LISBON SCHOOL OF ECONOMICS & MANAGEMENT
UNIVERSIDADE DE LISBOA

Master’s in management - MBA

Supply Chain Traceability using Blockchain

Pedro Ricardo Granjo de Azevedo

October – 2019

ISEG

LISBON
SCHOOL OF
ECONOMICS &
MANAGEMENT
UNIVERSIDADE DE LISBOA

Master's in management - MBA

Supply Chain Traceability using Blockchain

SUPERVISION

Professor Doutor Mário José Batista Romão

Pedro Ricardo Granjo de Azevedo

October – 2019

LIST OF ABREVIATIONS

API: Application Programming Interface

BC: Blockchain

CoC: Chain of Custody

CPU: Central Processing Unit

DSR: Design Science Research

EOA: Externally Owned Account

EPC: Electronic Product Code

ERC: Ethereum Request for Comments

EVM: Ethereum Virtual Machine

IT: Information Technology

IoT: Internet Of Things

KYC: Know Your Customer

PDO: Protected Designation of Origin

PKI: Public Key Infrastructure

PoC: Proof of Concept

QR: Quick Response

RFID: Radio-Frequency Identification

SC: Smart Contract

SCA: Supply Chain Actor

SCM: Supply Chain Management

ABSTRACT

Traceability is the ability to trace the origin, processing history, and the distribution of products in a Supply chain. In order to implement a complete traceability system, it is crucial to establish a chain of custody. Chain of Custody is typically defined as a sequence of procedures that validates the ownership and control of products along the supply chain. In the current global marketplace supply chains can span a huge number of countries, cross many borders and require interoperation of a multitude of organizations. This vastness of supply chains impacts business competitiveness since it adds complexity and can difficult securing traceability (ability to trace product attributes), chain of custody (chronological sequence of control) and transparency. In this work it is proposed that assurance of chain of custody is a complete approach for organizations to be able to demonstrate traceability, provenance (proof of origin) and product integrity and compliance. Blockchain technology with its attributes of decentralization, transparency and immutability has been touted to revolutionize several industries, and most recently has been proposed for supply chain management (SCM). The present study reviews the published literature to find the aspects that influence the problem and then follows the Design Science Research Methodology to analyze the requirements and propose a solution to a more complete traceability in SCMs. The results of this thesis were architectural artifacts, including an Ethereum SC (Smart Contract) and a certificate-based authentication system. These deliverables would allow implementation of a supply chain system over the Ethereum Blockchain that can provide decentralized and trustful assurance of the provenance, chain of custody and traceability functionalities for the participants and consumers.

KEYWORDS: Chain of Custody; Provenance; Traceability; Supply Chain; Blockchain; Ethereum, Smart contracts; Certificates; Design Science Research.

RESUMO

Rastreabilidade é a capacidade de rastrear a origem, a história e a distribuição de produtos numa cadeia logística. Para implementar rastreabilidade completa, é crucial estabelecer uma cadeia de custódia, normalmente definida como uma sequência de procedimentos que valida a propriedade e o controle de produtos ao longo da cadeia de logística. No mercado atual globalizado, as cadeias de logística podem abranger um grande número de países e fronteiras e exigir a interoperabilidade de numerosas organizações. Esta vastidão e complexidade impacta a competitividade dos negócios e dificulta a segurança, e a transparência da cadeia de logística. A implementação da rastreabilidade é fundamental para que as organizações possam posteriormente demonstrar a rastreabilidade, proveniência e integridade e conformidade do produto. A tecnologia Blockchain, com os seus atributos de descentralização, transparência e imutabilidade, tem sido apontada como destinada a revolucionar vários setores, com aplicação ao gerenciamento de cadeias de logística. O presente estudo começa pela revisão da literatura publicada para encontrar aspetos que influenciam o problema e segue a Metodologia de Pesquisa de Projeto para analisar os requisitos e propor uma solução para um sistema de gestão de cadeia de logística com melhor rastreabilidade. Os resultados da tese são artefactos de arquitetura, incluindo um contracto inteligente para Ethereum e um sistema de autenticação baseado em certificados, que permitem a implementação de um sistema de cadeia de logística suportado em Ethereum Blockchain que providencia aos seus utilizadores e ao consumidor final, as funcionalidades de proveniência, rastreabilidade e cadeia de custódia.

Palavras-chave: Cadeia de custódia; Proveniência; Rastreabilidade; Cadeia Logística; Ethereum Blockchain, Contratos Inteligentes; Certificados; Metodologia de Pesquisa de Projeto.

TABLE OF CONTENTS

INDEX OF FIGURES

INDEX OF TABLES

1 Introduction

Blockchain (BC) is a recent technology that was first introduced with the Bitcoin cryptocurrency. However, BC is not only applicable to cryptocurrency and it can and is being applied in other applications. BC results from the combination of several other technologies namely cryptography algorithms, peer to peer networking, consensus algorithms and software programming. The adoption of this young technology has had its fair share of hype and according to some authors it is currently around the peak of the hype cycle as reported by O'Marah, K. (2017). More than creating the trend of cryptocurrency, BC (sometimes also called distributed ledger technology) proposes the following main features to any application: decentralization, trust, transparency, irrevocability, immutability and computational logic. According to many authors most of these features seem to make a perfect fit to supply chains since they support the key basic objectives of supply chains: quality, speed, dependability, cost and flexibility (Casey et al. 2017). In addition to the mentioned traditional supply chain objectives a recent duo of aspects: traceability and provenance have gained more importance to allow the industries and customers to become assured of the products and processes sustainability (Kshetri, 2018). While it is common nowadays for logistics operators to accurately track packages at the transportation stages, that type of granularity is either lost or many times not possible at all stages of the supply chains since they have become much more complex, interorganizational and international spanning (Kim et al., 2018). This loss of provenance information creates much impact in sustainability and compliance efforts so the current focus on traceability has become crucial. Traceability permits the optimization of supply chains which has always been one of the most preeminent topics for businesses as it influences highly a firm's success. The optimization of the supply chain is then the main driving reason that has led some companies to make trials for Supply Chains using BC for traceability. Such is the case with Maersk – tracking global shipping, Alibaba – reduce food fraud, Lockheed Martin – improve cybersecurity, Everledger – implement diamonds and wine certificates, Walmart – monitor pork produce in China, Modum – safe drug delivery, Intel – track seafood supply chain, Bext360 – bring transparency into the coffee bean supply chain (as reported in Kshetri,2017). The initial target of this work was to learn as much as possible on the BC technology which evolved to analyze the current state of the art in supply chain implementations over BC and select the most important aspects related to traceability to be able to propose a solution to the traceability problem. This thesis proposes that in order to effect true traceability a complete approach is to connect both the Supply Chain Actors (SCAs) and products identifications using digital certificates. The BC will be used to manage the traceability and validation of the identities. In order to handle the importing and verification of certificates another existing architecture – WalliD[1] (as defined in Tavares, M., et al - 2018) has been selected to be reused by the proposed solution. In order to create, validate the certificates and setup the chain of trust an appropriate PKI (Public Key Infrastructure) was designed as part of the proposal. To better understand the problem and apply the solution an example was taken from a real food supply chain that uses provenance certificates. In summary this thesis work aims to provide a concrete answer to the supply chain traceability problem for the use case of certifiable actors and products. The answer is a complete traceability system that provides both SCAs and the customers the highest level of traceability by assuring provenance, chain of custody and traceability verifiability and visibility to the SCAs and customers. The solution proposal consists of a set of artifacts (architecture diagrams and workflows, Ethereum SC and a PKI infrastructure) that followed the DSR methodology.

[1] WalliD product: https://wallid.io/

2 Literature Review

In order to understand the historical and technological details of the BC technology and the possible business and technological applications to supply chains a comprehensive study and literature review was conducted.

2.1 Business adoption aspects

2.1.1 Business adoption value drivers

According to Angelis et al. (2019) the adoption of BC is promoted by the value it creates for firms. Four value drivers were identified in their study for the adoption of BC. The first is the decrease in transaction cost both in financial sense and by eliminating the need for a central authority and middlemen. This value driver statement assumes that the adoption of BC was already performed however this might not be the case if we take into account the complete implementation cost of a BC system. The second value driver is via the introduction of SCs (Smart Contracts) that allows for creating rules knowledge and establishment of trust between unknown parties. This value driver endows the ledgers with business logic and allows to leverage the possibility of integration with other IT (Information Technology) systems added value. The third is the introduction of Distributed Applications (also referred by DApps) that allow for new parties to be incorporated into firm's functions without the organization's direct control lowering the organizational barriers to service innovativeness. Finally, the adoption of BC together with other emerging technologies such as Artificial Intelligence and Internet of Things (IoT) could allow for increased productivity. The increase productivity would come via automatic decision systems and reduction of overhead in micromanaging systems and stocks.

2.1.2 Business adoption topics

According to Hughes et al. (2019) there are four important features managers need to consider in the analysis of adoption of BC: the implementation of a trust engine via the BC consensus protocol, the higher uptime since no single point of failure exists, the adoption of immutability for the records and the transaction speed variability. The last point can be problematic since the speed of transactions depends on the BC implementation from 7 transactions per second in the case of Bitcoin to 3000 transactions per second in the case of EOS[2]. Angelis et al. (2019) propose a structured framework of four questions that firms should answer to access the feasibility and impacts of BC in their business. The questions proposed in the study were: “What kind of value is sought?” (Angelis et al. 2019, p. 311) – that is which of the value drivers and features of BC are crucial to the organization. “Is it a feasible and viable option to adopt the technology” (Angelis et al. 2019, p. 312) – the firm needs to identify a strong expected benefit and there is access to sufficient IT knowledge to deploying the selected BC solution. “Why is BC preferable to a centralized ledger?” (Angelis et al. 2019, p. 312) – this means that the benefits of having a decentralized ledger should be valued against than the risks of data ownership, transaction time and susceptibility to the 51% or majority attacks. “What combination of technologies align with pursued value?” (Angelis et al. 2019, p. 312) – That is the firm has to define which features of BC it will use in combination with the existing IT systems.

2.1.3 BC adoption obstacles

Morkunas et al (2019) discussed and listed the BC adoption problems. At the top of the list was the general perception that BC operations are slow and costly compared with other centralized transaction systems. This perception issue however does not

[2] EOS BC proposal is to emulate the attributes of a real computer: https://eos.io/

translate into reality since there is no comparable technology that provides the same set of properties of trust, decentralization, programmability and immutability. The correct approach should be to only choose to apply and implement BC to problems that require its very specific capabilities and strengths and not to blindly try for BC to replace all existing systems. Additionally, the costs imputed to BC generally include the total implementation costs against other competing and more mature technologies which is not favorable to BC (due to its novelty) requires costlier IT skills, expertise and hardware. The next obstacle is that several news on BC trading platform data breaches have been reported which contrasts with the expectation (and requirement) for data security and integrity in IT systems. Finally, the last main adoption obstacle is that no standardization exists. The current state is there exist more than 6,6K active BC projects (and growing) all based on some different implementation of consensus protocol or coding language which difficult the integration between architectures and organizations. Nonetheless, some of these obstacles seem to be possible to overcome by recent initiatives. In what regards performance several new consensus mechanisms are being developed (e.g. such as Ripple, R3, Stellar reported by Morkunas et al (2019) that reduce processing time from seconds to milliseconds. Also, two main standardization efforts have appeared: Enterprise Ethereum Alliance[3] with more than 600 members and Hyperledger foundation[4] with over 250 organizations. The cost and complexity of BC are also decreasing via the appearance of major IT firms template BC commercial offerings (Amazon, IBM and Microsoft[5]).

[3] EEA is a member-led industry organization to drive adoption of Ethereum: https://entethalliance.org/
[4] Hyperledger is a Linux Foundation sponsored open source BC effort: https://www.hyperledger.org/
[5] As reported in: https://franciskim.co/blockchain-as-a-service-azure-vs-aws-vs-ibm/

2.2 Supply chain aspects

According to Chang et al. (2019) BC adoption in Supply Chain Management (SCM) is expected to boom over the next 5 years and is one of the BC applications with more growth potential where the market is estimated to grow at a compound annual growth rate of 87%. What follows is a literature review summary of the main aspects influencing BC and SCM adoption.

2.2.1 BC aspects impacting Supply Chains

According to Litke et al. (2019) there are several BC features that offer tradeoffs in SCM. Scalability may be improved since all actors participate in a common ledger without a single point of interaction. There may also be a performance increase measurable in a reduced time for assurance of transaction verification compared to centralized and escrow services (e.g. bank payment liquidity or manual verification of a bill of lading) and also possible due to automatic execution of contracts. The consensus mechanism provides trust to all actors in the chain. Offers privacy since although the transactions are verified the actor's identity might be kept private via the addressing scheme. Location dependency becomes more flexible by effectively allowing to make transactions autonomous from country regulations and laws. Reduced cost by allowing faster payments and with SCs allowing for faster dispute resolution. Wang et al (2019) summarize the generic benefits of BC to SC in 3 main topics: improvement of SC visibility, ensuring secure information sharing and trust, increased operational effectiveness.

2.2.2 BC Benefits to SCAs

Perboli et al. (2018) used a lean approach to design real world use cases that combine BC and SC. In their analysis there are specific benefits to each actor in the supply chain: Producer, Transporter, Distributer/Warehouse, Final user/customer. For the

Producer the value propositions of BC are the improvement of production planning and certification via Enterprise resource planning (ERP) integration, introducing Stock Keeping Unit (SKU) certificates into BC and the reduction of the bullwhip effect (improving supply chain visibility allows for increased production requirements accuracy). For the Distributor the visibility of the whole supply chain allows for better inventory update and the reduction of counterfeit, theft, wrong delivery, product recalls, paperwork and the increase in ease of compliance. For the Transporter/Carrier the benefits are the forecast improvement and the time slot reservation by using more real time information on the actual state of the product location and processing phase. For the final User the benefits depend on the segment: Business-to-Business or Business-to-consumer. Regarding the first, it will benefit more of easier stock management and expiration/recall management while the later will benefit more in better brand value by providing the consumer better health protection and more transparent sustainability or compliance claims.

2.2.3 BC adoption path in supply chains

Dobrovnik et al. (2018) propose an adoption path for BC in supply chains and logistics. They propose that companies first focus on single use cases to minimize risks of adoption and to start with proof on concepts that require little coordination with 3rd parties and that allow for IT skills to be developed and learn the technology nuances. Specifically, they mention the use case of reconciling multiple companies' internal databases since it is a contained problem that brings major benefits. The second proposed adoption approach it to tackle the transactions across boundaries as, in example, reducing the paperwork by migrating the bills of lading (responsibility ownership documents used in shipping industry) into BC. Thirdly they recommend focusing on replacing functionalities that do not require that end users significantly change their behavior. As an example, replacing paper certificates in the diamond industry. Finally, the introduction of new business models or new logics of value creation over BC, as for example using SCs to prioritize air corridors.

2.2.4 Problems and challenges of SC over BC

A particularly challenging aspect for supply chains over BC has been reported by Weber et al. (2016) and is the latency and latency variance. In a public Ethereum platform the average latency for a modeled supply chain scenario is about 23s. This problem however is reported to be mitigated in a private customized BC with average latency around 2.8 seconds. Another answer to the low performance problem of BC is advanced by Xu et al. (2019). Their study focused on providing traceability assurance via improving certificate traceability systems. These systems receive the certificates issued by inspection authorities (that verify the quality and originality of the products) and store and expose them to other interested parties for accountability purposes. The authors proposed and implemented a proof of concept that moved the centralized certificate traceability system to a decentralized system over BC in order to avoid the risk of tampering by unreliable employees or firms. Their answer to the lower performance problem is that it is acceptable in this use case since the number of certified suppliers and products is low and therefore acceptable. Another problem that undermines the effectiveness of supply chains over BC is that the number of stakeholders in global supply chains tend to undermine any traditional type or mechanism for enforcing security. Xu et al. (2018) in their work proposed to enhance the security of said supply chains via the binding of the physical and cyber worlds using certificates for both employees, devices and products that are responsible to enter and check the product data in the supply chain.

In order to understand which functionalities are required for SCAs it is important to define their reported weaknesses and limitations. From the presented review of literature of Supply chain aspects resulted the list of SCA limitations and problems which are presented in Table 1- SCA problems (see Annex 2 - SCA problems).

2.3 Traceability aspects

After the literature review it became clear that there the main problem in SCM and the one that emerged as the most interesting candidate for a solution with the adoption of BC was: assuring traceability. What follows is a summary of the literature review on traceability aspects that necessitate BC and the derived conceptual framework for implementing a SCM with more complete traceability.

2.3.1 BC application in SCM

Wang et al. (2019) conducted a series of interviews with supply chain experts and provided a supply chain challenges frame where the experts indicated the areas where they expected BC might penetrate. The areas at the top of the list were “providing visibility and traceability to stakeholders”, followed by “disintermediation” and “simplification, digitalization and optimization of SCM operations in a global context”. The areas where the experts perceived more challenges to BC usage in SCM were “cultural, procedural, governance and collaboration issues” and “cost, privacy, legal and security issues”. Dobrovnik et al. (2018) in their analysis to identify the potential BC application in logistics followed the Roger’s innovation framework which comprises 5 dimensions: relative advantage, compatibility, complexity, trialability and observability. In the result of this analysis the following topics related with traceability emerged as the primary factor in three of the dimensions. In the dimension of relative advantage, the major factor was provenance - prove that products originate from safe/sustainable sources. In the dimension of compatibility, the major factor was more accurate info in movements and time of delivery of products. In the dimension of observability, the most important issue was allowing to make more effective the tracking of fleet and vehicle performance history. Montecchi et al. (2018) make the case that supply chain transparency leads to provenance knowledge which will result in the reduction of perceived risks for the consumer and SCAs. According to the authors customers perceive risk when there is information that is not shared with them and this influences their perceived purchase decisions and attitudes towards the brands. By improving supply chain transparency and especially improving the provenance visibility aspect of products the participating firms can increase customer trust and reduce the perception of risk. In summary BC can offer powerful solutions to enhance customer provenance knowledge by tracing origin, certifying authenticity, tracking the custody and verifying the integrity of products.

2.3.2 Traceability conceptual framework

Many different aspects of traceability have already been mentioned and it is then important to provide clear definitions and context to their use and relationship to supply chains in order to have a conceptual framework on how to build a SC with more complete traceability. As mentioned by John G. Keogh[6], GS1 supply chain industry expert with 35 years of experience in the SC field the terms Provenance, Traceability and Chain of Custody (CoC) are often misused but their understanding and differentiation provides a stepwise framework on how to understand and approach traceability network.

Provenance: Even before BC was developed it had already been identified that provenance management was a cross-cutting "hard" problem in science, industry and society. In Cheney et al. (2009) provenance was defined as the metadata about the origin, context and history of change of origin in associated objects and processes. In order to assure provenance, there has to be some metadata that identifies the item

[8]Described in "Blockchain, Provenance, Traceability & Chain of Custody": https://bit.ly/2LaJ6x7

and its geographic characteristic and some functionality that transmits that information along the supply chain. At the time of the rise of the web and search engines it seemed that it was possible to make the claim that all metadata could be indexed, and provenance could be assured. However, several problems with the reality of provenance in SCM were pointed out: provenance was incomplete, unreliable, insecure, heterogeneous, difficult to integrate and non-portable across systems. At the time no real complete solution for provenance assurance was possible although the combination of sematic web and detailed causal graphs was suggested as a path forward. In order to make evident the difference of applying BC to the provenance problem Montecchi et al. (2019) used the slogan “It’s real, trust me” and proposed a framework where the traceability, certifiability, trackability and verifiability aspects of BC are set to contribute to increase provenance knowledge. This increase in provenance knowledge comes from providing provenance assurances: origin tracing, authenticity certification, custody tracking and integrity verification. These will in turn benefit firms by reducing business risks (real or perceived) which can be further categorized in physical, performance, social, psychological and financial risks.

Chain of custody - According to GS1 (2017) - chain of custody or cumulative tracking in the context of a supply chain is a time-ordered registry of the sequence of parties who take physical custody of an object or collection of objects as it moves through a supply chain network. Chain of custody historically comes from legal requirement perspective to provide proof of the tracking process. In highly regulated sectors (such as food, arms and drugs) chain of custody is critical and serves as the basis of both provenance and traceability assurance. According to Alliance, I. S. E. A. L. (2016) - global membership association for credible sustainability standards - the key propositions of a chain of custody system are to: identify the origin of a product (final or intermediate), ensure a custodial sequence along the supply chain, ensure that a certified product matches the certification characteristics, link, monitor and protect a claim at a certain stage of the chain with a claim at another point of the chain and finally to improve transparency. ISEAL proposes several custody models where the choice of the model depends on the claims the system or the actors wish to make. The models (in decreasing order of connectivity with a certain provenance claim) are identity preservation, segregation, mass balance overview and certificate trading.

