, some concepts need to be addressed:

- Gas: Refers to the fee, or pricing value, required to conduct a transaction or execute a contract successfully.
- Gas price/gwei: Is a denomination of the cryptocurrency ether (ETH) used on the Ethereum network to buy and sell goods and services. Gwei is the most used unit of ether because gwei can specify Ethereum gas prices easily.
- Gas limit: The maximum amount of Gas that a user is willing to pay for a given action.

Other than these base concepts, we need to understand how the Ethereum Virtual Machine works and processes costs. Without going into specific calculations, some base costs are fixed, and others depend on the contracts’ deployment and execution. We can divide them into two categories:

- Base transaction costs: are based on the cost of sending data to the blockchain. Four items make up the full transaction cost: 1) Base cost of a transaction 21000 gas; 2) Base cost of a contract deployment 32000 gas; 3) Cost for every zero bytes of data or code for a transaction; 4) Cost of every non-zero byte of data or code for a transaction.
- Execution costs: are based on the cost of computational operations executed as a result of the transaction.

Based on this we can verify the current gwei and Ehtereum’s token price to estimate the price of a given transaction or SmartContract deployment. The gwei is not static. It will depend on the network usage. The Ethereum token price is also variable according to the Market. We will use the first eth/gwei prices we have tested and the current ones to have 2 distinct scenarios in Table 1.

Table 1: Ethereum gas and token prices for September 16th, 2020 and March 18th, 2021

Date	high priority gwei	avg priority gwei	low priority gwei	Ethereum \$
September 2020	139	146	160	365
March 2021	79	91	99	3410

The priority represents how fast the transaction can be processed when compared to the other network request. For simplicity, we will use the avg priority values in our calculations. Based on an IDE such as remix¹, we can check the cost in Gas of each smart contract deployment or call. For this cost evaluation purpose, we will use the smart contract mentioned above.

Table 2: Transaction costs for sample SmartContract deployment and interaction

	SC deployment	SC interaction
base cost	71632 gas	21464 gas
execution cost	51705 gas	21101 gas
total cost	123337 gas	42565 gas

Table 2 represents the costs associated with the sample SmartContract. With this data, we can calculate the actual cost of interaction in USD. To do so we can use the following formulas:

$$EthCost = GasPrice * GasConsumed * 10^9$$

$$Cost \$ = EthCost * EthPrice$$

With this information, we can create a table with the overall costs for this SmartContract deployment and interaction on both dates.

Table 3: Cost per interaction for the deployment and Interaction scenarios at 2 distinct dates

Date	Scenario	Cost
September 2020	Deployment	6.57 \$
	Interaction	2.26 \$
March 2021	Deployment	38.27 \$
	Interaction	13.2 \$

At this point, with the data available in Table 3 we can already see the costs of Ethereum network usage are at a non-competitive level. However, if we extrapolate a single

smart contract deployment and an increasing number of users with 1 interaction/day we have the following results:

Table 4: Daily cost estimation for N users with SmartContract deployment included

users	September 2020	March 2021
1	7.83 \$	51.47 \$
10	28.17 \$	170.27 \$
100	231.57 \$	1358.27 \$

In Table 4 we can verify that the daily costs in September 2020 were already noncompetitive. However, with the current situation, an Ethereum Mainnet based solution is completely unpractical from the cost standpoint.

1.6. Hyperledger Sawtooth

Unlike Ethereum, Hyperledger sawtooth does not have a direct token cost associated with a transaction. Sawtooth doesn't even have a token. This is because unlike Ethereum there is not a publicly available Mainnet. The stakeholders of the system must do the hardware deployment with multiple nodes/servers to construct the network. So, instead of counting the cost of an estimated number of transactions, we will evaluate the cost of deployment per month of N nodes. Note that the greater N is, greater the cost, but also more decentralized the network will become. Later on, we will also need to check the scalability of the solution based on the number of users. Since running a sawtooth node using Proof of elapsed time is not computationally intensive, we can assume that all the actors participating in the validation process will not need expensive hardware. Some research articles even managed to have a Sawtooth node running in Raspberry Pi's (Kromes et al. 2019).

For the sake of simplicity, we will assume DigitalOcen's listing of 2 CPU cores, 4GB of memory, and 80GB of storage at 20\$ monthly. This system is more than enough to run a Sawtooth Node. Tests executed in-house in a Virtual Machine system with half of those requirements allowed for 200+ transactions per second.

