Blockchain-based Traceability of Inter-organisational Business Processes

Claudio Di Ciccio, Alessio Cecconi, Jan Mendling, Dominik Felix, Dominik Haas, Daniel Lilek, Florian Riel, Andreas Rumpl, and Philipp Uhlig

Vienna University of Economics and Business, Vienna, Austria
{alessio.cecconi,claudio.di.ciccio,jan.mendling}@wu.ac.at,
{dominikfelix,dominik.haas1993,daniel.lilek90}@gmail.com,
{flori0n.riel,andreasrumpl,philippuhlig1204}@gmail.com

Abstract. Blockchain technology opens up new opportunities for Business Process Management. This is mainly due to its unprecedented capability to let transactions be automatically executed and recorded by Smart Contracts in multi-peer environments, in a decentralised fashion and without central authoritative players to govern the workflow. In this way, blockchains also provide traceability. Traceability of information plays a pivotal role particularly in those supply chains where multiple parties are involved and rigorous criteria must be fulfilled to lead to a successful outcome. In this paper, we investigate how to run a business process in the context of a supply chain on a blockchain infrastructure so as to provide full traceability of its run-time enactment. Our approach retrieves information to trace process instances execution solely from the transactions written on-chain. To do so, hash-codes are reverse-engineered based on the Solidity Smart Contract encoding of the generating process. We show the results of our investigation by means of an implemented software prototype, with a case study on the reportedly challenging context of the pharmaceutical supply chain.

Keywords: Blockchain · Ethereum · Smart Contracts · Supply Chain · Business Process Management

1 Introduction

Integrating processes that extend throughout the supply chain requires multi-party collaborations and an intense exchange of information along multiple channels. The multitude of passages entails data redundancy and lack of full knowledge on how, when and where tasks were conducted, and products used through their life cycle [24]. This becomes an issue of particular concern for inter-organisational processes involving untrusted parties [6]. In a pharmaceutical supply chain, for instance, critical goods are handled and the verifiability of their origin and processing is needed to prevent illegal actions, e.g., the distribution of counterfeit drugs. Multiple actors work together on delivering a product that needs to have very specific qualities to be fulfilled. Thus *traceability* is required.

Pre-print copy of the manuscript published by Springer (available at link.springer.com)
identified by DOI: 10.1007/978-3-319-94214-8_4

Classic approaches against counterfeiting, like the one proposed in [25] are not always sufficient, because supply chains usually lack trust, transparency, and documentation. The key properties of the blockchain technology offers a promising solution to this problems without the need of a third party authority.

Blockchain has emerged as an open, distributed ledger that can record transactions between parties efficiently and in a verifiable and permanent way. This is enabled by a combination of peer-to-peer networks, consensus-making, cryptography, and market mechanisms [5]. Blockchains thus ensure data integrity and transparency [22]. Furthermore, they support so-called Smart Contracts, that is, fully executable pieces of decentralised code expressing how business is to be conducted among contracting parties, e.g., transfer cryptocurrencies and digital assets after a condition is fulfilled [28,7]. These characteristics make it particularly suitable to the execution of inter-organisational business processes along supply chains [30].

In this paper we investigate the application of the Ethereum blockchain to enable traceability of inter-organisational business process, focussing on a case study taken from the pharmaceutical domain. We remark that we abstract from the sole product traceability in supply chains [17,29] and aim at extending the concept towards full traceability of the entire process execution [16]. To that extent, we rely on an existing platform for running business processes on the Ethereum blockchain [4,13] and devise a framework to trace the execution information solely based on the transactions recorded on-chain. Our approach is implemented in a software prototype and its usage demonstrated on an exemplifying process.

The remainder of the paper is structured as follows. Section 2 describes the preliminary notions upon which out approach is based. Section 3 illustrates our approach in detail. Section 4 concludes the paper and draws future research plans.

