future internet

MDPI

Article

Blockchain and Self Sovereign Identity to Support Quality in the Food Supply Chain

Luisanna Cocco *, Roberto Tonelli and Michele Marchesi

Department of Mathematics and Computer Science, University of Cagliari, 9124 Cagliari, Italy; roberto.tonelli@dsf.unica.it (R.T.); marchesi@unica.it (M.M.)
* Correspondence: cocco@unica.it

Abstract: This work presents how a digital identity management system can support food supply chains in guaranteeing the quality of the products marketed and the compliance of the several supply-chain's nodes to standards and technical regulations. Specific goal of this work is to present a system that provides full visibility of process/food certifications, which nowadays are issued by accredited and approved certification bodies (issuers) and delivered and stored in paper version by the several participants (holders) of the supply chain. The system is designed and implemented by combining the latest most innovative and disruptive technologies in the market—Self Sovereign Identity system, Blockchain, and Inter Planetary File System. The crucial aspects that it aims to hit are the storage and access of food/process certifications, and the proper eligibility verification of these certifications exploiting the concepts of the Self Sovereign Identity-based models. The proposed system, realized by using standards that are WWW Consortium-compatible and the Ethereum Blockchain, ensures eligibility, transparency, and traceability of the certifications along a food supply chain, and could be an innovation model/idea that the companies that adopt the Open Innovation paradigm might want to pursue.

Keywords: self sovereign identity; blockchain; food supply chain; food/process certifications

Citation: Cocco, L.; Tonelli, R.; Marchesi, M. Blockchain and Self Sovereign Identity to Support Quality in the Food Supply Chain. *Future Internet* **2021**, *13*, 301. https://doi.org/10.3390/fi13120301

Academic Editors: Luis Javier Garcia Villalba and Paolo Bellavista

Received: 7 October 2021
Accepted: 24 November 2021
Published: 26 November 2021

1. Introduction

Today, in an increasingly globalized world, companies connect ever more closely to their customers and partners around the world. Companies expand beyond national borders, aiming for continuous improvement and for increasing the trust and loyalty of their customers. They standardize their processes/procedures in such a way to be able to guarantee their customers to work methodically to achieve a specific result, so that the quality of the products marketed can always be guaranteed. In addition, they increasingly use certifications as direct or indirect guarantee tools in supply chains.

For a company, having one or more certifications means being visibly placed on the market, being considered a reliable company, entering supply chains, and being able to export its products. Certifications help to simplify the production and to minimize the risks and consequences associated with them. Having a certification implies the implementation of strict controls in internal processes, the monitoring and verification of processing methods and compliance with hygiene requirements along the entire production chain. Behind a certification there is a Hazard Analysis and Critical Control Points (HACCP) system, procedures to define precise processing standards to be followed and a document management system that allows companies to manage, organize, share, and archive documents/files.

Food/process certifications are not mandatory but they are increasingly requested by the market [1,2]. They give to companies a competitive advantage in the sector in which they operate. The certifications allow to assess compliance with the requirements on food safety, quality and legality and are essential for companies to be able to operate in international markets, where each country operates according to its own current legislation.

Thus, given that for the food industry there are not standards valid around the world, it is very useful to own more than one certification.

These certifications help to guarantee and increase trust and transparency to consumers, but also to suppliers, according to their target. There are certifications created to "communicate" with the final consumer, the so called *Business to Consumer certifications* and certifications created to "communicate" with the Large-Scale Retail Trade, the so called *Business to Business certifications*.

Food certifications are third-party verifications that products, processes, or systems in the food supply chain must meet to satisfy food standards [3]. By means of these certifications, a third independent party, which can be for example a certification body, a test laboratory, a certification center or a calibration center, certifies with a sufficient level of confidence that a particular product, service, or process complies with a given standard or technical regulation. This is a typical third-part model, in which if an/a actor/participant along the supply chain wishes to own a certification has to start a request process to obtain the desired certification signed by a trusted authority. On the other side if a participant along the supply chain wishes to verify the validity of a given certification, to ascertain that the certification is issued by a trusted body, he/she has to start a verification process that could take hours or days since food supply chains are hardly digitalized and food documentation are typically stored on paper or private databases [4,5]. In light of these considerations, our research question is that of studying and analyzing whether the SSI concepts applied within a supply chain ensure eligibility, transparency, and traceability of the certifications along the chain.

This work deals with the storage and the retrieval of food supply chain certifications, and presents a system in which the validity of all certifications owned by the several participants along the supply chain is verified against a decentralized and trusted registry of information.

Specifically, we present a Self Sovereign Identity (SSI) system that relies on the Blockchain technology and, by means of smart contracts, implements both the certification registry and the logic to verify the validity of a certification. The certification registry creates a binding between a certification, its owner, its issuer, and its associate off-chain location. Thus, it maps each certification by an identifier to an interplanetary file system (IPFS) storage location, hence to an IPFS hash. Every time that a participant along a supply chain needs to obtain a certification signed by an authorized body, he/she has to complete an off-chain procedure, as we see in detail in Section 3 The Proposed SSI System. If the procedure is completed successfully, the body/issuer creates a certification, and registers, into a blockchain, a cryptographic proof of the certification's issuance, together with the certification's ownership, the status of the certification, and the off-chain location in which the complete and plain text of the certification can be inspected by a human or by a computer.

Let us conclude this section underlining the link between our work and the so called *open innovation* paradigm. Open Innovation is a paradigm that states that to implement new technologies and business opportunities, companies can and must make use of both external and internal ideas, so as to reduce the risks and costs associated with innovation and share the benefits. Our work, which includes first of all the model proposal, and then a possible first implementation of the model, could be an innovation model/idea that companies in the agrifood sector, which put Open Innovation at the center of their strategic choices, could adopt/ pursue. Within the SSI model each certification can become a verifiable attestation/presentation, hence also a food certification could be dealt as verifiable attestation and be used within a given food supply chain to add value to the chain. Of course maybe only in future we could see complete SSI model in which individuals share their data by using verifiable attestations/presentations.

2. Review

Globalization has significantly transformed the agri-food sector and provenance, quality, transparency and safety have become key concepts within food supply chains.

Today the agri-food supply chain is operating in a market characterized by different quality standards and by subjects with different interests that, to remain competitive on the market, increasingly retain the consumer, looking for tools to make the production process transparent, since consumers are increasingly attentive to the origin and the quality of the products they buy. In addition, due to the numerous episodes of fraud and food counterfeiting, today more than ever there is a need for efficient coordination between the operators involved in the supply chain, and for new management models that can lead to a reduction in the management costs that companies have to bear to comply with the various international regulations aimed at protecting health and safety.

2.1. Certifications in Food Supply Chain

Let us start with the definition of supply chain and food product.
As defined by S. and Meindl [6]

> . . . a supply chain is a sequence of processes and flows that take place within and between different stages and combine to fill a customer need for a product. . . The processes in a supply chain are divided into a series of cycles, each performed at the interface between two successive stages of a supply chain. Given the five stages of a supply chain (Customer, Retailer, Distributor, Manufacturer, Supplier), all supply chain processes can be broken down into the following four process cycles: Customer order cycle, Replenishment cycle, Manufacturing cycle, Procurement cycle. Each cycle occurs at the interface between two successive stages of the supply chain. Not every supply chain will have all four cycles clearly separated. For example, a grocery supply chain in which a retailer stocks finished-goods inventories and places replenishment orders with a distributor is likely to have all four cycles separated. . .
>
> A cycle view of the supply chain clearly defines the processes involved and the owners of each process. This view is useful when considering operational decisions because it specifies the roles and responsibilities of each member of the supply chain and the desired outcome for each process.

As regards the definition of food, at a European level, a food product is first of all a food and as such it must respond to the basic principles of this type of good which are expressed by Regulation (EC) 178/2002, which involves the entire food chain including the primary phase (Ref. https://ec.europa.eu/info/food-farming-fisheries_en (accessed on 15 November 2021)) upstream and the final market downstream.

