



Enforcing a Cross-Organizational Workflow: An Experience Report

Susanne Stahnke^(✉), Klym Shumaiev, Jorge Cuellar,
and Prabhakaran Kasinathan

Siemens AG - Corporate Technology, Otto-Hahn-Ring 6, 81739 Munich, Germany
{susanne.stahnke,klym.shumaiev,jorge.cuellar,
prabhakaran.kasinathan}@siemens.com

Abstract. Today business processes often exceed organizational boundaries and the participants may not fully trust each other. In the past, this trust was guaranteed solely through legal contracts. To digitally establish trust without requiring a trusted third party, Blockchain technology together with Smart Contracts is used to ensure that involved organizations can not break their agreements. This experience report introduces a workflow enforcement method for a particular use case, the certification of the construction of industrial plants. The method comprises the modelling of the workflow, the verification of the models and the translation into Smart Contracts.

Keywords: Business process modelling · Cross-organizational processes · Workflow enforcement · Blockchain · Smart Contracts · Experience report

1 Introduction

Globalization, novel technologies, fast-changing environments, the need for cost reduction, and the rapid evolution of information and communication technologies introduce challenges for enterprises, which are tackled by the cross-organizational collaboration between organizations, as stated in [9]. Establishing such cross-organizational collaboration requires the involved enterprises to specify the different interoperations and a common goal beforehand. This can be done by modelling these cooperative processes.

Already in 1998, before emergence of mature business process modelling standards, like BPMN, van der Aalst suggested how inter-organizational workflows could be modelled and verified. In [18], he describes the concept of loosely coupled workflow processes, which use asynchronous communication. A workflow consists of one or more processes, each one assigned to a given organization. The processes operate independently from each other but synchronize at specified points to ensure the correct execution of the whole workflow. However, in the same paper van der Aalst did not provide any details on how such an approach can be implemented in practice.

Inter-organizational business processes require a certain level of trust between the participating organizations. This trust is traditionally guaranteed solely through legal contracts, which specify a collection of responsibilities and obligations agreed upon all participating parties. With emergence of Blockchain, Smart Contracts assure that involved organizations cannot break their agreements and therefore, serve as a machine enforceable and verifiable protocol for inter-organizational workflows. This establishes trust between the participating parties without requiring a trusted third party.

This experience report demonstrates how to digitalize a cross-organizational business process on the basis of the use case “Certification of the Construction of Industrial Plants”. Coloured Petri Net modelling notation is used to capture and verify inter-organizational business processes. It serves as a base for the implementation of Smart Contracts running in the *Hyperledger Fabric* environment. Petri Nets are easy to understand and use, and offer formal semantics and analysis techniques. Therefore, different behavioural properties can be analysed and verified. Implementing the business process as Smart Contracts on a Blockchain enforces all involved parties to execute the workflow correctly. Since we require that the identity of the participants is known and secured, a private permissioned Blockchain is used to implement the use case. In this experience report we discuss an approach on how to enforce cross-organizational business processes, and we present the lessons learned to be considered by the practitioners in the future.

The report is structured as follows. In Sect. 2, the use case is described. Section 3 presents the background to this work: we compare different Workflow Modelling Languages and introduce private permissioned Blockchain networks. The workflow enforcement method, comprising the modelling of the use case, the verification of the models, the translation into Smart Contracts and the evaluation of the Smart Contracts, is described in Sect. 4. In Sect. 5, encountered problems and best practices are discussed. Related work is presented in Sect. 6 and conclusions drawn from this experience report and an outlook to future work are described in Sect. 7.

2 Use Case: Certification of the Construction of Industrial Plants

Particular industrial plants, designed and built under the global supervision of a company called the *EPC* (abbreviation for Engineering, Procurement and Construction), require a certification by a governmental entity. In this report we take a detailed look at the certification process for a power plant, which is also valid in other types of industrial construction. The certification process requires the governmental body to issue approvals throughout the different stages of the power plant functional safety life cycle (consisting of design, fabrication, mounting, commissioning and test approval). In Fig. 1 we illustrate high-level phases of the life cycle.