2 Background

An ever-increasing number of organisations both in the private and public sectors identify their business processes as a key asset. Business processes, or *processes* hereinafter, regulate the inter-relation, assignment, and execution of tasks and decisions that ultimately yield an outcome that adds value for a customer [6]. Business Process Model and Notation (BPMN) is among the most prominent modelling languages for processes [21]. The integration of business processes along the supply chain, has been found to contribute both to better operational and business performance [15,30].

A supply chain is defined as the set of activities and independent organisations whose cooperation delivers a product from its production to the end consumer [12,26]. A pharmaceutical supply chain, for instance, encompasses the stages pertaining to a pharmaceutical product from the raw material processing to the end consumer acquisition of the final product. More in detail, it includes the following tasks: *(i)* manufacturers produce raw materials, *(ii)* pharmaceutical manufacturers produce the final product, *(iii)* hospitals and pharmacies store and

Pre-print copy of the manuscript published by Springer (available at `link.springer.com`)
identified by DOI: `10.1007/978-3-319-94214-8_4`

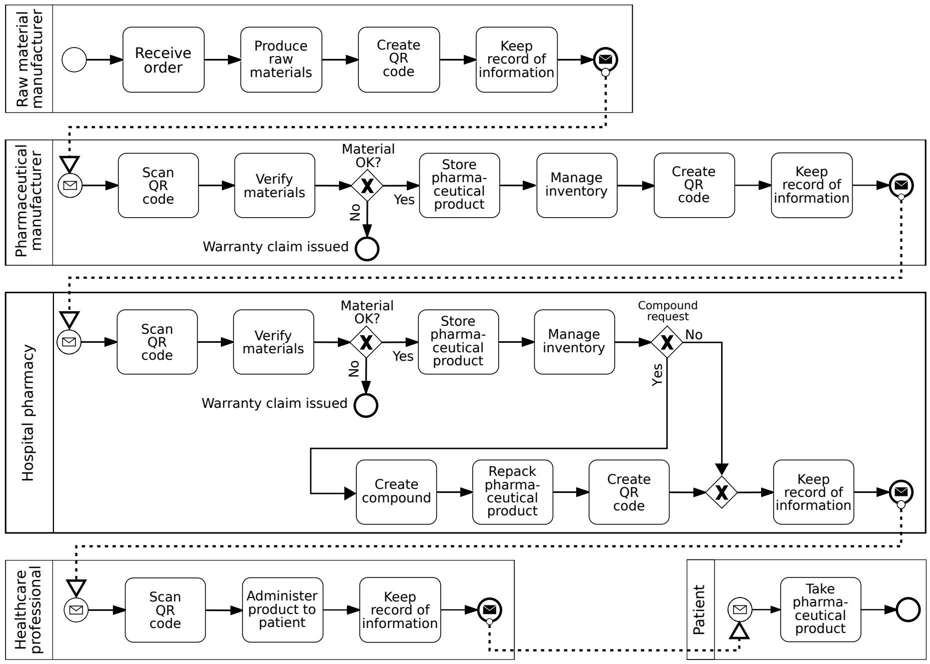


Fig. 1: Simplified pharmaceutical supply chain modelled as a BPMN collaboration diagram.

dispense the final product dispatched by wholesalers and distributors, *(iv)* patients take the final product. The coactive work of the many parties involved is paramount as the efforts lead to a successful outcome only if the internal coordination, information exchange, and material flow is efficient. A weakness anywhere in the chain results in an overall failure to achieve the competitive potential of the whole chain. In other words, the chain is only as strong as its weakest link [27]. Standards exist to regulate both the activity flow and the communication between the actors of supply chains. A simplified model of the pharmaceutical supply chain, e.g., is illustrated in Fig. 1 as a BPMN collaboration diagram. It is based upon the model proposed in [10] by GS1, a non-profit organisation that develops and maintains global standards for business communication. The diagram consists of five pools, represented as wrapping labelled boxes, each depicting one of the different parties involved: raw material manufacturer, pharmaceutical manufacturer, hospital pharmacy, healthcare professional, and patient. Within each pool, processes are represented as tasks (rounded boxes) connected by control-flow (arcs), starting and ending with events (circles, drawn with narrow and bold lines respectively). Envelopes represent a message exchange, and dotted arcs connect sending events (black envelope) to catching events (transparent envelope). For instance, pharmaceutical products are delivered to the hospital pharmacy from the pharmaceutical manufacturer after task Keep record of information is

