applied sciences

MDPI

Article

ID-Service: A Blockchain-Based Platform to Support Digital-Identity-Aware Service Accountability

Luciano Argento [1], Francesco Buccafurri [2,*], Angelo Furfaro [3], Sabrina Graziano [1], Antonella Guzzo [3], Gianluca Lax [2], Francesco Pasqua [1] and Domenico Saccà [3]

[1] Open Knowledge Technologies, 87036 Rende, Italy; luciano.argento@okt-srl.com (L.A.); sabrina.graziano@okt-srl.com (S.G.); francesco.pasqua@okt-srl.com (F.P.)
[2] Department of Information Engineering, Infrastructure and Sustainable Energy (DIIES), University Mediterranea of Reggio Calabria, 89122 Reggio Calabria, Italy; lax@unirc.it
[3] Department of Informatics, Modeling, Electronics and Systems Engineering (DIMES), 87036 Rende, Italy; angelo.furfaro@unical.it (A.F.); antonella.guzzo@unical.it (A.G.); sacca@unical.it (D.S.)
* Correspondence: bucca@unirc.it

Abstract: Accountability refers to the need of individuals or organizations to account for their activities, accept responsibility, and disclose results in a transparent manner. Nowadays, the pervasivity of digital systems is making increasingly critical security, reliability, and trustworthiness of such services. When a service is delivered by involving different (eventually conflicting) parties, accountability could be achieved by including in digital transactions a trusted third party (TTP). Blockchain decentralizes trust, thus avoiding to rely on a single TTP. However, to deal with accountability in concrete solutions, the issue of securely integrating digital identity and Blockchain should be solved. The paper describes the results of a three-year research project merging academic and industrial expertise, to design and implement a Blockchain-based platform for service accountability integrating eIDAS-compliant Public Digital Identity. The platform has been used in several real-life contexts made available by industrial project partners, which demonstrated the effectiveness and novelty of the solution.

Keywords: Cross-Organizational Workflows; smart contracts; DLT; eIDAS; certification

Citation: Argento, L.; Buccafurri, F.; Furfaro, A.; Graziano, S.; Guzzo, A.; Lax, G.; Pasqua, F.; Saccà, D. ID-Service: A Blockchain-Based Platform to Support Digital-Identity-Aware Service Accountability. *Appl. Sci.* **2021**, *11*, 165. https://doi.org/10.3390/app11010165

Received: 19 November 2020
Accepted: 21 December 2020
Published: 26 December 2020

1. Introduction

In today's business, globalization has leading new markets and opportunities and, at the same time, the need to address rapidly changing market conditions in a more flexible and effective way than ever before. To this aim, the emerging business model adopted by organizations is to focus on their competitive advantage, performing only those functions for which they have expert skills and they complement their offering through partners and suppliers. In this context, Cross-Organizational Workflows (COWs) [1–3] came into play. However, the design and management of this kind of workflows are more difficult than traditional ones, also because the participation of possibly conflicting parties enforces higher levels of assurance about security and trustworthiness features. This is complicated by the intrinsic lack of global workflow schemes (managed by the engine of the system) and central coordinating bodies, due to the decentralized nature of COWs.

One of the features that may provide digital services with a competitive advantage is certainly accountability. A service is accountable if trusted mechanisms are enabled to trace actions and link them in a secure way with the involved actors, in such a way that responsibility can be claimed with an appropriate level of assurance, and, thus, actions cannot be repudiated or falsely attributed. Obviously, accountability is strictly related to reliable identity management. No responsibility can be claimed if the involved digital entities are not securely linked to reliable digital identities. The other need is the ability to trace workflows at least in the sensitive points (e.g., inter-organization actions) in a way that integrity and authenticity are securely preserved.

The approach followed in our project to attribute certain identity to the critical actors of services is to rely on a Public Digital Identity Systems compliant with the EU regulation eIDAS [4]. It is worth noting that the trusted third parties required by the eIDAS ecosystem (i.e., the Identity Providers) are definitely not in conflict with the distributed nature of COWs because they can be view as orthogonal entities provided *for free* in real-life contexts (at least in the European Union). Instead, to accomplish accountability, without involving in the COWs dedicated central TTPs, we leverage the Blockchain technology as a platform for both secure tracing and trusted execution. Blockchain has proved to lend itself well to the implementation of cooperative processes that involve different organizations [5–7], because:

1. Blockchain enables these processes to be executed in a distributed manner without delegating trust to central authorities nor requiring mutual trust between each pair of parties.
2. Blockchain can serve as an immutable public register in which to store the history of the interactions, structured as a feedback control mechanism of information exchange between a sender and a receiver, at a certain point in time, providing a clear and openly verifiable indication of unexpected behaviours.
3. The logic of interaction of COWs could be codified within smart contracts ensuring the correct execution of the shared process. Smart contracts are a further element of control of the process execution, as they only accept coded interactions and only if executed by the participants who have the necessary authorizations.

Unfortunately, no native mechanism in Blockchain exists to manage securely the identity of the involved actors.

Although the use of the blockchain in the implementation of organizational processes is currently a growing trend, as shown by the various applications and experiments in the field of academic and industrial research, there are still critical issues for which efficient and robust solutions must be found. One of them concerns the certification of the identity of the participants in the process, which is referred to by the term *accountability*. Indeed, in cooperative processes, the entities belong to different organizations so that they are intrinsically "untrusted" and each of them assumes the obligation and responsibility to carry out actions whose outcome also depends on the obligation and responsibility of the other parties involved in the process. Thus, accountability requires that the parties involved in the process (and their actions) must be identifiable in a secure way: other requirements are non-repudiation of activities/tasks involved in the process, logging, and monitoring of events. It is worth noting that the importance of accountability has been recently remarked also by the European General Data Protection Regulation in the context of privacy.

In this paper, we present an accountability system able to appropriately integrate the Blockchain technology with the Public Digital Identity Ecosystem within a multi-layer architecture providing a platform, called *Id-Service*, implementing a set of APIs for the design, management, and execution of accountable COWs. A nice feature of the system is to associate an action with the digital identity of the actor performing this action and that guarantees the integrity and non-repudiation of this association with respect to the attacks currently known. Specifically, our solution provides a secure, certified, traced, and unmodifiable service capable of supporting any organization in which Identity and Accountability represent hard requirements. Observe that, even if blockchain has been already exploited to achieve reliability and security in other contexts, the assumption of responsibility of the participants in the different tasks of the process cannot be guaranteed by only the blockchain because, by its very nature, it is hard to establish a direct link to the real-world identities of the users involved in a given transaction. This is one of the main issues solved by our system. We describe a high-level view of this platform which actually collects a number of different innovative solutions from a scientific point of view, as witnessed e.g., by [8–11]. We also provide some details about the architecture and the implementation of our solution, whereas a real-life example of its use for accountability is described in Section 3.

The structure of the paper is as follows. In the next section, we introduce our proposal and present the architecture of our platform, which is composed of two modules. In Section 3, we show the application of our solution to a real-life scenario to certify the operations performed by different organizations that cooperate in the production and delivery of high-quality biomass. The related work is discussed in Section 4. Finally, in Section 5, we draw our conclusions.

2. ID-SERVICE Framework

The architecture of our platform, which is depicted in Figure 1, is composed of two modules, the *Accountability Certification* and the *Accountability Workflow*. The former implements an aptly modified Ethereum [12] permissioned blockchain and serves as a distributed solution with immutability and non-repudiation features. The latter provides a higher level support for Cross-Organization Workflows and uses functionalities offered by the Certification platform to obtain all the certified information required for a strongly accountable operation. Both the modules provide client applications with suitable APIs, to offer services such as integrated private key management, transaction signature support, and seamless identity provider integration.

