A simulation-based and data-driven framework for enabling the analysis and design of business processes based on blockchain and smart contracts solutions

Luciano Argento[1], Sabrina Graziano[1], Alfredo Garro[2], Antonella Guzzo[2],
Francesco Pasqua[1], and Domenico Saccà[2]

[1] OKT
{luciano.argento, sabrina.graziano, francesco.pasqua}@okt-srl.com
[2] University of Calabria
{alfredo.garro, antonella.guzzo, sacca}@unical.it

Abstract

Blockchain and smart contracts solutions can represent key enablers for implementing and enacting flexible and secure business processes, particularly those crossing organizational boundaries (i.e. inter-organizational business processes). Even though organizations foresee enormous potential benefits in using such solutions, the percentage of projects involving Blockchain and Distributed Ledger Technologies (DLT) that reach the operational stage is still quite low. Among the several reasons that are limiting this growth, there is the lack of well-founded and integrated approaches that are able to help organizations in mitigating the risks of the concrete adoption of DLT-based business processes. To fill this lack, the paper proposes a methodological approach for the dynamic assessment of inter-organizational business processes in which virtual environments, combined with simulation and analytical techniques, are used to create, validate and subject to stress tests business processes relying on blockchain and smart contracts solutions. In application of the model continuity paradigm, the simulation models and the data-driven techniques used for analyzing the business process under definition, represent the starting point for the design of an enactable business process as well as for implementing monitoring, corrective and evolutionary maintenance services related to its operation. In order to show the effectiveness of the proposal, a case study concerning a biomass production chain is presented.

1 Introduction

Blockchain technology has been gaining increasing popularity and, currently, is widely considered as a breakthrough invention which could change many everyday activities and business processes in different application domains, like supply chain, healthcare and financial sectors[5, 6]. Currently there is a growing trend in the use of the blockchain as infrastructure for deploying inter-organizational processes, i.e. processes involving different organizations, as shown by the various applications and experiments in the field of academic and industrial research[7, 10]. In fact, as reported in [12], blockchain can help organizations in implementing and executing business processes across organizational boundaries. The major features are:

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 (e.g. state change message) that in turn is made immediately accessible to the participants for the identification of any errors.

3. Logic of interaction of the inter-organizational processes could be codified within smart contracts ensuring the correct execution of the shared process. Smart contracts are a further element of control of the execution of process, as they only accept coded interactions and only if executed by the participants who have the necessary authorizations.

In[6] the current state of the art for the model-driven design and implementation of blockchain-based process has been presented by comparing the features of two concrete approaches, namely Caterpillar and Lorikeet. Here, as also outlined in[12], the authors pointed out that more investigation on how to support organizations in embracing the blockchain technology to implement their interorganizational processes is definitely needed[6]. Even tough organizations foresee enormous potential benefit in the deployment of blockchain-based processes, including: the disintermediation of low value-added middlemen, increasing level of collaborations, improved traceability, and enhanced transparency, there are still technical challenges to be addressed, such as scalability, integrity of network participants, distribution of computational power, reaching of consensus that require to research for a more robust solutions. According to the data of the ”Blockchain & Distributed Ledger Observatory” of the Politecnico di Milano, in the last 3 years, only 14% of the projects involving Blockchain and DLT based platforms and services have became already operational, denoting also a distruth or caution in embracing such application project.