Table 5: Cost estimation for Hyperledger Sawtooth based on the number of nodes

Deployed nodes	Daily cost	Monthly cost
3	2 \$	60 \$
10	6.6 \$	200 \$
100	66.6 \$	2000 \$

¹ <https://ethgasstation.info/>

Table 5 shows a cost estimation for hyperledger sawtooth based on the number of nodes. When comparing to Ethereum, the costs on a Sawtooth network are significantly more competitive. The minimum number of nodes needed to run a Poet consensus algorithm is 3. So, we cannot deploy the network with less than 3 Nodes. Also, it is important to mention that, unlike Ethereum, the cost only increases with the number of deployed nodes. Not the number of users. Nevertheless, there are several implementations of sawtooth capable of 1000 transactions per second (TPS) (Ampel et al. 2019). This would easily be able to handle 10000+ active users. In our specific case, due to the possible complexity of the transactions, however, there are no guarantees we would achieve the same.

Another interesting aspect of this network is that due to the volume capabilities we could even enable IoT devices integration to track aspects such as humidity, temperature, location or even light. It is currently not in the scope of this project, but it is a nice future development.

V. PROPOSED SOLUTION

In this section, we will discuss details of the sawtooth implementation for SmartAgriChain. In Hyperledger Sawtooth the network nodes deployment can be progressive. This means that as soon as the network has the required 3 nodes it will be operational and extra nodes can be added over time.

Fig.3 shows a basic representation of a sawtooth network deployment. There we can clearly see the current node and the connections with other nodes representing the entire distributed network. A node can also contain a REST API to connect with clients and serve as a gateway to the blockchain contents. The Transaction Processor is the core part where we have the SmartAgriChain field-specific logic. It contains the rules of our system and processes the logic needed for our application. The Sawtooth design allows several transactions processors in the same network, allowing several applications to be used. For instance, our network deployment could also accommodate other application running in the same blockchain but utilizing a different Transaction Processor.

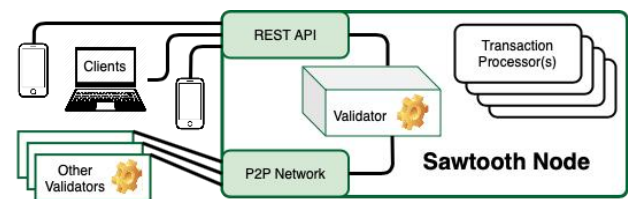


Fig.3: Hyperledger Sawtooth network layout and interactions

Given this network morphology now we need to specify the transaction workflow and structure to be able to map it to the logic we need for the supply chain and certification use-cases.

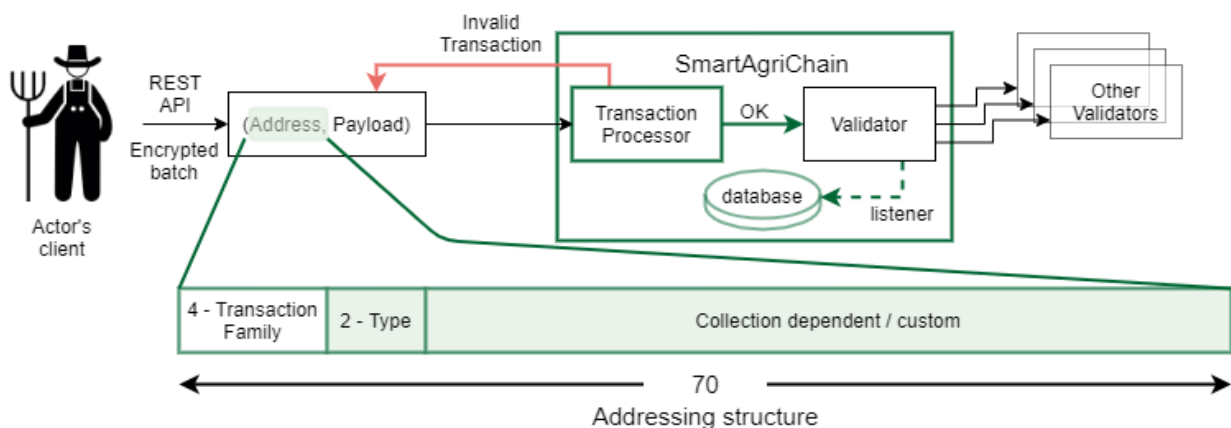


Fig.4: High-level overview of the proposed sawtooth implementation

Sawtooth stores the transactional data in a Merkle-Radix tree – using LMDB² as the underlying database – where each node can be accessed by an address of 70 hex characters. So, for this purpose, we need to map all existing records in this address space. **Error! Reference source not found.** represents a high-level overview of the proposed sawtooth implementation.