Pre-print copy of the manuscript published by Springer (available at link.springer.com)
identified by DOI: 10.1007/978-3-319-94214-8_4

concluded. The pharmacy thereafter scans the QR code of the package to register the delivery. Routing choices are drawn as diamonds with a cross. Considering the process of the pharmaceutical manufacturer, e.g., if raw materials are corrupted the process ends with a warranty claim issued, otherwise it proceeds with the production of pharmaceutical product.

A cornerstone of Business Process Management (BPM) is the integration of IT systems in the management of processes to favour their automation. Blockchains can provide an underlying technological infrastructure for business process management.

2.1 Blockchain

The concept of blockchain was introduced in 2008 to support the creation and exchange of the *Bitcoin* cryptocurrency without the need for a controlling central authority [20,14]. While indeed the primary driver of blockchain technology has been cryptocurrencies, it has later found a broader application field outside the sole finance scope. The foundation of a blockchain is the decentralised storage of data that describe transactions. When we think of a traditional bank, the bank is a centralised node of all transactions, the ledger. By contrast, the blockchain network stores the transactions on the different peers that are part of the network, in a *distributed ledger* fashion. This allows for a direct exchange between different participants without a third party that observes the transactions. The transactions are collated in blocks and several blocks together result in a chain. In this way, a blockchain describes a logical sequence of transactions. Adding new blocks in a trustworthy manner is enabled by a combination of peer-to-peer networks, consensus-making, cryptography, and market mechanisms [19]. Blockchains ensure data integrity and transparency, such that the blockchain network stays operational even under byzantine faults. A new block can only be added to the chain if there is a *consensus* on the majority of the network. The most common consensus techniques are Proof-of-Work (PoW) and Proof-of-Stake (PoS) [2]. Ethereum [32] is among the most used blockchain implementations.

All transactions are signed. Nevertheless, the identity of the signee remains anonymous. Consensus and anonymity are facilitated by hashing. A *cryptographic hash function* (henceforth simply *hash function*) is a deterministic one-way function mapping any given input to a fixed-size output. It means that given the same input, the same output is always returned. It is however not feasible to retrieve the original input from from the sole output. Keccak [3] is the hash function used in Ethereum. Every block is associated with a hash generated from its content *and* the hash value of the previous block in the list. Hash values thus uniquely represent not only the transactions within blocks but also the ordering of every block. This mechanism is at the basis of the chain. In case somebody tried to alter a transaction, this would change the hash value of its block, and thus break the chain. Different types of blockchains exist, but they can be generally classified by two properties, i.e., verifiability (*private* or *public*), and access grant (*permissioned* or *permissionless*) [33]. The difference between *public* and *private blockchains* is the visibility of the network: in the former the

Pre-print copy of the manuscript published by Springer (available at `link.springer.com`)
identified by DOI: `10.1007/978-3-319-94214-8_4`

network can be joined and inspected by everyone, while in the latter only by those who are invited. Instead, *permissionless* and *permissioned blockchains* are differentiated by the capability to add blocks to the chain: in the former every peer in the network can add blocks, i.e., participate to the consensus, while in the latter only allowed ones by the network.

Blockchains allow for the execution of Smart Contracts, i.e., software programs that react upon triggers provided by users or generated by the environment, running within the blockchain [28,7,9]. Distributed ledger systems constitute computational platforms then, thus going beyond mere distributed databases. The *Ethereum* blockchain for instance provides a Turing-complete language to encode Smart Contracts, namely *Solidity*[1] [4]. Computational effort comes at a price for Smart Contracts. Operations are associated to a cost measured in *gas*, a unit of work that is meant to be paid in *Ether*, the Ethereum cryptocurrency. In order for a Smart Contract to complete its execution, enough Ether have to be in the wallet of the invoker.