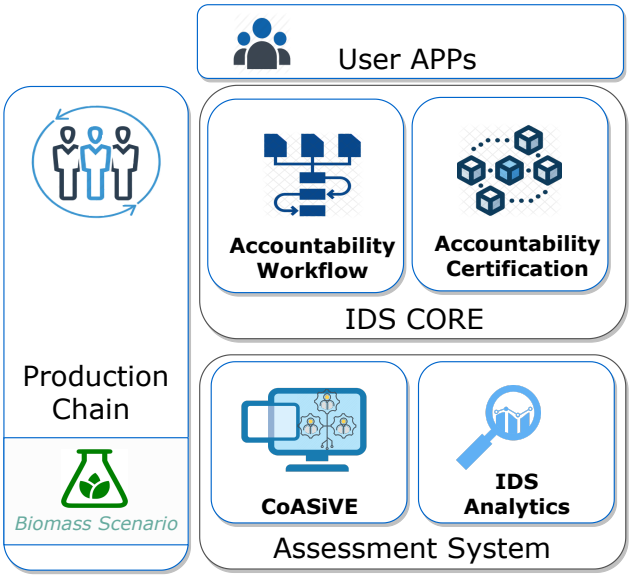


Figure 1. The architecture of the platform.

We also included the *Assessment System* module into the framework, which offers tools developed for analytics support (*IDSAnalytics*) and (*CoASiVE—Combined Agent-based Simulation and Virtual Environments*) for the validation of complex processes and systems. Both sub-modules have also been used as a development and validation tool to aid all functionalities in the framework itself to great proficiency and effect.

2.1. Accountability Certification

This module holds the two technological pillars on which our solution relies, which are digital identity and Blockchain. As for the identity, this module is capable of handling different authentication protocols and identities from different Identity Providers. In particular, to make our platform compliant with the emergent EU ecosystem, we provided the integration with the Italian Public Digital Identity System, called SPID, which is eIDAS compliant [13].

Our modifications to Ethereum encompassed as little as possible the native Golang level [14]. We restrained ourselves by implementing features using smart contracts where applicable. We used the Solidity programming language to implement smart contracts in order to preserve as much as possible the original open-source code, and not to introduce

any incompatibility with known tools for Blockchain and smart contract development (for example, Truffle). One of the main specific features introduced is the presence of *Accountability Nodes* (ANs), specialized and trustworthy master nodes [15]. ANs are considered service providers w.r.t. users and operate as user's delegates, in such a way that any interaction of the users with the Blockchain happens through them and the gas they provide.

Any miner with enough funds can become an AN by paying collateral using native token trading features provided by the Blockchain platform or a custom token developed according to the ERC-223 standard. As compensation for their efforts, ANs receive a higher block reward than all other miners. Any unintended behaviour of an AN is carried out with the explicit risk of losing their paid collateral forever, with no way to cash it back.

One of the purposes of this module is to provide user identification. To this aim, users are asked to generate a Blockchain private/public key pair, and the tuple $\langle UID, PK \rangle$, where *UID* is a user identifier and *PK* is the public key, is written in the Blockchain by ANs as a registration transaction. This way, any transaction done by this key pair can be associated with a real-life user. The *UID* is generated by the module in such a way that we can unambiguously refer to the very same identity regardless of specific Identity Providers (IdPs) or protocol employed. Registered users may have a different set of permissions, granted by their respective companies, and they may or may not interact among themselves (i.e., they can exchange between each other signed messages containing arbitrary data).

Since ANs generate the registration transactions, their public key must be well-known. To this aim, we assume that every AN is identified by a human-readable string (for example, the web domain associated with AN) and we exploit an identity-based encryption approach [16] to implement the integration into the Blockchain. This approach allows us to associate a string (in our case the AN's identifier) with an asymmetric (IBE) public key (IBE key), which can be computed by anyone. The IBE private key can be obtained only by who is able to prove to be the owner of the given identifier. Specifically, any AN generates a Blockchain transaction containing (1) the reference to its Blockchain public key and (2) the signature of this reference by the IBE private key. This way, any user can obtain the Blockchain public key of an AN, starting from its string identifier.

2.2. Accountability Workflow

The Accountability Workflow Module is another IDService core module, as it interacts with the participants of the process (top layers) through appropriate APIs for controlling and managing the flow of information in and out the Blockchain. Each state of the workflow is associated with unique mapping data contained within an instance of an information trade smart contract, in order to keep track of the process execution. In other words, when a transaction is submitted to the Blockchain, a unique identifier is stored on the chain for such an execution. Any event emitted by the smart contract will contain a reference to the previous state in the process, in order to maintain an execution trace chain. Another responsibility of this module is the automatic creation, deployment, and triggering of smart contracts, but it is also responsible for transaction validation and verification. When a new transaction (e.g., a purchase order) has to be store by the platform, the workflow engine is invoked to handle and guide all operations through the architecture. Its main responsibility is to store the transaction data, such as the address of the sender, the address of the receiver, the timestamp, the value of the transaction, and other information on the Blockchain ledger. Moreover, accountability is achieved by assuring a direct connection with the Certification Platform, since each participant of the workflow has to be first authenticated from the Certification Platform and, then, an authentication token to access the workflow platform through API invocation is obtained.

As a quick reminder, a *synchronization point* is a certain step in a multi-organization workflow wherein a critical information exchange happens between two parties belonging to the workflow itself. This particular step receives way further value and significance when it gets certified by our framework: neither of the two parties can, in the future,

claim anything different than what has already been written on chain. This kind of immutable non-repudiation is key in delivering true value and enriching any, already set and experience-proven, workflow, without requiring extensive adaptations in order to comply with our framework. Any information exchange is modeled as a finite state machine, in which one participant, named *sender*, must relay some information (a hash linking to an off-chain resource), to another participant, an intended *receiver*, before an intended expiration date. We do not store any real information that's being traded on chain, only the hash reference. Users are free to choose any method they see fit, be it e-mail or instant messaging, in order to communicate the actual information between one another. Once a trade has been initiated, the intended receiver can decide to accept the information trade as legitimate or reject it if anything does not check out, like the hash reference not correctly matching the received data off-chain. Either way, any action taken by the two participants is recorded forever on the blockchain, and as such, in the future, neither of them will be able to deny what truly happened.

IDS API

By leveraging the `web3.js` library, an API layer was built on top of the Blockchain core, charged with reducing friction between higher, web GUI layers, and the bare chain node. This API structure holds almost no state, other than useful metrics for diagnosis, and operates as a `node.js` endpoint collection.

Web GUIs and mobile applications can be easily built on top: the only hard requirement is a proper implementation of any chosen IdP protocol scheme. Private key management, as a means of transaction signature, is always considered as an end-user responsibility in the Blockchain world, and we decided to keep this feature intact. Any change would have been a critical weakening of non-repudiation, only to have enhanced usability. Every signature check is done on Blockchain, by using cryptographic primitives offered by the solidity language and the EVM, such as *ecrecover()*. The API layer only does value format checks for transactions, that is, every supplied parameter for a given transaction request should be present and accounted for. We have willingly chosen to check as much as possible on chain, and that means both semantic preconditions for a given transaction and syntactic adherence for parameters.

2.3. Assessment System

The *Assessment System* layer supports the assessment of the set of techniques, algorithms, and methodologies used to assess complex processes and systems. Being able to evaluate and predict the impact of one or more potential changes on a system or process of interest may be extremely helpful for a company. As a consequence, the Assessment layer provides a very important tool for Change Management activities that focus on the identification of actions to take before, during, and after a change of interest. Assessment is achieved by means of a methodology that consists of combining *digital virtual environments* (*DVE*), *simulation* and *analytical techniques* to create, validate and perform stress tests on new scenarios related to complex production chains that involve different organisations which use digital systems and services concurrently (see [17] for details). The layer comprises two components, namely *CoASiVE* and *IDS Analytics*. The former embodies the combination of DVEs and simulation and is responsible for validating processes and systems. The idea behind the digital virtual environments is to be able to assess the performance of a system, among other things, without the need of deploying the real system and face significant costs [18,19]. A DVE has great potential, in that it offers a very flexible tool for building a virtual system. Indeed, depending on the study to conduct, there may be no need to recreate the entire system inside the virtual environment. In that case, only a subset of the system's components would be deployed inside the virtual environment, which could make the deployment process lighter and easier to perform. Simulation is the element that creates dynamic and complex scenarios for evaluating processes and systems. When combined with DVEs, simulation allows for the reproduction of entities characterised by complex be-

haviours inside a virtual environment. At the moment these entities interact with each other and the virtual system, they create scenarios of interest. Specifically, we decided to resort to agent-based simulation [20,21]. Agents are autonomous, social, reactive, and proactive software entities that live and act in an environment within which they may be stimulated by events. These entities are able to reproduce complex behaviours of both humans and systems. The above-mentioned features, as well as others, make the agent a powerful tool for building complex and dynamic scenarios. Simulation is then exploited to conduct extensive experimental campaigns in order to validate processes and/or test the behaviour of a target system. At the end of the campaign, we get a set of measurable indicators. Typically these indicators undergo an analytical study whose goal is to gain new knowledge and draw conclusions on the experiments conducted. IDS Analytics is responsible for analysing the data produced by both the simulator and the Blockchain. It is developed on top of `Kibana` and `Elastic search`, for analysing simulation and Blockchain-related data, and `Fluxicon Disco` [22], for process mining, after converting said data in the XES format. Both intended workflow and trace exception detection are expected outputs of this activity.