One of the major issues in the deployment of business processes through blockchain solutions is to assess the suitability of applying a blockchain-based infrastructure against the requirements of use cases and to clearly analyze and define its boundary. When, among the others, a requirement is the deployment of inter-organizational processes, there is also the issue of assuring monitoring and analysis of the process itself. To fill the lack of well-founded and integrated approaches that are able to help organizations in mitigating the risks of the concrete adoption of DLT-based business processes, the paper proposes a methodological approach for the dynamic assessment of inter-organizational business processes in which virtual environments, combined with simulation and analytical techniques, are used to create, validate and subject to stress tests business processes relying on blockchain and smart contracts solutions. The combination of simulation models with process mining techniques allow to perform both predictive and prescriptive analysis and to effectively grasp the requirements of the system/infrastructure that has to deliver the process/service under analysis. In application of the model continuity paradigm, the simulation models and the data-driven techniques used for analyzing the business process under definition, represent the starting point for the design of an enactable business process as well as for implementing monitoring, corrective and evolutionary maintenance services related to its operation. The rest of the paper is structured as follows: In Section 2 the proposed simulation-based and data-driven framework for enabling the analysis and design of business processes based on blockchain and smart contracts solutions is presented; in order to show the effectiveness of the proposal, Section 3 reports a case study, which concerns a biomass production chain, developed in the context of the “Id-Service: Digital Identity and Service Accountability” project, funded by the Ministry of Economic Development (MISE); finally, conclusions are drawn and future work delineated.

2 The Methodological Framework

The proposed methodological framework for enabling the analysis and design of business processes based on blockchain and smart contracts solutions is depicted in Figure 1. The method combines simulation-based and data-driven approaches and techniques with the aim to provide a full-fledged analysis of the business process under consideration that could benefit from the exploitation of DLT-based solutions; in particular, four main phases can be identified:

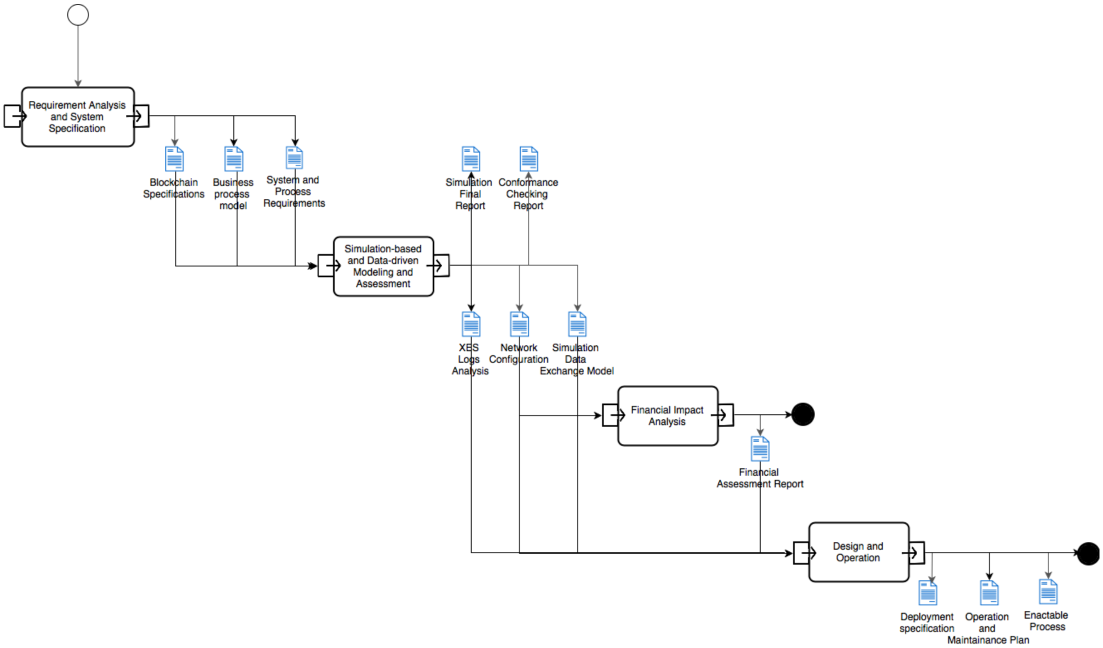


Figure 1: The Methodological framework: main phases and workproducts

- Requirement Analysis and System Specification, in which the requirements of both the business process and the underlying system are derived along with its high-level model and the blockchain specifications;

- Simulation-based and Data-driven Modeling and Assessment, in which, by using agent-based simulation and process mining techniques, a “Simulation Final Report” and a “Conformance Checking Report” are produced. Moreover, a “Simulation Data Exchange Model”, a “Network Configuration” document, and a report concerning the analysis of XES (eXtensible Event Stream)[3] logs are also delivered.