The implication of using blockchain for BPM has been discussed under diverse viewpoints. Mendling et al. [16] highlight the challenges and opportunities of blockchain for BPM in relation to the six BPM core capability areas [23] and in relation to the traditional BPM lifecycle [6]. The first of seven future research directions listed in [16] relates to our work in particular, namely the development of a diverse set of execution and monitoring systems on blockchain. The fundamental problem of trust in collaborative process execution is highlighted in [30]. The authors develop a technique to integrate blockchain into the choreography of processes in order to avoid the need for a central authority and still maintain trust. A first method to compile a process model directly into an Ethereum Smart Contract is presented in [8] by focussing on three areas: initialisation cost for process instances, task execution cost by means of a space-optimised data structure and improved runtime improvements for maximised throughput. The implemented software prototype, Caterpillar [13], is used in the context of our research.

3 Approach

Enabling traceability of inter-organisational business processes implies the capability of tracking the status of ongoing instances and reconstructing the history of its execution. Such a view transcends the monitoring of sole business objects, information artefacts, or carriers, hence transitioning to a more holistic approach. To that extent, in this work we focus on the identification and linkage of all transactions that report on activities of running processes. The blockchain opens up the opportunity of retrieving that information, which is digitally stored and shared by different parties in a decentralised manner.

Our approach begins with the encoding of supply chain business process models into executable Solidity programs. To that extent, we resort on the Caterpillar

[1] https://solidity.readthedocs.io

Pre-print copy of the manuscript published by Springer (available at link.springer.com)
identified by DOI: 10.1007/978-3-319-94214-8_4

Table 1: Addresses of Smart Contracts (SCs) deployed by Caterpillar on the blockchain.

	Global Factory SC address:	`0x8cdaf0cd259887258bc13a92c0a6da92698644c0`
	Process Factory SC address:	`0x345ca3e014aaf5dca488057592ee47305d9b3e10`
	Worklist Factory SC address:	`0xf12b5dd4ead5f743c6baa640b0216200e89b60da`
Prc. Instance	**Worklist SC address**	**Process Instance SC address**
1	`0x6512a267ad28dfe41a5846e7ad0b2501633cb3f2`	`0xf2beae25b23f0ccdd234410354cb42d08ed54981`
2	`0x0ebe109b4ac5de65d63f7d7e5a856dcd77dc58fd`	`0xaa8f61728cb614f37a2fdb8b420c3c33134c7f69`
3	`0x22029e89e1d1f79d8e57c9af2fb9bf653bdf4be1`	`0xf21cf97429e6f7338ae989135ff9aa0225719347`

tool [13]. We assume that activities executed by Solidity Smart Contracts are registered as transactions. This is typical in collaborative business processes run on the blockchain [30,8]. Thereupon, we resort on the Solidity encoding of the process model to reverse-engineer the hash-codes and reveal *(i)* the process instance committing the transaction, and *(ii)* the operation signature, denoting the task. We remark here that the latest step requires knowledge on the encoded process model because hash-codes cannot be reverted into original input signatures. In the following, we detail the aforementioned passages.

3.1 Caterpillar: From Process Models to Smart Contracts

Caterpillar acts as a Business Process Management System (BPMS) operating on the blockchain by means of Smart Contracts (SCs). The Caterpillar tool [13] relies on a *Global Factory* SC, acting as a container for all processes and deployed on the blockchain as soon as Caterpillar is launched. Given a BPMN process model like the one of Fig. 1, the tool generates a *Process Factory* SC and a *Worklist Factory* SC. In turn, the Worklist Factory SC generates new *Worklist* SCs. Each of them is associated to a *Process Instance* SC, generated by the Process Factory SC at every start of the process. The Worklist SC routes the execution of process activities as dictated by the workflow depicted in the process model. In other words, the control-flow logic is embedded in every Worklist SC. Every activity corresponds to a function of the Worklist SC, which then forwards the call to the corresponding Process Instance SC. Notice that the Worklist SC registers a transaction for every activity execution. Table 1 shows examples of addresses assigned by Caterpillar to the Smart Contracts described so far.