In each step of the process, multiple documents are sent to the governmental body, that checks the documents, if needed asks for revisions and eventually

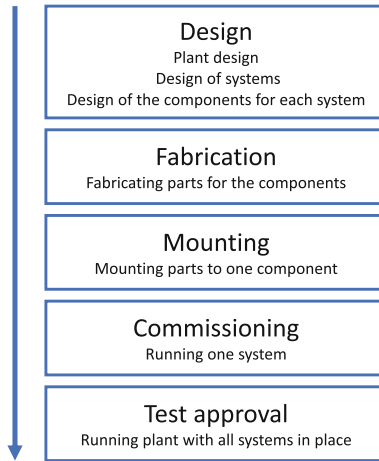


Fig. 1. High-level phases of the industrial plant functional safety life cycle

issues a certificate with its statement. This process includes different parties with different goals. Therefore, various steps of the process should be secured, to assure to all parties that the others execute the workflow correctly, and therefore, certain invariants (safety properties in the sense of temporal logic) hold. In case one of those invariants does not hold, all parties should be able to ascertain the responsible entity. Digitalizing this certification process results in an acceleration of the overall certification process, a secure management of all versions of the published documents, an easy search through the documents, a clear overview of all requirements and tasks during the entire plant life cycle, and finally, a paperwork reduction.

The following stakeholders are involved in the certification process. The *Owner* issues the plant, checks that the participating parties proceed correctly, and gives its approval throughout the entire life cycle. *EPC*¹ represents the service provider, which designs and builds the industrial plant. In our case, the *EPC* was represented by Siemens AG. *NOBO*, short for *Notified Body*, is a governmental agency responsible for examining and eventually approving the result of different stages of the construction of the industrial plant throughout the overall plant life cycle. In our case, the participating *NOBO* was TÜV Rheinland. *Manufacturer* produces parts of a component and delivers them to *EPC*. The certification process, described in this experience report, is an industrial standard.

¹ <https://www.marquard-bahls.com/de/news-info/glossar/detail/term/epc-engineering-procurement-and-construction.html>.

3 Background

3.1 Modelling Inter-organizational Workflows

A *workflow* can be defined as a set of activities or tasks, which is executed by certain entities, see [8]. The participating entities execute the workflow according to a defined set of rules to achieve, or contribute to an overall goal, as described in [6]. If the workflow is constructed to achieve business goals, it is also called business process. An *inter-organizational business process* is a workflow which involves multiple different, untrusted parties. To define such a business process Workflow Modelling Languages (WML) can be used. WMLs give the opportunity to not only define a workflow, but verify its *correctness*. A workflow is called correct, when it achieves its pre-defined goals by executing the defined partially ordered set of activities.

YAWL, BPMN, UML activity diagrams and Petri Nets are four WMLs, which are evaluated in this section. *YAWL*² (Yet Another Workflow Language, see [16]), inspired by Petri Nets, offers various workflow patterns, like basic control flow patterns, advanced branching and synchronization patterns, structural patterns and cancellation patterns. Since YAWL does not support PNML (Petri Net Markup Language), the possibility to export and import nets from/into different tools is very restricted. Since *BPMN*³ (Business Process Modelling Notation) and UML activity diagrams lack a complete formal semantics and analysis techniques, the papers [3, 12, 13] suggest mapping workflow modelling languages, which lack rigorous semantics, to Petri Nets.

An ordinary *Petri Net* (P/T-Net) is a directed graph, in which the nodes are places or transitions, and the arcs connect places to transitions or vice versa. Informally, places can be seen as local states and transitions as local activities. Additionally, tokens mark the current state of the Net. A *Coloured Petri Net* (CPN) is a combination of a Petri Net and a programming language, called Standard ML. Petri Nets allow the definition of concurrency, control structures, synchronization and communication. The addition of Standard ML extends this definition with the ability to specify colour sets (data types) and markings, arc expressions and guards. For further definition and a concrete example see [7]. Petri Nets are easy to understand and use, and offer formal semantics and analysis techniques. Therefore, different behavioural properties can be analysed and verified. That is the main reason why Petri Nets are preferred to other modelling languages. Petri Nets are general, easy and abstract models, but on the same time very clean semantics.

3.2 Private Permissioned Blockchain Networks

Distributed Ledger Technologies (DLTs) enable a verifiable and transparent way to enforce an inter-organizational business process. It implements a decentralized

² <http://yawlfoundation.org/>.