- Financial Impact Analysis, in which a report concerning the financial impact of the solution under evaluation is produced;

- Design and Operation, in which the work products produced in the previous phases are refined to produce the detailed design of the enactable business process as well as for implementing monitoring, corrective and evolutionary maintenance services related to its operation.

In the following, the above introduced phases and work-products are discussed more in details, using as reference the main steps provided by the IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP)[1], from conceptual analysis to simulation execution, data analysis and result evaluation.

2.1 Requirements Analysis and System Specification

This first work phase requires *domain specific analysis* and *requirements engineering techniques* in order to assess the *AS-IS* situation for the process at hand.

Here the modelling effort starts with identifying the main concepts that need to be represented and with mapping high-level and domain-specific features of the application process; the technical details—the main components of the operational perspective—are taken into account in the configuration phase of the business process simulation.

Moreover, since that phase guides the process implementation, more precise specifications are required in the software layer and thus specified and formalized requirements should come in the form of a standard *BPMN document*[14, 8], which is eventually enriched with a set of annotations of expected performance as qualitative indicators.

According to said formal bounds, a *Blockchain Specification* must also be produced in this phase: this document contains information about what kind of specific blockchain platform and technology is best suited for the given field of analysis.

2.2 Simulation-based and data-driven modelling and assessment

Technology assessment, which can be described as the set of techniques, algorithms and methodologies used to assess complex processes and systems, is an element that, especially when combined with *Financial assessment*, positively affects the *Value proposition* of a company.

The simulation-based and data-driven modeling and assessment phase of the proposed methodological framework encompasses a series of goals : Design and build a testbed blockchain infrastructure, according to the *Blockchain Specification* produced before. Then, implement data and Smart Contracts in accordance to an ad-hoc defined Platform Indipendent Model (*PIM*), called *Process-Agnostic Smart Contract Specification.* Finally, break down each step found in the *Business Process Model* as a series of chained informational exchanges among clearly identified actors, in order to power conformance checking based on process mining techniques.

The first activity changes heavily according to what specific blockchain platform has been chosen after the analysis conducted before. Implementation of a test network with an API layer varies a great deal from platform to platform, be it *Corda*, *Hyperledger Fabric* or *Ethereum.* It also encompasses technical, DevOps oriented details, definitely not suited for an extensive and in detail discussion in this paper, other than recommending the use of container technology, given how widespread, mature and handy to deploy it has become.

The second activity deals with implementing a process agnostic smart contract to then deploy on the test network, while the third readies an interpretation layer for logs extracted using process mining techniques from said contract's event data, thanks to a mapping with the XES-metamodel[3].

All the information produced during the last activity results in a *Process Model* document, useful later on in reconstructing process-relevant knowledge from mining results.

More details on the above mentioned activities and work products are reported in Section 3 with reference to the selected case study.

2.3 Financial Impact Analysis

If the results of the previous phase were positive, i.e. we were able to find a system configuration that fits the process requirements, then we move forward with the *Financial Impact Analysis* phase. Here our objective is to evaluate the financial feasibility of the blockchain system configuration, from the point of view of the commissioning organization.

We conducted an in-depth study to determine how much hardware resources cost for building up a network. We compared listings of blockchain resources offered by different cloud service providers to estimate the cost of a blockchain configuration. Let us consider a single node with 250 GB of storage, if we look at the listings of Microsoft Azure, for say, a tentative Ethereum like solution. We can conclude that a single node has a monthly cost of 195,786 EUR. The cost is equal to 0,00447 EUR per node * 43.800 minutes per month of usage. For Storage resources we have a monthly cost of 10,75 EUR per single node. The cost is equal to number of nodes * 250 GB * 0,043 EUR per GB.