Traceability has been defined in many different standards (EU Regulation (EC) No 178/2002, ISO 9000:2015, FAO CODEX Alimentarius CXG 60-2006) and it can be summarized by: “the origin of materials and parts, the processing history, and the distribution and location of the product after delivery”. Traceability comes from a business requirement perspective of tracking the movement of products and when origin information is preserved it is said to include provenance information. According to the most recent GS1 Global Traceability Standard, V2.0[7] these traceability concepts (Provenance, Traceability and CoC) when implemented correctly can be used to provide different levels of traceability functionality in supply chains. According to Sermpinis et al. (2018) there are two types of traceability: forward traceability the ability to find the locality at any point of the supply chain and backward traceability which is the ability to find the origin of any product given certain search criteria. Providing traceability is important for the food industry as is recommended[8] by the European Parliament in GMOs and GM free products. In order to provide traceability using BC in supply chains an approach is to tokenize the goods and use Smart Contracts (SCs) to model their transformation (Westerkamp et al., 2019). The BC in SCM traceability model has also been considered for risk management when supporting a Hazard Analysis and Critical Control Points System (HACCP) as described by Rahmadika et

[7] Latest GS1 standard at: https://www.gs1.org/standards/traceability/traceability/2-0

[8] EU traceability recommendations at: https://ec.europa.eu/food/plant/gmo/traceability_labelling_en

al. (2018). BC enabled traceability using SCs is also well adapted to the post supply chain and has been proposed in a Product Ownership Management System (POMS) that detects counterfeits via combining the Radio Frequency Identification (RFID) product tags with a Ethereum BC system as described by Toyoda et al. (2017).

2.3.3 GS1 Traceability data

The already mentioned GS1 V2.0 standard proposes to make the bridge between physical products and their digital counterparts. According to GS1, traceability data that can be collected can be defined to answer the following five questions at each point of any business process role. “Who” – is typically identified by a Global Location Number (GLN) code (constituted by Company Prefix, Location Reference and Check Digit). “What” – can be a combination of identifiers based on Global Trade Identification Number (GTIN) with increased traceability granularity: class-level (GTIN), lot-level (GTIN + batch/lot ID - Identification) or instance level (GTIN + serial ID). When in transport process the GTIN may be coupled with the Serial Shipping Container Code (SSCC) – this is a pallet IDs that is created in during packing (by the shipping party) and loses the context and value after receipt by (the receiving party). “Where” – is typically identified by a GLN but can be extended by a GLN extension component to identify internal locations within a site, the Serial GLN (SGLN). “When” can be answered via a time stamp which should include date and time (including time zone and Coordinated Universal Time (UTC) time offset). Finally, “Why” should state the role of the party in the chain. The typical roles are harvesting, manufacturing, shipping, transporting, receiving and selling. Some additional information might be added if shipping is required: Global Shipment Identification Number (GSIN) or Global Identification Number for Consignment (GINC) when a bill of lading requires that the logistic unit has common delivery or shipping.

2.3.4 Types of traceability networks

Besides the type of traceability information, it is also possible to categorize the type of information sharing across the supply chain. GS1 defines four types of traceability networks in a supply chain.: the “one up-one down network” whereby the traceability information is compartmentalized and shared only with the neighbor. The “cumulative tracking network” where information is encapsulated downstream the supply chain in a “Russian doll” process. This type of network is driven by regulations and was usually applied to preserving CoC information in traditional supply chain networks. The “single source database network” where all participants use a database (or even an ERP system) and supply it with the traceability information that is required. This network applies more to limited communities and suffers from centralization and lack of transparency and scalability problems. The last type is the most recent, the “distributed information sources” where actors in the supply chain network provide traceability information in a Peer to peer fashion (e.g. supported by BC). So as explained by Martindale et al. (2018) the BC brings about a different type of traceability model. From the traditional “Regulation mediated transparency model” to the “Technology mediated transparency model” where any type of traceability information can be supported across the network in a decentralized and replicated fashion.

2.3.5 SCM over BC state of the art

The literature review search for the state of the art on supply chain implementations with traceability over BC, revealed some publications that try to solve one or several aspects of the problem and that make concrete proposals for a solution. Xu et al. (2018) present a proposal for maritime cargo transportation (work conducted under a grant of the US department of Homeland Security) that addresses the problem maintaining the chain of custody that leads to cargo losses due to theft and a high burden in cargo inspections. Their proposal advances the idea of using digital identities for employees (in all subsidiaries involved in the shipping process) and the cargo tracking devices where all are signed by government agencies and trusted CAs. In this way they bind the registration of digital information with the physical world and thus resolve the chain of custody problem. Toyoda et al. (2017) focused on providing value to the post supply chain by proposing a Product Ownership Management System (POMS) that combats counterfeiting. The main problem they tackle is the cloning of RFID tags that allows for counterfeits to be introduced to the supply chain. Their proposal is based on implementing four functionalities over BC: authentication of legitimate producers, enrollment of products in SCM only by legitimate producers, maintaining the chain of custody during transfer of ownership in a two-step approach and advancing an incentive mechanism for the other SCAs to follow the proposal. At the post supply chain, the user can verify the current ownership of the product he buys, and this become assured of no counterfeiting. Westerkamp et al. (2019) propose to address the traceability maintenance across the supply chain using a lightweight implementation of the OpenZeppelin[9] ERC721[10] tokenization of the products. ERC721 (as is ERC20) are defined contract interfaces for the implementation of tokenized entities inside a SC. In ERC721 the token is non-fungible meaning it is unique and not exchangeable with another token. Another distinguishing characteristic of this proposal is that the transformation aspect is taken into account where a set of tokens can be used as input by a factory SCA to be transformed to another set of output tokens thus maintaining the traceability in this use case. The proposal defines a set of actions that are available to each SCA including the addition of products certificates to be stored inside SCs. It is proposed that the product certificates can be associated to the product tokens in 2 ways: either a certifier SC that holds all references to certified product

[9] Available at: https://bit.ly/2OtQPZ9

[10] ERC (Ethereum Request for Comment) are Ethereum standards: https://eips.ethereum.org/erc

tokens – would require compliance with the defined token reference and the defined architecture details, or a certificate SC that would hold the certificate information but would increase processing or storage overhead.

3 Research Methodology

The choice of methodology was Design Science Research (DSR) defined by Hevner et al. (2004) which is fundamentally a proactive problem-solving paradigm with the objective to create, apply and evaluate useful artifacts that have as objective to forward the human business and social capabilities in the context of information and management systems. This research project follows the DSR methodology since the aim of this work is to help solving the known difficult problem of providing complete traceability in supply chains for both the SCAs and the consumers. The DSR requires that the result of applying the methodology are artifacts and these can be defined either as constructs, models, methods or instantiations. In order to support DSR and to make sure that DSR is well carried Hevner et al. (2004). established a set of 7 rules or guidelines: (1) problem relevance, (2) research rigour, (3) design as a research problem, (4) design as an artifact, (5) design evaluation, (6) research contributions and (7) communication of the research. Details on the DSR methodology can be read in Hevner, A., & Chatterjee, S. (2010). In this thesis the process of investigation was literature review, definition of problems and requirements, definition of architecture and functionality, interview with use case SCAs, production of artifacts, application of use case and finally an interactive review of artifacts. The produced artifacts were a system architecture, the BC SCs required to support it, a PKI and digital certification scheme. It is important to clarify that within the principles of DSR this proposed solution is still subject of further reviews and improvements and also it is not a final implementation that can be deployed live.

3.1 Research objective

This research proposal intends to address the traceability problem by formulating a solution that leverages existing BC functionalities and certificate validation architectures and allows SCAs to gain confidence and verifiable knowledge on a product’s traceability in a decentralized manner. In order to provide context and guide the analysis and design of the solution an alimentary traceability case study has been selected and studied. The research problem then can be formulated as follows: “How to implement a supply chain traceability system using a certificate validation architecture using blockchain?”.

3.2 Research questions

The research problem can be partitioned into the following specific research questions: (1) “What are the relevant attributes and functionality that have to be considered to implement complete traceability in Supply Chain?”. (2) “What is an applicable architecture to implement a traceability system using certificates?”, (3) “What are the required smart contracts and their functionality to implement the Supply Chain System?”,(4) ”Understand and define the required functionality for business logic to import and retrieve ID and certificates into the SC system”, (5) ”How to validate the ID data and certificates”, (6) “How the end user can validate the certificates at the post supply chain?”, (7) “How can this system apply to a alimentary supply chain (case study)?”

3.3 Guideline application

In this proposal the DSR guidelines defined above were applied as follows. Regarding the first guideline it was clear in via the review of literature that SC traceability is one of the most important topics to all organizations that work in a SC that have to certify the provenance and chain of custody of certified products. The proposed artifacts for the solution have to support the implementation of a system that can help businesses and end users to find and verify associations between the product and its certificate in each step of the supply chain. This association will allow all actors to further process this data to make or verify claims on the products certificate properties. For businesses this will allow to detect problems on the certificate and the supply chain management (be it fraud, losses, failure of integrity) and for end user to assure that his purchase is compliant to the stated certificate. The second and fourth guidelines to be applied are the rigor and the artifact itself. The architecture design artifacts will follow good architecture guidelines as defined by Martin, R. C. (2017). The Smart Contract will have to follow described SC patterns by Bartoletti et al. (April 2017) and Antonopoulos et al. (2018) and is to be defined in solidity code that can be compiled into Ethereum bytecode. The third guideline that applies to this problem is Design as a Research Problem. Here the heuristic search strategy is performed by conducting a wide range literature review on the subject and collecting inputs form field specialists to find a solution that follows the state of the art and is as close to implementation as possible according to the available time. The fifth guideline is to make an evaluation of the artifacts. During this study the artifacts were reviewed via one round of review process with two focus groups of specialists one for each area: BC architecture and solidity (WallId team) and the PKI infrastructure (engineering specialist with experience in the technology). From each review improvements were incorporated into the final artifacts. The sixth guideline is research contribution and this study proposes a model for a system that combines certificate management and supply chain management over BC – which according to the literature review performed is unique. The seventh guideline will be this study itself that will be written with the additional goal of conveying all the research aspects and results in an accessible form to both management and technology audiences.

4 Results

4.1 Analysis of requirements

Based on the review of literature the SCM problems related to traceability are access control, impersonation, counterfeiting, theft and wrongful delivery, uniqueness of products, visibility, product recalls and brand value. Also according to the review of literature to solve these problems it was found to be crucial to improve the 3 aspects of the defined traceability conceptual framework: provenance (metadata about the origin and associated objects, processes and users), traceability (ability to trace the history, application or location of an object) and the chain of custody (time-ordered sequence of parties with physical custody of an object). The state-of-the-art literature review of SCM over BC provided indications on the required functionalities to implement more complete traceability namely:

- Manage the SCA access authorization via a certification mechanism;
- Bind the physical and digital worlds by restricting access to supply chain product data only to certified actors and devices;
- Use of a lightweight tokenization of products for representation of the products.
- Allow the import of certificates and verify the true identity of both SCAs and products using said certificates;
- Allow for certification data to be univocally linked with the SCAs and product tokens;
- Allow processing and transfer of ownership procedures while maintaining the identity chain of custody and respective certificate linkages;
- Reduce supply chain perceived risks in the post supply chain by allowing the customers to view certification information;

So, in order to verify a digital identity and ensure that only that participant, device or product can use that identity is an essential functionality that is required in supply chains that implement traceability. This functionality has 3 parts: being able to verify the digital signature, being able to verify the certificate of a CA has the correct attributes and that the participant is the correct owner of the certificate. This functionality was translated to a requirement to setup a PKI involving the SCAs, the CAs and their certified products. According to the literature review it was also possible to summarize the required attributes for a SCM with more complete traceability with verifiable: "user identity" (SCA ID and certificate), "product identity" (Product ID and certificate), "transfer of custody" (two-sided verification of SCA and product IDs), "uniqueness" (ledger of unique product IDs, "location of products" (geographic reference), and "timestamp of operations". From the required functionality and attributes, it was possible to select a supporting applicable technology and derive the corresponding improvement of the traceability aspect. The summary of this analysis is presented in Table 2 (see Annex 3 - Requirements).

4.2 Proposed functionality

This proposal uses both digital cryptographic certificates to establish SCA and product identity and authenticity inside/post the supply chain. The digital representation of supply chain products is supported by a lightweight tokenization of the products and their associated processes in a SC. However, in this proposal the certificates are only linked to the tokens and there is minor increase of BC storage and cost for the import and validation of the certificates in the SCM. To implement the previously defined requirements a set of SCM traceability functions were defined to be implemented in the Ethereum SC (with the business logic). The SCAs can operate the SCM by calling the SC functions according to Figure 1 which presents also the mapping of each function to the SCA that can call it.

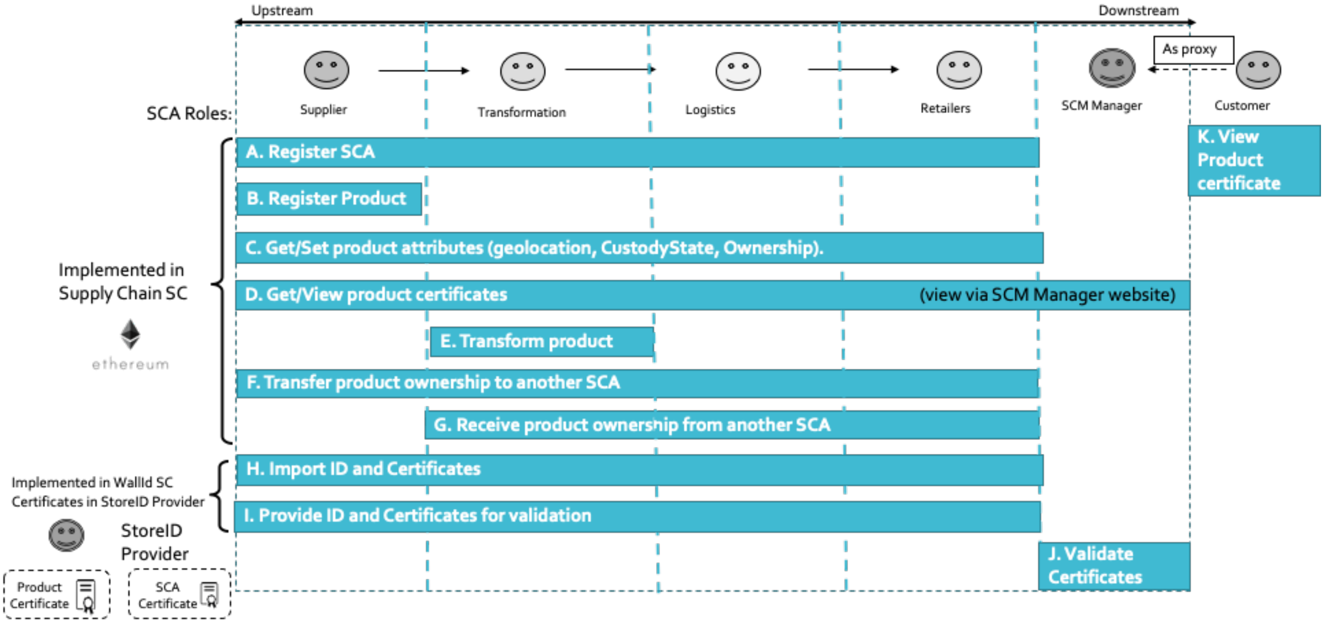

Upstream
Downstream
SCA Roles:
Supplier
Transformation
Logistics
Retailers
SCM Manager
As proxy
Customer
K. View Product certificate
A. Register SCA
B. Register Product
C. Get/Set product attributes (geolocation, CustodyState, Ownership).
Implemented in Supply Chain SC
ethereum
D. Get/View product certificates
(view via SCM Manager website)
E. Transform product
F. Transfer product ownership to another SCA
G. Receive product ownership from another SCA
Implemented in WallId SC Certificates in StoreID Provider
H. Import ID and Certificates
I. Provide ID and Certificates for validation
StoreID Provider
J. Validate Certificates
Product Certificate
SCA Certificate

Figure 1 - Proposed functionality

A. Register SCA – Any authorized SCA can register itself with the Supply Chain SC. In order to register a SCA must provide proof of its identity that has been made with the governmental or supply chain organizational entities. The objective is to provide decentralized authentication and avoid impersonation of the SCA. The SCA ID certificates are stored via function H. and validated via function I. (reusing WallId architecture).

B. Register Product - Any previously authorized SCA can register a product token with the Supply Chain Smart Contract. The objective is to allow for the minimum set of attributes to support traceability stored in BC (SC storage). The selected product attributes are the EPC, quantity, geolocation, ownership, custody state and if a certificate was provided. The addition of a product certificate is optional so the supply chain could if required operate with no product certificates but still provide product traceability with validated SCAs. The product certificate is imported in a storage provider (a specific role in the WalliD architecture) in the same way as for the SCA certificate and that SCA will remain the product certificate owner.

C. Get/Set Product Attributes – for an owned product it is possible to get/set some of the product token attributes namely: ownership, location and custody state.

D. Get Product Certificate – the current SCA product owner can request the SCA product certificate owner to retrieve the certificate from StoreId Provider and provide it to the Certificate Validator (SCM Manager). The certificate will then be available in the Certificate Validator website for the current SCA Owner to view. It shall also be possible for the consumer to use the SCM Manager as proxy to request to view the product certificate. This is the only SCM functionality that is provided to non SCAs (via SCM Manager proxy).

E. Transform Product – A SCA with transformation role can transform an existing product, copy the existing product attributes to a new product (new EPC) thus updating the inventory of products.

F. Transfer ownership to– A SCA with ownership of a product can tentatively change ownership to the a SCA (destination address is set), waiting for other SCA to commit (logistics handoff).

G. Receive ownership from-The previous process is only complete when the receiver SCA calls "Transfer ownership from" and the sender and receiver BC address are verified.

H. Import ID and Certificates (operates over WalliD SC and architecture) – These functions operate using events that are sent by the SC in 2 occasions: when a SCA registers and when a product certificate is added. Both events initiate a sequence of actions from the SC to the SCA that has the certificate (see Figure 14 – Import certificate architecture in Annex 5 - Certificate import and validation) and ends when the certificates are stored in "Store Provider" and validated in SC by "Certificate Validator".

I. Provide ID and Certificates for validation (operates over WalliD SC and architecture) – These functions operate using the events “perform KYC” / ”perform KYP” subsequent to the previous import. KYC (Know Your Customer) / KYP (Know Your Product) are similar use cases where the identity is verified either for user or for product. These flows end when “Certificate Validator” validates either the SCA or EPC (Electronic Product Code).

J. Validate Certificates – The SCM “Certificate Validator” must perform the hashing algorithm and verify the provided ID and certificate information and chain of trust are valid and not revoked in order to make proof to the Supply Chain SC that the registered SCA has a true identity and thus can operate on the supply chain. This also applies to a product so that the product identity provided is certified and its certificate is available to be viewed.

K. View EPC Certificate – SCAs and customers can use the Certificate Validator website address and call Get Product Certificates (EPC). In the case of customers, they use “Certificate Validator” as a proxy to access and retrieve the product certificate. This function is only available to “Certificate Validator” when the product is in custody state: “sold”

These functionalities are implemented in the SC artifact. For details on the SC code see Annex 6 - Ethereum Smart Contract and for details on interworking with WalliD architecture see Annex 5 - Certificate import and validation. For details on Public Key Infrastructure (PKI) see Annex 7 - Public Key Infrastructure and for the establishment of the PKI using OpenSSL[11] certificates see Annex 10 - PKI setup.