³ http://yoann.nogues.free.fr/IMG/pdf/07-04_WP_Intro_to_BPMN_-_White-2.pdf.

record of transactions. Blockchain is a particular type of DLTs. A blockchain is a replicated and distributed data structure, called ledger, that verifies and stores transactions occurring in a peer-to-peer network, as defined in [14]. The ledger is a timestamped list of blocks. Each block contains multiple transactions, is identified by its cryptographic hash and has a reference to the previous block's hash. This results in a linear, chronological chain of blocks. Each change on a block results in a change of the block's hash. This results in an immutable ledger, which allows the storage of transactions/assets in a tamper-proof way. Since each node holds a copy of the ledger, all nodes are able to verify all prior transactions. This decentralization grants transparency about every transaction for all participants of the network. Private permissioned Blockchain technologies, which restrict the participation in the network and the access to assets within the network, enforce access control. This enables a trustless network, conceding a transaction between parties, who do not trust each other, described in [2].

4 Workflow Enforcement Method

In order to digitalize the certification process for the construction of power plants, the following steps are performed (Fig. 2). First, the use case is modelled using a Workflow Modelling Language. In the second step, the derived models are translated into one coherent model to verify its correctness. Third, the model is translated into Smart Contracts in Hyperledger Fabric, and in the fourth and last step, the Smart Contract functionality is validated in regard to the model, which was verified in the second step.

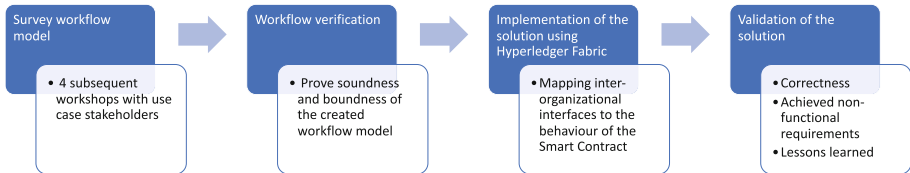


Fig. 2. Overview of our method

4.1 Modelling of the Workflow

The first step for the digitalization of the certification process was modelling the workflow. Four workshops each with three representatives from the involved organizations are performed; two domain experts from the company, in charge of the design and construction of the power plant, and one expert from a governmental authority. We, as researchers and facilitators, analysed several aspects of the process together with the domain experts: the different stages of the safety life cycle, the involved parties, the documents to be certified by the governmental

entity, and the overall workflow. Since this initial modelling phase required participation of domain experts, who are not IT experts, ordinary P/T-Nets were used to make sure the models are easy to understand and adapt.

Dealing with an inter-organizational business process, van der Aalst's method was used for modelling and verifying this type of workflows; the concept of loosely coupled processes, which use asynchronous communication. This method is described in [18], and explained in more detail in [5]. Using this concept, shared places were introduced to the P/T-Nets. This can be seen as an assumption-commitment specification of a contract, which defines the interfaces between the participating parties. In the later implementation only those interfaces are known by everyone, internal business logic stays secret. Therefore, the confidentiality of internal states and processes by all participants is guaranteed, but nevertheless there is an understanding that a certain activity has been done but not how it has been done. Additionally, special operators (i.e. XOR-Joins/Splits) were added to enable decision-making in the Net.

Since the tasks of the certification process of a power plant are bound to documents, only the publication of the so-called documentation packages were modelled. The tool *WoPeD* was used to model the different stages of the certification process in ordinary P/T-Nets with special operators and shared places extension. After two workshops with the domain experts two meta-models emerged. Those meta-models were applied to all models of the different stages of the life cycle. This resulted in 7 models (Plant Design, System Design, Component Design, Fabrication, Mounting, Commissioning, Test Approval), 200 places and 83 transitions. Since a plant consists of multiple systems, which consist of multiple components, and each component is composed of several parts, an overall model was introduced to represent the parallel execution of designing the systems and components, manufacturing the different parts of a component, mounting the components, running each system, and finally combining the systems and running the whole plant.