Obviously, such expenses per node vary heavily based on the choices discussed in the *Blockchain Specification* document, as different technologies mandate different price structures due to power consumption requirements. A proof of work based blockchain, such as Ethereum, is going to require a lot more power to run, compared to other solutions, such as Corda.

2.4 Design and Operation

The simulation models and the data-driven techniques used for analyzing the business process under consideration can be reused to support its operation by enabling the implementation of monitoring, corrective and evolutionary maintenance services. In particular, the (agent-based) simulation model used during the assessment phase can be refined and transformed into an enactable process model running, as an example, on a BPMN-based environment (see [9] for details); whereas the process mining algorithms, which during the previous phase were applied on data coming from simulation, can be fed with data coming from the operation to monitoring the process enactment for diagnosis, prognosis, treatment and evolution purposes.

3 Case study: a Biomass production chain

This section discusses the case study that was considered for validating the methodology. The discussion is structured as a function of the methodology phases.

3.1 Requirement Analysis and System Specification

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 [4][2], and how adversely these impact the end product's quality. Accountability is a property of utmost importance in this domain due to the rigourous necessity of 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.

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 3.2.2 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 in 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: *Preliminary Activity*, *Building Site Activity* and *Delivery Activity*. The above list shows a clear breakdown of activities, in order of execution. Below we briefly describe each of these activities, assuming that the supplier also acts as 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 elaborates on the case study from the perspective of the two central macro phases of the proposed methodology.

3.2 Simulation-based and Data-driven Modeling and Assessment

3.2.1 Process model design

The process model design is the starting point of the Simulation-based on Data-driven Modelling and Assessment phase, along with the blockchain operational model design. 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

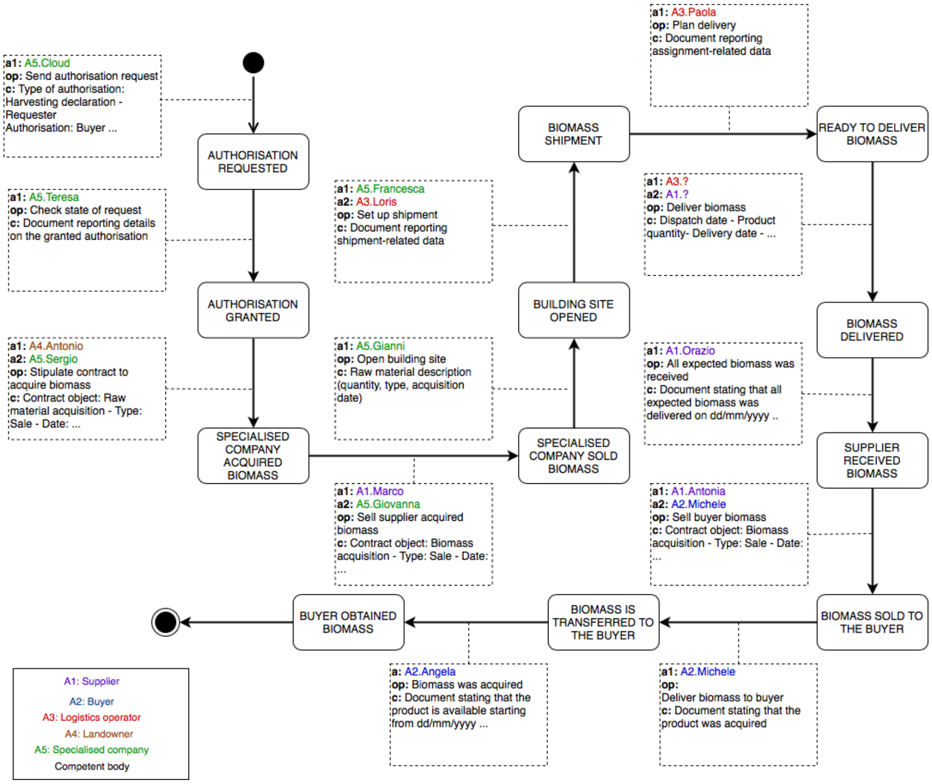


Figure 2: Model of a simple biomass production chain process

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 there are five actors that 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 (*Specialised company sold biomass*), whose purpose is to sell the biomass to the buyer. 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*).