[11] OpenSSL is a general-purpose cryptography library licensed under an Apache-style license that is used to both secure communications and implement PKE and PKI: https://www.openssl.org/

4.3 Design aspects

Due to BC's unique capabilities and features several design aspects and tradeoffs have to be considered when designing a SCM over BC. The decision to use Private/consortium vs. public BC systems will depend of the selected industry use case, its requirement for global public access and will have impacts on the decentralization, the scalability and latency/latency variance of transactions. A public BC like Ethereum is global, fully decentralized and has higher availability due to the number of nodes and so is less prone to failures or malicious take over. One trade-off of a public BC is that its transactions are expected to have higher latency and latency variance and the scalability is not under control of the organizations. As an example, an operation over Ethereum public BC is expected to take tens of seconds, varying much on the load on the system which is not under control of the SCM participants. This latency and latency variance problem can be mitigated to a few seconds per transaction if a private or consortium BC is used however as trade-off it will lose some of the global access and availability features. In what regards scalability a private or consortium BC can scale its throughput without having to increase the number of nodes since it is possible in this case to specifically configure the consensus mechanism to allow for faster transaction validation. For the use case in analysis in this thesis and for the proposal both public or private BC networks are possible to be used and there is no dependency to any BC public/private flavour. Another design aspect to consider is how to store some or all of the SCM data: on-chain or off-chain. When data is stored on-chain (as a variable in a SC) it is more costly (2x more) but also more performant. Data can be stored off-chain and the SC interacts with it with the use of events but is less performant since it requires for queries to the log event and of course a more complex and latency prone software architecture. The proposal in this thesis aims to keep as little supply chain data on-chain as possible while retaining the traceability functionalities thus following a light tokenization approach. Reducing BC storage costs is important since it could make the solution too costly and hinder the business adoption. Following this approach, the certificates are stored/retrieved off chain via a store provider and events and only the information on the validity of SCA and product certificates are stored in BC. The SCA addresses and roles and a token representing the product is stored in BC. This token has a universal identifier (Electronic Product Code – EPC) which provides an immediate link to the physical world via Bar codes/QR codes/RFID tags. This EPC will also provide the link to the product certificate which is stored off-chain. Additional token attributes are the ownership link between EPC and SCA, the custody state (e.g. “owned”, ”inTransfer” or ”sold”) and the current geographic location of the product. The SCA access is implemented via SC logic by verification of the SCA certificates. The same pattern is used both for SCA and product certificates. An SCA can import SCA or product certificate into certificate storage and afterwards retrieve the certificate to validate the identity (SCA or Product) with the SC Manager. If the deployment is to a private/consortium BC it would be possible to remove the strict enforcing of authentication in the SC via certificates and have other methods of access control such as LDAP[12] (Lightweight Directory Access Protocol) and Kerberos[13] to prevent attacks on the BC consensus. It could be argued that private BCs provide better security from the start since only allowed users can use the BC. However, one of the main security risks to any BC is the protection of the private keys and the integrity of the SC code, which is non-dependent on network infrastructure control but on good security practices and hardened and well-maintained SC code. In order for an SCA to interact with the SCM it must create a BC account. A BC account is considered to be the public

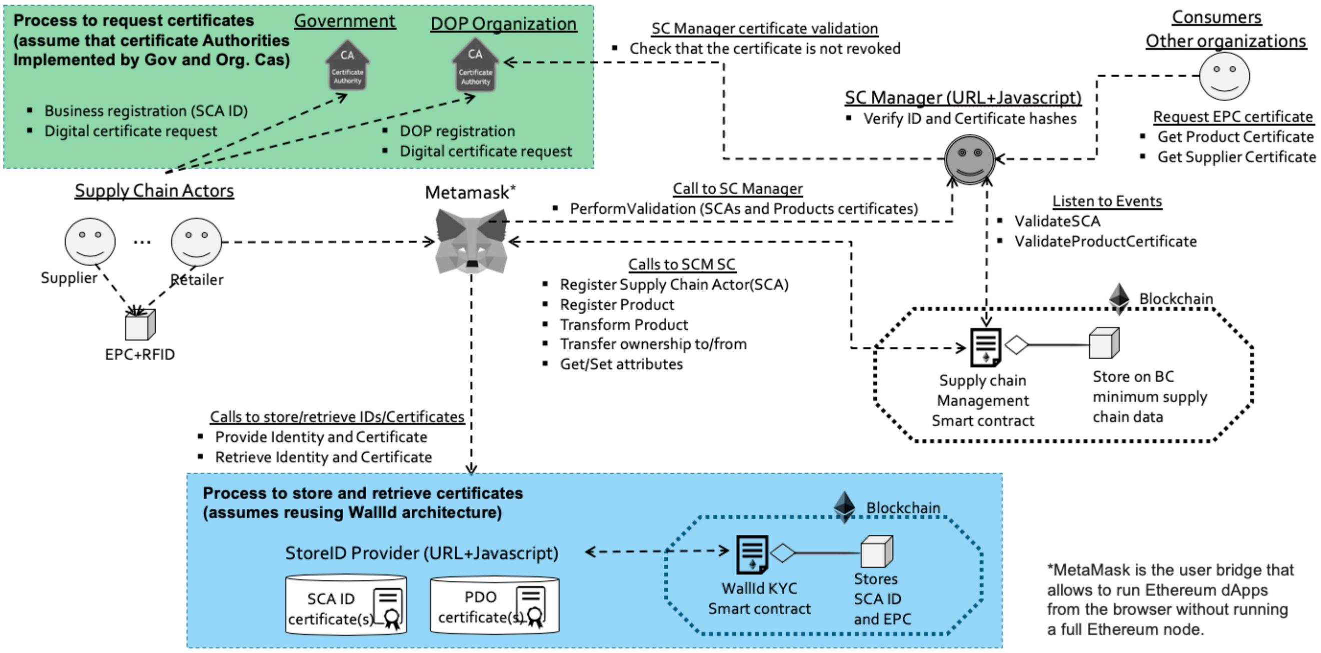
[12] An application protocol for accessing and maintaining distributed directory information services.

[13] A computer-network authentication protocol that uses tickets to allow nodes to prove their identity.

and private keypair that are connected to the user address and the funds (both stored in the network). To manage the account more easily an interface/wrapper is used, and this is generically called a Wallet. Interworking with BC with the user wallet can be performed in several ways. For this proposal Metamask[14] was selected since it allows to streamline the end-user experience by having an interaction with a responsive website while also allowing the use of both hardware wallets (e.g. Ledger[15] with Metamask) and online wallets (e.g. MyEtherWallet[16] with Metamask).

4.4 Proposed architecture

The proposed architecture uses a Ethereum BC SC to implement the decentralized supply chain functionalities. As already mentioned, the SCAs interact with the BC using their wallets via Metamask with calls to the SC code using a Javascript based interface (Web3 API[17]) – see Figure 2.


Process to request certificates
(assume that certificate Authorities
Implemented by Gov and Org. Cas)
Government
DOP Organization
CA
Certificate Authority
CA
Certificate Authority
▪ Business registration (SCA ID)
▪ Digital certificate request
▪ DOP registration
▪ Digital certificate request
SC Manager certificate validation
▪ Check that the certificate is not revoked
Consumers
Other organizations
SC Manager (URL+Javascript)
▪ Verify ID and Certificate hashes
Request EPC certificate
▪ Get Product Certificate
▪ Get Supplier Certificate
Supply Chain Actors
Supplier
...
Retailer
EPC+RFID
Metamask*
Call to SC Manager
▪ PerformValidation (SCAs and Products certificates)
Listen to Events
▪ ValidateSCA
▪ ValidateProductCertificate
Calls to SCM SC
▪ Register Supply Chain Actor(SCA)
▪ Register Product
▪ Transform Product
▪ Transfer ownership to/from
▪ Get/Set attributes
Blockchain
Supply chain
Management
Smart contract
Store on BC
minimum supply
chain data
Calls to store/retrieve IDs/Certificates
▪ Provide Identity and Certificate
▪ Retrieve Identity and Certificate
Process to store and retrieve certificates
(assumes reusing Wallld architecture)
StoreID Provider (URL+Javascript)
SCA ID
certificate(s)
PDO
certificate(s)
Blockchain
Wallld KYC
Smart contract
Stores
SCA ID
and EPC

*MetaMask is the user bridge that allows to run Ethereum dApps from the browser without running a full Ethereum node.

Figure 2 - Proposed architecture

[14] Metamask documentation. Retrieved from https://bit.ly/2ATNqeE

[15] Ledger is one example of a hardware Ethereum wallet: https://www.ledger.com/

[16] MyEtherWallet is one example of a software Ethereum wallet: https://www.myetherwallet.com/

[17] Web3 API is Ethereum Javascript API that allows to interact with an Ethereum node: https://web3js.readthedocs.io/en/v1.2.4/

The state of the supply chain, all its actors and products are tokenized in Ethereum via the SC. This proposal uses digital certificates for user and product authentication and so requires that recognized organizations implement a PKI (in order to generate digital certificates and provide the chain of trust). It is also proposed that governments and certification organizations are among the best candidates for establishing a PKI for businesses operating on global supply chains. It is possible to have the PKI implemented directly by consortium organizations when the use cases are restricted to specific businesses or industry sectors. This proposal requires that only validated SCAs (the ones which can provide an ID and certificate that match and verifies) can register products into the SC system. The SCA and products certificates are digital certificates that attest of the business identity (e.g. when registering with national government agency) or attest that the product has unique distinguishing and certified characteristics (such as in the case of PDO (Protected Designation of Origin) where the products and processes are verified and certified by a selected PDO regulator organization. The importing and retrieval of the SCA and product certificates is performed with the reuse of the KYC architecture from WalliD adapted for other identities such as organizations and products. Products in the supply chain are referenced by an industry referencing standard, the EPC – Electronic Product Code which is a unique identifier commonly used in supply chains as described in the case of livestock supply chain by Hartley et al. (2014). The supply chain can operate with products without certification (since it is optional to provide them) but the main value proposal is for certified products. The validation of certificate hashes for both SCA identity and products is performed off-chain by the Supply Chain Manager entity (that includes the role of “Certificate Validator”). This decision to have certificate validation off chain comes because at this time it is not feasible to perform crypto hashing on-chain in the current Ethereum EVM. The SC Manager (“Certificate Validator”) is also responsible to perform the verification of the certificate revocation. Additionally, SC Manager also provides a website for SCA actors and consumer to request and view the product certificate given a product EPC (which can be read from the physical product RFID or QR/Serial code tag). After SCA certificate validation the Supply chain management and traceability functionality is provided using the SCA BC addresses via transactions in a trusted and decentralized way with no further validation required. The SCM-SC is deployed in the EVM by the SC Manager which is the owner and ultimate responsible for the security and maintenance of the SC code. A more detailed description of the architecture workflows is available in Annex 4 - Architecture workflows.

4.5 Use case application

As already mentioned, an example about alimentary supply chain use case was selected in order to better understand the requirements and correct application of the proposed solution. The selected use case was the production and transformation of certified livestock produce of the “Carne Mirandesa” type due to ease of access to the SCAs. From an interview with a producer and retailer 3 document samples were collected: the certificate that links the Government ID of the animal (SNIRA[18] ID) with the PDO ID and certificate of the brand “Mirandesa” (Geneology ID), the transformation identifier that is shipped with the carcass that shows the reference to the animal (SNIRA ID) and the sale point invoice attests to the purchase of the carcass and served meals with the reference to the batch (SNIRA ID/Lote). The unique identifier in the supply chain is the Electronic Product Code (EPC) that is linked with both the SNIRA ID and the Genealogy ID (EPC-SNIRA ID-Genealogy ID). The product certificate to be

[18] SNIRA (SISTEMA NACIONAL DE INFORMAÇÃO E REGISTO ANIMAL) is the national Portuguese Government livestock registry: https://www.ifap.pt/snira

generated will have these 3 fields in the X509[19] certificate request as mandatory fields to be certified together with the public key hash of the producer (supplier actor in the SCM). In summary each product will be issued a X509 certificate by the producer. This particular CA chain of trust requires three SCA certificates: one for the root CA (the Government CA), another for the intermediate CA (the PDO association CA) and finally one for the SCA producer. In order to streamline the production of X509 certificates the request for product certificates can be automated by an application at the producer side that requires that the producer inputs the triad of X509 attributes that need to be certified (EPC, SNIRA-ID, PDO-ID) and then issues the certificate request and at the CA's side a backend IT system that issues the X509 certificate after the verification processes have been validated. It should also be noted that as a product is processed in the supply chain its EPC code may change from animal EPC (type SGTIN) to carcass EPC (type SSCC) and carton tag EPC (type SGTIN). However even in this case the SCM SC will maintain reference to the original certified EPC and its owner SCA and maintains certificate traceability. This EPC code change is also described by GS1 in Hartley et al. (2014) where a livestock traceability proof of concept (PoC) was implemented. In that PoC the EPC codes are read from RFID tags inserted into a centralized SCM application to provide the required traceability metadata. In the PoC the EPCs suffer change due to processing the meat thus the requirement for "transform product" functionality in the SCM SC to keep the EPC and certificate link. For details on use case samples see Annex 8 - Use case data. For details on the EPC details see Annex 9 - EPC detail. For details on the certificate request and revocation see Annex 10 - PKI setup.

[19] X.509 is a cryptographic standard defined by ITU-T standardization body that defines the format of public key certificates (used in https and electronic signatures): https://tools.ietf.org/html/rfc5280

5 Conclusions and future work

5.1 Summary

In order to fully answer the research problem, the research questions were addressed and answered during the results chapter. Following are summarized answers:

1) “What are the relevant attributes and functionality that have to be considered to implement complete traceability in Supply Chain?” Answer: the EPC and digital certificate data and the SC functionality implement complete traceability.

2) “What is an applicable architecture to implement a traceability system using certificates?” Answer: the proposed architecture uses Ethereum SC to implement SCM traceability and interworks with both a PKI infrastructure and an identity and certificate management system (from WallId) to resolve all the defined requirements.

3) “What are the required SCs and their functionality to implement the Supply Chain System?” Answer: a single SCM SC is required for traceability functions and another identity/certificate management SC is reused for both SCAs and products’ certificates.

4) ”Understand and define the required functionality for business logic to import and retrieve ID and certificates into the SC system”. Answer: the SCM SC implemented the required functions and applied the events pattern of WallId SC.

5) ”How to validate the ID data and certificates”. Answer: the proposal advances a solution of an offchain “certificate validator” using openSSL for the chain of trust.

6) “How the end user can validate the certificates at the post supply chain?”. Answer: the “SC Manager” validates the certificates and allows the customer to view them.

7) “How can this system apply to an alimentary supply chain (case study)?”. Answer: by setting up the PKI and creating the certificate chain of the “Mirandesa” livestock breed the system is able to provide complete traceability until the final customer.

5.2 Contributions

With this thesis work it was possible to understand the BC and Ethereum technologies, improve knowledge on supply chain systems, on traceability requirements and of certificate-based authentication and verification systems. The research contribution is a solution that aims to answer the research problem and questions. The main benefits of this proposed solution are that it allows for a group of independent participants to implement a more decentralized supply chain system with a complete traceability model for certified products. Another benefit is that it allows for both SCAs and customers to import and view the product certificates thus providing trust among participants. The proposed architecture builds on using digital certificates produced by trusted organizations and on reusing a KYC system to both store and validate these certificates. If any certificate fails validation (e.g. expiration) it is added to a CRL (Certificate Revocation List) by the CA and the SC Manager sets the certificate flag in the SC to invalid. If a SCA certificate is revoked, the SCA will not be able to access the SC until it provides a valid certificate. If the product certificate is revoked the product token can still be managed in the SCM but the certificates will be shown as expired or invalid to SCAs and customers. During operation the SCAs have access to the ownership status, the operations timestamps, the chain of certificates and the transfer locations which can be further processed by SCAs own internal SCMs to analyze how to optimize the supply chain.

5.3 Limitations

The main drawbacks of the solution are its dependency on a PKI (for SCAs and products) and the dependency on a central entity (SC Manager) for certificate validation. Regarding the dependency on PKI the minimum requirements are its setup by a national or regional or any trusted authority and the SCAs have to subscribe to an authentication scheme for their identities and products. Regarding the SC Manager it has to deploy the proposed SC into the Ethereum BC while becoming responsible to be the “certificate validator”. Instead of digital certificates some other solutions propose the import of scanned paper certificates (in a PNG or PDF format). This is the case of “originChain” implementation, but it is not a step forward since such verification is more complex to implement and may require intervention. A better way forward is sure to be the use of IoT devices and RFID tags with digital certificates. In this approach all authentication is verification is automated with higher granularity of secure processes. An alternative for the SCAs certification-based authorization is to replace it with a simpler although less decentralized and autonomous authentication system (e.g. LDAP). but increases complexity and reduces the security in cases of global SCMs. If only a sectorial industry approach is required and the global decentralization and autonomy is not necessary and it would be more advantageous to operate over a consortium BC, with the benefits of lower latencies/latency variances and centralized SCM control.

5.4 Business aspects

For this proposal the main value drivers (following the previously mentioned framework of Angelis et al. (2019)) in the adoption of the BC technology are: to establish trust between unknown parties and the increase of productivity due to the possibility of automated interaction of the BC trust engine with other SCs, IoT devices and Artificial Intelligence technologies. As described in the literature review the authors Angelis et al. (2019) also propose a framework to access the feasibility and impacts to business adoption. In what regards this proposal the main value that is sought is “decentralized traceability assurance”. The feasibility and viability aspect is still an open question for future exploration. For the type of use cases that are targeted (global, complex supply chains with many SCAs) BC is preferable to a centralized ledger for businesses since it should lower the cost and simplify adoption while providing SCM traceability. The aligned technologies would be existing SCMs, IoT (e.g. RFID via MQTT - Message Queuing Telemetry Transport) and Big data which could use the all provided data to optimize lead times and inventories

5.5 Future work

There remain open points that are left for possible future work and development of a Proof of Concept (PoC). The first open point is that the SC lacks the functionality for all SCAs to add product certificates (currently only suppliers) in the case of transformation of products. Additionally, it is still undefined when to delete product references after the products go to the post supply chain. Some attention should be given also to the issue of certificate validation outside of the BC. One of the possible criticisms of the proposed solution is that of not achieving complete decentralization since due to current EVM limitations the certificate validation mechanism must be implemented off chain. However, this problem may be close to be solved in future via EVM upgrades[20]. Due to time restriction on the thesis delivery dates there was no time to fully develop the concept up to the level of a PoC. For future work, besides the open issues it is left the development of a PoC that would allow the deployment of a working SCM with this architecture. Also, for future work would be the design of an incentive scheme that would allow for the implementation of the "Certificate Validator" function. This functionality should be possible to implement by third parties in the public Ethereum BC. A future PoC requires the implementation of a test framework (in Javascript) to interact with the SC via the Web3 API and the companion browser or IoT software add-ons to facilitate user interaction with the SCM. In addition to validation a PoC would allow to measure operating and information storage costs plus the operational feasibility and business competitivity (in terms of required IT infrastructure).

[20] Se for example the: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-152.md

Annexes

Annex 1 - Supply chain stakeholders

In order to understand the stakeholders needs in the supply chain traceability problem it is important to understand their identity and context in the supply chain. In Figure 3 all the relevant stakeholders were mapped according to their use and impact on the supply chain.