4.2 Workflow Verification

Extending the ordinary P/T-Nets with shared places and special operators, the Petri Nets could not be verified by the tool *WoPeD*. Therefore, the Petri Nets, resulting from the initial modelling process, were translated to Coloured Petri Nets and verified in *CPN Tools* supporting these two extensions (special operators and shared places). During the translation from P/T-Nets to a CPN, the Net was verified for its behavioural correctness through the token game (simulation), which already eliminated several errors. To verify the resulted CPN, the state space (reachability graph) of the Net was calculated in *CPN Tools*. A report was generated from the calculated state space in *CPN Tools* and analysed in regard to the following properties:

1. Termination: The report contains at least one dead marking, which is also a home marking. (A marking is called dead if it does not enable any transition in the Net. A home marking is reachable from each marking of the reachability graph.)

2. Deadlock-freeness: The overall state space is finite.
3. Boundness and Safeness: The upper bound of each place is 1.
4. Soundness: The report contains only one dead marking.

The first state space, calculated from the finished CPN, was infinite and therefore, only the partial state space was presented in the first report (Fig. 4). The state space was calculated for ten minutes, which resulted in a state space with almost 300000 nodes. Four unbound places (Fig. 3) and over 500 dead-markings existed.

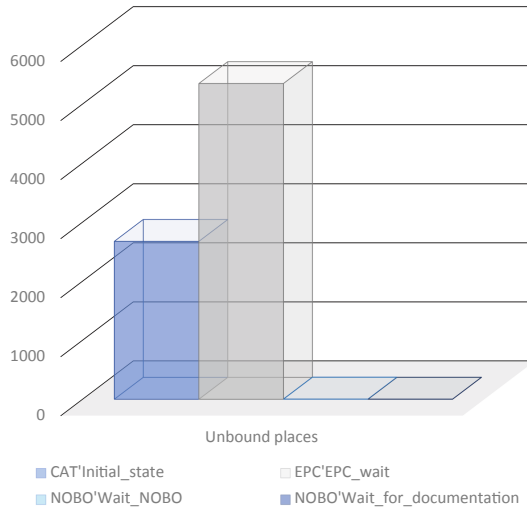


Fig. 3. Statistics about unbound places

Fixing the discovered unbound places, resulted in a finite state space with 186 nodes, 225 arcs and two dead markings (*final report* in Fig. 4). Analyzing the generated state space for the before defined properties resulted in the following: the CPN model terminates, is deadlock-free, holds no unbound places, and is therefore bound. The Net is not sound, since the state space contains two dead markings. Since both markings are valid end states, only distinguished by the placement of tokens in the final places, no further adjustments were made to the model.

4.3 Implementation of the Solution Using Hyperledger Fabric

Cross-organizational business processes require a certain level of trust between the involved organizations. This trust, traditionally guaranteed solely through legal contracts, can be established through the use of Blockchain, together with Smart Contracts. Smart Contracts assure that involved organizations cannot

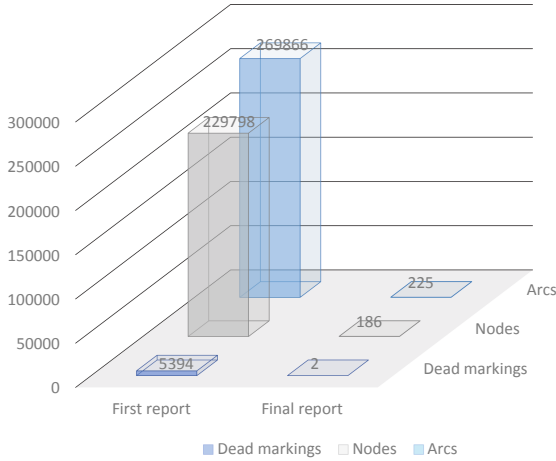


Fig. 4. Statistics about first and final report

break their agreements and therefore, serve as a machine enforceable and verifiable protocol for inter-organizational workflows. Implementing the business process as Smart Contracts on a Blockchain enforces all involved parties to execute the workflow correctly and therefore, establishes trust between all workflow participants without requiring a trusted third party. The following requirements were considered in the choice of a suitable Blockchain platform. All participants of the network must be identifiable at any time. Unauthorized access of non-involved organizations must be prevented. Privacy and confidentiality of transactions and data must be guaranteed at all times. The private permissioned Blockchain platform *Hyperledger Fabric* supports all before mentioned requirements, and is therefore used to implement the use case.

The CPN model was manually translated into Chaincode using *Golang* and the *IBM Blockchain Platform*⁴ extension in *Visual Studio Code*. Interfaces, involving the publication of documents, were extracted from the CPN model and the surrounding transitions were examined. Internal places and transitions, framing the interfaces, were merged to one transition. Figure 5 displays the original part of the CPN model covering the publication of the Plant Design documentation package by EPC.