3. Case Study: A Biomass Production Chain

Among the real-life cases in which our platform has been tested, we describe here its application to the field of biomass production. We chose this scenario mainly because the addressed problem is representative, complex, and timely. This case study is a good test-bed as it inherently involves Cross-Organizational Workflows, in which different and conflicting organisations cooperate for achieving a common goal (i.e., producing and delivering high-quality biomass to the final customer). Moreover, due to legal requirements, there is a strict need of accountability with respect to subjects with assured and publicly verifiable identity. Therefore, this domain collects all the requirements we fulfill in our platform. As a matter of fact, one of the industrial partners of the project works in this field (specifically, in the agro-energy sector), and the above considerations were the basis of the project partnership composition, also to test our solution in a real-life environment. The experimentation has been conducted during the run of the project, from September 2019 to March 2020.

Selection of test objects. We identified a process that could benefit from its integration with a blockchain-based technology. Specifically, we chose to model a biomass production chain, which involves many different organisations that cooperate to pursue a common goal, i.e., producing and delivering high-quality biomass. The process was chosen due to the existence of accountability and traceability issues [23,24], and how adversely these impact the end product's quality. Accountability is a property of utmost importance in this domain due to the strict need for compliance to norms and regulations as well as the high level of legal and non-legal responsibilities related to actions performed by the participants. Traceability is another very important property, which is related to visibility, controllability, and security requirements. The production chain is characterised by intra and inter-organisation interactions.

Research Methodology. The chosen methodology included: (1) implementation of the selected process; (2) simulation and assessment; (3) performance analysis of blockchain features; (4) process mining analysis, after extraction of simulation logs from the blockchain.

The accountability issue arises with inter-organisation interactions: while the identities of a group of employees are clearly defined within an organisation, thus enabling accountability, the same does not necessarily hold outside an organisation. The main idea behind the case study was to address the above-mentioned issues by integrating the process with a blockchain-based technology, combined with digital identities. See Section 3.2 for additional details on the implemented system. The result of such integration is a process in which actors that participate in inter-organisation interactions use an identity-based digital service to certify the operation(s) performed. The certification consists of writing an association between the actors' identity and operation-related documents on the blockchain

so that a permanent record is created. The record will serve as a verifiable and immutable trace of what has happened between two participants.

The domain process that was considered for the case study is briefly described below.

The biomass production chain involves a lot of different actors which are the following: *landowner*, *supplier*, *wood company*, *transformer operator*, *electricity company*, *competent bodies*. Some of the mentioned actors may be represented by a single organisation: for example, supplier, wood company, and transformer operator may coincide.

The process can be described with three macro activities, listed in order of execution: *Preliminary Activity*, *Building Site Activity* and *Delivery Activity*. Below we briefly describe each of these activities, assuming that the supplier also acts as a wood company and transformer operator. In the Preliminary Activity, the supplier sends a competent body an authorisation request for harvesting raw materials and stipulates a contract with one or more landowners. If everything goes well, the Building Site Activity may start. The supplier opens the building site wherein biomass is produced. At the end of the transformation and production activities, the supplier prepares the shipment. At this point, the last macro activity begins. One or more logistics operators take part in the process to deliver biomass produced by the supplier. The process ends when the biomass is delivered and sold to the electricity company. The remainder of this section provides details on the usage of the platform.

3.1. Process Model Design

The process model design is the starting point of the Simulation and Assessment phase. The product of this phase supports both the process mining analysis and the simulation-based modelling and assessment process.

We modelled the production chain in order to emphasise the entities (participants and actors) that participate in the process and the interactions that are relevant from the accountability and traceability perspective. The result of the modelling phase is depicted in Figure 2. The model captures the most important actors, participants, operations that need to be made accountable and states of the production process. Upon the successful execution of the operations the process moves from the current state to the next one. Process states are represented as rectangles with rounded angles, whereas the operations as arrows that connect two states. The actors are listed inside the rectangle at the bottom left of the figure. For the sake of clarity, we refer to organisations as actors and employees as participants. Each actor's name is written with a distinct colour and preceded by a label that is used to indicate the organisation a participant works for. Each operation has a description, reported inside a dashed rectangle, about the agents participating in the operation (note the preceding label), the operation's name and a few examples of operation-related data. Participants that are not directly involved in the process, like employees of a competent body, were not included in the representation. The model assumes that five actors participate in the process: *Buyer*, *Supplier*, *Logistics operator*, *Landowner* and *Specialised company*. The remaining organisations that were previously mentioned are not directly involved in the process, because they do not need the operations performed to be both traceable and accountable. The model assumes that Specialised company serves as both wood company and transformer operator. Keep in mind that upon the completion of an operation the participants use the certification service to certify the operation they've just performed in order to make the process move from the current state to the next one. The following description specifies the state triggered by each operation between parenthesis. The first four states are part of the Preliminary Activity. The base scenario considered assumes that the specialised company first has to ask for permission to harvest raw materials and then it can stipulate a contract with a landowner. The specialised company sends a competent body an authorisation request (*Authorisation requested*). After receiving a positive response (*Authorisation granted*), the company stipulates a contract with a landowner (*Specialised company acquired biomass*). The Preliminary Activity ends with the specialised company selling the biomass to be produced to the supplier, whose purpose is to sell the biomass to

the buyer (*Specialised company sold biomass*). In the Building Site Activity the specialised company opens a building site (*Building site opened*) wherein raw materials are transformed into biomass. Upon the end of the production of biomass one or more logistics operators are contacted for the shipment (*Biomass shipment*). This is where the last macro activity begins. The logistics operator elaborates a delivery plan (*Ready to deliver biomass*). Upon the delivery of the entire product (*Biomass delivered*), a participant working for the supplier verifies if all expected biomass was received (*Supplier received biomass*). Thereafter, the supplier sells the buyer the biomass received in the previous phase (*Biomass sold to the buyer)*. The process concludes with the acquisition of biomass by the buyer (*Biomass is transferred to the buyer* and *Buyer obtained biomass*).

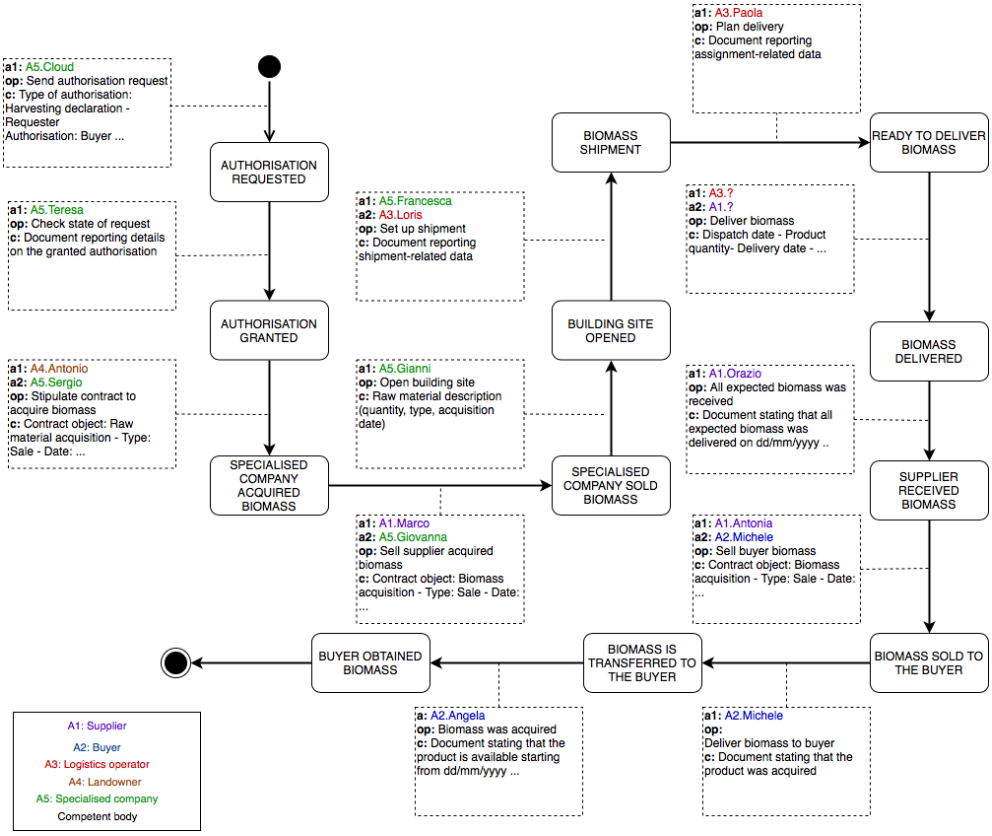


Figure 2. Model of a simple biomass production chain process.