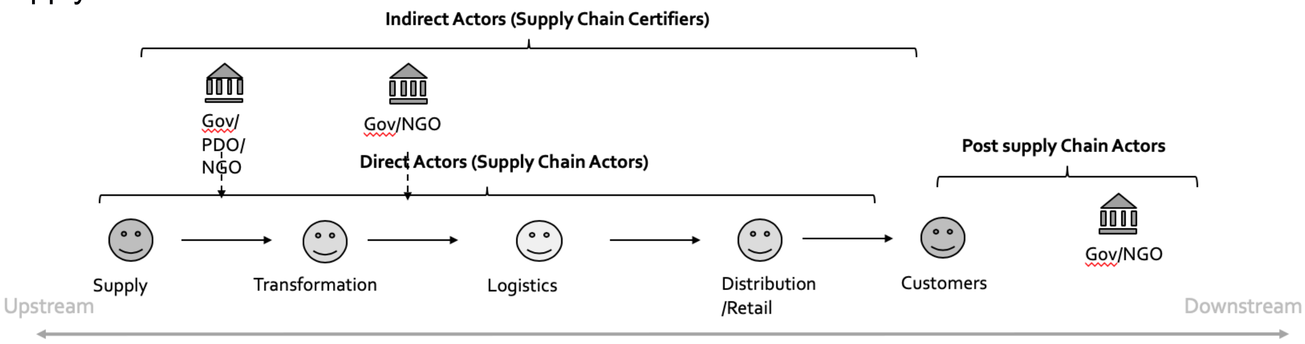


Figure 3 – Supply chain organization

The actual supply chain users can be grouped in categories where they are involved in the same business use case:

- **Supplier:** creates products/goods without any that will become part of the resulting product.
- **Transformation:** creates products/goods that require physical inputs
- **Logistics:** receives, transports and delivers the products
- **Distribution/retail:** purchases and sells the products

The users of the products after the retail/seller are not considered to be actors of the supply chain in the business aspect since they are not directly responsible in the normal functioning of the supply chain and are considered to belong to the Post Supply Chain. These are either end-user consumers or organizations related to consumer interests:

- **Consumers:** purchases and uses the products
- **Consumer groups/Environmental groups:** reviews and influences public opinion on product attributes and impacts
- **Governmental agencies:** Verifies product safety and regulations

There are also groups or organizations that influence the working of the supply chain (require that processes or documentation follow guidelines) but that do not participate directly. In what influences the traceability problem and the certificates it is possible to group the supply chain certifiers according to the type of certificate:

- **Government:** certifies products/goods that are in accordance to governmental regulations
- **PDO:** certifies products/goods in accordance to PDO regulations
- **NGO:** certifies products/goods in accordance to Non-Governmental Organization regulations

Annex 2 - SCA problems

Actor	Weakness/Limitation	Consequent problems	Aspect to improve	Requirements
Supply	Ability to prove globally the origin, authenticity and quality of the products and producers	Counterfeiting Loss of brand equity	Provenance	Register valid SC Actor Register products with information and proof of origin
Transformation	Difficulty to monitor the quality and origin of supplies. Limitations in monitoring the product to the final destination.	Contamination Loss of quality Loss of brand equity	Traceability CoC	Register valid SC Actor Transform products while maintaining certificate traceability Register valid SC Actor Register transfer of ownership
Logistics	Lack of visibility and trust of the transfers of ownership (internal or external).	Delays and theft No attribution of responsibility Interoperability costs	Traceability CoC	Register valid SC Actor Register transfer of ownership Provide visibility to certified product inventory, location, owner
Distribution/ **Retail**	Ability to verify the inventory, origin and authenticity of certified products Lack of visibility and trust of the transfers of ownership (internal or external).	Counterfeiting Misrepresentation of quantities Customer Legal action Loss of brand image	Traceability CoC	
Final customer	No independent confirmation of the quality, origin and sustainability of products	Health and monetary impacts Distrust in business Concern for environment and sustainability	Provenance Traceability Chain of custody	Provide visibility to supply chain trace and certificates.

Table 1- SCA problems

Annex 3 - Requirements

From the previous analysis of SCA problems and the review of published implementations that are described in traceability state of the art (see 2.3.5) a list of requirements was derived. For each requirement a solution is proposed using an applicable technology. Each proposed solution supports a traceability concept.

ID	Requirement	Applicable technology – Proposed solution	Problem that it solves	Supported Concept
1	SCA registration validation and access control	Public Key Infrastructure (PKI) and SCs – uses certificate to establish and maintain assurance of the identity.	**Access control**- only allowed SC actors can interact with SCM. Requires registration/verification of the certificates	Support Traceability
2	SCA sign on operations	Ethereum BC and SC logic– associate EBC addresses to validated identities. Any EBC has to interact by using a signed transaction.	**Impersonation** - The validated participants are required to sign all operations and make proof of their identity	Support Traceability
3	Register products certificates	SC logic and PKI – associate product identifiers with their certificates	**Counterfeiting** - only original products information is introduced into the SCM	Assure Provenance
4	Correct transfer of ownership	SC logic - provides 2-sided transfer of ownership.	**Theft and wrongful delivery** – register of each transfer of ownership is registered	Implement Chain of custody
5	Verify ownership and product certificate validity	SC logic and PKI - to verify the current ownership of a product and if the certificate is valid or has been revoked.	**Product ownership and certificate validation** - requires a check of ownership and if the certificate is valid.	Implement Traceability and assure provenance
6	Transform products	SC logic - use of SC functions to tracks the transformation of certified products	**Certificate and inventory management** – requires that transformed products maintain the certification source.	Support traceability, provenance
7	Product certificate retrieval.	Javascript, SC logic and PKI- use of SCs and an external URL for certificate visibility.	**Standards, health, compliance, brand value** - requires controlled access to the chain of product certificates.	Implement Provenance visibility

Table 2- Proposal requirements

Annex 4 - Architecture workflows

Use case A: Add a new SCA to the supply chain management system

A SCA must firstly authenticate himself before being able to interact with the SCM. For this he must register with a trusted entity that makes sure he has the correct credentials and authorization to interact with the SCM. Following the requirement of having a decentralized system and in accordance to the selected use case (PDO alimentary supply chain) the trusted entities were considered to be the Governmental/Organizational agencies. In the case of a centralized SCM (e.g. over private/consortium BC) another central entity could be chosen, and the certificate validator could also become the certificate issuer (CA).

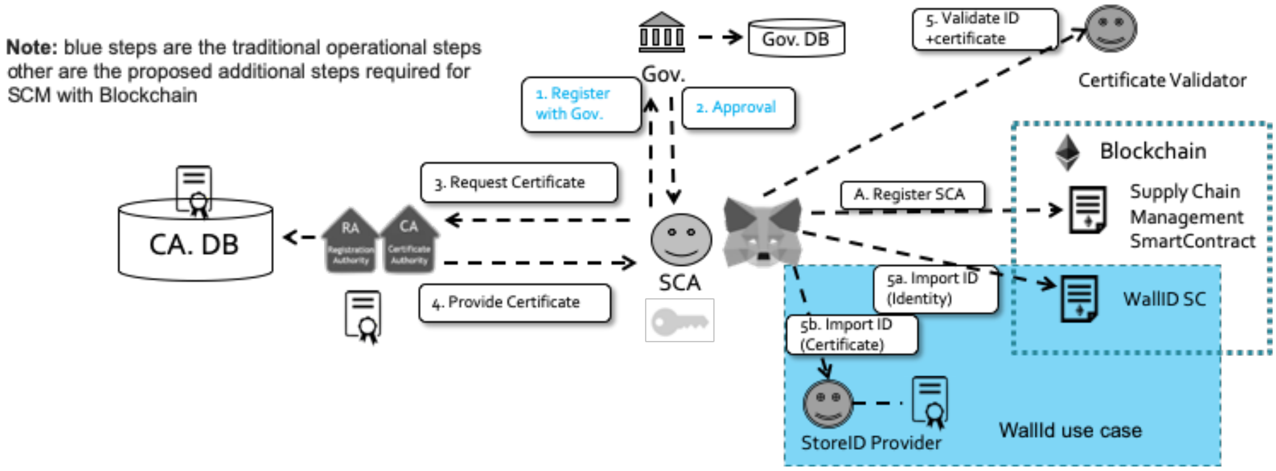
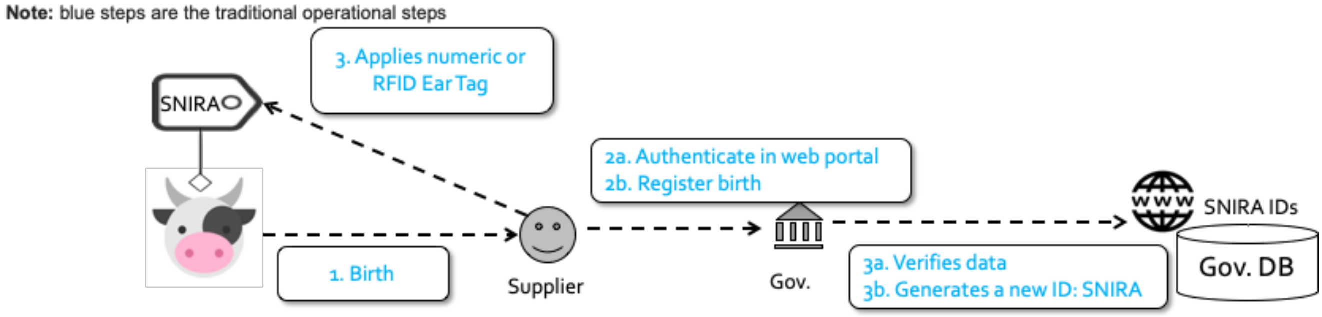

Note: blue steps are the traditional operational steps other are the proposed additional steps required for SCM with Blockchain

Figure 4 – Use case A – Add a new SCA

Use case B1: Add a new product to supply chain


Note: blue steps are the traditional operational steps

SNIRA
3. Applies numeric or RFID Ear Tag
2a. Authenticate in web portal
2b. Register birth
1. Birth
Supplier
Gov.
www
SNIRA IDs
3a. Verifies data
3b. Generates a new ID: SNIRA
Gov. DB

Figure 5 – Use case B1 – Add a new product

Use case B2: request certificate for product

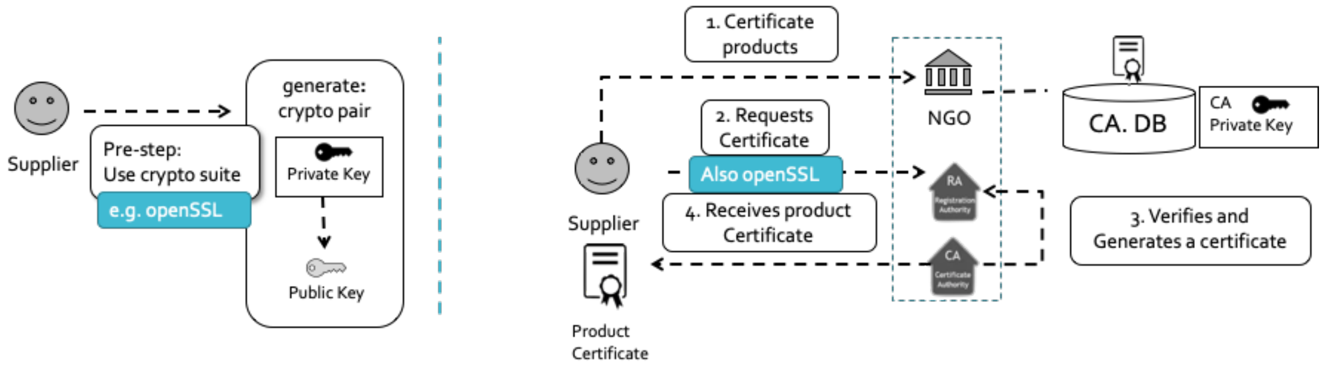


Figure 6 – Use case B2 – Add a new product

Use case B3– Register a new product in SCM

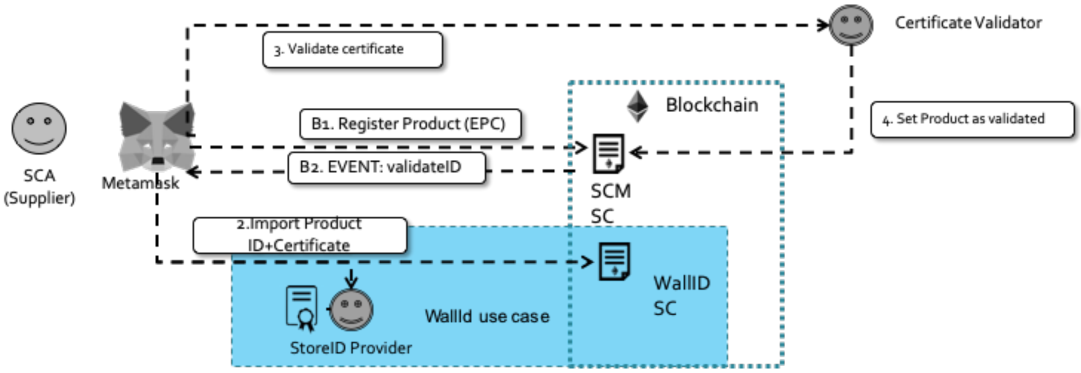


Figure 7 – Use case B3 – Register a new product

Use case C – Get/Set product attributes

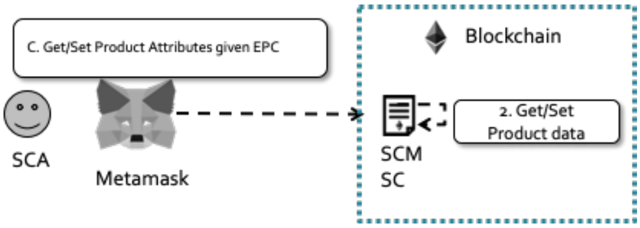


Figure 8 – Use case C – Get/Set product attributes

Use case D – Get product certificate

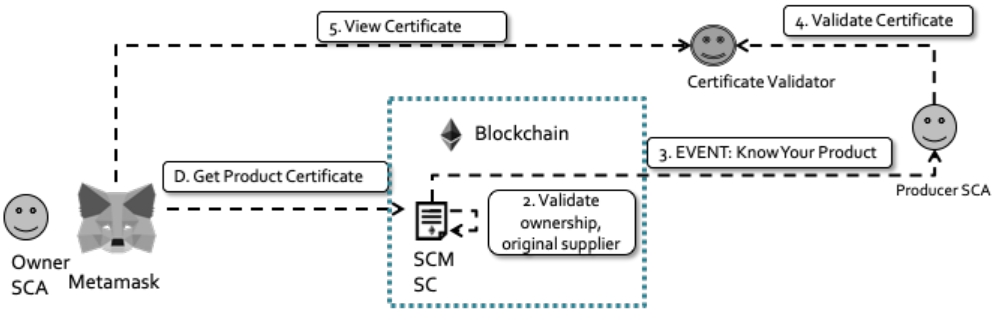


Figure 9 – Use case D – Get product certificate

Use case E– Transform product

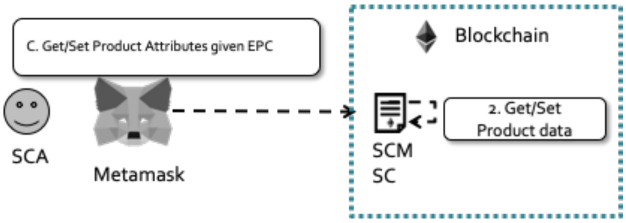


Figure 10 – Use case E – Transform product

Use case F/G– Transfer ownership to/from

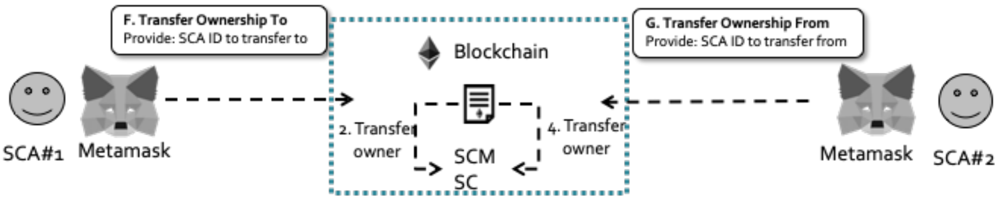


Figure 11 – Use case F/G – Transfer ownership

Annex 5 - Certificate import and validation

In order to import and validate the certificates the SCM solution interacts with the WalliD architecture for 2 use cases: import of certificates into the WalliD store provider and afterwards certificate retrieval from the store provider. These actions are run in sequence with 2 events: ImportID and following the correct import RequestKYC/RequestKYP. The same pattern is used for both the registration of SCAs and the registration of products.

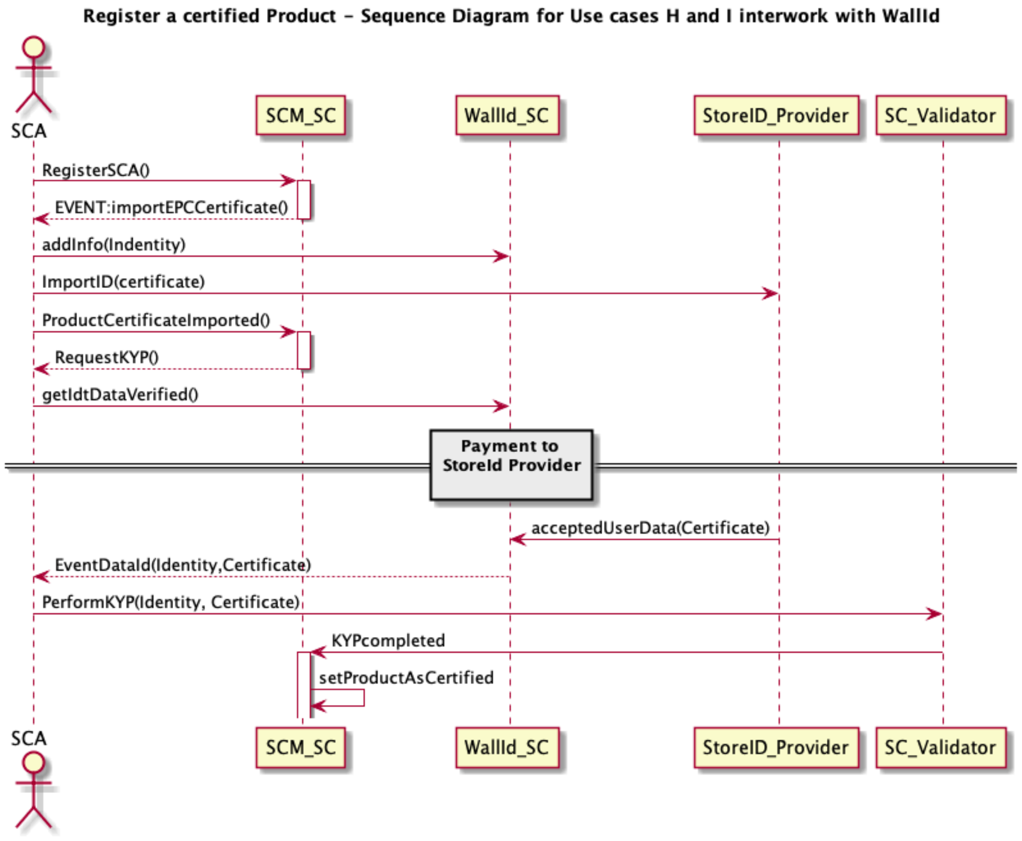

Register a certified Product – Sequence Diagram for Use cases H and I interwork with WallId
SCA
SCM_SC
WallId_SC
StoreID_Provider
SC_Validator
RegisterSCA()
EVENT:importEPCCertificate()
addInfo(Indentity)
ImportID(certificate)
ProductCertificateImported()
RequestKYP()
getIdtDataVerified()
Payment to StoreId Provider
acceptedUserData(Certificate)
EventDataId(Identity,Certificate)
PerformKYP(Identity, Certificate)
KYPcompleted
setProductAsCertified