The internal places and transitions (creating, processing and publishing the documentation package) were merged to one atomic transition (Fig. 6). This transition is only representing the publication of the documentation package itself. This ensures that the internal processes and states of each participant in the workflow, and later in the Blockchain, are kept secret. The transition is surrounded by incoming/outgoing places, constituting the before extracted interfaces. The incoming places are named preconditions, whereas the outgoing

⁴ <https://marketplace.visualstudio.com/items?itemName=IBMBlockchain.ibm-blockchain-platform>.

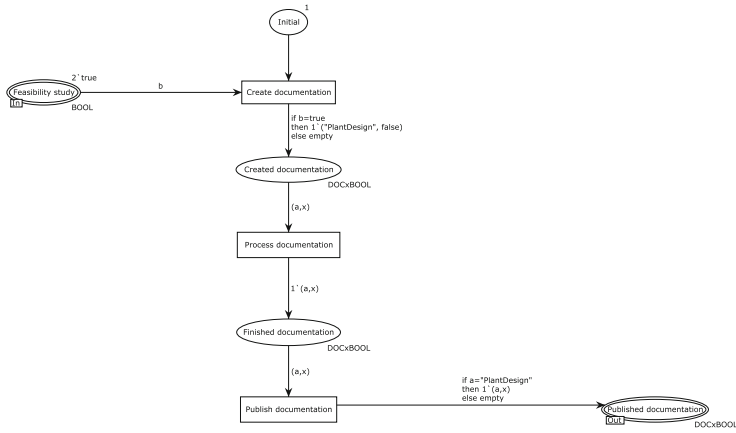


Fig. 5. Part of the CPN model covering the publication of the plant design

places are postconditions, see Fig. 6. All pre- and postconditions are represented as interfaces in the CPN model. In the end, all interfaces constituting published documents were represented in the Smart Contracts, which determined a sequence of tasks.

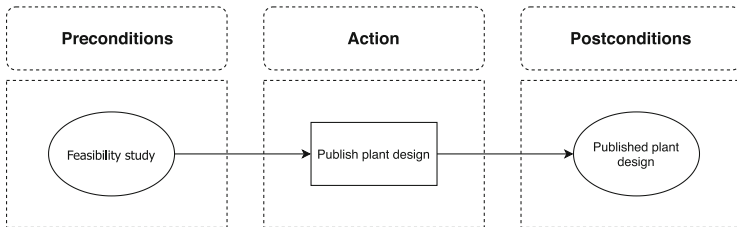


Fig. 6. Simplification of the publication of the plant design

For simplicity, *LevelDB*, a simple key-value store, was used. Additional to *LevelDB*, composite keys were created to be able to improve the search through the database. Only a *Document* structure was stored in the ledger; the actual documentation packages were stored off-chain, in a separate database. The *Document* structure contained a hash of the published documentation package, as well as a link to the actual documents.

The *IBM Blockchain Platform* extension provides a fast way to write, debug and test Chaincode in Hyperledger Fabric without the requirement of an elaborate network setup.

4.4 Validation of the Solution

To evaluate that the Smart Contracts implement the workflow correctly, the token game was run on the model, which resulted in a set of executed tasks, together with their initial and current, local markings. These tuples were put into logical orders:

- **Prerequisites:**
 $publish(FeasibilityStudy) \rightarrow SUCCESS$
- **Positive test case:**
 $published(FeasibilityStudy),$
 $publish(PlantDesign) \rightarrow SUCCESS$
- **Negative test case:**
 $\neg published(FeasibilityStudy),$
 $publish(PlantDesign) \rightarrow ERROR$

The resulting orders were manually translated into unit tests, which were run against the Smart Contracts. The partial state of the Net should be equal to the current state of the ledger.

Formulating unit tests from the extracted logic resulted in 87 tests with a test coverage of 44.4%. To cover also simple checks for wrongly inserted (number of) arguments, additional unit tests were formulated. A regression test was written to cover the overall execution. The written tests cover 56.6% of all statements. Certain error cases (i.e. JSON encoding/decoding failures, errors while getting/putting Document structures from/on the ledger) were not considered in the unit tests, which explains the low test coverage.