According to article 2 of the aforementioned regulation, food is defined as "any substance or product processed, partially transformed or not processed, intended to be ingested, or which is reasonably expected to be ingested, by humans". To be marketed, food products must comply with a canon of legality, which includes safety for the health of consumers and technical compliance with current regulations in terms of identity, process, and communication, which must meet the requirements of truthfulness and loyalty.

Having respected the mandatory configuration of the food products defined by this regulation, it is possible to add voluntary requirements. These additional requirements must not conflict with the mandatory standards and must be substantial and objective to pursue the quality of the product. So, pursuing the product quality to increase it is therefore intended as an addition of requirements to those defined by the mandatory regulation.

The pursuit of quality requires a precise and specific management that goes from planning to implementation, by the definition of methods, rules/procedures that have to be adapted to the real needs and business reality. A method that aims at the continuous improvement of processes and is at the same time a continuous guarantee of quality, is the Deming cycle. It is based on four moments—Plan, Do, Check, Act, (PDCA cycle)—and aims to promote a culture of quality to obtain continuous improvements of processes and the optimal use of resources [7].

Within a quality system we find the standards that define the requirements to be met, and the certifications that certify that the requirements are actually met. Certifications

are issued by third parties completely independent of the company that, after precise and accurate certifications processes, certify that a product, service, process, or company complies with the specified requirements. The certifications are divided into mandatory and voluntary [3,8]. The former is regulated by laws enacted at a national or community level; the latter is freely chosen by the company and, in any case, they refer to quality systems based on rules of a legislative nature, or on rules of a technical nature. The mandatory certifications are certificates that a company is forced to obtain to be able to operate. For example, the certificates that define the use of a valid plan self-control and implementation of the HACCP plan to reduce the risk of food contamination are mandatory certificates.

The certifications can then be classified as product or process certifications. Product certification tends to enhance and differentiate certain qualities of a product food. Instead process certifications have as their object the production phases, which carry out the transformations and/or the sale of products or services, and often have the purpose of ascertaining a hygienic-sanitary quality. They attest the ability of an organization to structure and manage its resources and production processes to ensure, for example, the hygienic and sanitary quality through traceability systems and HACCP methodologies.

Among the main voluntary certification standards [3,9] there are:

- *ISO 9001* describes the organizational structure, procedures, processes, and resources necessary to implement quality management in a company [10].
- *ISO 22000* is specific for food safety, and is applicable to all companies involved in the agri-food chain. This standard establishes the Prerequisite Programs (PRP), which integrate with the HACCP principles. These programs are all those prerequisites and activities to be implemented to maintain a hygienic environment throughout the agri-food chain, capable of producing, managing, and supplying products that are safe for consumption.
- *ISO 22005* certifies traceability in the food and feed chain. Organizations wishing to enhance and communicate the origin of the products and raw materials used often resort to this certification.
- *IFS FOOD* (International Food Standard) differently from the aforementioned standards is a product certification, that is, it certifies that agri-food productions are carried out guaranteeing food safety and the healthiness of the products marketed [11]. For a company to be certified IFS it must perform an accurate risk analysis (microbiological, biological, chemical, physical, allergens), know, evaluate, and control all its suppliers, both of raw materials and of primary packaging and material in contact with Foods.
- *BRC GLOBAL STANDARD* offers a set of guidelines for the production of safe food and for the management of product quality, to fully meet customer expectations. The main purpose of this standard is to strengthen and promote food safety throughout the supply chain. The standard applies to food processing and preparation companies and identifies the specific elements of a management system focused on the quality and hygienic-sanitary safety of products, which take as reference the HACCP methodology for the planning and the implementation.

IFS and BRC are recognized by the Global Food Safety Initiative (GFSI), which is a nonprofit foundation created under Belgian law in May 2000 to address the issue of differences in standards on a global scale and reduce resultant audit inefficiency. All the voluntary certifications, described above, guarantee an independent and reliable assessment of a company's food safety systems, given that the appropriate checks are performed by certification bodies external and independent of the company.

2.2. SSI, Blockchain and IPFS

Recently, many research papers investigated Blockchain's potential in combination with Internet of Things (IoT) [12–18] and/or with distributed systems such as IPFS [19–24], for enhancing the integrity, auditing, information flows, and traceability of the supply

chain, and contributing to scalability, security, and interoperability of the supply chain management systems.

In this work, we propose a model that exploits Blockchain, IPFS, and the SSI concepts, thereby aiming at superior management and storage of the food/process certifications. The model is conceived in such a way to be easily integrated, in the near future, into complete and functioning SSI systems, in which each individual manages their entire identity, and all their claims, independently, without relying on a third party. Let us give some details about this new identity management system (ref. https://ecas.ec.europa.eu/cas/login (accessed on 15 November 2021)and [25]), that entails (may entail) numerous advantages in the supply chain management.

Traditionally, the identification of people is performed through centralized systems, that uniquely identify and store people's data. Nowadays, thanks to digital transformation, data represents individuals, and the data owner no longer has control over their own data (Ref. https://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html (accessed on 15 November 2021)). Whenever individuals use the network, by using social networks or carrying out economic transactions, third parties the so called "service providers" request them the "user authentication" to grant access to their services. In this way individuals leave on the network virtual footprints, that are their digital identities. The SSI models want to give back to individuals the possession and control of their data, and use digital wallets as a possible solution to collect all the individuals' data/attestations that each identity can boast. Thanks to blockchain and/or other distributed systems, these models could become a reality. Blockchain technology, and in general distributed systems, allow a new form of identity management. Precisely they allow the management of the so called decentralized identities (DIDs). A decentralized identity is a set of characteristics/claims that uniquely define natural persons, legal entities or things.

Unlike conventional methods, a DID remains under the control of the individual. In this way, personal data are solely owned by the individual and are not registered with other parties. In addition, with encryption, the individual can demonstrate possession of the claims/attestations without sharing other information that is not strictly necessary, and thus ensure maximum respect for their privacy. The concepts of Zero-Knowledge Proof and Selective Disclosure, are focused both on the principle of data minimization and are used to reduce the exposure of individuals' personal information. So with SSI, the individual returns to having full control over their own identity, deciding if and which attributes/claims to make available to external parties.

The European Blockchain Services Infrastructure (EBSI) (EBSI is a network of distributed nodes across Europe that will deliver cross-border public services.) works on the implementation of an European Self-Sovereign Identity framework (ESSIF). This working group drafted the technical specification of this framework and describes it through a very explicit UML (unified model language) class diagram, which shows that each entity can have multiple DIDs, and verifiable IDs, attestations, and presentations, see Figure 1 (Extracted from the EBSI's technical specification in web site https://ecas.ec.europa.eu/cas/login, (accessed on 15 November 2021)).