Figure 12 - Certify product

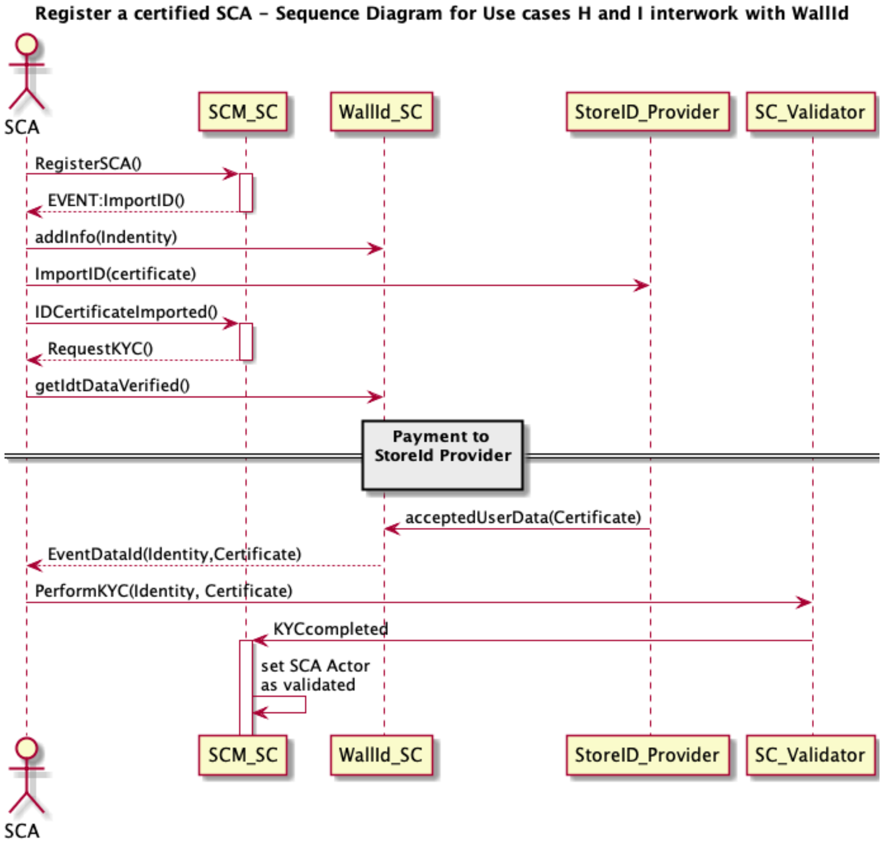
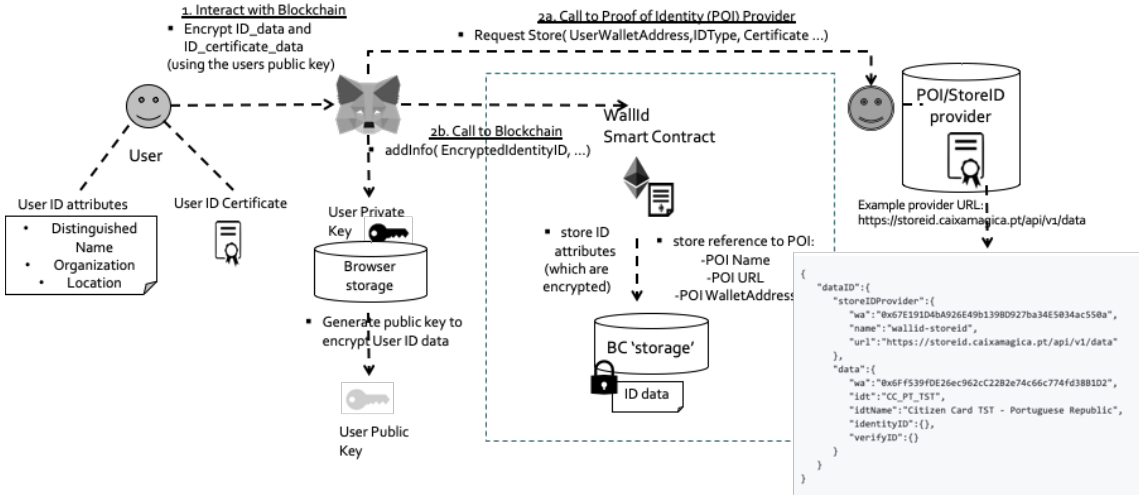

Register a certified SCA – Sequence Diagram for Use cases H and I interwork with WallId
SCA
SCM_SC
WallId_SC
StoreID_Provider
SC_Validator
RegisterSCA()
EVENT:ImportID()
addInfo(Indentity)
ImportID(certificate)
IDCertificateImported()
RequestKYC()
getIdtDataVerified()
Payment to StoreId Provider
acceptedUserData(Certificate)
EventDataId(Identity,Certificate)
PerformKYC(Identity, Certificate)
KYCcompleted
set SCA Actor as validated

Figure 13 - Certify SCA

WallId Import certificate


1. Interact with Blockchain
▪ Encrypt ID_data and ID_certificate_data (using the users public key)
2a. Call to Proof of Identity (POI) Provider
▪ Request Store(UserWalletAddress,IDType, Certificate ...)
User
User ID attributes
• Distinguished Name
• Organization
• Location
User ID Certificate
2b. Call to Blockchain
▪ addInfo(EncryptedIdentityID, ...)
WallId Smart Contract
User Private Key
Browser storage
▪ Generate public key to encrypt User ID data
User Public Key
▪ store ID attributes (which are encrypted)
▪ store reference to POI:
-POI Name
-POI URL
-POI WalletAddress
BC 'storage'
ID data
POI/StoreID provider
Example provider URL: https://storeid.caixamagica.pt/api/v1/data

```
{
   "dataID":{
      "storeIDProvider":{
         "wa":"0x67E191D4bA926E49b139BD927ba34E5034ac550a",
         "name":"wallid-storeid",
         "url":"https://storeid.caixamagica.pt/api/v1/data"
      },
      "data":{
         "wa":"0x6Ff539fDE26ec962cC22B2e74c66c774fd38B1D2",
         "idt":"CC_PT_TST",
         "idtName":"Citizen Card TST - Portuguese Republic",
         "identityID":{},
         "verifyID":{}
      }
   }
}
```

Figure 14 – Import certificate architecture

WallId Validate certificate

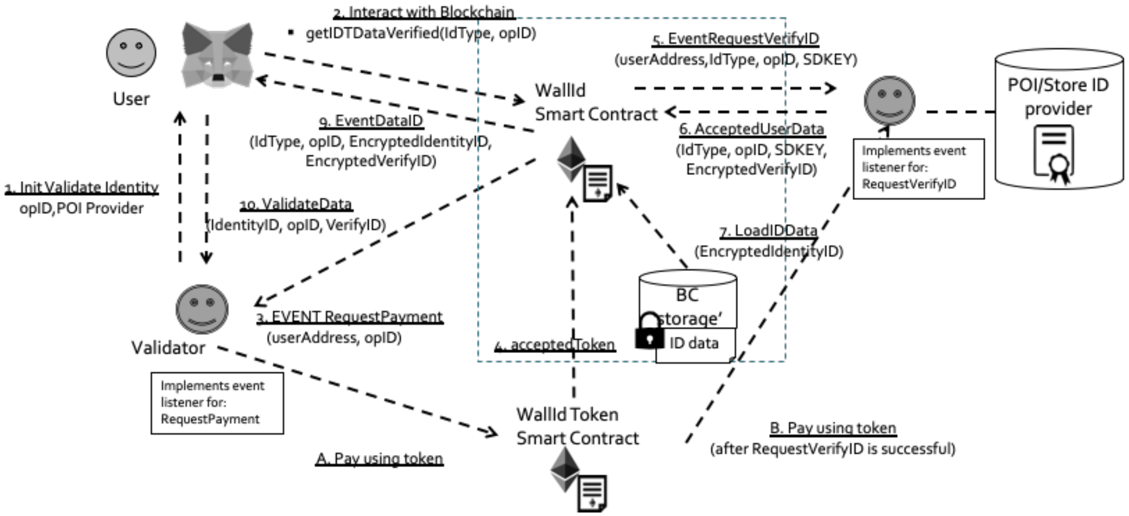
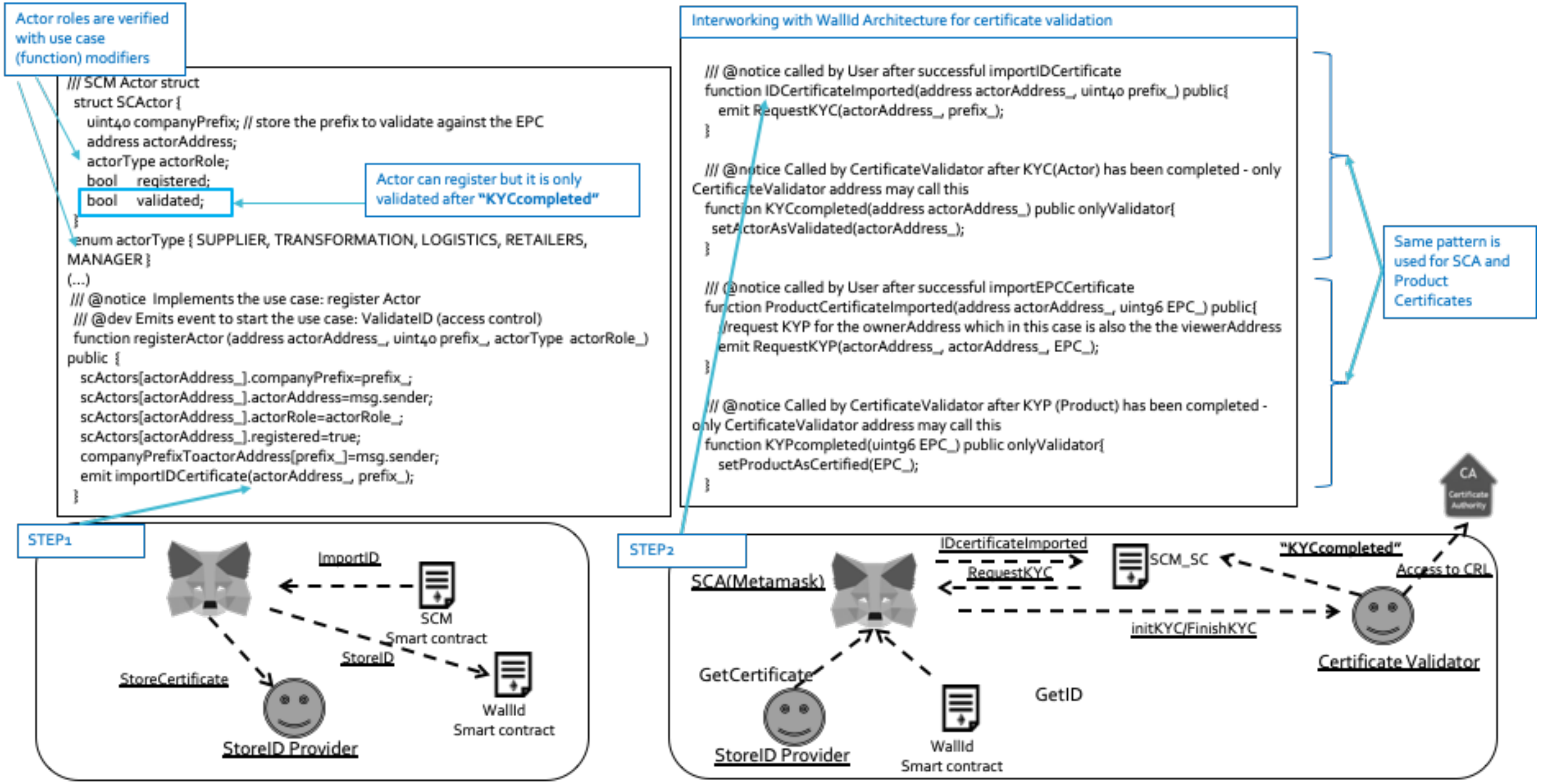

User
2. Interact with Blockchain
▪ getIDTDataVerified(IdType, opID)
5. EventRequestVerifyID
(userAddress,IdType, opID, SDKEY)
WallId Smart Contract
POI/Store ID provider
9. EventDataID
(IdType, opID, EncryptedIdentityID, EncryptedVerifyID)
6. AcceptedUserData
(IdType, opID, SDKEY, EncryptedVerifyID)
Implements event listener for: RequestVerifyID
1. Init Validate Identity
opID,POI Provider
10. ValidateData
(IdentityID, opID, VerifyID)
7. LoadIDData
(EncryptedIdentityID)
BC 'storage'
ID data
Validator
3. EVENT RequestPayment
(userAddress, opID)
4. acceptedToken
Implements event listener for: RequestPayment
WallId Token Smart Contract
A. Pay using token
B. Pay using token
(after RequestVerifyID is successful)

Figure 15 – Validate certificate architecture

Annex 6 - Ethereum Smart Contract

The most updated version and complete SC code is available at: https://github.com/prgazevedo/DLT_Masters/tree/master/SCM_SmartContracts. This code compiles for solidity version 0.5.11.

Detail on SCA and Validation



Actor roles are verified with use case (function) modifiers

```
/// SCM Actor struct
 struct SCActor {
   uint40 companyPrefix; // store the prefix to validate against the EPC
   address actorAddress;
   actorType actorRole;
   bool    registered;
   bool    validated;
 }
enum actorType { SUPPLIER, TRANSFORMATION, LOGISTICS, RETAILERS,
MANAGER }
(...)
 /// @notice  Implements the use case: register Actor
  /// @dev Emits event to start the use case: ValidateID (access control)
 function registerActor (address actorAddress_, uint40 prefix_, actorType  actorRole_)
public {
   scActors[actorAddress_].companyPrefix=prefix_;
   scActors[actorAddress_].actorAddress=msg.sender;
   scActors[actorAddress_].actorRole=actorRole_;
   scActors[actorAddress_].registered=true;
   companyPrefixToactorAddress[prefix_]=msg.sender;
   emit importIDCertificate(actorAddress_, prefix_);
 }
```

Actor can register but it is only validated after **"KYCcompleted"**

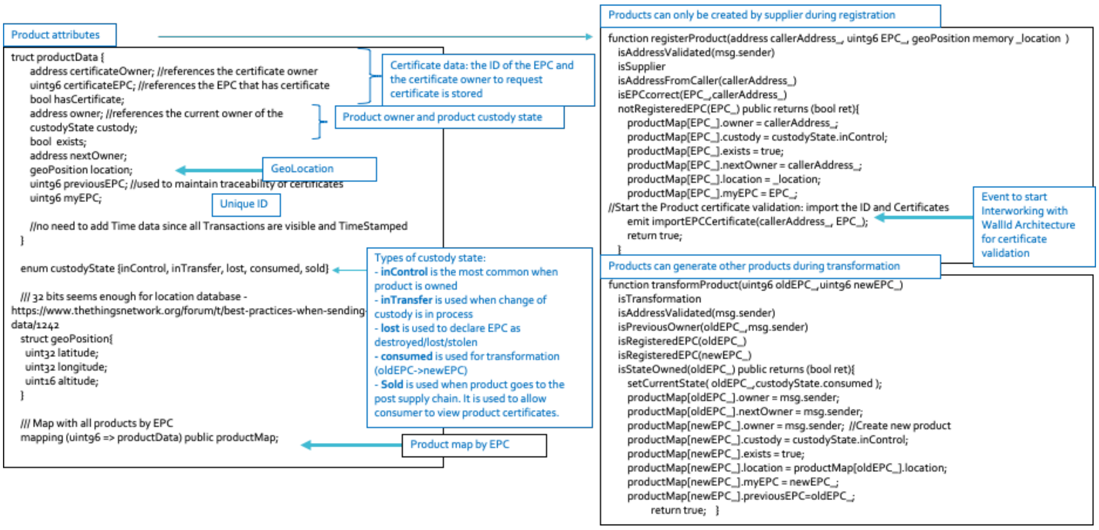
Interworking with WallId Architecture for certificate validation

```
  /// @notice called by User after successful importIDCertificate
  function IDCertificateImported(address actorAddress_, uint40 prefix_) public{
    emit RequestKYC(actorAddress_, prefix_);
  }

  /// @notice Called by CertificateValidator after KYC(Actor) has been completed - only
CertificateValidator address may call this
  function KYCcompleted(address actorAddress_) public onlyValidator{
   setActorAsValidated(actorAddress_);
  }

  /// @notice called by User after successful importEPCCertificate
  function ProductCertificateImported(address actorAddress_, uint96 EPC_) public{
    //request KYP for the ownerAddress which in this case is also the the viewerAddress
    emit RequestKYP(actorAddress_, actorAddress_, EPC_);
  }

  /// @notice Called by CertificateValidator after KYP (Product) has been completed -
only CertificateValidator address may call this
  function KYPcompleted(uint96 EPC_) public onlyValidator{
    setProductAsCertified(EPC_);
  }
```

Same pattern is used for SCA and Product Certificates

Figure 16 – SCA registration and validation

Detail on Product registration and transformation



Certificate data: the ID of the EPC and the certificate owner to request certificate is stored

Product owner and product custody state

```
truct productData {
    address certificateOwner; //references the certificate owner
    uint96 certificateEPC; //references the EPC that has certificate
    bool hasCertificate;
    address owner; //references the current owner of the
    custodyState custody;
    bool  exists;
    address nextOwner;
    geoPosition location;
    uint96 previousEPC; //used to maintain traceability of certificates
    uint96 myEPC;

    //no need to add Time data since all Transactions are visible and TimeStamped
  }

  enum custodyState {inControl, inTransfer, lost, consumed, sold}

  /// 32 bits seems enough for location database -
https://www.thethingsnetwork.org/forum/t/best-practices-when-sending-
data/1242
  struct geoPosition{
   uint32 latitude;
   uint32 longitude;
   uint16 altitude;
  }

  /// Map with all products by EPC
  mapping (uint96 => productData) public productMap;
```

Types of custody state:
- **inControl** is the most common when product is owned
- **inTransfer** is used when change of custody is in process
- **lost** is used to declare EPC as destroyed/lost/stolen
- **consumed** is used for transformation (oldEPC->newEPC)
- **Sold** is used when product goes to the post supply chain. It is used to allow consumer to view product certificates.

Products can only be created by supplier during registration

```
function registerProduct(address callerAddress_, uint96 EPC_, geoPosition memory _location )
  isAddressValidated(msg.sender)
  isSupplier
  isAddressFromCaller(callerAddress_)
  isEPCcorrect(EPC_,callerAddress_)
  notRegisteredEPC(EPC_) public returns (bool ret){
    productMap[EPC_].owner = callerAddress_;
    productMap[EPC_].custody = custodyState.inControl;
    productMap[EPC_].exists = true;
    productMap[EPC_].nextOwner = callerAddress_;
    productMap[EPC_].location = _location;
    productMap[EPC_].myEPC = EPC_;
//Start the Product certificate validation: import the ID and Certificates
    emit importEPCCertificate(callerAddress_, EPC_);
    return true;
  }
```

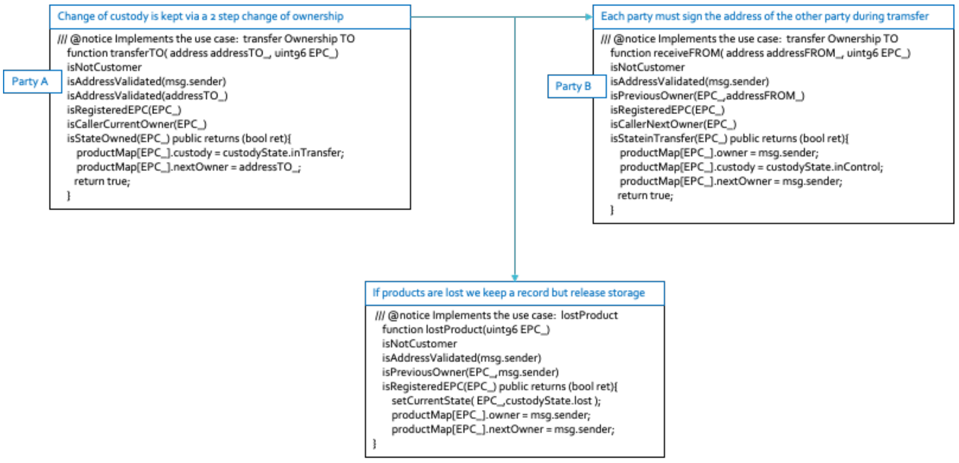
Event to start Interworking with WallId Architecture for certificate validation

Products can generate other products during transformation

```
function transformProduct(uint96 oldEPC_,uint96 newEPC_)
  isTransformation
  isAddressValidated(msg.sender)
  isPreviousOwner(oldEPC_,msg.sender)
  isRegisteredEPC(oldEPC_)
  isRegisteredEPC(newEPC_)
  isStateOwned(oldEPC_) public returns (bool ret){
    setCurrentState( oldEPC_,custodyState.consumed );
    productMap[oldEPC_].owner = msg.sender;
    productMap[oldEPC_].nextOwner = msg.sender;
    productMap[newEPC_].owner = msg.sender;  //Create new product
    productMap[newEPC_].custody = custodyState.inControl;
    productMap[newEPC_].exists = true;
    productMap[newEPC_].location = productMap[oldEPC_].location;
    productMap[newEPC_].myEPC = newEPC_;
    productMap[newEPC_].previousEPC=oldEPC_;
        return true;   }
```

Figure 17 – Product registration and transformation

Detail on transfer of custody and loss of Product



Change of custody is kept via a 2 step change of ownership

```
/// @notice Implements the use case:  transfer Ownership TO
  function transferTO( address addressTO_, uint96 EPC_)
  isNotCustomer
  isAddressValidated(msg.sender)
  isAddressValidated(addressTO_)
  isRegisteredEPC(EPC_)
  isCallerCurrentOwner(EPC_)
  isStateOwned(EPC_) public returns (bool ret){
    productMap[EPC_].custody = custodyState.inTransfer;
    productMap[EPC_].nextOwner = addressTO_;
   return true;
  }
```