3.2. Test Blockchain

For this case study, domain analysis pointed towards an Ethereum-like implementation with blockchain 3.0 elements, in order to integrate master nodes, aptly named *Accountability Nodes* (ANs).

ANs serve as service providers w.r.t. users, and deploy user signed writes onto the blockchain. We generated a Docker swarm-based deployment suite, capable of instantiating a full test blockchain network in one go, complete with an implementation of our PIM smart contract, with other contracts designed to handle user login, and distributed public key management.

3.3. PIM Smart Contract Implementation

The *platform-independent model specification*, PIM in short, gave clear guidance in the implementation efforts to be carried out, once a definite technological choice had been made, according to domain-specific criteria.

We used Solidity, the premier programming language for Ethereum Smart Contracts at the moment. The implementation created retains its property of being process agnostic. That means: regardless of the process' specific field of application, a smart contract mod-

elled according to the specified requirements, supports subsequent process mining efforts easily. The *finite state machine* is also a kind of behaviour that can easily be created using language-specific features, like invocation modifiers.

Code must, not only, support the PIM, but also respect the platform uniqueness. For example in Ethereum, every transaction or, specifically, smart contract call, costs a certain amount of gas. Gas is a way to measure the resources needed to accomplish a certain computation on the distributed, EVM based network. The deployment itself of a smart contract also costs gas, and such price heavily depends on code size: keeping code short, concise, and reusable is considered a definite plus.

So, it is preferable to keep code as simple as possible, avoiding loops, redundant writes, and unnecessarily complicated data structures.

Also, it is very unwieldy to update smart contract code, while retaining all the information contained within. Standard techniques should be employed, such as the usage of specific design patterns (e.g., *Check-Effects-Interactions* and *Create-Read-Update-Delete*), and code must then be thoroughly tested.

It should be noted that we used a slightly modified version of the classical Check-Effects-Interactions pattern: we moved away from the construct *require* using a simple *if* check to emit an event logging the unexpected condition (Listing 1).

Listing 1. smart contract code.

```
if(log.isLogged(_sender)==false){
    emit TradeFailed(_sender, _receiver, _valueTradedHash, msg.sender);
    return false;
}
```

In order to correctly implement our PIM, we created a function for each edge connecting one state to another, and, for each function, exactly one event describing a correct execution, and a certain number of other events emitted whenever execution encountered a particular error. Data were laid out to support the multi-FSM behaviour: the Create-Read-Update-Delete pattern was used to keep track of every new information exchange performed by actors. Structs and enums, instead, were put in place in order to contain relevant data, such as actors' UIDs, and timestamps. As an example, here's the signature of the *OfferTradeSigned* function in Listing 2.

Listing 2. smart contract code of OfferTradeSigned function.

```
function offerTradeSigned(bytes32 _sender,bytes32 _receiver, bytes32
    _valueHash,
uint _tmout, uint8 sigV,bytes32 sigR, bytes32 sigS) returns(bool success)
    public  {}
```

This function implements the very first edge found in the PIM, with blockchain native signature checking. A correct execution of this particular function results in the emission of an instance of the event in Listing 3.

Listing 3. smart contract code of TradeOfferPlaced event.

```
event TradeOfferPlaced(bytes32 indexed sender, bytes32 indexed receiver,
    bytes32 valueTradedHash, address indexed callingAn)
```

On the other hand, if the sender tries to re-enact an information exchange performed already in the past, execution stops, and the event in Listing 4 is emitted.

Listing 4. smart contract code of ValueAlreadyTraded event.

```
event ValueAlreadyTraded(bytes32 indexed sender,bytes32 indexed
    valueTradedHash)
```

By applying the same principles for the other remaining functions, a fully functional implementation is obtained.

3.4. Simulation and Assessment System

In order to validate the integration, we needed to create dynamic and complex scenarios that could help us reproduce many different situations to deeply understand if the chosen underlying blockchain technology provides satisfying responses. This requirement led us to choose the agent-based simulation approach [20,21] for simulating the biomass production chain.

Agent-based simulation has proved to be a very powerful tool when it comes to manipulating the model. This type of model can be seen as an agent society that lives inside an environment where agents perform actions, interact with each other, and respond to events according to their current state. By altering one or more of these elements (e.g., specific interactions, the set of agents) it is possible to explore many different scenarios.

The process model presented previously helped us building the Conceptual Model and laying out the scenarios to simulate. Figure 2 refers to a base scenario that is characterised by a linear execution without cycles and/or alternate paths. However, such a scenario serves as a starting point for laying out more complex what-if useful scenarios, two of which are shown in Figures 3 and 4, from the agent perspective. The scenarios are named *Multiple stakeholder* and *Partial delivery*, respectively. The agent-based representation decorates every operation (arrow) with a set of agents, which includes agents that either participate in the certification process or provide a service that contributes to complete the operation. For instance, let us consider the very first operation in Partial delivery scenario. Claudia is responsible for sending a competent body an authorisation request. As can be seen in Figure 2, Claudia is presented as the only participant. There may be operations that need to be made accountable that involve only one participant. In this case, it would be risky to let only one participant use the certification service because she could alter the content of operation-related documents for personal gain. In order to address this situation, the Conceptual Model ensures that the participant cannot use the service without engaging a manager from the same organisation. This constrain introduces new participants that act as witnesses. There are also agents that reproduce the behaviours of external participants that do not participate in the certification process. An example of these agents is Gennaro, an employee of a competent body.

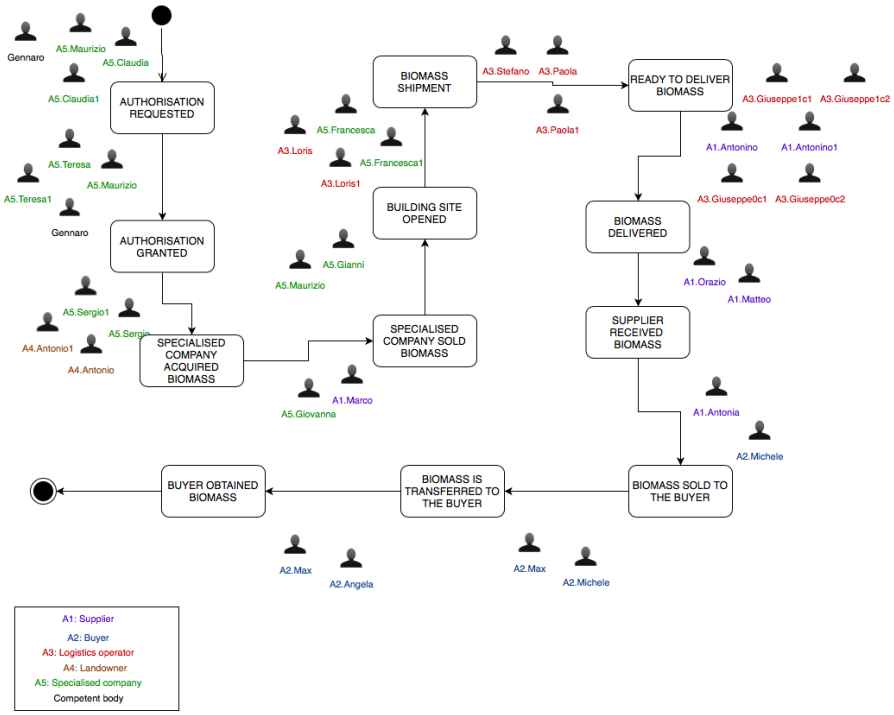


Figure 3. Agent-based representation of what-if scenario where multiple landowners and logistics operators participate in the process.