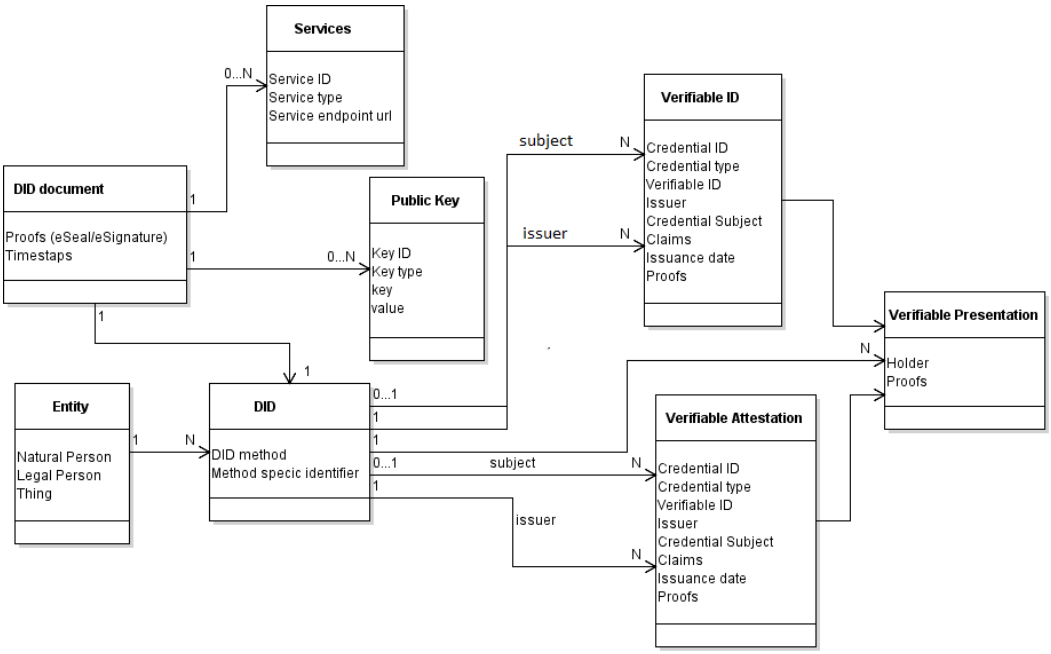

Figure 1. SSI data model proposed by EBSI.

This working group gives the following definitions to specify the parties and the objects involved in the framework:

- Digital Identity: a Digital Identity—and corresponding identifiers (see DIDs)—allow for unique identification of an entity within an SSI ecosystem, such as ESSIF. Also, every entity (i.e., natural person, legal entity) can have multiple digital identities. Simply speaking, Digital Identities require (a) identifiers (e.g., for discovery) and (b) identity-related information about an entity, such as about individual attributes or properties (see Verifiable IDs and/or Verifiable Attestations).
- Decentralized Identifier (DID): a DID is in its basic form/in principal and according to SSI-principals nothing more than an address, which is mainly used to identify and find other parties, establish relations and sessions to exchange data. Consequently, a DID, viewed in isolation, says nothing about the respective entity with which it is associated.
- Verifiable IDs: in contrast to a DID, a Verifiable ID contains information about an entity which it can put forward as evidence of whom he/she/it is (comparable with, e.g., a passport or physical ID card) and, thus, allows for identification and authentication. A Verifiable IDs is issued by an entity, called an Issuer. The entity holding a Verifiable ID is called a Holder.
- Verifiable Attestation: a Verifiable Attestation contains information about an entity which it can put forward as evidence of certain attributes or properties, or as evidence of e.g., a permit, attestation, authorization received. A Verifiable Attestation is issued by an Issuer to a Holder. The terminological differentiation between Verifiable IDs and Verifiable Attestations is purely functional, thus, no relevant differences on the technical level are to be expected.
- Verifiable Presentations: an entity that was issued a Verifiable ID and/or Attestation, should not share it directly with another party (which may need certain information for conducting a transaction), called a Verifier or a Relying Party. To share the information contained in Verifiable IDs and/or Attestations, an entity must construct a Verifiable Presentation by transforming information from one or more underlying Verifiable IDs and/or Attestations. Simply speaking, a Verifiable Presentation is identity data (taken from Verifiable IDs and/or Attestations) that the Holder chooses to share with a specific Verifier for a specific purpose.

and adopted the following W3C specification as DID document features:

- Contexts: JSON-LD contexts define the terminology used to describe data, ensuring that both producers and consumers of data have a shared understanding of the semantics.
- DID Subject: this is the subject (individual, organization, thing, animal, etc.) identified by the DID.
- Public Keys: Public Keys associated with a DID are a prerequisite for secure and authenticated communication between DID Subjects.
- Authentication: the Authentication block in a DID Document simply references the DID Document's Public Key (see above) that is intended for proving control/ownership of a DID. This is used when two parties (e.g., a Holder and a Verifier) connect and exchange data and messages.
- Proof: this can be added to a DID Document to prove integrity or correctness or other security and trust aspects of a DID Document. DID Resolution (obtaining a DID Document for a given DID) is itself intended to be a trusted process; therefore, explicitly adding a Proof object to a DID Document is not strictly required, but can optionally be used to add data for additional trust characteristics.
- Extensibility: additional elements can be added to a DID Document with additional metadata about the DID Subject.

On 3 June 2021 this working group made a proposal for an amending Regulation (EU) No 910/2014 about the ESSIF [26]. The amending regards the legal effects of electronic ledgers and may entail significant implications for all applications that make use, or may in the future make use, of these registers. Specifically, as reported on page 43 of above quoted work, the group write:

> Legal effects of electronic ledgers:
>
> 1. An electronic ledger shall not be denied legal effect and admissibility as evidence in legal proceedings solely on the grounds that it is in an electronic form or that it does not meet the requirements for qualified electronic ledgers.
>
> 2. A qualified electronic ledger shall enjoy the presumption of the uniqueness and authenticity of the data it contains, of the accuracy of their date and time, and of their sequential chronological ordering within the ledger.

As already mentioned, in this work we present a system that manages the food/process certifications in a decentralized public system within a SSI system based on the concepts just described. Our system was designed exploiting the two main elements for identity management, that are the decentralized registers and digital wallets as pointed out by López [25] that we quote in the following:

> "SSI leverages two essential elements for identity management: decentralized registers of information and digital wallets. Decentralized ledgers: the SSI model relies on decentralized registers of information, in which the proofs of ownership of decentralized identifiers and the verifiable credentials are stored within a decentralized ledger. Unlike the centralized, third-party, federated, and user-centric models, which require the verifying entity to somehow reach out to the issuer to verify digital credentials presented to them by the subjects, the SSI model allows the issuer to leave all necessary proofs (cryptographic proofs such as digital signatures and timestamps) in a decentralized public ledger so that anyone can verify them against it... Digital wallets: digital wallets are portable and secure personal repositories. Ideally in the form of a mobile app, they allow us to manage our identifiers, authenticators, data, and verifiable credentials within our phones, which are completely protected and under our control. We decide what information we disclose to whom in the form of verifiable presentations..."

Our work aims to manage the food/process certifications along the supply chain, and is designed in such a way to be easily integrated in a supply chain management system

in which both the identities of the participants, and those of the IoT devices, are a type of DID. We propose a system characterized by the requirements of interoperability, portability, pseudonymity, recovery, scalability, security, and usability, compatible or easily adoptable to the standards of DIDs and verifiable credentials (VCs) under development by the WWW Consortium (Ref. https://w3c.github.io/did-core/) (accessed on 15 November 2021).

2.3. Related Works

As regards the literature on this topic, recently Nguyen et al. [27] wrote a work dealing with the problem of fake educational certificates in Vietnam and proposed a blockchain-based authentication system to manage it, but without taking into account the SSI concepts. They designed a complete system for the educational certificate management, including business processes, data mapping structure, and the decentralized application, in such a way to meet the specific Vietnamese requirement. They tested their system by using Hyperledger Fabric and deploying their dApp on the Amazon EC2 cloud. Bouras et al. [28] presented a decentralized capability-based access control architecture designed for IoT consortium networks to tackle for example data leakage and single-point failure that are typical problems of centralized traditional access control schemes. Weingaertner and Camenzind [29] faced the problem of Identity of the IoT devices whose reliable identification is crucial, proposing an approach based on blockchain and decentralised identifiers (DID); hence, based also on the concepts of SSI and on bootstrapping of remote secure key infrastructures (BRSKI). Bartolomeu et al. [30] discussed the advantanges of the SSI systems in combination with industrial IoT applications, and Niya [31] proposed a Know Your IoT device platform (KYoT), which enables the self-sovereign identification of IoT devices on the Ethereum BC applicable to use cases, such as supply chain tracing, smart cities, and IoT data marketplaces. However, the proposed system is based on the ERC 734 and ERC 735 Ethereum identity standards (Ref. https://hackernoon.com/first-impressions-with-erc-725-and-erc-735-identity-and-claims-4a87ff2509c9 (accessed on 15 November 2021)) and so it is not compatible with the standards of DIDs and VCs under development by the WWW Consortium (W3C). Liu et al. [32] proposed key management patterns, decentralized identifier management patterns, and credential design patterns as valid support in SSI management system design.