Each party must sign the address of the other party during tramsfer

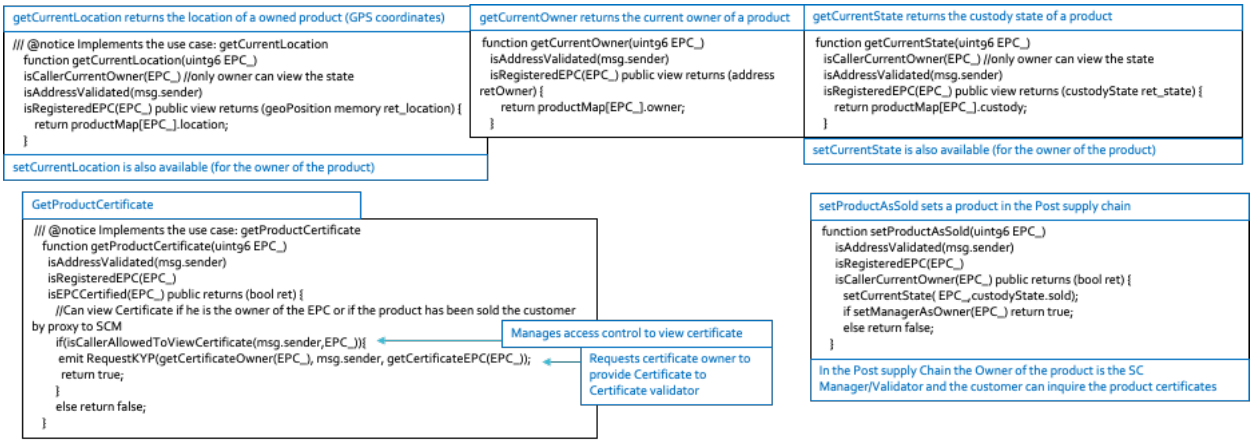
```
/// @notice Implements the use case:  transfer Ownership TO
  function receiveFROM( address addressFROM_, uint96 EPC_)
  isNotCustomer
  isAddressValidated(msg.sender)
  isPreviousOwner(EPC_,addressFROM_)
  isRegisteredEPC(EPC_)
  isCallerNextOwner(EPC_)
  isStateinTransfer(EPC_) public returns (bool ret){
    productMap[EPC_].owner = msg.sender;
    productMap[EPC_].custody = custodyState.inControl;
    productMap[EPC_].nextOwner = msg.sender;
   return true;
  }
```

If products are lost we keep a record but release storage

```
/// @notice Implements the use case:  lostProduct
  function lostProduct(uint96 EPC_)
  isNotCustomer
  isAddressValidated(msg.sender)
  isPreviousOwner(EPC_,msg.sender)
  isRegisteredEPC(EPC_) public returns (bool ret){
    setCurrentState( EPC_,custodyState.lost );
    productMap[EPC_].owner = msg.sender;
    productMap[EPC_].nextOwner = msg.sender;
}
```

Figure 18 – Transfer of custody

5.5.1 Detail on Get/Set functions



getCurrentLocation returns the location of a owned product (GPS coordinates)

```
/// @notice Implements the use case: getCurrentLocation
  function getCurrentLocation(uint96 EPC_)
  isCallerCurrentOwner(EPC_) //only owner can view the state
  isAddressValidated(msg.sender)
  isRegisteredEPC(EPC_) public view returns (geoPosition memory ret_location) {
    return productMap[EPC_].location;
  }
```

setCurrentLocation is also available (for the owner of the product)

getCurrentOwner returns the current owner of a product

```
 function getCurrentOwner(uint96 EPC_)
  isAddressValidated(msg.sender)
  isRegisteredEPC(EPC_) public view returns (address
retOwner) {
    return productMap[EPC_].owner;
  }
```

getCurrentState returns the custody state of a product

```
function getCurrentState(uint96 EPC_)
  isCallerCurrentOwner(EPC_) //only owner can view the state
  isAddressValidated(msg.sender)
  isRegisteredEPC(EPC_) public view returns (custodyState ret_state) {
    return productMap[EPC_].custody;
  }
```

setCurrentState is also available (for the owner of the product)

GetProductCertificate

```
/// @notice Implements the use case: getProductCertificate
  function getProductCertificate(uint96 EPC_)
   isAddressValidated(msg.sender)
   isRegisteredEPC(EPC_)
   isEPCCertified(EPC_) public returns (bool ret) {
     //Can view Certificate if he is the owner of the EPC or if the product has been sold the customer
by proxy to SCM
     if(isCallerAllowedToViewCertificate(msg.sender,EPC_)){
      emit RequestKYP(getCertificateOwner(EPC_), msg.sender, getCertificateEPC(EPC_));
       return true;
     }
     else return false;
  }
```

Manages access control to view certificate

Requests certificate owner to provide Certificate to Certificate validator

setProductAsSold sets a product in the Post supply chain

```
function setProductAsSold(uint96 EPC_)
   isAddressValidated(msg.sender)
   isRegisteredEPC(EPC_)
   isCallerCurrentOwner(EPC_) public returns (bool ret) {
     setCurrentState( EPC_,custodyState.sold);
     if setManagerAsOwner(EPC_) return true;
     else return false;
  }
```

In the Post supply Chain the Owner of the product is the SC Manager/Validator and the customer can inquire the product certificates

Figure 19 – Get/Set functionality

Annex 7 - Public Key Infrastructure

The main components of a PKI are Public Key Certificates and Certificate Authorities.

PKC - public key certificate, also known as a digital certificate or identity certificate, is an electronic document used to prove the ownership of a public key

The certificate includes information on the public key of the subject, the identity of the subject and the digital signature of the issuer that has verified the certificate's contents.

In a PKI there are 3 main roles and procedures for a certificate: authenticating the identity carried out by the RA (Registration Authority), issuance of certificate carried out by the CA (certification authority) and validation of certificates carried out by the VA (validation authority. A distrusting 3rd party can trust the subscriber when the digital signature (PKC) is valid and the 3rd party trusts the issuer (CA). A certificate binds the public key with the identity (distinguished name) of an entity (subscriber).

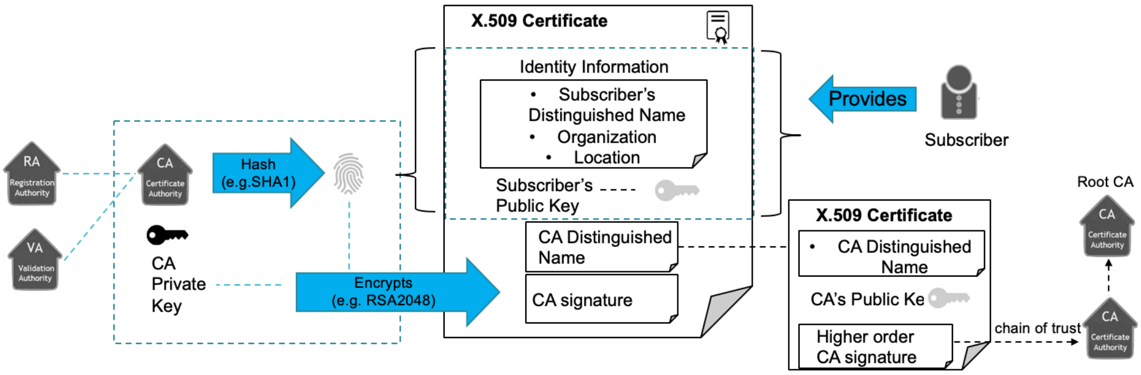
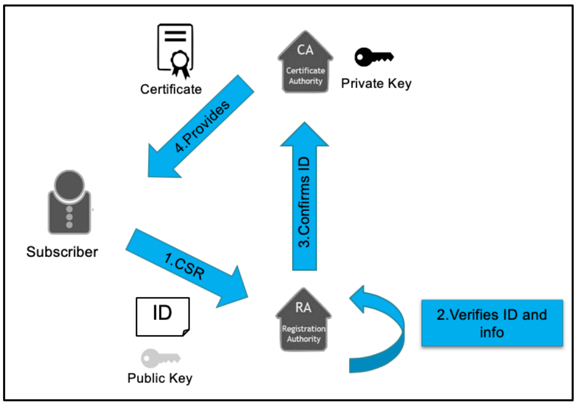


Figure 20 – X.509 certificate

Registration and certification procedures: a Registration Authority (RA) receives a request for the digital certificate (CSR) from the subscriber that needs a certificate. The RA verifies the identity of the user and the information provided. After verification it triggers the CA to sign a certificate based on that information using the information provided by the user and it's private key. The certificates and the CA's public keys are made publicly available.



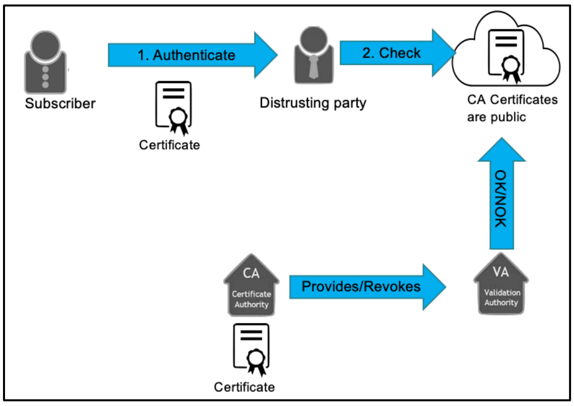


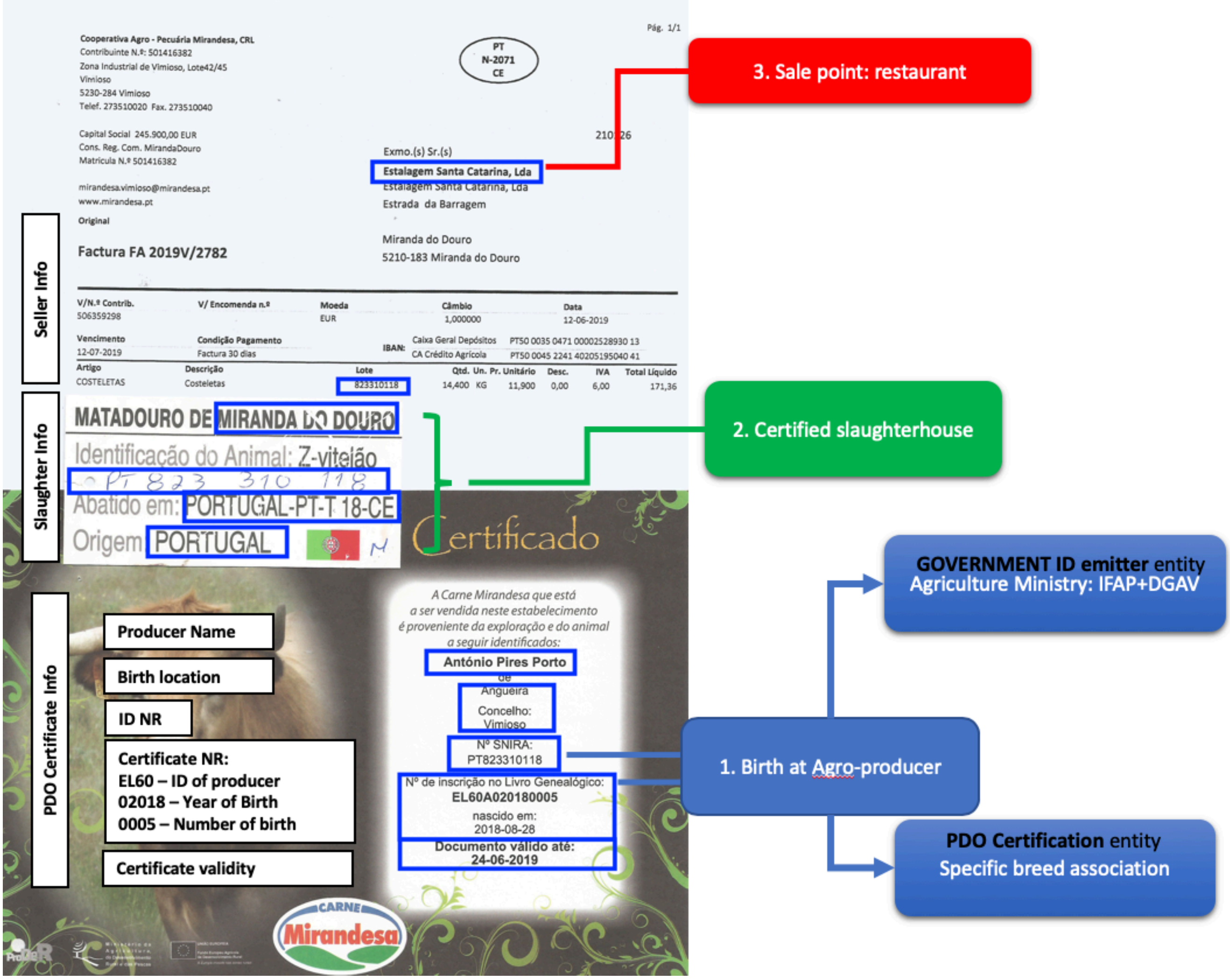
Figure 21 – PKI certification procedure

Figure 22 – PKI validation procedure

The validation step is performed online by the Validation Authority (VA). It is possible for a Certification Authority (CA) to merge all 3 functionalities.

Annex 8 - Use case data

In the next figure is presented the linkage between the livestock (bovine) government assigned ID and the PDO organization assigned certificate ID. These 3 documents presented were collected at the local supply chain exemplify the main data and attributes that are required to establish traceability for this use case.



Pág. 1/1

Cooperativa Agro - Pecuária Mirandesa, CRL
Contribuinte N.º: 501416382
Zona Industrial de Vimioso, Lote42/45
Vimioso
5230-284 Vimioso
Telef. 273510020 Fax. 273510040

Capital Social 245.900,00 EUR
Cons. Reg. Com. MirandaDouro
Matricula N.º 501416382

mirandesa.vimioso@mirandesa.pt
www.mirandesa.pt

Original

Factura FA 2019V/2782

PT
N-2071
CE

210 26

Exmo.(s) Sr.(s)
Estalagem Santa Catarina, Lda
Estalagem Santa Catarina, Lda
Estrada da Barragem

Miranda do Douro
5210-183 Miranda do Douro

V/N.º Contrib.	V/ Encomenda n.º	Moeda	Câmbio	Data
506359298		EUR	1,000000	12-06-2019

Vencimento	Condição Pagamento	IBAN:	
12-07-2019	Factura 30 dias	Caixa Geral Depósitos CA Crédito Agrícola	PT50 0035 0471 00002528930 13 PT50 0045 2241 40205195040 41

Artigo	Descrição	Lote	Qtd.	Un.	Pr. Unitário	Desc.	IVA	Total Líquido
COSTELETAS	Costeletas	823310118	14,400	KG	11,900	0,00	6,00	171,36

MATADOURO DE MIRANDA DO DOURO
Identificação do Animal: Z-vitelão
PT 823 310 118
Abatido em: PORTUGAL-PT-T 18-CE
Origem PORTUGAL

Certificado

A Carne Mirandesa que está a ser vendida neste estabelecimento é proveniente da exploração e do animal a seguir identificados:
António Pires Porto
de
Angueira
Concelho: Vimioso
Nº SNIRA: PT823310118
Nº de inscrição no Livro Genealógico: EL60A020180005
nascido em: 2018-08-28
Documento válido até: 24-06-2019

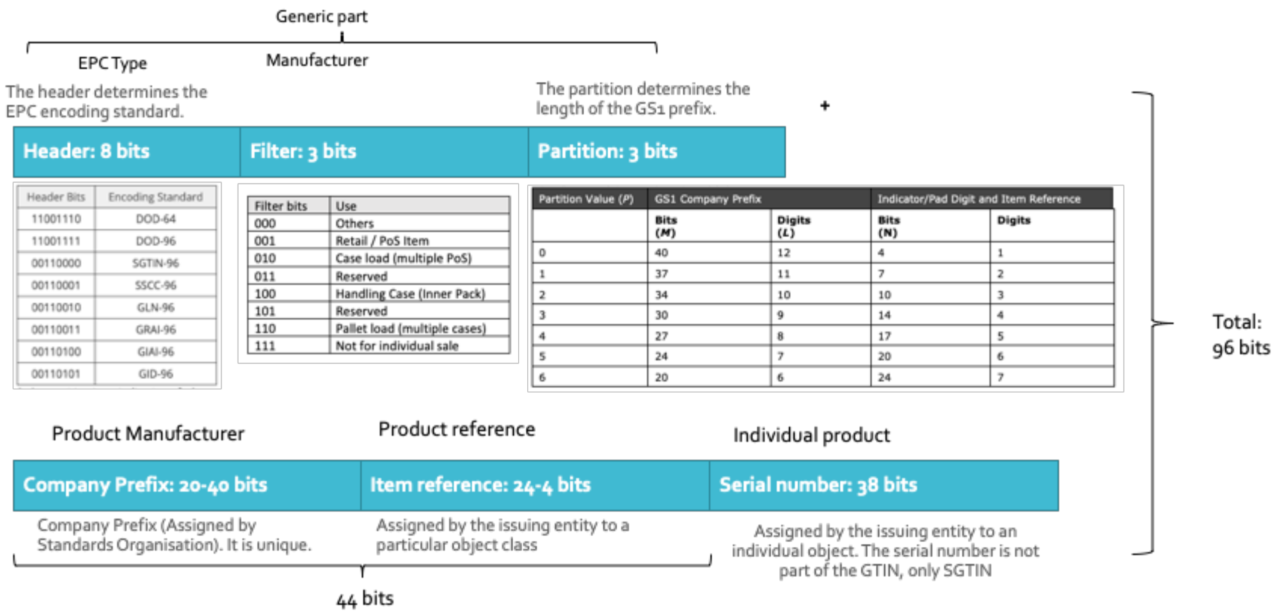
Figure 23 – Use case certificate

The government assigned ID of Bovine (SNIRA ID) is attributed at birth by DGAV and stored in Sistema Nacional de Informação e Registo Animal (SNIRA)by IFAP (more details at https://www.ifap.pt/web/guest/snira-regras). At the same time of birth the genealogy of calf (bull ID and Cow ID) is recorded by the PDO organization (in this case the “Mirandesa” association) and is also recorded in (SNIRA) by IFAP[21]. When the bovine is ready, it is then sent to a certified slaughterhouse where the registry of both SNIRA ID and certificate linkage is assured. At this time the carcass is assigned a EPC code and a physical tag with the ID of the slaughter house (PT-T 18-CE). The carcass is then shipped to the retailers or seller of the end product that can be either a consumer beef produce (in the case of butcher or supermarket) or a prepared meal at a restaurant or hotel. Each of the SCAs receive the PDO paper certificate together with the invoice on each carcass.