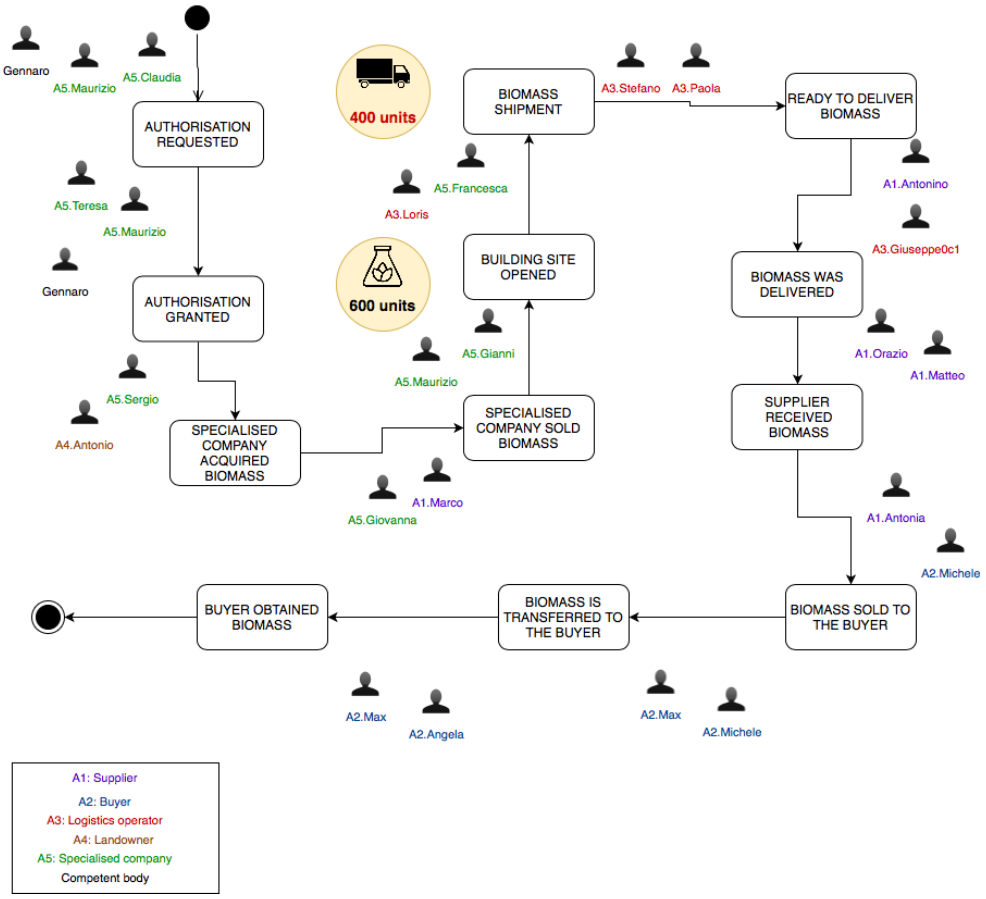


Figure 4. Agent-based representation of what-if scenario where the logistics operator is not able to dispatch all biomass produced by the specialised company.

The scenario shown in Figure 3 is characterised by the presence of two logistics operators and landowners. Increasing the number of actors has an impact on the number of operations to perform and, as a consequence, the number of participants required may rise. This type of scenario could be even more complex and could set up a basis for stress testing the infrastructure to evaluate if it can scale well up to the number of deployed agents. On the other hand, a scenario like the one shown in Figure 4 is interesting to explore rare, unusual, or alternate situations to see how the process behaves from different perspectives. For instance, the Partial delivery scenario was designed to simulate a situation wherein the logistics operator is not able to dispatch the entire product. It would be interesting to see what is the impact of delivering only a partial amount of biomass that is equal to the total transport capacity provided by the operator. This scenario could lead the simulation team to lay out other what-if scenarios, where, for example, recovery procedures are explored.

The simulation environment derived from the Conceptual Model and other internal products was first subject to testing to verify that the integrated member applications worked as intended and then was executed to evaluate the performance of the infrastructure with respect to a set of parameters of interest. The simulation execution and analysis were driven by the sets of dependent and independent variables described below, both from the point of view of a single AN, or the entire system as a whole.

The following dependent variables were identified: (i) Average Response Time, the average time taken by an AN or the entire infrastructure to fulfill a transaction request from an end-user; (ii) Throughput of the number of requests per time unit fulfilled by an AN, or by the entire infrastructure.

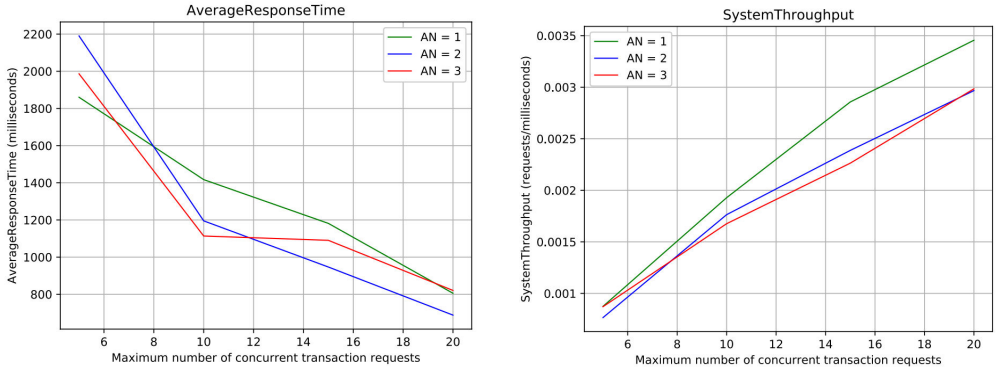
The evaluation was conducted based on the following two independent variables: (i) Number of ANs (AN_w), the number of AN that can actually write transaction on behalf of an end-user; (ii) Number of concurrent transaction requests (R).

The above-mentioned sets of variables were determined based on our needs; other behavioural aspects could be interesting for future analysis like system and blockchain usage. AN_w takes values from the set $S_1 = (1, 2, 3)$, whereas R from $S_2 = (5, 10, 15, 20)$. The number of ANs was determined based on the number of available ANs on the network, while the number of concurrent transaction requests was based on available computational and memory resources. We tested the behaviour of the infrastructure by combining the values of AN_w and R. In order to better understand how the evaluation was performed, it is important to note that there is a one-to-one relationship between transaction requests and smart contract execution, i.e. given a user transaction request, the infrastructure executes only one transaction (in other words only one atomic write operation). Moreover, a user request triggers a number of calls within the infrastructure, a few of which do not interact with the blockchain. When we measure the behaviour of the system as a whole we take into account both blockchain and API-related operations. When it comes to variables that describe the behaviour of a single AN, we also actually get readings on how the infrastructure behaves when it interacts solely with the blockchain. At last, the measurements were taken only after a transaction got confirmed, which means that we had to wait for a certain number of blocks to be mined (in our case 23 blocks). We conducted 12 simulation campaigns, which were divided into three groups. In each group, we execute a validation campaign for each level assigned to AN_w. Specifically, we first validate the infrastructure with only one AN that can write transactions on behalf of users, then with two and at last with three. The difference between the groups lies in the maximum number of concurrent transactions that can be observed. For instance, in one group we can observe up to 5 transaction requests, in another 10 requests and so on.