2.4. Food Certifications as Verifiable Attestation

Our SSI model aims to provide more transparency and traceability in the supply chains giving to users/participants free access to food/process certifications issued by authorized entities, which regularly monitor and inspect supply chain nodes, issuing, revoking, and updating the certifications to ensure regulatory compliance.

Certifications are hence, within this system, verifiable attestations, and as such, they can be modeled following the indications of the EBSI that proposes the format for Verifiable ID and Attestation signed with a DID Key shown in Figure 2. This format is in according to the DIDs standard developed by the W3C.

```
{
 "@context": [
  "https://www.w3.org/2018/credentials/v1",
  "https://essif.europa.eu/schemas/vc/2019/v1",
  "https://essif.europa.eu/schemas/eidas/2019/v1"],
 "id": "did:ebsi-eth:00000001/credentials/1872",
 "type": ["VerifiableCredential", "EssifVerifiableID"],
 "issuer": "did:ebsi-eth:00000001",
 "issuanceDate": "2019-06-22T14:11:44Z",
 "credentialSubject": {
  "id": "did:ebsi-eth:00000002",
  "currentFamilyName": "Franz",
  "currentGivenName": "Hinterberger",
  "dateOfBirth": "1999-03-22T00:00:00Z",
  "placeOfBirth": "Salzburg, Austria"
 },
 "proof": [{
  "type": "EcdsaSecp256k1Signature2019",
  "created": "2019-06-22T14:11:44Z",
  "proofPurpose": "assertionMethod",
  "verificationMethod": "did:ebsi-eth:00000001#key-1",
  "jws": "eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..TCYt5X
   sITJX1CxPCT8yAV-TvkIEq_PbChOMqsLfRoPsnsgw5WEuts01mq-pQy7UJiN5mgRxD-Wuc"
 }
```


Figure 2. Format for Verifiable ID and Attestation signed with a DID Key.

So a Verifiable Credential is a file (The recommended formats for credentials are JWT, JWS, and JSON-LD.) containing the following information (ref. work by López [25]):

- The context defines URI to identify terminology and protocols that allow parties to read the credential. It enables different systems to exchange data with each other.
- URI to identify the credential type.
- URI to identify the issuer.
- Issuance date of the claim.
- URI (URI stands for Uniform Resource Identifier.) to uniquely identify the credential and/or the subject of the credential.
- Cryptographic proof of the issuer.

The W3C specification provides that the data model can be extended. The following features can be added as indicated in the work by López [25]:

- Claims data or metadata.
- Expiration conditions for defining when the claim is no longer valid.
- Location of the credential status.

In general, in an SSI model, it is recommended that the individuals/holders use private repositories/digital wallet to store and manage all their credentials to guarantee data protection requirements. Since our system aims to be as transparent as possible the off-chain model was chosen to meet only transactionability and storage requirements. The database chosen for storing the food/process certificates is the IPFS, that is publicly accessible. The blockchain is queried for verification of the certifications. The subject and issuer DIDs can be found and resolved in the blockchain registries, as well as the credential status (certification status) and the identifier of the certificaton. The certification identifier, that allows to access its location, as we are going to see in the next sections, is the IPFS hash. In fact, when uploading a document on IPFS, the system directly computes and returns the hash of the document and use it as a unique identifier to retrieve the document. We implemented the proposed system using the Ganache tool to simulate a local Ethereum blockchain so the queries for the verification of certifications do not generate transactions, hence costs. In the future, when there will be complete and functioning SSI models, less data may be stored in blockchain, as suggested by EBSI and López [25].

All claims data, including the food/process certifications data, could be stored in the same registry.

2.5. DID Standards and Platforms

We already mentioned that European countries constituted a working group to collaborate towards the realization of the ESSIF. In addition, many start-ups and enterprises are currently exploring how to implement SSI solutions. We can quote some of the most popular DID platforms, such as uPort, Sovrin and Civic, which are based on Ethereum, Hyperledger, and Bitcoin blockchain, respectively; the Bloom platform and the Ethereum standards, such as ERC735 and ERC734/ERC725 (Ref. https://github.com/ethereum/EIPs/issues/780, https://eips.ethereum.org/EIPS/eip-1056, https://github.com/ethereum/eips/issues/734, and https://github.com/ethereum/EIPs/issues/735 (accessed on 15 November 2021)) [33,34].

uPort consists of identity and messaging protocols that together form a layer of interoperable identity for the decentralized web to return ownership of the identity to the individual (see web site https://www.uport.me/ (accessed on 24 September 2021) and [33,34]. It is part of the Decentralized Identity Foundation (DIF), which is a working group created to establish a very simple, common framework for making claims about identities using different kinds of decentralized technologies, and hence, blockchain and nonblockchain technologies. uPort uses ERC1056 and ERC780 as standards. Specifically it uses ERC1056 as DID standard, and ERC780 as standard for claims registry. uPort architecture is W3C compatible, hence it complies with the universally accepted standard and allows for two types of claims: on-chain (using ERC780) and off-chain (using Json Web Tokens, JWTs) Braendgaard [35]. Any address, wallet, or identity is compatible with the ERC1056. It is not necessary to create a contract to access the benefits of the DID.

Sovrin is a standard for identity, solving endless passwords, insecure databases on the internet, and building a user-friendly digital identity as the analog identity, such as driver's licenses and ID cards (see web site https://sovrin.org/ (accessed on 15 November 2021) and [34].

Civic is an identity ecosystem decentralized, which allows the user to create the their virtual identity and store it together to personal information on the device. Verifications are made using selfie and Optical Character Recognition (see web site https://www.civic.com/ (accessed on 15 November 2021)and [34]. *Bloom* is an end-to-end protocol for proof of identity, evaluation risk and credit score so facilitating the issuing of loans. It is entirely based on blockchain (see web site https://bloom.co/ (accessed on 15 November 2021) and [34].

ERC 725/734 standard stipulates that only smart contracts can be identities. An Identity is defined implementing ERC 725/734 standard and is identified by the address of the ERC725/734 Ethereum Smart Contract. Claims instead are implemented by using ERC735 Ethereum Smart Contract and are associated to a precise ERC725/734 identity. So while ERC 725/734 incorporates everything into a single contract, uPort, which is based on Ethereum as the just quoted standards, this realizes a levels architecture in which almost any layer can be modified and updated without the need to have to update the contiguous levels (ref. [35–38]).

Table 1 describes the comparison among the related works quoted in this section and our. From the comparison the main novelties that emerge are the following.

- Our dApp is designed and implemented following a design method recently developed specifically for decentralized application accordingly to the software engineering principles. It is the so-called Agile Block Chain Dapp Engineering (ABCDE) method.
- Our model exploits the public blockchain concepts in combinations with those of the SSI to ensure eligibility, transparency and traceability of the food certifications along a food supply chain.
- In our work, the food certification registry is implemented using ERC 780 standard, hence using an Ethereum standard that, contrary to other standards, such as ERC734

and ERC735, is W3C compatible, so it complies with the universally accepted standards.

Table 1. A comparative table of related works.