[21] More details in https://tradicional.dgadr.gov.pt/pt/cat/carne/carne-de-bovino/235-carne-mirandesa-dop

Annex 9 - EPC detail

In order to have unique global identification at instance level granularity a EPC: Electronic Product Code – GS1 SGTIN (Serialized GTIN) or SSCC (Serial Shipping Container Code) identifier is required The next figure presents the different fields in a SGTIN EPC


Generic part
EPC Type
Manufacturer
The header determines the EPC encoding standard.
The partition determines the length of the GS1 prefix.
+
Header: 8 bits
Filter: 3 bits
Partition: 3 bits
Total: 96 bits
Product Manufacturer
Product reference
Individual product
Company Prefix: 20-40 bits
Item reference: 24-4 bits
Serial number: 38 bits
Company Prefix (Assigned by Standards Organisation). It is unique.
Assigned by the issuing entity to a particular object class
Assigned by the issuing entity to an individual object. The serial number is not part of the GTIN, only SGTIN
44 bits

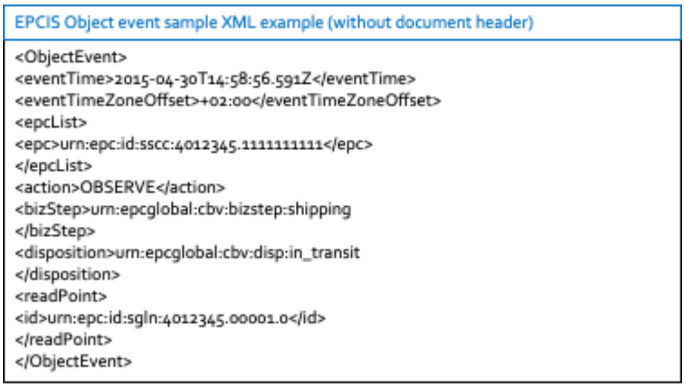
Header Bits	Encoding Standard
11001110	DOD-64
11001111	DOD-96
00110000	SGTIN-96
00110001	SSCC-96
00110010	GLN-96
00110011	GRAI-96
00110100	GIAI-96
00110101	GID-96

Filter bits	Use
000	Others
001	Retail / PoS Item
010	Case load (multiple PoS)
011	Reserved
100	Handling Case (Inner Pack)
101	Reserved
110	Pallet load (multiple cases)
111	Not for individual sale

Partition Value (*P*)	GS1 Company Prefix		Indicator/Pad Digit and Item Reference	
	Bits (*M*)	**Digits (*L*)**	**Bits (N)**	**Digits**
0	40	12	4	1
1	37	11	7	2
2	34	10	10	3
3	30	9	14	4
4	27	8	17	5
5	24	7	20	6
6	20	6	24	7

Figure 24 – EPC structure

In the case of SGTIN it is composed of a GTIN (Global Trade Identification Number) plus a serial ID for unique identification of each product. SSCC is also EPC and is similar to SGTIN but is it is mostly used for identifying shipping units uniquely, for example a pallet or handling unit. When EPC codes are transmitted into traditional centralized supply chain systems it is generally within the framework of Electronic Product Code Information Services (GS1 EPCIS standard) a standard used to create and share event data collected along the 4 dimensions: what, when, where, why for trade objects. This data standard follows the framework: identify (e.g. GS1 EPC), capture (e.g. using barcodes or RFID) and share (e.g. via SOAP/XML) and is applied regularly within logistics companies supply chain systems. However as mentioned by Tröger, R., & Alt, R. (2017) the volume of data that is generated in single company EPCIS SCM systems although still under the terabyte it is rising and progressively necessitating cloud and big data. The volume of data is consequence of the verbosity of the standard (XML) as can be viewed in the excerpt below.


EPCIS Object event sample XML example (without document header)

```
<ObjectEvent>
<eventTime>2015-04-30T14:58:56.591Z</eventTime>
<eventTimeZoneOffset>+02:00</eventTimeZoneOffset>
<epcList>
<epc>urn:epc:id:sscc:4012345.1111111111</epc>
</epcList>
<action>OBSERVE</action>
<bizStep>urn:epcglobal:cbv:bizstep:shipping
</bizStep>
<disposition>urn:epcglobal:cbv:disp:in_transit
</disposition>
<readPoint>
<id>urn:epc:id:sgln:4012345.00001.0</id>
</readPoint>
</ObjectEvent>
```

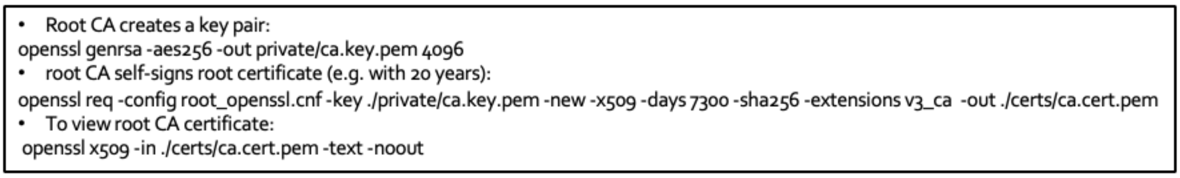
Figure 25 – EPCIS XML sample

It is then clear that the EPCIS data format is not suitable for BC and this is thus a further reason to use a much more succinct representation in the tokenization of products as single EPCs in the proposed SCM SC.

Annex 10 - PKI setup

In order to provide products certificates a PKI needs to be setup. It is recommended that an hierarchical PKI structure is setup in order to improve security (e.g. by having the root CA offline) and distribute responsibility. The proposed PKI for the use case shall have 3 levels the root CA, intermediate CA and end user. The root CA needs to be a most trusted entity, in this case it can be the Portuguese governmental institution IFAP (Instituto de Financiamento da Agricultura e Pescas) which is ruled in Portugal by the Agriculture Ministry. The intermediate CA needs to be a trusted certifier entity that is verified and trusted by the IFAP, in this case the PDO association and certificer “Mirandesa”. The end user shall be the certificate requester and in the sample use case is “AgroGranjo” which is the producer/supplier in the supply chain. In order to establish the PKI each CA must validate and sign certificates in a chain of trust as follows. In order to implement the PKI and generate the certificates openSSL application was used. The openssl program is a vast library with a big number of commands, each of which often with many options and arguments. Many commands use an external configuration file where the user specifies a configuration file.

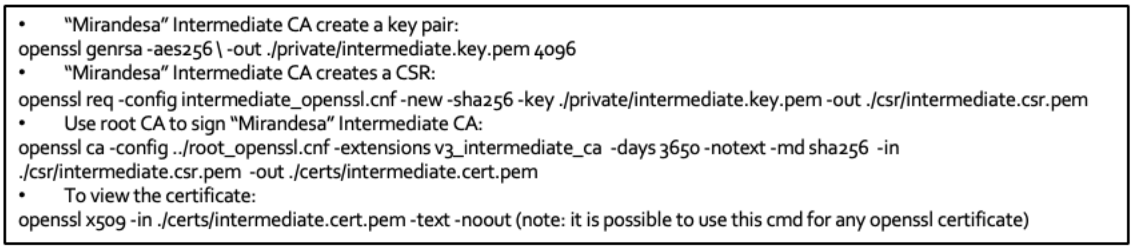
To establish the PKI we establishing the root CA, next the intermediate CA and finally the Producer certificate requests. For CA root establishment the entity responsible needs to run following commands.



```
•   Root CA creates a key pair:
openssl genrsa -aes256 -out private/ca.key.pem 4096
•   root CA self-signs root certificate (e.g. with 20 years):
openssl req -config root_openssl.cnf -key ./private/ca.key.pem -new -x509 -days 7300 -sha256 -extensions v3_ca  -out ./certs/ca.cert.pem
•   To view root CA certificate:
 openssl x509 -in ./certs/ca.cert.pem -text -noout
```

Figure 26 – root CA certificate commands

For intermediate CA establishment we need to run following commands:



```
•       “Mirandesa” Intermediate CA create a key pair:
openssl genrsa -aes256 \ -out ./private/intermediate.key.pem 4096
•       “Mirandesa” Intermediate CA creates a CSR:
openssl req -config intermediate_openssl.cnf -new -sha256 -key ./private/intermediate.key.pem -out ./csr/intermediate.csr.pem
•       Use root CA to sign “Mirandesa” Intermediate CA:
openssl ca -config ../root_openssl.cnf -extensions v3_intermediate_ca  -days 3650 -notext -md sha256  -in
./csr/intermediate.csr.pem  -out ./certs/intermediate.cert.pem
•       To view the certificate:
openssl x509 -in ./certs/intermediate.cert.pem -text -noout (note: it is possible to use this cmd for any openssl certificate)
```

Figure 27 – intermediate CA certificate commands

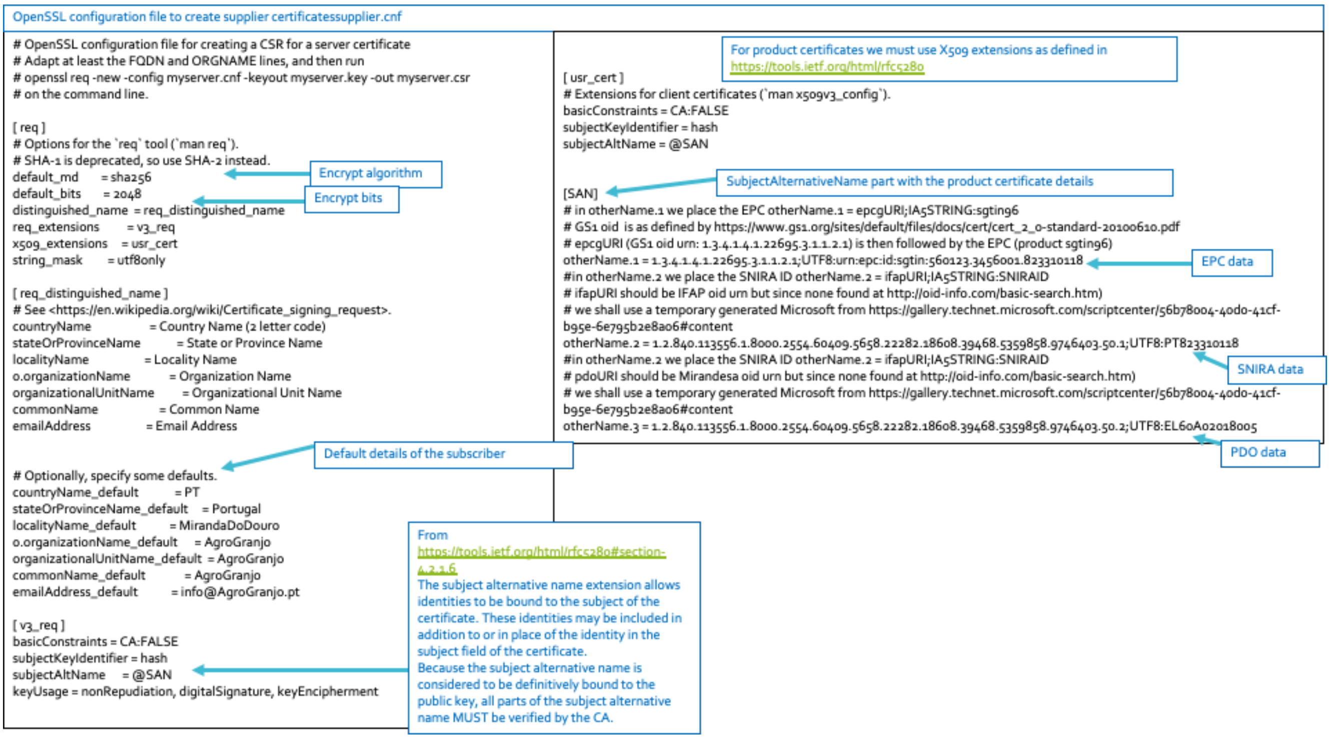
The resulting intermediate.cert.pem will be used to sign the product certificate after a certificate signing request is sent from the end user “Agrogranjo”

Product certificate generation

As described in order to univocally associate the PDO certificate with the product identification the digital certificate should include: a EPC global identifier, the governmental identifier and the PDO identifier. A sample EPC global identifier for the use case can be created[22] to a Tag URI: urn:epc:tag:sgtin-96: 2.560123.3456001.823310118 or pure URI: urn:epc:id:sgtin: 560123.3456001.823310118 which is a valid global product identifier that can be used in any supply chain or EPCIS system. For the case of the bovine PDO we must add the SNIRA ID: PT823310118 and the Genealogy ID: EL60A02018005. The validity of the digital certificate should follow the rules of the physical certificate (e.g. 15 days). In order

[22] EPC converter at http://convert.erfideo.com/Home/

to use X509 extensions (as defined in OpenSSL X509 V3) we use a configuration file for the CA authority (the Mirandesa organization issuing PDO certificates on their products). The Producer “Agrogranjo” generates a certificate request using: *open ssl genrsa -aes256 \ -out ./private/Supplier.key.pem 4096*. In order to create the Product CSR it is pratical to use a configuration file which includes the EPC Tag URI/SNIRA ID/Genealogy ID as follows.



OpenSSL configuration file to create supplier certificatessupplier.cnf

```
# OpenSSL configuration file for creating a CSR for a server certificate
# Adapt at least the FQDN and ORGNAME lines, and then run
# openssl req -new -config myserver.cnf -keyout myserver.key -out myserver.csr
# on the command line.

[ req ]
# Options for the `req` tool (`man req`).
# SHA-1 is deprecated, so use SHA-2 instead.
default_md      = sha256
default_bits     = 2048
distinguished_name = req_distinguished_name
req_extensions       = v3_req
x509_extensions    = usr_cert
string_mask       = utf8only

[ req_distinguished_name ]
# See <https://en.wikipedia.org/wiki/Certificate_signing_request>.
countryName               = Country Name (2 letter code)
stateOrProvinceName          = State or Province Name
localityName              = Locality Name
o.organizationName          = Organization Name
organizationalUnitName        = Organizational Unit Name
commonName                = Common Name
emailAddress              = Email Address


# Optionally, specify some defaults.
countryName_default          = PT
stateOrProvinceName_default    = Portugal
localityName_default         = MirandaDoDouro
o.organizationName_default     = AgroGranjo
organizationalUnitName_default  = AgroGranjo
commonName_default           = AgroGranjo
emailAddress_default         = info@AgroGranjo.pt

[ v3_req ]
basicConstraints = CA:FALSE
subjectKeyIdentifier = hash
subjectAltName     = @SAN
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
```

```
[ usr_cert ]
# Extensions for client certificates (`man x509v3_config`).
basicConstraints = CA:FALSE
subjectKeyIdentifier = hash
subjectAltName = @SAN

[SAN]
# in otherName.1 we place the EPC otherName.1 = epcgURI;IA5STRING:sgting6
# GS1 oid  is as defined by https://www.gs1.org/sites/default/files/docs/cert/cert_2_0-standard-20100610.pdf
# epcgURI (GS1 oid urn: 1.3.4.1.4.1.22695.3.1.1.2.1) is then followed by the EPC (product sgting6)
otherName.1 = 1.3.4.1.4.1.22695.3.1.1.2.1;UTF8:urn:epc:id:sgtin:560123.3456001.823310118
#in otherName.2 we place the SNIRA ID otherName.2 = ifapURI;IA5STRING:SNIRAID
# ifapURI should be IFAP oid urn but since none found at http://oid-info.com/basic-search.htm)
# we shall use a temporary generated Microsoft from https://gallery.technet.microsoft.com/scriptcenter/56b78004-40d0-41cf-
b95e-6e795b2e8a06#content
otherName.2 = 1.2.840.113556.1.8000.2554.60409.5658.22282.18608.39468.5359858.9746403.50.1;UTF8:PT823310118
#in otherName.2 we place the SNIRA ID otherName.2 = ifapURI;IA5STRING:SNIRAID
# pdoURI should be Mirandesa oid urn but since none found at http://oid-info.com/basic-search.htm)
# we shall use a temporary generated Microsoft from https://gallery.technet.microsoft.com/scriptcenter/56b78004-40d0-41cf-
b95e-6e795b2e8a06#content
otherName.3 = 1.2.840.113556.1.8000.2554.60409.5658.22282.18608.39468.5359858.9746403.50.2;UTF8:EL60A02018005
```

Encrypt algorithm

Encrypt bits

Default details of the subscriber

From https://tools.ietf.org/html/rfc5280#section-4.2.1.6 The subject alternative name extension allows identities to be bound to the subject of the certificate. These identities may be included in addition to or in place of the identity in the subject field of the certificate. Because the subject alternative name is considered to be definitively bound to the public key, all parts of the subject alternative name MUST be verified by the CA.

For product certificates we must use X509 extensions as defined in https://tools.ietf.org/html/rfc5280

SubjectAlternativeName part with the product certificate details

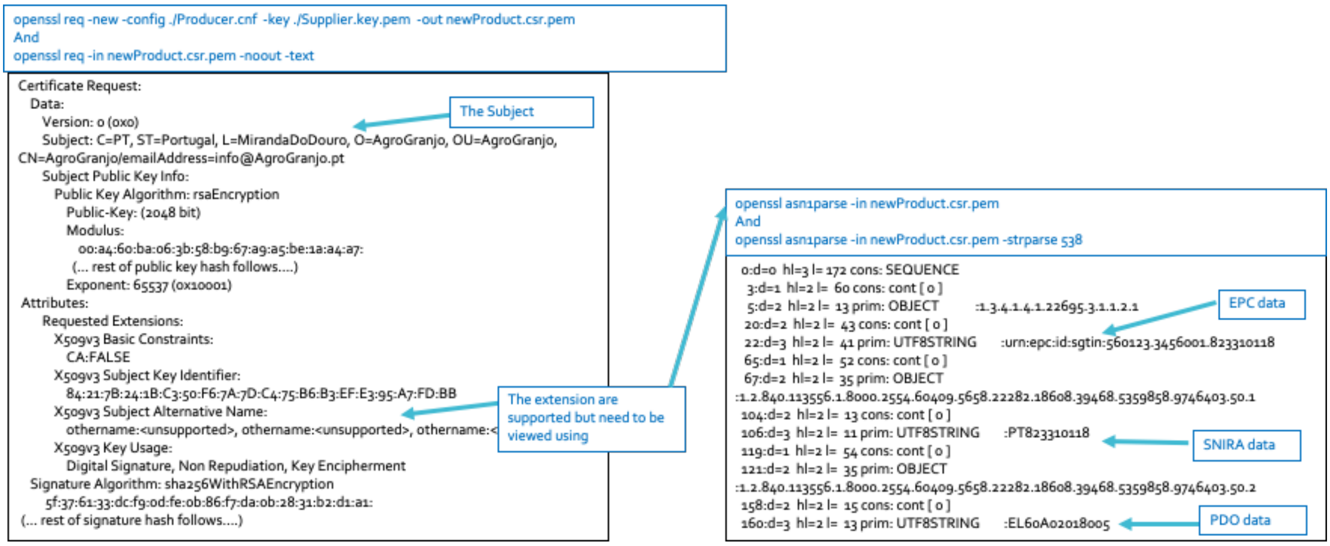
EPC data

SNIRA data

PDO data

Figure 28 – CSR configuration file

Note that to include the product data as a subjectAltName the otherName format is used. This is defined in RFC4043 that requires extra data should be prepended with a OID (as defined by GS1 EPCglobal Certificate Profile Specification 23. In the case of SNIRA and PDO IDs a private sample generated OID was provided via Windows script24. The Producer “Agrogranjo” can create a CSR as follows



```
openssl req -new -config ./Producer.cnf  -key ./Supplier.key.pem  -out newProduct.csr.pem
And
openssl req -in newProduct.csr.pem -noout -text
```

```
Certificate Request:
  Data:
    Version: 0 (0x0)
    Subject: C=PT, ST=Portugal, L=MirandaDoDouro, O=AgroGranjo, OU=AgroGranjo,
CN=AgroGranjo/emailAddress=info@AgroGranjo.pt
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
        Public-Key: (2048 bit)
        Modulus:
          00:a4:60:ba:06:3b:58:b9:67:a9:a5:be:1a:a4:a7:
         (... rest of public key hash follows....)
        Exponent: 65537 (0x10001)
 Attributes:
    Requested Extensions:
      X509v3 Basic Constraints:
        CA:FALSE
      X509v3 Subject Key Identifier:
        84:21:7B:24:1B:C3:50:F6:7A:7D:C4:75:B6:B3:EF:E3:95:A7:FD:BB
      X509v3 Subject Alternative Name:
        othername:<unsupported>, othername:<unsupported>, othername:<
      X509v3 Key Usage:
        Digital Signature, Non Repudiation, Key Encipherment
  Signature Algorithm: sha256WithRSAEncryption
     5f:37:61:33:dc:f9:0d:fe:0b:86:f7:da:0b:28:31:b2:d1:a1:
(... rest of signature hash follows....)
```

The Subject

The extension are supported but need to be viewed using

```
openssl asn1parse -in newProduct.csr.pem
And
openssl asn1parse -in newProduct.csr.pem -strparse 538
```

```
 0:d=0  hl=3 l= 172 cons: SEQUENCE
  3:d=1  hl=2 l=  60 cons: cont [ 0 ]
  5:d=2  hl=2 l=  13 prim: OBJECT        :1.3.4.1.4.1.22695.3.1.1.2.1
 20:d=2  hl=2 l=  43 cons: cont [ 0 ]
 22:d=3  hl=2 l=  41 prim: UTF8STRING     :urn:epc:id:sgtin:560123.3456001.823310118
 65:d=1  hl=2 l=  52 cons: cont [ 0 ]
 67:d=2  hl=2 l=  35 prim: OBJECT
:1.2.840.113556.1.8000.2554.60409.5658.22282.18608.39468.5359858.9746403.50.1
 104:d=2  hl=2 l=  13 cons: cont [ 0 ]
 106:d=3  hl=2 l=  11 prim: UTF8STRING     :PT823310118
 119:d=1  hl=2 l=  54 cons: cont [ 0 ]
 121:d=2  hl=2 l=  35 prim: OBJECT
:1.2.840.113556.1.8000.2554.60409.5658.22282.18608.39468.5359858.9746403.50.2
 158:d=2  hl=2 l=  15 cons: cont [ 0 ]
 160:d=3  hl=2 l=  13 prim: UTF8STRING     :EL60A02018005
```