3.2.2 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.

3.2.3 PIM Smart Contract implementation

The *platform independent specification* 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 modelled 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, each and 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 (*Check-Effects-Interactions and CRUD*) 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 *require* construct, using a simple if check, in order to emit an event logging the unexpected condition.

```
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, exactly one event describing a correct execution for each function, and a certain number of other events emitted whenever execution encountered a particular error. Data was laid out in order to support the multi-FSM behaviour: the CRUD pattern was used in order 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 for the *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 following 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 this other event is emitted.

```

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

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

3.2.4 Simulation-based Modelling and Assessment

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 [13][11] for simulating the biomass production chain. 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 the desired scenarios.

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 which 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 lay out the scenarios to simulate. Figure 2 refers to a base scenario that is characterised by a linear

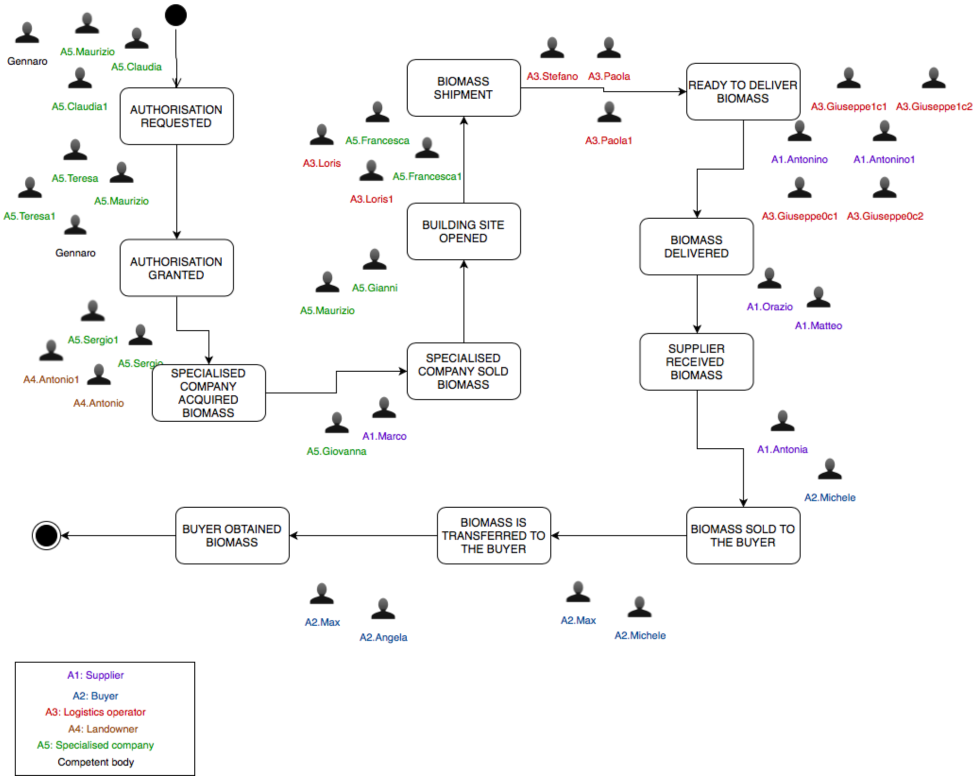


Figure 3: Agent-based representation of a what-if scenario where multiple landowners and logistics operators participate in the process.

execution without cycles and/or alternate paths. However, such 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 completing 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.