Each address then contains data in a format specified by the system. There is no specific method for building these addresses in Sawtooth. However, as recommended in their documentation, some structure is desirable since it can provide an intuitive way of uniquely mapping an object to an address. Typically, this is done using either the hex-encoded hash values of each stored object's main fields or

² <http://www.lmdb.tech/doc/>

a unique pre-defined hexstring for each type. Independent of the chosen method, all addresses must contain a unique prefix of 6 hex-characters corresponding to the transaction family. This will define which Transaction Processor is used to process the transaction. One way of doing this is to hex-encode the name of the developed transaction family.

Table 6: Addressing model for SmartAgriChain collections

Addressing space #70				
Collection	Transaction Family	Collection ID	Dependent on the collection	
Organization	4c0aba	ae	orgID	
		#2	#62	
Agent	#6	ed	orgID	agentID
		#2	#31	#31
Certification	#6	ac	orgID	agentID certID
		#2	#20	#20 #22
Product	#6	ec	agentID	productID
		#2	#31	#31

To start explaining how the addresses are generated we will define the collections needed for our application. We defined 4 core collection types: Organizations, Agents, Certifications and Products. These collections represent the core logic and information needed by the system to hierarchically control access to the collections and allow actors to execute certain actions according to their role. For instance, a certification process can only be validated by a certification entity. To be able to access data based on the addressing structure of Sawtooth we defined addressing model available in **Error! Reference source not found.**

We can easily explain this table with the following product example:

Table 7: Address generation example for a product

4c0aba	ec	sha(agent,31)	sha(product,31)
4c0abaec9009bd1cfaa91b4e87ca81746bfb3994a70d6a1bd			
9f087414804c914694d			

As seen in Table 7 we use a cryptographic hash function with the agentID and productID and use the first 31 characters to generate a given part of the address. Note that the total length of the address is always 70! If we wanted, for instance, to query all products of a specific agent we would use a partial address without the final product part, and the Sawtooth API would return a list of available products in that partial address.

Since queries for sawtooth use the first variable number of elements of an address this structure easily allows us to have access to the following queries:

- All records from an agent
- All certifications assigned to an organization
- All agents of an organization
- All certifications of a specific agent

Other complex queries can also be executed but not directly with Sawtooth addressing. If in the future the queries needed become too complex or unpractical a local database listening to ongoing accepted transactions can be implemented at each accessing node and replicate the blockchain state. This database can be verified with the blockchain at any point in time to assure coherence.

The addressing part of the implementation is then covered. However, we still need to explain exactly how these addresses will map to collections and data. To do so, in our implementation we used Protobuf³ and created a ProtoFile for each collection we use. Protobuf is an open-source, platform-independent tool used to serialize data structures like JSON. However, Protobuf offers several advantages when it comes to processing time and data volume. To prevent the unlikely scenario where we may have to change the same address for 2 different collection instances, we implemented a hash collision failsafe system that will store a list of collections on each address.

Currently, the system is being tested for deployment, so we can only provide preliminary results when it comes to the network capabilities regarding transactions per seconds (TPS) and response time. Tests revealed an average transaction confirmation time of less than 1 second and a maximum of 3 seconds. Also, in tests, the transactions throughput achieved was well above 200 TPS but note that this number may have significant variation based on the

³ <https://developers.google.com/protocol-buffers>

verifications, access and queries needed by the Transaction Processor for each specific request.

VI. CONCLUSION AND FUTUTRE WORK

With the study and architecture presented along with this article for SmartAgriChain, we believe blockchain technology should be part of the future of agri-food supply chain management and certification. It can provide all the features needed while adding value to the solution itself and the actors. It can also be done with concrete and acceptable costs for system usage and implementation. The current tools and platforms are not yet fully matured but are evolving rapidly and already allow for a complete implementation. Public blockchains do not provide the scalability and cost prediction needed for a no-compromise solution such as this one. This article represents an ongoing effort of development for SmartAgriChain, with a partially implemented solution that allows legacy systems to interact with our system via a rest API. The logic on the transaction processor is also not yet fully implemented but already allow for most operations needed in Agri-food management and certification of the use-cases. SmartAgriChain combines blockchain technology with the management of producers' supply chains, transforming the current chain into a more modern one with guarantees of security, transparency, traceability, and tracking of products in all its phases. In turn, consumers will have at their disposal a platform where they can confirm, quickly and easily (for example, using a smartphone), whether the products they buy respect the principles of sustainable agriculture and conscious consumption, principles that can be attested via a certification. It is increasingly proven that bringing consumers and producers closer together through technology will allow new food consumption forms/products. Besides, the platform will have mechanisms to simplify and improve the agricultural-based products' certification process, not being tied to a single type of certification. In the following phase, SmartAgriChain also intends to be a meeting point between producers and points of sale looking for certification services, the certifying entities responsible for all the management of the certification process in force, the respective certification experts and/or consulting entities in the area of certification or production. A kind of marketplace that aims to provide innovative components and blocks to serve as a basis for producers and sellers in the agri-food sector to improve their supply chain and the chances of certifying their products.