Figures 5 report the most significant results of the campaign. Figure 5a shows the Average Response Time computed for requests processed by both the blockchain and high-level API. The maximum number of concurrent transaction requests is reported on the x-axis, whereas the Average Response Time, expressed in milliseconds, on the y-axis. Each curve describes how the system behaves with a specific number of enabled ANs. As can be seen from the graph, the greater the number of requests is, the lower the response time gets. From the minimum to the maximum level of R evaluated, the response time improves by about 1 second. It is interesting to note that a few configurations with a smaller number of enabled ANs may perform better than configurations with a higher number of ANs. This behaviour is due to transaction-related computational costs, and higher consensus cost per update, given a higher number of nodes to sync. Regarding the System Throughput for blockchain and high-level API interactions, the results are reported in Figure 5b. Similarly to the previous graph, the maximum number of concurrent transaction requests is reported on the x-axis, while the System Throughput is expressed in milliseconds/requests, on the y-axis. The behaviour observed fits our expectations: the more simultaneous transaction requests arrive the better the throughput gets. Configurations with two and three ANs achieve 180 requests per minute of throughput. The configuration with only one AN achieves 200 requests per minute, with the same number of requests. The System Throughput tends to decrease as we enable new ANs, except for a few cases, even though the difference is negligible. This trend becomes much more evident when the configuration with only one AN is compared with the remaining two configurations. This behaviour is caused by synchronisation costs which increase when new nodes are added to the network.

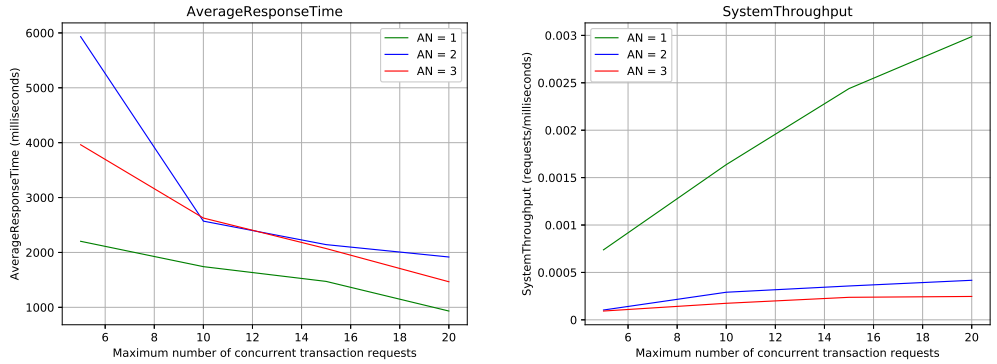
The graphs reported in Figure 6a,b illustrate what happens when we consider only blockchain request processing. The way the variables behave is consistent with what we observed in the previous discussion. However, when we look at the values computed, there are significant differences that deserve to be highlighted. Blockchain-only response time is much greater than the system response time. This result reflects the absence of requests that don't translate into a transaction execution, hence they are very light from a computational perspective and contribute heavily to lowering the overall score. The blockchain throughput is worse than the system throughput, especially for configura-

tions with two or three ANs, while the configuration with only one AN behaves comparably with what is reported in Figure 5. Once again, keep in mind that the difference observed between the latter and the former two configurations is due to synchronisation costs.



(**a**) (**b**)

Figure 5. (**a**) Average Response Time and (**b**) System Throughput computed for requests processed by both the blockchain and high-level API.



(**a**) (**b**)

Figure 6. (**a**) Average Response Time and (**b**) System Throughput computed for requests processed only by the blockchain.

The analysis led us to choose a configuration with three ANs enabled for writing transactions on behalf of end-users, forming the *Network Configuration*. At first glance, this choice might seem odd, because the configurations with only one AN have often proved to offer better performances. However, there are two reasons that support our decision: *(i)* there is no neat difference with the results observed for configurations with a higher number of nodes, and *(ii)* the fewer the nodes are available, the weaker the infrastructure gets from a security perspective. The financial evaluation determined that the total yearly cost of the blockchain infrastructure would be about 7426 EUR.

3.5. Conformance Checking

After executing a set of campaigns, we extracted *Simulation Logs* from the blockchain itself, by using the implemented data transformation tool. From the 10865 events extracted, distributed among 5236 blocks of data generated by the simulation suite, the tool created a corresponding *XES Log* showed in Listing 5. The format is compliant with the XES mapping PIM, with some information added in order to track which *AN* was responsible for a specific computation. We used *Disco* in order to analyze this XES log, and, in the *map* view, we obtained a trace graph showed in Figure 7, in which every interaction either starts with an error event, thus ending the trace there and then, or with a *Trade Offer Placed*

event, that is only emitted by a successful execution of the *OfferTrade* function. In the vast majority of cases, a trace then moves from this event, towards either a successful *TradeOfferAccepted*, or other error states, blocking the trace from progressing further. It is important to understand that error events are related to invalid attempts at calling a specific function when the necessary preconditions are not true. And those can either be signals of malicious activity, aimed at subverting the business process, or benign, honest mistakes by actors. The map view, along with the other views present in the Disco tool, are instrumental in finding out whether or not the implemented process adheres to the model, or in other words, is compliant with business criteria analyzed during the modelling phase. Visual reports concerning this property of adherence form the *Conformance Checking Report*, while other, possibly non-visual and not strictly conformance checking related forms of information extracted from the XES file form the *XES Logs Analysis*, useful in the Design and Operation phase.

Listing 5. XES formatted blockchain log.

```
<event>
    <string key="callingAn" value="0x7A60C4f6788Bf85bE3B6350ff8A9619a08ED0C0E
        "/>
    <string key="concept:name" value="TradeOfferPlaced"/>
    <string key="receiver" value="0
        X7465737431406832692e636f6d0000000000000000000000000
        0000000000000"/>
    <string key="sender" value="0
        X7465737431406832692e636f6d0000000000000000000000000
        00000000000"/>
    <int key="blockNumber" value="819356"/>
    <string key="valueHash" value="0
        x636f736131000000000000000000000000000000000000000
        00000000000000"/>
    <id key="identity:id" value="E58B5862-8606-43EA-94F9-1DB2C0DB33F6"/>
</event>
```

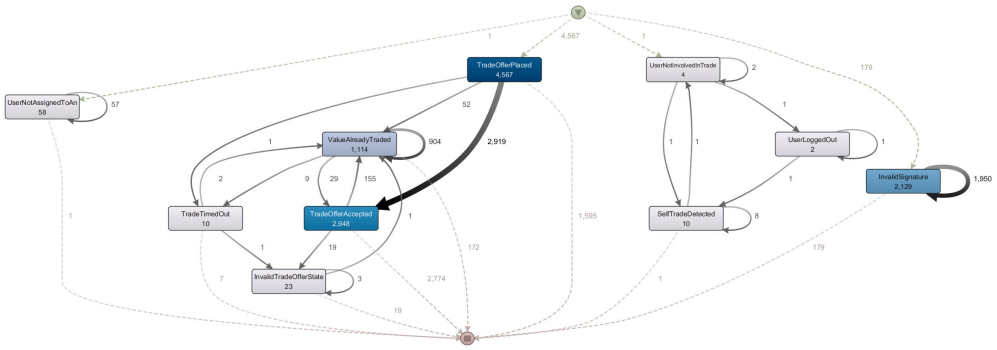


Figure 7. Trace Graph.

4. Related Work

The topics of Blockchain and digital identity have been investigated in recent literature. The collection of relevant research on Digital Identity on Blockchain technology implemented in a smart city environment is reported in [25]. This study aims to understand the challenges and future directions of these areas from the technical point of view.

How to obtain assured identities based on face-to-face proofing that can then be validated against a record on a blockchain is discussed in [26]. To provide anonymity, the authors define a new scheme that stores a commitment against which one can perform zero-knowledge proofs of identity. The schema has been implemented on Bitcoin's blockchain and exploits grouping commitments using Merkle trees to minimize the number of Bitcoin transactions that have to be sent.

Zero-Chain is a Blockchain-based system for the development of a digital infrastructure for smart city management [27]. It focuses on a key component of digital city management in the form of secure identification of individual residents. The system collects user attributes and securely transmits them to other system components for verification. Upon successful completion of the verification process, a digital identity is created for the applying resident and the set of transactions leading to the ID creation are stored in a blockchain.