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

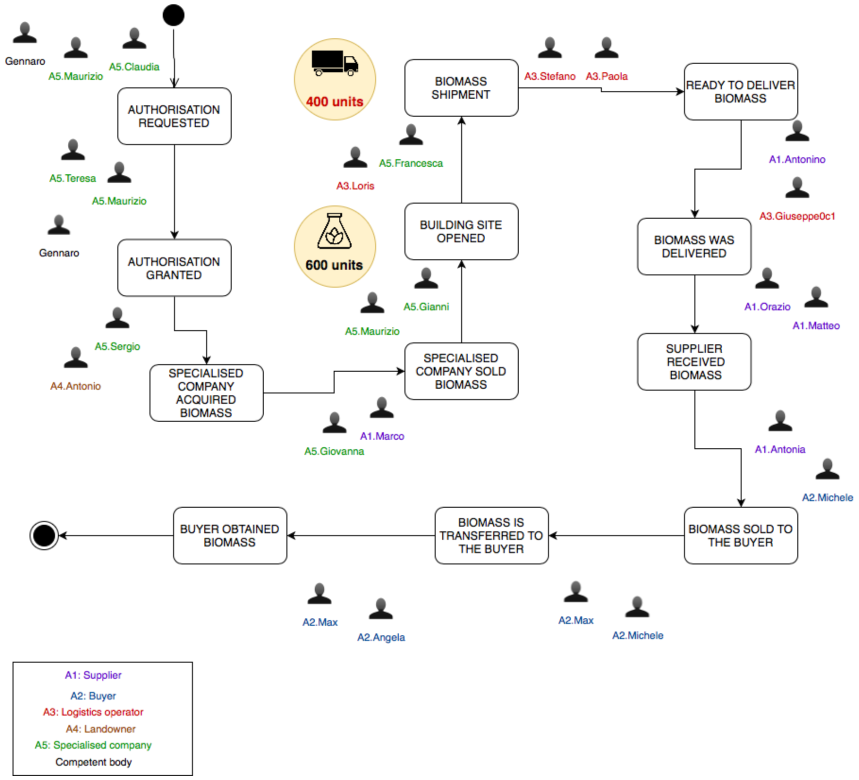


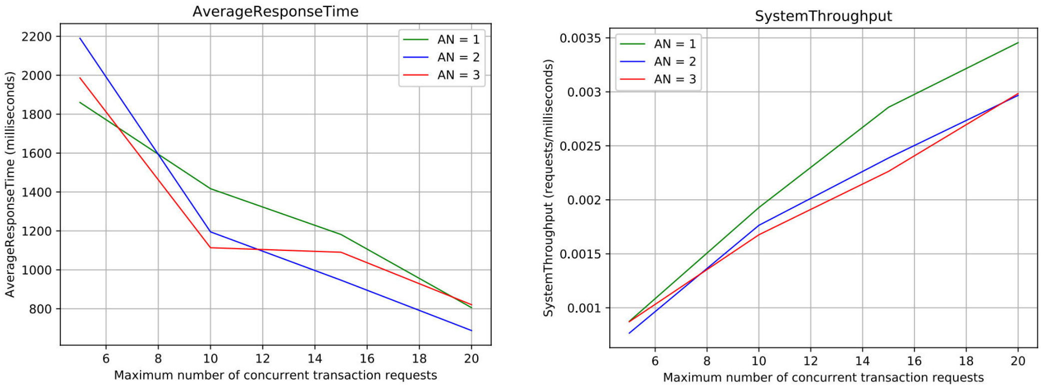
Figure 4: Agent-based representation of a what-if scenario where the logistics operator is not able to dispatch all biomass produced by the specialised company.