5 Results and Discussion

The introduced method is a new approach of digitalizing cross-organizational business processes, displayed in Fig. 7. In the first step of digitalizing the “Certification of the Construction of Industrial Plants”, the workflow of the certification of a power plant was modelled as a simple P/T-Net with extensions that are required for this process. This first modelling process included domain experts from multiple participating organizations. Using ordinary P/T-nets, together with the tool *WoPeD*, during the initial modelling phase was well-received by all participating entities. With their simplicity on the one hand and formal semantics on the other, Petri Nets were accepted from all involved stakeholders. We explained the semantics of Petri Nets to the domain experts while running the token game. Consequently, the domain experts described Petri Nets as “easy to understand and adjust”.

In the second step, the resulting P/T-Nets were translated to CPN and verified in CPN Tools. During the process of translating the simple P/T-Nets into a CPN, the model was verified by simulation (token game) and calculation of the partial state space (reachability graph). The finished Net was then verified by calculating and analysing the whole state space. Behavioural properties

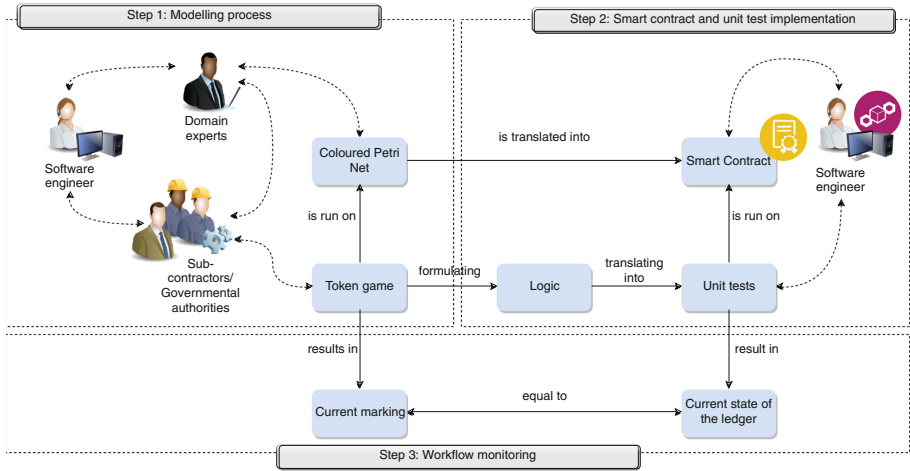


Fig. 7. Suggested development approach

(i.e. termination, deadlock-freeness, boundness and soundness), which should be fulfilled by the Net, were specified beforehand. Since the specified properties were not fulfilled, the CPN was adapted accordingly. Using Petri Nets as a modelling language for cross-organizational business processes, enables a formal verification of the workflow from early stages of the modelling process on.

In this experience report, application specific properties for the models were not considered in the verification process. This should be done in future work to guarantee not only behavioural properties, but make sure that the business process is modelled in the intended way. In our case, an application specific property would look like the following: “the system design phase can only be entered after the plant design is finished and accepted by NOBO”.

After the Net fulfilled all properties, a suitable Blockchain platform was chosen and the CPN model was translated into Smart Contracts. To evaluate that the Smart Contracts implement the workflow correctly, the token game was run on the model and the resulted current markings were translated into logical orders, which were then translated into unit tests. These unit tests were run against the Smart Contracts. It was checked that the partial state of the Net was equal to the current state of the ledger. Translating the extracted logic into unit tests and running them against the Smart Contracts, resulted in a low test coverage, since edge and error cases are not considered in the calculated state space. Therefore, additional unit and regression tests were formulated.

Running the resulted Smart Contracts on Hyperledger Fabric ensures that all participating stakeholders execute the workflow correctly and therefore, establishes trust between the participants.

Although this method, including the elaborate modelling process takes time and contains multiple revisions, the method, introduced in this report, supports a correct implementation of the cross-organizational business process. Unfortu-

nately, a model often cannot represent every detail of the workflow. Therefore, information that is not captured by the model has to be recorded in textual form, can not be verified and has to be considered in the implementation and testing phase additionally.