To provide identity within the context of mutual distrust, the work published in [28] presents a blockchain-based digital identity solution. Without depending upon a single trusted third party, the proposed solution achieves a passport-level legally valid identity. For making identities Self-Sovereign, this solution exploits a generic provable claim model for which attestations of truth from third parties need to be collected. The claim model is then shown to be both blockchain structure and proof method agnostic. Four different implementations in support of these two claim model properties are shown to offer sub-second performance for claim creation and claim verification. Other solutions for Self-Sovereign Identity [28] that exploit the blockchain technology are presented in [29].

In the field of business Cross-Organizational Workflows (COWs), the authors of [30] propose and analyze a taxonomy of blockchain-based trust design patterns from a process-centric perspective. This taxonomy is useful to establish how the blockchain technology can enhance trust in collaborative processes. In [31], a framework for the execution of cross-organizational process collaborations and the implementation of inter-organizational processes on a Blockchain is presented. This framework includes a voting mechanism for process deployment as well as a subscription service to facilitate process handovers between participants more efficiently. The study presented in [32] explores the intersection of multi-agent simulations and consortium blockchain technology in the context of enterprise applications by devising architectures and technology stacks for both off-chain and on-chain agent-based simulation in the context of blockchain-based business process execution.

Weber et al. [5] published a landmark paper, where blockchain is the shared process execution platform. Specifically, the control flow and business logic of inter-organizational business processes is compiled from process models into smart contracts which ensure the joint process is correctly executed. Architecture in [5] has been implemented in Caterpillar [33], that is a system by which a well defined BPMN model can be translated into a set of smart contracts, enacting the model itself. Such an approach creates a strong bond between the smart contract implementations to the business process, and its ontology, increasing costs whenever any change has to be implemented, on either side of the equation. As the follow-up works have demonstrated, the Caterpillar system works by implementing multiple smart contracts [7], which, depending on the type of Ethereum network used (public or permissioned/private), can be either expensive and/or time-consuming to operate and test.

Even by employing Caterpillar, there's always the necessity to link each process' instance with the corresponding blockchain transactions sequence [34]. Instead, we've opted to employ the hashed documentation accompanying each asset as an on-chain reference.

Another similar tool, in terms of model-driven engineering with blockchain implementations, geared more towards asset management, is Lorikeet [35]. Lorikeet is a full-stack solution, together with a web GUI, used to construct the business process model itself. This model is then used to generate the smart contract code and then deploy it on chain. Similarly to Caterpillar, the same heavy interdependency between the model and its blockchain data and process representation is present here, making it difficult (albeit less so than in Caterpillar) to make any changes, especially during enactment.

Our work aims towards a different direction in this sense: we sought to decrease intensity of any business process to smart contract bond, in order to generalize code, make it more reusable and more maintainable. Creating or modifying any smart contract code, even by employing automation, can always introduce bugs, that might go undetected, or can have costly side effects. Fixing such problems, on the Ethereum blockchain, can either be very expensive, or outright impossible, without loss of data. We decided to mitigate this

problem by keeping our core smart contract code essential and well tested, and ready to be used in any business process implementation.

Another related and important research direction concerns the formal verification of the smart contracts [36] (e.g., by means of model-checking techniques [37]).

Here, as also outlined in [38], authors pointed out that more investigation on how to support organizations in embracing the blockchain technology to implement their interorganizational processes is definitely need [6]. With respect to the state of the art, the proposed platform takes into account the presence of already established, even public, IdPs (Identity Providers) and provides the following important properties: Control, Access, Transparency, Interoperability, Consent, and Protection.

5. Conclusions

ID-Service is a platform to support an innovative design, implementation, and execution of services implemented through Cross-Organization Workflows. It supports the principle of security by design, for what concerns the attributes of trust, accountability, non-repudiation, and ability of the system to provide forensic evidence of workflow traces, critical actions, and actors' responsibilities and to preserve these features during the execution. Our approach also leverages code re-usability at the smart contract level in order to mitigate deployment and maintenance costs, whilst keeping all the important properties mentioned beforehand intact. The Assessment layer provides the designer with powerful tools to control design and change management w.r.t. the intended features, which is the way to achieve reliable software. The platform provides the user with a set of APIs, which are built on top of a complex substrate that integrates different paradigms and technologies, including Blockchain, smart contracts, and eIDAS-compliant Public Digital Identities. Master nodes execute transactions on behalf of users, thus improving user-friendliness of our platform.

We identified some limitations of our solution. The first limitation is that our platform does not implement the Self Sovereign Identity model because public Identity Providers are exploited for identity management. Another limitation derives from a design choice: we decided to renounce to a centralized yet more controlled engine for the process execution in flavor of flexibility and dynamic adaptation.

Our approach has still further avenues of improvement in the future: we would like to increase the amount of information presented in the assessment layer and make it easier for a user analyst to submit new business processes.

Author Contributions: Conceptualization, F.B., A.F., A.G., G.L. and D.S.; Data curation, L.A. and F.P.; Funding acquisition, F.B., A.F. and D.S.; Investigation, F.B., A.F., S.G., A.G. and G.L.; Methodology, F.B., S.G., A.G. and D.S.; Software, L.A.; Supervision, F.B., A.G. and G.L.; Validation, S.G.; Visualization, L.A. and F.P.; Writing, original draft, L.A., A.F., S.G., A.G. and F.P.; Writing, review & editing, L.A., F.B., A.G., G.L., F.P. and D.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the project "Id-Service: Digital Identity and Service Accountability" funded by the Ministry of Economic Development (MISE), project code number F/050238/01-03/X32. This paper aims to describe the overall scientific contribution of the above research project. However, terms and conditions enforced by the project regulation do not allow us to make public the source code of the software platform.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AN	Accountability Node
API	Application Programming Interface
BPMN	Business Process Model and Notation
CoASiVE	Combined Agent-based Simulation and Virtual Environments

COWs	Cross-Organizational Workflows
DLT	Distributed Ledger Technology
DVE	Digital Virtual Environment
eIDAS	electronic IDentification, Authentication and trust Services
EVM	Ethereum Virtual Machine
GUI	Graphical User Inteface
IBE	Identity Based Encryption
IdP	Identity Provider
IM	Instant Messaging
TTP	Trusted Third Party
PIM	Platform Independent Model specification
XES	eXtensible Event Stream