3.2 Executing Processes through Smart Contracts

Let us consider that a unit of medicines is handed over from the manufacturer to the hospital pharmacy, as in the example of Fig. 1. A transaction is required to confirm that the hand-over took place. Nowadays, a centralised system would register this information and involved parties require access to that. This naturally extends to the blockchain environment, which makes the information readily

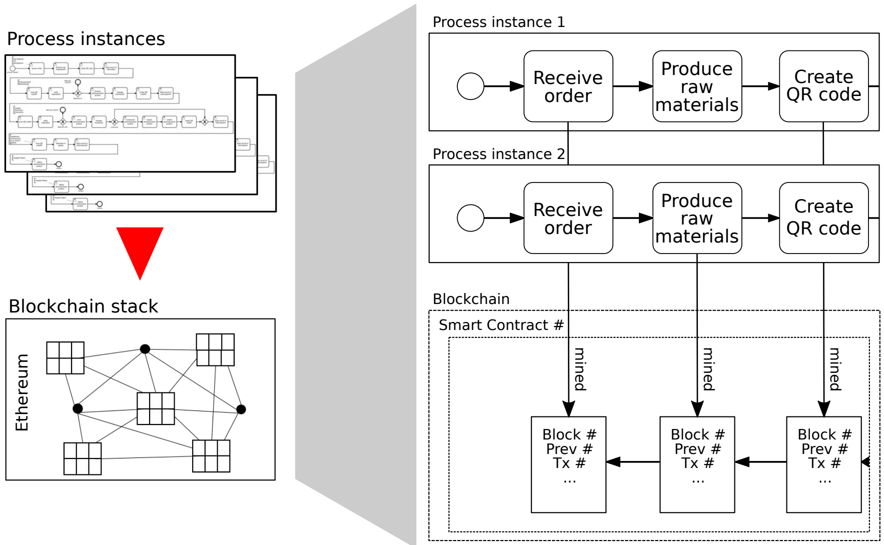


Fig. 2: The execution of process instances on the blockchain

available to all counterparts in a decentralised manner. Figure 2 depicts how process instances are deployed and executed on the blockchain through Caterpillar.

Once a process instance is initiated through Caterpillar, a new block is added to the chain each time an activity is executed. Inside the block, the transaction reporting on the activity execution is collated. Transactions record, among other things, *(i)* the *contract address* defining the called contract, i.e., the Worklist SC, *(ii)* the *sender address* of the user carrying out the activity, and *(iii)* the *transaction data*, containing the hash-code of the function signature corresponding to the activity. They are the fundamental building blocks for traceability.

3.3 Tracking Activities in the Blockchain

Different process instances may run at the same time. Accordingly the resulting blockchain will be a sequence of blocks originating from different instances. Furthermore, each block is identified by its hash-code, which does not explicitly disclose the process instance that generated the block itself. Therefore we present a procedure to differentiate the process instances in order to allow for their tracking, as depicted in Fig. 3.

Through Ethereum modules it is possible to extract the blocks of a blockchain, along with the individual transactions and all the information held by them. In particular we are interested in the aforementioned *contract address* and *transaction data*. Interestingly, operational transactions generated from the process instance execution are such that the *contract address*es correspond to the address

Pre-print copy of the manuscript published by Springer (available at `link.springer.com`)
identified by DOI: `10.1007/978-3-319-94214-8_4`