In case of a change in the model, the Smart Contracts are easily adaptable. Backwards compatibility is not guaranteed. Also, once the Smart Contracts are running, they can't be updated, since in our case keys to get earlier performed transactions from the ledger are saved locally by each node in the Blockchain network and depend on the currently running Smart Contracts. This approach does not force all participants to use the same SDK, but gives each stakeholder the flexibility to use any technology they want behind the API.

6 Related Work

In [4], the authors demonstrate the use of Blockchain technology in the context of inter-organizational business processes for the use case of a documentary letter of credit. Unfortunately, no method was introduced to specify a workflow. Since we are dealing with a large workflow, a method to specify the workflow and translate this specification into Smart Contracts is necessary. Routis investigates the use of Case Management Modelling and Notation for collaborative processes [15]. A method on how to translate the defined processes in Smart Contracts is not given. The authors in [11] introduce the notion of Enforceable Business Processes (EBP), a cooperative stable process, managed by multiple mutually independent organizations with a common goal. Challenges regarding running EBPs as Smart Contracts are analysed. A method on how to map such processes onto Smart Contracts is not introduced.

Caterpillar [10] and Lorikeet [17] translate BPMN to Solidity. Caterpillar is an open-source Business Process Management System running on Ethereum. Models in BPMN are translated to Solidity with the compilation tools provided by Caterpillar. Furthermore, the execution engine enables the deployment of the generated Smart Contract by the compilation tools, as stated in [10]. Lorikeet is a model-driven engineering tool to implement business processes on a Blockchain, described in [17]. The tools create Solidity Smart Contract code from BPMN specifications, which is demonstrated by an industrial use case. These tools restrict the modelling of the business process to BPMN. Since we use Petri Nets to model our use case, because of the above mentioned advantages, those systems are not applicable.

Nakamura presents in [1] a similar approach to ours. Defining the cross-organizational business processes in statecharts, applying statechart reduction algorithms to optimize the size of the statechart and generating software artifacts, Smart Contracts, running on Blockchain. Since we are dealing with an inter-organizational business process that comprises obligations which are agreed upon by all parties beforehand, a reduction of the size of the specified workflow, and therefore, a possible loss of information, is not permitted. Furthermore, statecharts lack formal semantics and therefore, can not be formally verified. The proposed solution in [1] does not include a verification of the statecharts.

7 Conclusion

Today business processes often exceed organizational boundaries and therefore, trust between the organizations that participate in this process has to be established. In the past, this trust was guaranteed solely through legal contracts. To digitally establish trust without requiring a trust third party, Blockchain technology together with Smart Contracts is used to ensure that involved organizations can not break their agreements. Smart Contracts can serve as a machine enforceable and verifiable protocol for inter-organizational workflows. This experience report introduced a solution on how to model, verify and implement an enforcement of cross-organizational business processes. This was done on the basis of the use case “Certification of the Construction of Industrial Plants”. Using different types of Petri Nets and verifying the resulted models in CPN Tools, assured the correct representation of the workflow. The verified models were then translated into Smart Contracts, which were evaluated based on the simulation and reachability graph of the models.

To enhance the outcome of this report, the following points could be considered in the future. Building a generator that generates Smart Contracts from Petri Nets, Coloured Petri Nets in particular, could improve the cost for the method introduced in this report. Furthermore, versioning and updating the Smart Contracts have to be improved, and the admissibility of the ledger and the published documents as evidence in a court of law should be researched.

Acknowledgement. - This research work has been partially funded by the European Union’s H2020 Programme under the Grant Agreement No. 830929 (Cyber-Sec4Europe). Also, we would like to thank our colleagues from Siemens and TÜV who were involved.

References

1. B, S.M.G., Yakhchi, S., Beheshti, A.: Inter-organizational Business Processes Managed by Blockchain, pp. 161–177 (2018). <https://doi.org/10.1007/978-3-030-02922-7>, https://doi.org/10.1007/978-3-030-02922-7_11
2. Christidis, K., Devetsikiotis, M.: Blockchains and smart contracts for the Internet of Things. *IEEE Access* **4**, 2292–2303 (2016). <https://doi.org/10.1109/ACCESS.2016.2566339>
3. Dijkman, R.M., Dumas, M., Ouyang, C.: Formal semantics and analysis of BPMN process models using petri nets. Technical Report, Prepr. 7115, 1–30 (2007). <http://eprints.qut.edu.au/7115/>
4. Fridgen, G., Radszuwill, S., Urbach, N., Utz, L.: Cross-organizational workflow management using blockchain technology - towards applicability, auditability, and automation. In: Proceedings of 51st Hawaii International Conference System Science, vol. 4801 (2018). <https://doi.org/10.24251/hicss.2018.444>
5. Heckel, R.: Open petri nets as semantic model for workflow integration. In: Ehrig, H., Reisig, W., Rozenberg, G., Weber, H. (eds.) *Petri Net Technology for Communication-Based Systems*. LNCS, vol. 2472, pp. 281–294. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-40022-6_14