References

1. Grefen, P.W.P.J.; Aberer, K.; Ludwig, H.; Hoffner, Y. CrossFlow: Cross-Organizational Workflow Management for Service Outsourcing in Dynamic Virtual Enterprises. *IEEE Data Eng. Bull.* **2001**, *24*, 52–57.
2. Norta, A.; Grefen, P.; Narendra, N. A reference architecture for managing dynamic inter-organizational business processes. *Data Knowl. Eng.* **2014**, *91*, 52–89. [CrossRef]
3. Haki, M.K.; Forte, M.W. Inter-Organizational Information System Architecture: A Service-Oriented Approach. In *IFIP Advances in Information and Communication Technology, Proceedings of the Collaborative Networks for a Sustainable World, St. Etienne, France, 11–13 October 2010*; Camarinha-Matos, L.M., Boucher, X., Afsarmanesh, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 642–652.
4. Bender, J. eIDAS Regulation: EID—Opportunities and Risks. 2015. Available online: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/ElekAusweise/SmartCard_Workshop/Workshop_2015_Bender.pdf (accessed on 30 March 2020).
5. Weber, I.; Xu, X.; Riveret, R.; Governatori, G.; Ponomarev, A.; Mendling, J. Untrusted Business Process Monitoring and Execution Using Blockchain. In *Lecture Notes in Computer Science, Proceedings of the Business Process Management—14th International Conference, BPM 2016, Rio de Janeiro, Brazil, 18–22 September 2016*; Springer: Cham, Switzerland, 2016; pp. 329–347.
6. Di Ciccio, C.; Cecconi, A.; Dumas, M.; García-Bañuelos, L.; López-Pintado, O.; Lu, Q.; Mendling, J.; Ponomarev, A.; Tran, A.B.; Weber, I. Blockchain support for collaborative business processes. *Informatik Spektrum* **2019**, *42*, 182–190. [CrossRef]
7. López-Pintado, O.; García-Bañuelos, L.; Dumas, M.; Weber, I.; Ponomarev, A. Caterpillar: A business process execution engine on the Ethereum blockchain. *Softw. Pract. Exp.* **2019**, *49*, 1162–1193. [CrossRef]
8. Angiulli, F.; Fassetti, F.; Furfaro, A.; Piccolo, A.; Saccà, D. Achieving Service Accountability Through Blockchain and Digital Identity. In *Lecture Notes in Business Information Processing, Proceedings of the International Conference on Advanced Information Systems Engineering, Tallinn, Estonia, 11–15 June 2018*; Springer International Publishing: Cham, Switzerland, 2018; pp. 16–23.
9. Buccafurri, F.; Lax, G.; Russo, A.; Zunino, G. Integrating digital identity and blockchain. In *Lecture Notes in Computer Science, Proceedings of the OTM Confederated International Conferences "On the Move to Meaningful Internet Systems", Valletta, Malta, 22–26 October 2018*; Springer: Cham, Switzerland, 2018; pp. 568–585.
10. Furfaro, A.; Argento, L.; Saccá, D.; Angiulli, F.; Fassetti, F. An Infrastructure for Service Accountability based on Digital Identity and Blockchain 3.0. In Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 29 April–2 May 2019; pp. 632–637.
11. Felicetti, C.; Furfaro, A.; Saccà, D.; Vatalaro, M.; Lanuzza, M.; Crupi, F. Making IoT Services Accountable: A Solution Based on Blockchain and Physically Unclonable Functions. In *Lecture Notes in Computer Science, Proceedings of the Internet and Distributed Computing Systems, Naples, Italy, 10–12 October 2019*; Springer International Publishing: Cham, Switzerland, 2019; pp. 294–305.
12. Wood, G. Ethereum: A secure Decentralised Generalised Transaction Ledger. Ethereum Yellow Paper. 2014. Available online: https://ethereum.github.io/yellowpaper/paper.pdf (accessed on 30 March 2020).
13. AgID—Agenzia per l'Italia Digitale. SPID—Regole Tecniche. 2017. Available online: https://media.readthedocs.org/pdf/spid-regole-tecniche/latest/spid-regole-tecniche.pdf (accessed on 30 March 2020).
14. Andrawos, M.; Helmich, M. *Cloud Native Programming with Golang: Develop Microservice-Based High Performance Web Apps for the Cloud with Go*; Packt Publishing Ltd.: Birmingham, UK, 2017.
15. PIRL. Available online: https://pirl.io/en/about/(accessed on 30 March 2020).
16. Döttling, N.; Garg, S. Identity-based encryption from the Diffie-Hellman assumption. In *Lecture Notes in Computer Science, Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 20–24 August 2017*; Springer: Cham, Switzerland, 2017; pp. 537–569.
17. Argento, L.; Graziano, S.; Garro, A.; Guzzo, A.; Pasqua, F.; Saccà, D. A Simulation-based and Data-driven Framework for Enabling the Analysis and Design of Business Processes based on Blockchain and Smart Contracts Solutions. In Proceedings of the 3rd Distributed Ledger Technology Workshop Co-located with ITASEC, Ancona, Italy, 4 February 2020.
18. Furfaro, A.; Argento, L.; Parise, A.; Piccolo, A. Using virtual environments for the assessment of cybersecurity issues in IoT scenarios. *Simul. Model. Pract. Theory* **2017**, *73*, 43–54. [CrossRef]

19. Furfaro, A.; Piccolo, A.; Parise, A.; Argento, L.; Saccà, D. A Cloud-based platform for the emulation of complex cybersecurity scenarios. *Future Gener. Comput. Syst.* **2018**, *89*, 791–803. [CrossRef]
20. Macal, C.M.; North, M.J. Agent-based modeling and simulation. In Proceedings of the 2005 Winter Simulation Conference (WSC), Orlando, FL, USA, 4 December 2005; pp. 86–98.
21. Shoham, Y. Agent-oriented programming. *Artif. Intell.* **1993**, *60*, 51–92. [CrossRef]
22. Günther, C.W.; Rozinat, A. Disco: Discover Your Processes. *BPM (Demos)* **2012**, *940*, 40–44.
23. Bosona, T.; Gebresenbet, G.; Olsson, S.O. Traceability system for improved utilization of solid biofuel from agricultural prunings. *Sustainability* **2018**, *10*, 258. [CrossRef]
24. Abeyratne, S.A.; Monfared, R.P. Blockchain ready manufacturing supply chain using distributed ledger. *Int. J. Res. Eng. Technol.* **2016**, *5*, 1–10.
25. Rivera, R.; Robledo, J.G.; Larios, V.M.; Avalos, J.M. How digital identity on blockchain can contribute in a smart city environment. In Proceedings of the 2017 International smart cities conference (ISC2), Wuxi, China, 14–17 September 2017; pp. 1–4.
26. Augot, D.; Chabanne, H.; Clémot, O.; George, W. Transforming face-to-face identity proofing into anonymous digital identity using the Bitcoin blockchain. In Proceedings of the 2017 15th Annual Conference on Privacy, Security and Trust (PST), Calgary, AB, Canada, 28–30 August 2017; pp. 25–2509.
27. Asamoah, K.O.; Xia, H.; Amofa, S.; Amankona, O.I.; Luo, K.; Xia, Q.; Gao, J.; Du, X.; Guizani, M. Zero-Chain: A Blockchain-Based Identity for Digital City Operating System. *IEEE Internet Things J.* **2020**, *7*, 10336–10346. [CrossRef]
28. Stokkink, Q.; Pouwelse, J. Deployment of a blockchain-based self-sovereign identity. 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. 1336–1342.
29. Van Bokkem, D.; Hageman, R.; Koning, G.; Nguyen, L.; Zarin, N. Self-Sovereign Identity Solutions: The Necessity of Blockchain Technology. *arXiv* **2019**, arXiv:1904.12816.
30. Müller, M.; Ostern, N.; Rosemann, M. Silver Bullet for All Trust Issues? Blockchain-Based Trust Patterns for Collaborative Business Processes. In *Lecture Notes in Business Information Processing, Proceedings of the International Conference on Business Process Management, Seville, Spain, 13–18 September 2020*; Springer: Cham, Switzerland, 2020; pp. 3–18.
31. Klinger, P.; Bodendorf, F. Blockchain-based cross-organizational execution framework for dynamic integration of process collaborations. In Proceedings of the 15th International Business Informatics Congress, Potsdam, Germany, 8–11 March 2020.
32. Kampik, T.; Najjar, A. Simulating, off-chain and on-chain: Agent-based simulations in cross-organizational business processes. *Information* **2020**, *11*, 34. [CrossRef]
33. López-Pintado, O.; García-Bañuelos, L.; Dumas, M.; Weber, I. Caterpillar: A Blockchain-Based Business Process Management System. In Proceedings of the BPM Demo Track and BPM Dissertation Award Co-Located with 15th International Conference on Business Process Management (BPM 2017), Barcelona, Spain, 10–15 September 2017.
34. Di Ciccio, C.; Cecconi, A.; Mendling, J.; Felix, D.; Haas, D.; Lilek, D.; Riel, F.; Rumpl, A.; Uhlig, P. Blockchain-based traceability of inter-organisational business processes. In Proceedings of the International Symposium on Business Modeling and Software Design, Halifax, NS, Canada, 30 July–3 August 2018; pp. 56–68.
35. Tran, A.B.; Lu, Q.; Weber, I. Lorikeet: A Model-Driven Engineering Tool for Blockchain-Based Business Process Execution and Asset Management. In Proceedings of the 16th International Conference on Business Process Management, Sydney, Australia, 9–14 September 2018.
36. Bai, X.; Cheng, Z.; Duan, Z.; Hu, K. Formal Modeling and Verification of Smart Contracts. In Proceedings of the 2018 7th International Conference on Software and Computer Applications—ICSCA, Kuantan, Malaysia, 8–10 February 2018. [CrossRef]
37. Cicirelli, F.; Furfaro, A.; Nigro, L. Model checking time-dependent system specifications using Time Stream Petri Nets and Uppaal. *Appl. Math. Comput.* **2012**, *218*, 8160–8186. [CrossRef]
38. Mendling, J.; Weber, I.; Aalst, W.V.D.; Brocke, J.V.E.A. Blockchains for Business Process Management—Challenges and Opportunities. *ACM Trans. Manag. Inf. Syst.* **2018**, *9*, 1–16. [CrossRef]