Reference	Design Method	SSI Concepts	BlockchainTypology	Standards	Study Case
Nguyen et al. [27]	NA	NA	Hyperledger Fabric	NA	Fake certificate issues in Vietnam
Bouras et al. [28]	NA	NA	BigchainDB	JWT	Waste management in a smart city
Weingaertner and Camenzind [29]	NA	A	Ethereum blockchain	JWT	Devices' Identity in IoT networks
Bartolomeu et al. [30]	NA	A	NA	NA	Review and discussion of use cases
Niya [31]	NA	A	Ethereum blockchain	ERC734-ERC735	Know Your IoT device platform
Liu et al. [32]	NA	A	NA	NA	Design patterns
López [25]	NA	NA	NA	NA	Technical Report
Our dApp	ABCDE method	A	Ethereum blockchain	ERC1056	Food Certifications

3. The Proposed SSI System

The proposed system is a decentralized application (dApp) designed for Ethereum blockchain and the storage of the files on the IPFS external database. To design our dApp we conceived the supply chain as a chain of nodes. Each node (holder) can require one or more certifications to authorized bodies (issuers). Everyone (verifier) can access and read all the issued certifications and verify their validity. Each entity involved in this process, hence the nodes/participants in the supply chain, the certification bodies, and the consumers of the final product in exiting supply chain, are identified by a precise identifier, that is an Ethereum account, hence an address. If a participant, wherever they are in the food value chain, desires to acquire a process/food certification, starts an off-chain process/communication to ask authorized bodies to audit its processes and/or systems against the requirements of that precise process/food certification. The certification body, after audited the processes and/or systems of the node, has to issue a certification of conformity to precise standards, and the participant has to sign and store it.

Precisely, our system works as follows:

- A supply chain's participant asks an authorized body for a certification.
- If the performed audits by the certification body are successful, the body issues a certification.
- The certification body stores the issued certification in the IPFS.
- Then, the certification body signs the hash returned by IPFS, together with the status of certification and the name of the certification, and stores the signature and other certification data on-chain.
- A verifier that desires to verify the validity of a certification has to perform precise verification steps following the same scheme used to construct the signed data.

3.1. Methodology

Our dApp was developed applying the ABCDE method [39], that is a software development process specific for dApp. This process defines the steps needed to implement a dApp. It entails the definition of the system's goal, its actors, the system requirements, the subdivision of the system in two subsystems, the off-chain (external subsystem, side-server and/or side-client) and on-chain components (smart contract), and hence the design of the two subsystems, making appropriate security assessments and following appropriate patterns for integration and testing (see works [40,41] for more details about security assessments and appropriate patterns). We implemented the on-chain subsystem in Solidity language, and the off-chain in Javascript language, implementing a node.js application, that uses web3.js for the interaction with the on-chain subsystem and a react application for the interaction with the users (hence, the participants in the supply chain). We deployed and tested our system on Ganache, a local blockchain.

The on-chain subsystem is composed of smart contracts that have to implement the registry of the certifications and the needed logic to verify if a certification is valid. To implement the certification registry we referred to the uPort project. Precisely, we

referred to the Ethereum claims registry implemented by ERC780 (ref. https://github.com/ethereum/EIPs/issues/780 (accessed on 15 November 2021)).

Through this standard, a public distributed registry for claims is implemented, and all the on-chain claims can be found in this registry. The ECR780 provides an interface for managing claims, defining the functions *setClaim* and *setSelfClaim* for adding claims, *getClaim* for getting claims, and *removeClaim* for removing claims, and the type of the several entities and objects involved. Precisely, this standard provides that issuer and subject/holder are identified by a key, which is of type bytes32, and that the claims data is stored as type bytes32. The standard provides also the issuance of two events *ClaimSet* and *ClaimRemoved* to keep track of the claims issued and removed. The interface ERC780 above described provides the type *bytes32* for the *value* parameter in the functions *function setClaim* and *function setSelfClaim*, and encourages to set the *key* parameter, which is used to indicate the type of claim issued, equal to the hash of the claim type, such as making the function *keccak256('Owner-Address')*.

Our proposed certification registry has to allow certificate authorities to issue claims/certifications for a precise chain's participant. For the implementation of our certification registry we used the type *struct* in place of the type *bytes32* for the *value* parameter in the functions *function setClaim*, that manages the storage on-chain of all useful data about our food/process certifications (The authors of the ERC780 propose general entries to maintain the code base simple and suggest the implementation of libraries for the conversions between various solidity types.); and we used as the *key* parameter the variable named _IPFShash, a unique identifier as we are going to explain below.

The ERC780 provides that the identity of the claims' issuer are recovered through his/her address, *msg.sender*, to benefit from ethereums signature abstractions. In our system, the issuer's identity is recovered from the *v, r, s* signature parameters since our system has the signature verification in the contract CertificationVerifier.sol. The logic to retrieve the issuer's address used to sign a given certification was implemented following the procedure indicated in https://docs.soliditylang.org/en/v0.8.4/solidity-by-example.html (accessed on 15 November 2021).

3.2. On-Chain Subsystem

The on-chain subsystem is composed of two smart contracts, *CertificationRegistry.sol* and *CertificationVerifier.sol*. Both smart contracts were written in Solidity code and deployed in the Ethereum network. We assumed that each certification body takes care of the deployment of the *CertificationRegistry.sol* smart contract, and of the publication of both contract address and body address. The deployment of the *CertificationVerifier.sol* smart contract and the publication of its address is edited by a single body. In this work to test the proposed system we deployed the smart contracts in a local network, specifically Ganache, by a node.js application.

In the following, the code of the two smart contracts above quoted is shown. The smart contract named *CertificationRegistry.sol* (see Figure 3) implements a certification registry, in which each certification is represented by a variable of type *struct*.

The *struct* defines five variables. Two variables of type *address* that define the issuer's (issuer) and the holder's addresses (holder) of the certification. A variable named signature of type *bytes* that contains all useful data to verify the validity of a certification. Finally, the *struct* defines two variables of type *string*, certificationName that defines the type of the issued certification and dataState that defines the state of a certification that can be Active or Revoked. All other information related to a certification are stored in the IPFS system.

The *CertificationRegistry.sol* smart contract defines a mapping named registry to store all certifications. Mapping in Solidity is seen as hash tables, hence it contains keys and map each of them to a value. In our case, registry maps keys of type *string* to a value of type *struct*, so IPFS hash to certification data. In the code the keys are defined through the parameter _IPFShash that is the unique identifier of the certification. It is the hash that allows to uniquely identify the file stored in IPFS, hence to retrieve a precise certification.

```
contract CertificationRegistry {
  mapping (string => Certification) public registry;
  struct Certification {
    address issuer;
    address holder;
    bytes signature;
    string dataState;
    string certificationName;
  }
    modifier onlyHolder(string memory _IPFShash){
      require(registry[_IPFShash].holder == msg.sender, "access
      denied");
      _;
  }
    address owner;
  constructor() {
      owner = msg.sender;
  }
    modifier onlyIssuer(){
      require(msg.sender == owner, "access denied");
        _;
  }
    function addCertification(
    address _holder,
     bytes memory _signature,
    string memory _dataState,
    string  memory _IPFShash,
    string memory  _certificationName
    )
    public onlyIssuer
    {
        registry[_IPFShash].holder = _holder;
        registry[_IPFShash].issuer = msg.sender;
        registry[_IPFShash].signature = _signature;
        registry[_IPFShash].dataState = _dataState;
        registry[_IPFShash].certificationName = _certificationName;
    }
    function removeCertification(string memory _IPFShash) onlyHolder (
        _IPFShash) public{
        delete registry[_IPFShash];
    }
    function getCertification(string memory _IPFShash) public view
    returns(
    address,
    address,
    bytes memory,
    string memory,
    string memory) {
        return (registry[_IPFShash].holder ,
            registry[_IPFShash].issuer,
            registry[_IPFShash].signature,
            registry[_IPFShash].dataState,
            registry[_IPFShash].certificationName );
}
    function modifyCertificationState(string memory _IPFShash, string
    memory _dataState) onlyIssuer()  public {
    registry[_IPFShash].dataState=_dataState;
  }
}
```

Figure 3. Solidity smart contract CertificationRegistry.