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: (i) Average Response Time, the average time taken by an AN or the entire infrastructure to fulfil a transaction request from an end user; (ii) Throughput of 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 be-



(a) (b)

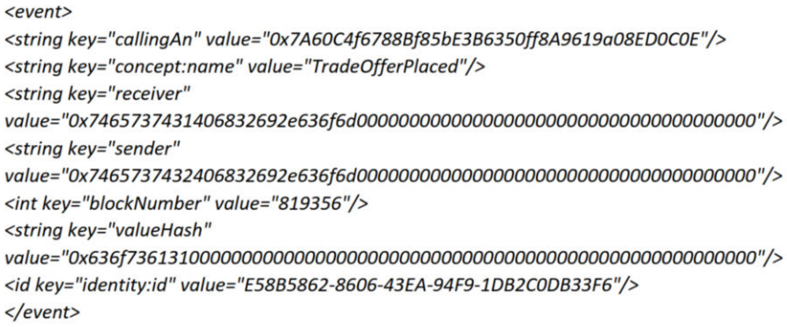
Figure 5: Average Response Time and System Throughput computed for requests processed by both the blockchain and high-level API.

havioural 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 in 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, in the perspective of finding the most suitable configuration that fit our needs. 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 as 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 as 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. If we consider the same number of concurrent requests, the configuration with only one AN achieves 200 requests per minute. 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 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 less 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.

3.2.5 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 Figure 6 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 obtained a trace graph, in the *map* view, shown in Figure 7. In the map every interaction either starts with an error event, thus ending the trace there and then, or with a *Trade Offer Placed* event, that, as we know, is only emitted by a successful execution of the *OfferTrade* function. And then 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 activities, 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.



```
<event>
<string key="callingAn" value="0x7A60C4f6788Bf85bE3B6350ff8A9619a08ED0C0E"/>
<string key="concept:name" value="TradeOfferPlaced"/>
<string key="receiver"
value="0x7465737431406832692e636f6d00000000000000000000000000000000000000"/>
<string key="sender"
value="0x7465737432406832692e636f6d00000000000000000000000000000000000000"/>
<int key="blockNumber" value="819356"/>
<string key="valueHash"
value="0x636f736131000000000000000000000000000000000000000000000000000000"/>
<id key="identity:id" value="E58B5862-8606-43EA-94F9-1DB2C0DB33F6"/>
</event>
```

Figure 6: XES formatted blockchain log

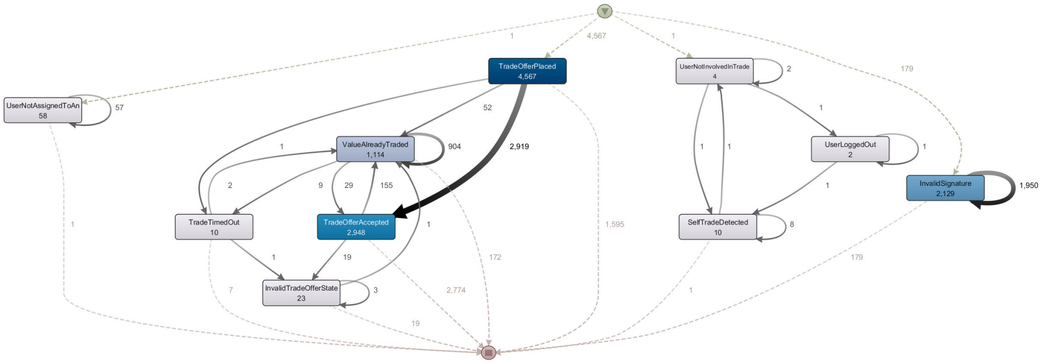


Figure 7: Trace Graph

4 Conclusions

To support organizations in mitigating the risks of the concrete adoption of DLT-based business processes, the paper proposed a methodological approach for the dynamic assessment of inter-organizational business processes in which virtual environments, combined with simulation and analytical techniques, are used to create, validate and subject to stress tests business processes relying on blockchain and smart contracts solutions. With reference to a specific business process under investigation that could benefit from the exploitation of DLT-based solutions, the proposed methodological framework, thanks to the combination of model-based and data driven techniques, allows not only to perform what-if/predictive analysis but also to get prescriptions on how to design, implement and configure the concrete process and the underlying infrastructure by also taking into account the financial impact of the development choices. Moreover, the simulation models and the data-driven techniques used for analyzing the business process under consideration can be reused to support its operation by enabling the implementation of monitoring, corrective and evolutionary maintenance services. Future research efforts will be devoted to further experiment the proposal with reference to inter-organizational business processes in several application domains as well as to improve its descriptive, predictive and prescriptive features.