EPC data

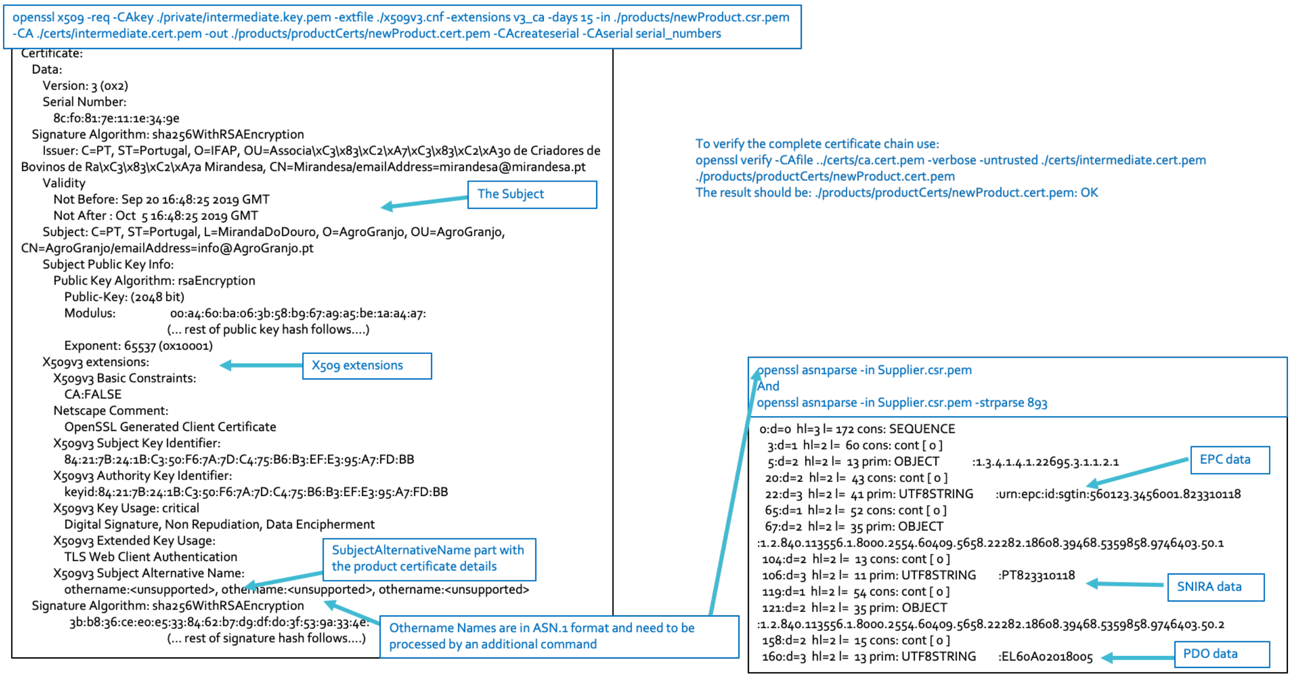
SNIRA data

PDO data

Figure 29 – Certificate Request with Product data

[23] Certificate profile specification available at: https://bit.ly/2QWsGMx
[24] OID generating script available at: https://bit.ly/37VxcB4

Now at the Intermediate CA “Mirandesa” we use following procedure to issue the certificate.



```
openssl x509 -req -CAkey ./private/intermediate.key.pem -extfile ./x509v3.cnf -extensions v3_ca -days 15 -in ./products/newProduct.csr.pem
-CA ./certs/intermediate.cert.pem -out ./products/productCerts/newProduct.cert.pem -CAcreateserial -CAserial serial_numbers
```

```
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number:
      8c:f0:81:7e:11:1e:34:9e
  Signature Algorithm: sha256WithRSAEncryption
    Issuer: C=PT, ST=Portugal, O=IFAP, OU=Associa\xC3\x83\xC2\xA7\xC3\x83\xC2\xA30 de Criadores de
Bovinos de Ra\xC3\x83\xC2\xA7a Mirandesa, CN=Mirandesa/emailAddress=mirandesa@mirandesa.pt
    Validity
      Not Before: Sep 20 16:48:25 2019 GMT
      Not After : Oct  5 16:48:25 2019 GMT
    Subject: C=PT, ST=Portugal, L=MirandaDoDouro, O=AgroGranjo, OU=AgroGranjo,
CN=AgroGranjo/emailAddress=info@AgroGranjo.pt
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
        Public-Key: (2048 bit)
        Modulus:            00:a4:60:ba:06:3b:58:b9:67:a9:a5:be:1a:a4:a7:
                            (... rest of public key hash follows....)
        Exponent: 65537 (0x10001)
    X509v3 extensions:
      X509v3 Basic Constraints:
        CA:FALSE
      Netscape Comment:
        OpenSSL Generated Client Certificate
      X509v3 Subject Key Identifier:
        84:21:7B:24:1B:C3:50:F6:7A:7D:C4:75:B6:B3:EF:E3:95:A7:FD:BB
      X509v3 Authority Key Identifier:
        keyid:84:21:7B:24:1B:C3:50:F6:7A:7D:C4:75:B6:B3:EF:E3:95:A7:FD:BB
      X509v3 Key Usage: critical
        Digital Signature, Non Repudiation, Data Encipherment
      X509v3 Extended Key Usage:
        TLS Web Client Authentication
      X509v3 Subject Alternative Name:
        othername:<unsupported>, othername:<unsupported>, othername:<unsupported>
  Signature Algorithm: sha256WithRSAEncryption
       3b:b8:36:ce:e0:e5:33:84:62:b7:d9:df:d0:3f:53:9a:33:4e:
                            (... rest of signature hash follows....)
```

To verify the complete certificate chain use:
openssl verify -CAfile ../certs/ca.cert.pem -verbose -untrusted ./certs/intermediate.cert.pem ./products/productCerts/newProduct.cert.pem
The result should be: ./products/productCerts/newProduct.cert.pem: OK

```
openssl asn1parse -in Supplier.csr.pem
And
openssl asn1parse -in Supplier.csr.pem -strparse 893
```

```
 0:d=0  hl=3 l= 172 cons: SEQUENCE
  3:d=1  hl=2 l=  60 cons: cont [ 0 ]
  5:d=2  hl=2 l=  13 prim: OBJECT        :1.3.4.1.4.1.22695.3.1.1.2.1
 20:d=2  hl=2 l=  43 cons: cont [ 0 ]
 22:d=3  hl=2 l=  41 prim: UTF8STRING     :urn:epc:id:sgtin:560123.3456001.823310118
 65:d=1  hl=2 l=  52 cons: cont [ 0 ]
 67:d=2  hl=2 l=  35 prim: OBJECT
:1.2.840.113556.1.8000.2554.60409.5658.22282.18608.39468.5359858.9746403.50.1
104:d=2  hl=2 l=  13 cons: cont [ 0 ]
106:d=3  hl=2 l=  11 prim: UTF8STRING     :PT823310118
119:d=1  hl=2 l=  54 cons: cont [ 0 ]
121:d=2  hl=2 l=  35 prim: OBJECT
:1.2.840.113556.1.8000.2554.60409.5658.22282.18608.39468.5359858.9746403.50.2
158:d=2  hl=2 l=  15 cons: cont [ 0 ]
160:d=3  hl=2 l=  13 prim: UTF8STRING     :EL60A02018005
```

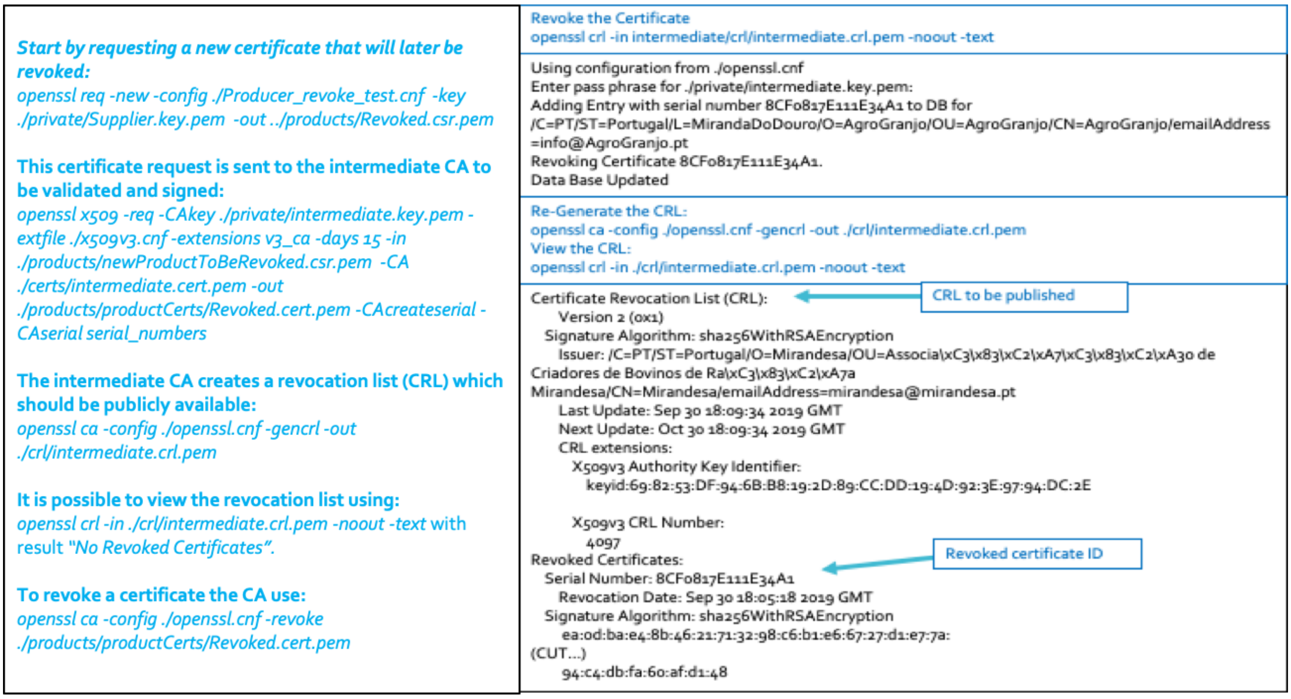
The Subject
X509 extensions
SubjectAlternativeName part with the product certificate details
Othername Names are in ASN.1 format and need to be processed by an additional command
EPC data
SNIRA data
PDO data

Figure 30 – Product Certificate

This valid certificate is now ready to be used in the SCM over BC, imported to WalliD provider and supplied to the SCM certificate validator for verification.

Certificate revocation

The complete revocation workflow is as follows:



Start by requesting a new certificate that will later be revoked:
openssl req -new -config ./Producer_revoke_test.cnf -key ./private/Supplier.key.pem -out ../products/Revoked.csr.pem

This certificate request is sent to the intermediate CA to be validated and signed:
openssl x509 -req -CAkey ./private/intermediate.key.pem -extfile ./x509v3.cnf -extensions v3_ca -days 15 -in ./products/newProductToBeRevoked.csr.pem -CA ./certs/intermediate.cert.pem -out ./products/productCerts/Revoked.cert.pem -CAcreateserial -CAserial serial_numbers

The intermediate CA creates a revocation list (CRL) which should be publicly available:
openssl ca -config ./openssl.cnf -gencrl -out ./crl/intermediate.crl.pem

It is possible to view the revocation list using:
openssl crl -in ./crl/intermediate.crl.pem -noout -text with result "*No Revoked Certificates”.*

To revoke a certificate the CA use:
openssl ca -config ./openssl.cnf -revoke ./products/productCerts/Revoked.cert.pem

```
Revoke the Certificate
openssl crl -in intermediate/crl/intermediate.crl.pem -noout -text
```

```
Using configuration from ./openssl.cnf
Enter pass phrase for ./private/intermediate.key.pem:
Adding Entry with serial number 8CF0817E111E34A1 to DB for
/C=PT/ST=Portugal/L=MirandaDoDouro/O=AgroGranjo/OU=AgroGranjo/CN=AgroGranjo/emailAddress
=info@AgroGranjo.pt
Revoking Certificate 8CF0817E111E34A1.
Data Base Updated
```

```
Re-Generate the CRL:
openssl ca -config ./openssl.cnf -gencrl -out ./crl/intermediate.crl.pem
View the CRL:
openssl crl -in ./crl/intermediate.crl.pem -noout -text
```

```
Certificate Revocation List (CRL):
    Version 2 (0x1)
  Signature Algorithm: sha256WithRSAEncryption
    Issuer: /C=PT/ST=Portugal/O=Mirandesa/OU=Associa\xC3\x83\xC2\xA7\xC3\x83\xC2\xA30 de
Criadores de Bovinos de Ra\xC3\x83\xC2\xA7a
Mirandesa/CN=Mirandesa/emailAddress=mirandesa@mirandesa.pt
    Last Update: Sep 30 18:09:34 2019 GMT
    Next Update: Oct 30 18:09:34 2019 GMT
    CRL extensions:
      X509v3 Authority Key Identifier:
        keyid:69:82:53:DF:94:6B:B8:19:2D:89:CC:DD:19:4D:92:3E:97:94:DC:2E

      X509v3 CRL Number:
        4097
Revoked Certificates:
  Serial Number: 8CF0817E111E34A1
    Revocation Date: Sep 30 18:05:18 2019 GMT
  Signature Algorithm: sha256WithRSAEncryption
     ea:0d:ba:e4:8b:46:21:71:32:98:c6:b1:e6:67:27:d1:e7:7a:
(CUT...)
     94:c4:db:fa:60:af:d1:48
```

Figure 31 – Revoke product certificate

The now revoked certificate is added to the CRL and can be accessed by any interested party (e.g. the SCM certificate validator)

Annex 11 - List of references

Alliance, I. S. E. A. L. (2016). Chain of Custody: Models and Definitions. London, UK.

Angelis, J., & da Silva, E. R. (2019). Blockchain adoption: A value driver perspective. Business Horizons, 62(3), 307-314.

Antonopoulos, A. M., & Wood, G. (2018). Mastering ethereum: building smart contracts and dapps. O'Reilly Media.

Bartoletti, M., & Pompianu, L. (2017, April). An empirical analysis of smart contracts: platforms, applications, and design patterns. In International Conference on Financial Cryptography and Data Security (pp. 494-509). Springer, Cham.

Casey, M., & Wong, P. (2017). Global supply chains are about to get better, thanks to blockchain. Harvard Business Review, 13, 1-6.

Chang, Y., Iakovou, E., & Shi, W. (2019). Blockchain in Global Supply Chains and Cross Border Trade: A Critical Synthesis of the State-of-the-Art, Challenges and Opportunities. arXiv preprint arXiv:1901.02715.

Cheney, J., Chong, S., Foster, N., Seltzer, M., & Vansummeren, S. (2009, October). Provenance: a future history. Proceedings of the 24th ACM SIGPLAN conference on Object oriented programming systems languages and applications (pp. 957-964). - ACM

Dobrovnik, M., Herold, D., Fürst, E., & Kummer, S. (2018). Blockchain for and in Logistics: What to Adopt and Where to Start. Logistics, 2(3), 18.

FAO CODEX Alimentarius CXG 60-2006. Retrieved from https://bit.ly/2meiPUS

GS1 (2017). Global Traceability Standard - GS1's framework for the design of interoperable traceability systems for supply chains.

GS1 EPCglobal Certificate Profile Specification. Retrieved from https://www.gs1.org/sites/default/files/docs/cert/cert_2_0-standard-20100610.pdf

GS1 EPCIS standard: Retrieved from https://www.gs1.org/standards/epcis

Hartley, G., & Sundermann, E. (2014) The Efficacy of Using the EPCglobal Network for Livestock Traceability: A Proof of Concept. – GS1 New Zealand

Hevner, A., & Chatterjee, S. (2010). Design science research in information systems. In Design research in information systems (pp. 9-22). Springer, Boston, MA.

Hughes, A., Park, A., Kietzmann, J., & Archer-Brown, C. (2019). Beyond Bitcoin: What blockchain and distributed ledger technologies mean for firms. Business Horizons.

ISO 9000:2015 Quality management systems – Fundamentals and vocabulary. Retrieved from https://www.iso.org/standard/45481.html

Kim, H. M., & Laskowski, M. (2018). Toward an ontology-driven blockchain design for supply-chain provenance. Intelligent Systems in Acc., Fin. and Mngt. 25(1), 18-27.

Kshetri, N. (2018). Blockchain's roles in meeting key supply chain management objectives. International Journal of Information Management, 39, 80-89.

Litke, A., Anagnostopoulos, D., & Varvarigou, T. (2019). Blockchains for Supply Chain Management: Architectural Elements and Challenges Towards a Global Scale Deployment. Logistics, 3(1), 5.

Martin, R. C. (2017). Clean architecture. Pearson Education.

Martindale, W., Hollands, T., Swainson, M., & Keogh, J. G. (2018). Blockchain or bust for the food industry?. Food Science and Technology, 33(4).

Montecchi, M., Plangger, K., & Etter, M. (2019). It’s real, trust me! Establishing supply chain provenance using blockchain. Business Horizons.

Morkunas, V. J., Paschen, J., & Boon, E. (2019). How blockchain technologies impact your business model. Business Horizons.

O’Marah, K. (2017). Blockchain: Enormous potential demands your attention. Supply Chain Digital. Retrieved from http://www.supplychaindigital.com/technology/blockchain-enormous-potential-demands-your-attention

OpenSSL X509 V3 - certificate extension configuration format. Retrieved from https://www.openssl.org/docs/manmaster/man5/x509v3_config.html

Perboli, G., Musso, S., & Rosano, M. (2018). Blockchain in logistics and supply chain: A lean approach for designing real-world use cases. IEEE Access, 6, 62018-62028.

Rahmadika, S., Kweka, B. J., Latt, C. N. Z., & Rhee, K. H. (2018, November). A Preliminary Approach of Blockchain Technology in Supply Chain System. In 2018 IEEE International Conference on Data Mining Workshops (ICDMW) (pp. 156-160) - IEEE.

Regulation (EC) No 178/2002 - European Food Safety Authority requirements. Retrieved from https://bit.ly/35il6Ra

RFC4043 - X509 PKI permanent identifier: https://tools.ietf.org/html/rfc4043

Sermpinis, T., & Sermpinis, C. (2018). Traceability Decentralization in Supply Chain Management Using Blockchain Technologies. arXiv preprint arXiv:1810.09203.

Solidity documentation. Retrieved from https://solidity.readthedocs.io/en/v0.5.11/

Tavares, M., Guerreiro, A., Coutinho, C., Veiga, F., & Campos, A. (2018, September). WalliD: Secure your ID in an Ethereum Wallet. In 2018 International Conference on Intelligent Systems (IS) (pp. 714-721) - IEEE.

Toyoda, K., Mathiopoulos, P. T., Sasase, I., & Ohtsuki, T. (2017). A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain. IEEE Access, 5, 17465-17477.

Tröger, R., & Alt, R. (2017). Design options for supply chain visibility services: learnings from three EPCIS implementations. Electronic Markets, 27(2), 141-156.

Wang, Yingli, Singgih, M., Wang, J., & Rit, M. (2019). Making sense of blockchain technology: How will it transform supply chains?. International Journal of Production Economics, 211, 221-236.

Weber, I., Xu, X., Riveret, R., Governatori, G., Ponomarev, A., & Mendling, J. (2016, September). Untrusted business process monitoring and execution using blockchain. In International Conference on Business Process Management (pp. 329-347). Springer Cham.

Westerkamp, M., Victor, F., & Kupper, A. (2019). Tracing manufacturing processes using blockchain-based token compositions. Digital Communications and Networks.

Xu, Lei, Chen, L., Gao, Z., Chang, Y., Iakovou, E., & Shi, W. (2018, October). Binding the Physical and Cyber Worlds: A Blockchain Approach for Cargo Supply Chain Security Enhancement. In 2018 IEEE International Symposium on Technologies for Homeland Security (HST) (pp. 1-5). IEEE.

Xu, Xiwei, Lu, Q., Liu, Y., Zhu, L., Yao, H., & Vasilakos, A. V. (2019). Designing blockchain-based applications a case study for imported product traceability. Future Generation Computer Systems, 92, 399-406.