The code shows how only the issuer of the certification can add/store a certification in the register on chain. The contract defines a variable named owner of type address. When deploying a contract, *msg. sender* is the owner of the contract. In our case, the owner is the certification body. The variable owner will have the address of the owner of the contract, hence of the certification body that originally deployed the contract. Through the onlyIssuer() *modifier*, we control the behavior of the addCertification smart contract's function. This modifier allows only the contract owner to run the addCertification function.

The solidity code implemented for the *CertificationVerifier.sol* smart contract to verify the validity of a certification is shown in Figure 4.

```
contract CertificationVerifier {
//ClaimHolder _identity is the address certificationRegistryContract
  function checkCertification(CertificationRegistry _registry, string
      memory  _IPFShash)
    public view
    returns (bool claimValid)
  {
      address holder;
      address issuer;
      bytes memory signature;
      string memory dataState;
      string memory certificationName;
    if (certificationIsValid(_registry, _IPFShash)) {
    // Fetch certification from registry
    ( holder, issuer, signature, dataState, certificationName ) =
        _registry.getCertification(_IPFShash);
    if(keccak256(abi.encodePacked(dataState)) == keccak256(abi.
        encodePacked("Active")))
      return true;
    } else {return false;}
  }
  function certificationIsValid(CertificationRegistry _registry, string
      memory _IPFShash)
    internal view
    returns (bool certificationValid)
    {
      address holder;
      address issuer;
      bytes memory signature;
      string memory dataState;
      string memory certificationName;
      // Fetch certification from registry
      ( holder, issuer, signature, dataState, certificationName ) =
        _registry.getCertification(_IPFShash);
    bytes32 dataHash = keccak256(abi.encodePacked(_IPFShash, dataState,
        certificationName));
      bytes32 prefixedHash = keccak256(abi.encodePacked("\x19Ethereum
          Signed Message:\n32",
      dataHash));
       // Recover address of data signer
       address recovered = getRecoveredAddress(signature, prefixedHash);
    if (recovered == issuer) {
       return true;
        } else{return false;}
    }
    function getRecoveredAddress(bytes memory sig, bytes32 dataHash)
    internal pure
    returns (address addr)
    {
    bytes32 ra;
    bytes32 sa;
    uint8 va;
  // Check the signature length
    if (sig.length != 65) {
    return address(0);
    }
  // Divide the signature in r, s and v variables
    assembly {
    ra := mload(add(sig, 32))
    sa := mload(add(sig, 64))
    va := byte(0, mload(add(sig, 96)))
    }
    if (va < 27) {
    va += 27;
    }
  address recoveredAddress = ecrecover(dataHash, va, ra, sa);
  return (recoveredAddress);
    }
}
```

Figure 4. Solidity smart contract CertificationVerifier.

A certification is valid if it is issued by an authorized body and if its state is *Active*. The checkCertification() function implements the logic just described, while the certificationIs-

Valid() and getRecoveredAddress() functions implement the logic to retrieve the issuer's address used to sign a given certification.

3.3. Off-Chain Subsystem

The off-chain subsystem of the proposed dApp is constituted by web applications running on the browser. They are the grapghic user interfaces (GUIs), hence client applications that allow the participants/consumers to store/read/trace a certification and verify the certification's validity. We used a JavaScript library, precisely React, to implement the GUIs. To interact with our GUIs participants/consumers must have Metamask enabled, to be connected to an Ethereum network node, and to be able to manage their Ethereum accounts, hence their private keys.

When a participant along the supply chain wishes to obtain a certification, they first needs to communicate this request to the authorized bodies. This part of the procedure is of course completely off-chain. The certification body performs the needed audits and if they are successful, it issues the certification. More specifically, the certification body creates a certification, hence a document containing all the useful information about the performed evaluation process of the food/process quality. Then, it stores this document in the IPFS and executes a Blockchain transaction to store all the useful information about it in the certification registry, as described in the previous section. Finally, it passes the IPFS hash returned by the system to the holder. This hash is the unique identifier (IPFShash) of that certification in the system.

The following javascript code describes the steps performed by the certification body to sign the certificate (Figure 5). The same steps have to be followed to construct the verification process of the validity of a certification, as described in the smart contract *CertificationVerifier.sol* that implements the logic of the verification process as illustrated in the previous section.

```
var dataState = "Active";
var IPFShash = "QmdWHcindTyowbPdbvPGEi1kYFC1Zwhn2mouh3xbatY8BP";
var certificationName = "IFS FOOD";
var hashedDataToSign = web3.utils.soliditySha3(IPFShash,dataState,certificationName);
var signature = await web3.eth.sign(hashedDataToSign, ISSUER);
```

Figure 5. Java code.

In our off-chain subsystem when a certification issuer wants to upload the file containing the issued certification into the IPFS, he/she has to install the IPFS client program and add the certification following the indicated procedure. Once the file is uploaded, the IPFS returns its hash. The issuer shares this hash since only with this hash a participant in the chain can have access to the file. Precisely the issuer shares this hash with its holder and store it on-chain to share it with everyone. We designed two GUIs: one for issuers and another for holders and verifiers. The first GUI allows to upload the certifications on the IPFS network and on Ethereum blockchain on the certifications registry. In this first work, the GUI is connected to an IPFS node via localhost, hence a participant that wishes to download a certification, knowing the *IPFS hash* of the certification, has to be connected to one of the IPFS gateways (https://gateway.ipfs.io/ipfs/+{IPFShash}#, accessed on 15 November 2021). The certification body runs the application the first time to deploy automatically the smart contracts. Every certification body has its own *CertificationRegistry.sol* smart contract, hence one contract for each body has to be deployed. Instead, the *CertificationVerifier.sol* can be deployed only once, and can be used by every participant to verify certifications issued by all the certification bodies.

By using this first GUI, issuers interact with the Certification Registry on-chain, and upload the certification on the IPFS. They can select the file to upload on the IPFS and can store on-chain the hash returned by IPFS after the storage of the file on the system, by executing the *CertificationRegistry.sol* smart contract method named addCertification(). They can retrieve some data about a precise certification upon setting in input the identifier of the

certification, that is the hash returned by the IPFS, as already mentioned, by executing the *CertificationRegistry.sol* smart contract method named getCertification(). The certification bodies can change the state of a certification. The state can be *Active* or *Revoked* as already described, and the body has to put in input the value of the state and the identifier of the certification. Hence, the GUI allows to execute the method named modifyCertification-State(). Finally, the GUI allows to verify if the certification is valid or not, upon setting in input the identifier of the certification, by executing the *CertificationVerifier.sol* smart contract method named checkCertification().