5 Acknowledgments

This paper has been partially supported 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. Terms and conditions enforced by the project regulation do not allow us to make public the source code of the software platform.

References

[1] IEEE STD 1730$^{\mathrm{TM}}$-2010. Ieee recommended practice for distributed simulation engineering and execution process (dseep). 2011.

[2] Saveen A Abeyratne and Radmehr P Monfared. Blockchain ready manufacturing supply chain using distributed ledger. 2016.

[3] G. Acampora, A. Vitiello, B. Di Stefano, W. M. P. van der Aalst, C. W. Günther, and H. M. W. Verbeek. IEEE 1849TM: The XES Standard: The Second IEEE Standard Sponsored by IEEE Computational Intelligence Society. *IEEE Computational Intelligence Magazine*, pages 4–8, 2017.

[4] Techane Bosona, Girma Gebresenbet, and Sven-Olof Olsson. Traceability system for improved utilization of solid biofuel from agricultural prunings. *Sustainability*, 10(2):258, 2018.

[5] Fran Casino, Thomas K. Dasaklis, and Constantinos Patsakis. A systematic literature review of blockchain-based applications: Current status, classification and open issues. *Telematics and Informatics*, 36:55–81, 2019.

[6] Claudio Di Ciccio, Alessio Cecconi, Marlon Dumas, and Luciano García-Bañuelos et al. Blockchain support for collaborative business processes. *Informatik Spektrum*, 42(3):182–190, 2019.

[7] Claudio Di Ciccio, Alessio Cecconi, Marlon Dumas, Luciano García-Bañuelos, Orlenys López-Pintado, Qinghua Lu, Jan Mendling, Alexander Ponomarev, An Binh Tran, and Ingo Weber. Blockchain support for collaborative business processes. *Informatik Spektrum*, 42(3):182–190, 2019.

[8] Alberto Falcone, Alfredo Garro, Andrea D'Ambrogio, and Andrea Giglio. Engineering systems by combining bpmn and hla-based distributed simulation. In *2017 IEEE International Systems Engineering Symposium (ISSE)*, pages 1–6. IEEE, 2017.

[9] Giancarlo Fortino, Alfredo Garro, and Wilma Russo. From modeling to enactment of distributed workflows: an agent-based approach. In *Proceedings of the 2006 ACM Symposium on Applied Computing (SAC), Dijon, France, April 23-27, 2006*, pages 128–129, 2006.

[10] Orlenys López-Pintado, Luciano García-Bañuelos, Marlon Dumas, Ingo Weber, and Alexander Ponomarev. Caterpillar: A business process execution engine on the ethereum blockchain. *Software: Practice and Experience*, may 2019.

[11] Charles M Macal and Michael J North. Agent-based modeling and simulation. In *Proceedings of the 2009 Winter Simulation Conference (WSC)*, pages 86–98. IEEE, 2009.

[12] Jan Mendling, Ingo Weber, Wil Van Der Aalst, and Jan Vom et al. Brocke. Blockchains for business process management - challenges and opportunities. *ACM Trans. Manage. Inf. Syst.*, 9(1):4:1–4:16, February 2018.

[13] Yoav Shoham. Agent-oriented programming. *Artificial intelligence*, 60(1):51–92, 1993.

[14] Mathias Weske. *Business Process Management - Concepts, Languages, Architectures.* Springer, 2007.