From a more technical point of view, it should be noted that this blockchain-based network provides high scalability to the system and guarantees security, transparency,

decentralization and traceability. Likewise, another focal point of this project is simplifying and digitalizing the procedural mechanisms of certifying an agri-food product. Since we provide a rest API based solution this can be achieved without dealing directly with blockchain logic, using a layer of abstraction instead. However, this mechanism still needs more research to find out exactly how information is shared between all stakeholders of the SmartAgriChain platform to guarantee its operation in a production environment. Blockchain technology already provides a mechanism to ensure that all stakeholders have access to information. Though, it is necessary to define mechanisms for access to information and associated knowledge. To this end, auxiliary mechanisms to the network based on blockchain will be investigated to provide controlled access to data, thus ensuring more transparency and security in the certification process.

ACKNOWLEDGEMENTS

This research received external funding from the project SmartAgriChain, co-funded by European Regional Development Fund (ERDF) from European Union (EU) through the Noth Portugal Regional Operational Program (NORTE 2020) with operation code 45416.

REFERENCES

- [1] Unnu, K., Pazour, J. A., Megahed, A., Narayanaswami, C. Blockchain Enabled Supply Chains: Classification, Decision Framework (2020). IIE Annual Conference. Proceedings , 483-488
- [2] Gonczol, P., Katsikouli, P., Herskind, L., Dragoni, N. (2020). Blockchain implementations and use cases for supply chains- a survey. Ieee Access, 8, 11856-11871.
- [3] Keogh, J. G., Rejeb, A., Khan, N., Dean, K., Hand, K. J. (2020). Data and food supply chain: Blockchain and GS1 standards in the food chain: a review of the possibilities and challenges. Building the Future of Food Safety Technology, 145.
- [4] Tse, D., Zhang, B., Yang, Y., Cheng, C., Mu, H. (2017, December). Blockchain application in food supply information security. In 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) (pp. 1357-1361). IEEE.
- [5] Casado-Vara, R., Prieto, J., De la Prieta, F., Corchado, J. M. (2018). How blockchain improves the supply chain: Case study alimentary supply chain. Procedia computer science, 134, 393-398.
- [6] Kamilaris, A., Fonts, A., Prenafeta-Boldó, F. X. (2019). The rise of blockchain technology in agriculture and food supply chains. Trends in Food Science & Technology, 91, 640-652.
- [7] Ghode, D., Yadav, V., Jain, R., & Soni, G. (2020). Adoption of blockchain in supply chain: an analysis of influencing factors. Journal of Enterprise Information Management.

- [8] Ferrag, M. A., Derdour, M., Mukherjee, M., Derhab, A., Maglaras, L., Janicke, H. (2018). Blockchain technologies for the internet of things: Research issues and challenges. *IEEE Internet of Things Journal*, 6(2), 2188-2204.
- [9] Wang, W., Hoang, D. T., Xiong, Z., Niyato, D., Wang, P., Hu, P., Wen, Y. (2018). A survey on consensus mechanisms and mining management in blockchain networks. *arXiv preprint arXiv:1805.02707*, 1-33.
- [10] Shahid, A., Almogren, A., Javaid, N., Al-Zahrani, F. A., Zuair, M., Alam, M. (2020). Blockchain-Based Agri-Food Supply Chain: A Complete Solution. *IEEE Access*, 8, 69230-69243.
- [11] Caro, M. P., Ali, M. S., Vecchio, M., Giaffreda, R. (2018, May). Blockchain-based traceability in Agri-Food supply chain management: A practical implementation. In *2018 IoT Vertical and Topical Summit on Agriculture-Tuscany (IOT Tuscany)* (pp. 1-4). IEEE.
- [12] Baralla, G., Pinna, A., Corrias, G. (2019, May). Ensure traceability in European food supply chain by using a blockchain system. In *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)* (pp. 40-47). IEEE.
- [13] R. Kromes, L. Gerrits and F. Verdier, "Adaptation of an embedded architecture to run Hyperledger Sawtooth Application," *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2019, pp. 0409-0415, doi: 10.1109/IEMCON.2019.8936264.
- [14] B. Ampel, M. Patton and H. Chen, "Performance Modeling of Hyperledger Sawtooth Blockchain," *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 2019, pp. 59-61, doi: 10.1109/ISI.2019.8823238.