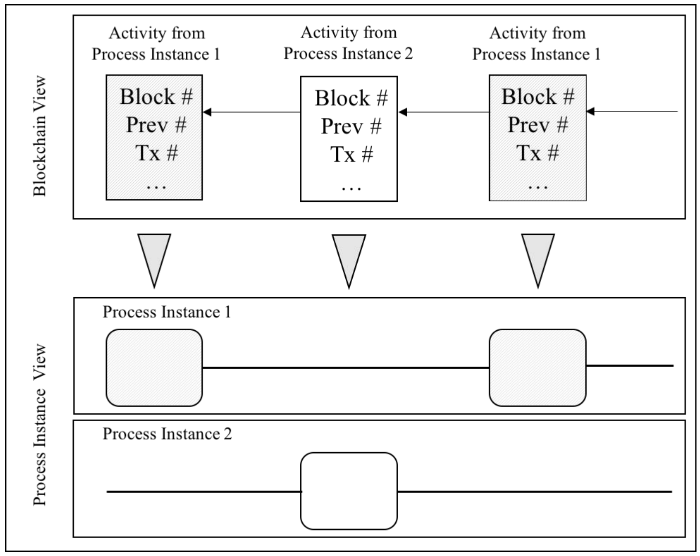


Fig. 3: Extraction of process instances

of the router of activities via function calls, i.e., of the Worklist SC, which can be retrieved via Application-Binary Interface (ABI). Therefore, to trace the execution of a process instance, we search for all the transactions resulting from the invocation of the corresponding Worklist SC: to that extent, we look up for matches between the value of the contract address field in the transaction payload and the Worklist SC registered in Caterpillar as depicted in Table 2.

To extract the specific task contained in a block, we fetch the content of the *transaction data* field of every transaction. Because it stores the function signature, it reveals a reference to the executed task once encoded. The *transaction data* consists in a hash-code which can be divided in two parts: the first 4 bytes represent the *function identifier*, while the following bytes enclose the actual parameters of the function. We thus identify the executed activity by parsing the *transaction data* hash-code, so as to compare it with the hashed function signatures.

To sum up, we compare the *contract address* of a transaction to the hash-code of the Worklist SC to identify the process instance, and we match the *function identifier* of the *transaction data* with the hash-code of function signatures to retrieve the activity.

3.4 Example

An excerpt of all addresses assigned to the Smart Contracts (SCs) deployed on the blockchain by Caterpillar is reported in Table 1. We consider three process instances, numbered from 1 to 3 in the table. We recall that to every process instance correspond two deployed SCs, namely a Worklist SC, generated by the Worklist

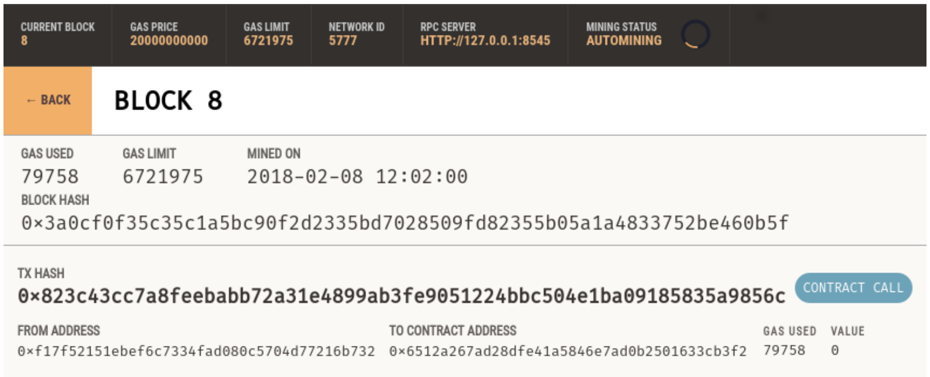
Table 2: Excerpt of recorded transactions and mapping to process instances via matching of the addresses listed in Table 1.