By using the second GUI, holders and verifiers can get the certification's data from blockchain, and verify the validity of the certifications downloaded from the IPFS network. The following lines describe the outputs of a nodejs application that performs the steps envisaged by the first GUI (Figure 6).

```
Issuer address: 0x691B027c7734cC9Ef32bd7B0e70c28c35cA8e7fa
Verifier address 0xeA735e7819267CE76A927C2C4099204A01D629E5
Holder 0x0c589dDfd4ad1542C0c74231939416bF9e20cb08
Deploying CertificationVerifier.sol by Issuer: 0x691B027c7734cC9Ef32bd7B0e70c28c35cA8e7fa
Contract deployed at address: 0xa35451772aB3820e0d5410174B7c86F1Eee359EC
Deploying CertificationRegistry.sol by Issuer: 0x691B027c7734cC9Ef32bd7B0e70c28c35cA8e7fa
 Contract deployed at address: 0xA248A284CcEFd8EDe5d1dF08D40f3AFd54624b16
Issuer prepares the signature...
dataState string
Certification State: Active
IPFS hash: QmQ7vnzAksEjU1SaJFczEyMR7NY9sjUb9ZPDDZrv9LG8kr
Certification Name: IFS FOOD
Data Signed: 0xc269bc727e18327ac8b51796ced10464416fc7448022369a066c8c1f39d0698759bd5f43da
1f25d496f10841f65433f55b483576c225a229e06b4510b97f89db00
Issuer adds a Certification...
transaction hash: 0xdb6d4952a705bf670c2ae849ce5d9329f21ec9a3ad2c043af28f7079c1618849
Issuer gets the on-chain certification: Result {
  '0': '0x0c589dDfd4ad1542C0c74231939416bF9e20cb08',
  '1': '0x691B027c7734cC9Ef32bd7B0e70c28c35cA8e7fa',
  '2':
   '0xc269bc727e18327ac8b51796ced10464416fc7448022369a066c8c1f39d0698759bd5f43da1f25d496f
   10841f65433f55b483576c225a229e06b4510b97f89db00',
  '3': 'Active',
  '4': 'IFS FOOD' }
Verifier checks the certification validity...
Is certification valid? true
Issuer modifies the state of certification....
Transaction hash: 0x3ee287f13377a47705ef990a404efa4d03cc58cd93d001e93933553cccf74672
Issuer gets the on-chain certification to verify the state change: Result {
  '0': '0x0c589dDfd4ad1542C0c74231939416bF9e20cb08',
  '1': '0x691B027c7734cC9Ef32bd7B0e70c28c35cA8e7fa',
  '2':
   '0xc269bc727e18327ac8b51796ced10464416fc7448022369a066c8c1f39d0698759bd5f43da1f25d496f
   10841f65433f55b483576c225a229e06b4510b97f89db00',
  '3': 'Active',
  '4': 'IFS FOOD' }
Certification data: Result {
  '0': '0x0c589dDfd4ad1542C0c74231939416bF9e20cb08',
  '1': '0x691B027c7734cC9Ef32bd7B0e70c28c35cA8e7fa',
  '2':
   '0xc269bc727e18327ac8b51796ced10464416fc7448022369a066c8c1f39d0698759bd5f43da1f25d496f
   10841f65433f55b483576c225a229e06b4510b97f89db00',
  '3': 'Revoked',
  '4': 'IFS FOOD' }
Is certification valid? false
```

Figure 6. Outputs of nodejs application.

First of all, the addresses assigned to the issuer, verifier, and holder are shown. Then, the addresses of the deployed smart contracts, together with the contract owner addresses are described. All data needed to prepare the signature, and to upload on-chain a certification, follow. Finally, the results of the tests performed to verify the validity of the uploaded certification and the correct change of the certification's state are illustrated.

4. Conclusions

Today, all the attributes that characterize any identity defining the so-called digital identity are under the control of third parties, who are external to the entity to which they refer. Web giants like Facebook or Google became the keepers of our digital identities.

The identity management system known as the SSI model was born from the need to have greater awareness of the processing of our personal identities and the protection of related data. In this model, technologies such as distributed ledgers, Blockchain, and IPFS allow the development of systems for efficient management of digital identities, and the user is at the center of their digital identity. The user can independently manage her attributes through zero knowledge proof algorithms, deciding from time to time which ones to share and with whom. In the SSI model, the user has a single identity, which is associated with various attributes (called claims) by subjects called issuers. The attributes can be verified in real-time and at any time by any person who has an interest (verifier).

SSI makes possible to develop and improve numerous applications/activities, which are destined to evolve over the next few years as national and international regulations provide a more precise and precise framework. Through the SSI systems it is possible to manage many activities, such as the issuance of an identity document or a university degree, without the need for intermediaries to certify the identity of the owner and the integrity of the digital document issued.

In this work, a system based on the concepts of the SSI is proposed to manage certifications along the supply chain. The proposed system assumes that the user (holder) requests a quality certification for their product/process, and that a certifying body issues a certificate by storing it in the IPFS and storing only some key information on the chain. Specifically, the information that allows a verifier to verify the validity of the certification that is stored on the chain. The system automatically manages the recovery of certificates and the verification of their validity. In future, the proposed system may be easily modified to be integrated into a real complete SSI system that manages all claims belonging to every individual.

Author Contributions: Conceptualization, L.C. and R.T.; methodology, L.C.; software, L.C.; validation, L.C.; formal analysis, L.C.; writing—original draft preparation, L.C.; writing—review & editing, L.C.; supervision, R.T. and M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been funded by the following grants from Regione Autonoma Sardegna: "C.A.S.C.O. Complete Assessment of Smart contracts and Coin Offers", Code: 2014IT16RFOP015, POR FESR Sardegna 2014-2020 Asse I–Azione 1.2.2; "CAFCha—Certification of AgriFood Chain"-POR FESR 2014–2020-Asse 1, Azione 1.1.3 Strategia 2; and by "ABATA Application of Blockchain to Authenticity and Traceability of Aliments" project, funded by Italian Ministry for Economic Development, National Operational Program "Enterprises and Competitiveness", project No. F/200130/01-02/X45. 2.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. EmergenResearch. Food Certification Market By Application (Processed Meat & Poultry, Organic Food, Dairy Products, Seafood, Beverages, Infant Food, Others), By Type (ISO 22000, Halal, Kosher, SQF, FSSAI, BRC, Others), By Risk (Low Risk, High Risk), and By Region, Forecasts to 2027. 2020. Available online: https://www.emergenresearch.com/industry-report/food-certification-market (accessed on 3 November 2021).
2. GlobeNewsWire. Global Food Certification Market (2020 to 2025)—Adoption of Halal, Organic, and 'Free-From' Food Certifications Presents Opportunities. 2020. Available online: https://www.globenewswire.com/news-release/2020/11/26/2134372/28124/en/Global-Food-Certification-Market-2020-to-2025-Adoption-of-Halal-Organic-and-Free-From-Food-Certifications-Presents-Opportunities.html (accessed on 3 November 2021).
3. GlobalFoodSafetyResource. The Importance of Food Safety Certification. 2021. Available online: https://globalfoodsafetyresource.com/food-safety-certification/ (accessed on 3 November 2021).