6. Hollingsworth, D.: Workflow management coalition: the workflow reference model. *Work. Manag. Coalit.* **59**(10), 904–913 (1993). <https://doi.org/10.1007/s00101-010-1752-4>
7. Jensen, K., Kristensen, L.M., Wells, L.: Coloured Petri Nets and CPN tools for modelling and validation of concurrent systems. *Int. J. Softw. Tools Technol. Transf.* **9**(3–4), 213–254 (2007). <https://doi.org/10.1007/s10009-007-0038-x>
8. Kasinathan, P., Cuellar, J.: Securing the integrity of workflows in IoT. In: *International Conference on Embedded Wireless Systems and Networks*, pp. 252–257 (2018)
9. Leal, G., Guédria, W., Panetto, H., Proper, E.: Towards a meta-model for networked enterprise. In: Schmidt, R., Guédria, W., Bider, I., Guerreiro, S. (eds.) *BPMDS/EMMSAD -2016. LNBIP*, vol. 248, pp. 417–431. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39429-9_26
10. López-Pintado, O., García-Bañuelos, L., Dumas, M., Weber, I.: Caterpillar: a blockchain-based business process management system. In: *CEUR Workshop Proceedings*, vol. 1920, pp. 1–5 (2017)
11. Migliorini, S., Gambini, M., Combi, C., La Rosa, M.: The rise of enforceable business processes from the hashes of blockchain-based smart contracts. In: Reinhartz-Berger, I., Zdravkovic, J., Gulden, J., Schmidt, R. (eds.) *BPMDS/EMMSAD - 2019. LNBIP*, vol. 352, pp. 130–138. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20618-5_9
12. Raedts, I., Petković, M., Usenko, Y.S., Van Der Werf, J.M., Groote, J.F., Somers, L.: Transformation of BPMN models for behaviour analysis. In: *Proceedings of 5th International Workshop on Modeling Simulation, Verification and Validation Enterprise Information Systems - MSVVEIS 2007; Conjunction with ICEIS 2007*, pp. 126–137 (2007)
13. Ramadan, M., Elmongui, H., Hassan, R.: BPMN formalisation using Coloured Petri Nets. In: *Proceedings of the 2nd GSTF Annual International Conference on Software Engineering and Application* (2011)
14. Rimba, P., Tran, A.B., Weber, I., Staples, M., Ponomarev, A., Xu, X.: Comparing blockchain and cloud services for business process execution. In: *Proceedings of 2017 IEEE International Conference on Software Architecture ICSA 2017 (Section III)*, pp. 257–260 (2017). <https://doi.org/10.1109/ICSA.2017.44>
15. Routis, I., Nikolaidou, M., Anagnostopoulos, D.: Modeling collaborative processes with CMMN: success or failure? An experience report. In: Gulden, J., Reinhartz-Berger, I., Schmidt, R., Guerreiro, S., Guédria, W., Bera, P. (eds.) *BPMDS/EMMSAD -2018. LNBIP*, vol. 318, pp. 199–210. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91704-7_13
16. Ter Hofstede, A., van der Aalst, W.M.P.: YAWL: yet another workflow language. *Inf. Syst.* **30**(4), 245–275 (2005)
17. Tran, A.B., Lu, Q., Weber, I.: Lorikeet: a model-driven engineering tool for blockchain-based business process execution and asset management. In: *CEUR Workshop Proceedings*, vol. 2196, pp. 56–60 (2018)
18. Van Der Aalst, W.: Loosely coupled interorganizational workflows: modeling and analyzing workflows crossing organizational boundaries. *Inf. Manag.* **37**(2), 67–75 (2000). [https://doi.org/10.1016/S0378-7206\(99\)00038-5](https://doi.org/10.1016/S0378-7206(99)00038-5)