4. Manyika, J.; Ramaswamy, S.; Khanna, S.; Sarrazin, H.; Pinkus, G.; Sethupathy, G.; Yaffe, A. *Digital America: A Tale of the Haves and Have-Mores*. Available online: https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/digital-america-a-tale-of-the-haves-and-have-mores (accessed on 7 October 2021).
5. Trienekens, J.H.; Wognum, P.; Beulens, A.J.; van der Vorst, J.G. Transparency in complex dynamic food supply chains. *Adv. Eng. Inform.* **2012**, *26*, 55–65. [CrossRef]
6. Chopra, S.; Meindl, P. *Supply Chain Management: Strategy, Planning, and Operation*; Prentice-Hall: Upper Saddle River, NJ, USA, 2001.
7. Dudin, M.N.; Smirnova, O.O.; Vysotskaya, N.V.; Frolova, E.E.; Vilkova, N.G. The Deming Cycle (PDCA) Concept as a Tool for the Transition to the Innovative Path of the Continuous Quality Improvement in Production Processes of the Agro-Industrial Sector. *Eur. Res. Stud. J.* **2017**, *20*, 283–293. [CrossRef]
8. CONTROLUNION. Food Safety. Available online: https://certifications.controlunion.com/en/industries/food-safety (accessed on 3 November 2021).
9. BTSA. The Main Certificates of the Food Industry. Available online: https://www.btsa.com/en/main-food-certificates/ (accessed on 3 November 2021).
10. ISO. Popular Standards ISO 22000 Food Safety Management. Available online: https://www.iso.org/iso-22000-food-safety-management.html (accessed on 3 November 2021).
11. IFS. Global Safety and Quality Standards. Available online: https://www.ifs-certification.com/index.php/en/ (accessed on 3 November 2021).
12. Aung, M.M.; Chang, Y.S. Traceability in a food supply chain: Safety and quality perspectives. *Food Control* **2014**, *39*, 172–184. [CrossRef]
13. Rejeb, A.; Keogh, J.G.; Treiblmaier, H. Leveraging the internet of things and blockchain technology in supply chain management. *Future Internet* **2019**, *11*, 161. [CrossRef]
14. Feng, T. A supply chain traceability system for food safety based on HACCP, blockchain & Internet of things. In Proceedings of the 14th International Conference on Services Systems and Services Management, ICSSSM 2017, Dalian, China, 16–18 June 2017. [CrossRef]
15. Baralla, G.; Pinna, A.; Corrias, G. Ensure Traceability in European Food Supply Chain by Using a Blockchain System. In Proceedings of the 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB '19), Montreal, QC, Canada, 27 May 2019; pp. 40–47. [CrossRef]
16. Baralla, G.; Pinna, A.; Tonelli, R.; Marchesi, M.; Ibba, S. Ensuring transparency and traceability of food local products: A blockchain application to a Smart Tourism Region. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e5857. [CrossRef]
17. Pranto, T.H.; Noman, A.A.; Mahmud, A.; Haque, A.B. Blockchain and smart contract for IoT enabled smart agriculture. *PeerJ Comput. Sci.* **2021**, *7*, e407. doi:10.7717/peerj-cs.407. [CrossRef]
18. Huang, H.; Zhou, X.; Liu, J. Food Supply Chain Traceability Scheme based on Blockchain and EPC Technology. Available online: https://www.semanticscholar.org/paper/Food-Supply-Chain-Traceability-Scheme-Based-on-and-Huang-Zhou/8f7c7759f8f9b67b6800879def1c76d08f88a9fd (accessed on 7 October 2021)
19. Xu, Q.; Song, Z.; Mong Goh, R.S.; Li, Y. Building an Ethereum and IPFS-Based Decentralized Social Network System. In Proceedings of the 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), Singapore, 11–13 December 2018; pp. 1–6. [CrossRef]
20. Chen, Y.; Li, H.; Li, K.; Zhang, J. An improved P2P file system scheme based on IPFS and Blockchain. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 2652–2657.
21. Ali, M.S.; Dolui, K.; Antonelli, F. IoT data privacy via blockchains and IPFS. In Proceedings of the Seventh International Conference on the Internet of Things, Linz, Austria, 22–25 October 2017; pp. 1–7.
22. Zheng, Q.; Li, Y.; Chen, P.; Dong, X. An innovative IPFS-based storage model for blockchain. In Proceedings of the 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), Santiago, Chile, 3–6 December 2018; pp. 704–708.
23. Norvill, R.; Fiz Pontiveros, B.B.; State, R.; Cullen, A. IPFS for Reduction of Chain Size in Ethereum. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 30 July–3 August 2018; pp. 1121–1128. [CrossRef]
24. Cocco, L.; Mannaro, K.; Tonelli, R.; Mariani, L.; Lodi, M.B.; Melis, A.; Simone, M.; Fanti, A. A Blockchain-Based Traceability System in Agri-Food SME: Case Study of a Traditional Bakery. *IEEE Access* **2021**, *9*, 62899–62915. [CrossRef]
25. López, M.A. *Self Sovereign Identity The Future of Identity: Self-Sovereignity, Digital Wallets, and Blockchain*. Available online: https://etradeforall.org/documents/idb-self-sovereign-identity-the-future-of-identity-self-sovereignity-digital-wallets-and-blockchain-2/ (accessed on 7 October 2021).
26. Anonymous. Proposal for a Regulation of The European Parliament and of the Council amending Regulation (EU) No 910/2014 as regards establishing a framework for a European Digital Identity. Available online: https://www.europeansources.info/record/proposal-for-a-regulation-amending-regulation-eu-no-910-2014-as-regards-establishing-a-framework-for-a-european-digital-identity/ (accessed on 7 October 2021).
27. Nguyen, B.M.; Dao, T.C.; Do, B.L. Towards a blockchain-based certificate authentication system in Vietnam. *PeerJ Comput. Sci.* **2020**, *6*, e266. doi:10.7717/peerj-cs.266. [CrossRef] [PubMed]

28. Bouras, M.A.; Xia, B.; Abuassba, A.O.; Ning, H.; Lu, Q. IoT-CCAC: A blockchain-based consortium capability access control approach for IoT. *PeerJ Comput. Sci.* **2021**, *7*, e455. doi:10.7717/peerj-cs.455. [CrossRef] [PubMed]
29. Weingaertner, T.; Camenzind, O. Identity of Things: Applying concepts from Self Sovereign Identity to IoT devices. *Peer Rev. Res.* **2021**. [CrossRef]
30. Bartolomeu, P.C.; Vieira, E.; Hosseini, S.M.; Ferreira, J. Self-Sovereign Identity: Use-cases, Technologies, and Challenes for Industrial IoT. In Proceedings of the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 10–13 September 2019; pp. 1173–1180. [CrossRef]
31. Niya, S.R.; Jeffrey, B.; Stiller, B. KYoT: Self-sovereign IoT Identification with a Physically Unclonable Function. In Proceedings of the 2020 IEEE 45th Conference on Local Computer Networks (LCN), Sydney, Australia, 16–19 November 2020.
32. Liu, Y.; Lu, Q.; Paik, H.Y.; Xu, X. Design Patterns for Blockchain-based Self-Sovereign Identity Share on. In Proceedings of the European Conference on Pattern Languages of Programs 2020 (EuroPLoP '20), Virtual Event, Germany, 1–4 July 2020; pp. 1–14. [CrossRef]
33. Lipińska, A. uPort Serto Ecosystems: Creating Trusted Data Networks between Businesses and Individuals. 2019. Available online: https://medium.com/uport/uport-serto-ecosystems-creating-trusted-data-networks-between-businesses-and-individuals-ff21c9368d3b (accessed on 14 June 2021).
34. Coutts, V. The Who's Who of Decentralized Identity Systems. 2018. Available online: https://medium.com/linum-labs/the-whos-who-of-decentralized-identity-systems-433b2dd9a195 (accessed on 14 June 2021).
35. Braendgaard, P. Different Approaches to Ethereum Identity Standards. 2018. Available online: https://medium.com/uport/different-approaches-to-ethereum-identity-standards-a09488347c87 (accessed on 14 June 2021).
36. Nelapati, Y. A Solution for Decentralized Identity. 2019. Available online: https://medium.com/makersplace/a-solution-for-decentralized-identity-a867d39b8726 (accessed on 14 June 2021).
37. Thorstensson, J. ERC1056 ERC780—An Open Identity and Claims Protocol for Ethereum. 2018. Available online: https://medium.com/uport/erc1056-erc780-an-open-identity-and-claims-protocol-for-ethereum-aef7207bc744 (accessed on 14 June 2021).
38. Santos, J. First Impressions with ERC 725 and ERC 735—Identity and Claims. 2018. Available online: https://hackernoon.com/first-impressions-with-erc-725-and-erc-735-identity-and-claims-4a87ff2509c9 (accessed on 18 May 2021).
39. Marchesi, L.; Marchesi, M.; Tonelli, R. ABCDE—Agile Block Chain DApp Engineering. *Blockchain Res. Appl.* **2020**, *1*, 100002. doi:10.1016/j.bcra.2020.100002. [CrossRef]
40. Marchesi, L.; Marchesi, M.; Pompianu, L.; Tonelli, R. Security checklists for Ethereum smart contract development: Patterns and best practices. *arXiv* **2020**, arXiv:2008.04761.
41. Marchesi, L.; Marchesi, M.; Destefanis, G.; Barabino, G.; Tigano, D. Design Patterns for Gas Optimization in Ethereum. In Proceedings of the 2020 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE), London, ON, Canada, 18 February 2020; pp. 9–15. [CrossRef]