Transaction	Contract address	Instance
1	0x8cdaf0cd259887258bc13a92c0a6da92698644c0	
2	0xf12b5dd4ead5f743c6baa640b0216200e89b60da	
...	...	
7	0x8cdaf0cd259887258bc13a92c0a6da92698644c0	
8	0x6512a267ad28dfe41a5846e7ad0b2501633cb3f2	1
9	0x0ebe109b4ac5de65d63f7d7e5a856dcd77dc58fd	2
10	0x6512a267ad28dfe41a5846e7ad0b2501633cb3f2	1
11	0x0ebe109b4ac5de65d63f7d7e5a856dcd77dc58fd	2
12	0x8cdaf0cd259887258bc13a92c0a6da92698644c0	
13	0x8cdaf0cd259887258bc13a92c0a6da92698644c0	
14	0x22029e89e1d1f79d8e57c9af2fb9bf653bdf4be1	3
15	0x6512a267ad28dfe41a5846e7ad0b2501633cb3f2	1
16	0x0ebe109b4ac5de65d63f7d7e5a856dcd77dc58fd	2
17	0x0ebe109b4ac5de65d63f7d7e5a856dcd77dc58fd	2
18	0x22029e89e1d1f79d8e57c9af2fb9bf653bdf4be1	3
19	0x22029e89e1d1f79d8e57c9af2fb9bf653bdf4be1	3
20	0x0ebe109b4ac5de65d63f7d7e5a856dcd77dc58fd	2
21	0x6512a267ad28dfe41a5846e7ad0b2501633cb3f2	1
22	0x6512a267ad28dfe41a5846e7ad0b2501633cb3f2	1
23	0x22029e89e1d1f79d8e57c9af2fb9bf653bdf4be1	3
24	0x6512a267ad28dfe41a5846e7ad0b2501633cb3f2	1

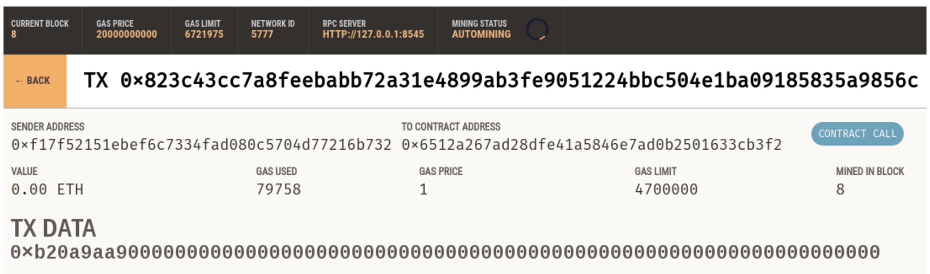
Factory SC, and a Process Instance SC, generated by the Process Factory SC. The hexadecimal address of the Worklist SC of process instance 1 in Table 1 is, e.g., `0x6512a267ad28dfe41a5846e7ad0b2501633cb3f2`, associated to the Process Instance SC having address `0xf2beae25b23f0ccdd234410354cb42d08ed54981`. The Worklist SC is the façade for the execution of tasks: single operations correspond to a task each, and re-route the call to the Process Instance SC once invoked. Considering the example process of Fig. 1, the signature of the function in the Worklist SC corresponding to Receive order is `receive_order(uint256)`. Every execution of Receive order corresponds to a transaction reporting the invocation of `receive_order(uint256)` on the Worklist SC of a process instance.

Figure 4 illustrates the information displayed by the Ganache tool[2] on the transactions recorded in the blockchain. More specifically, the block in Fig. 4(a) contains the transaction reporting a task execution of the example process. Transaction-specific data are illustrated in Fig. 4(b). Notice that the *contract address* field is `0x6512a267ad28dfe41a5846e7ad0b2501633cb3f2`, i.e., the same as the address of the aforementioned Worklist SC. This indicates that the running instance is the one we marked as instance 1 in Table 1. Table 2 shows a list

[2] `http://truffleframework.com/ganache/`



(a) The block containing the transaction of Receive order.



(b) The transaction in the block of Fig. 4(a) reporting on the execution of Receive order. The *transaction data* field is magnified for the sake of readability.

Fig. 4: Tracing of task execution on the blockchain

of such mappings between transactions and process instances based on data registered on the blockchain.

To understand what the executed activity is, we analyse the first 4 bytes of the hexadecimal *transaction data* field of the recorded transaction, e.g., `0xb20a9aa9` in Fig. 4(b). The Keccak hash of the signature of the operation corresponding to task Receive order, i.e., `receive_order(uint256)`, is the following: `0xb20a9aa9619b9dc0c6a76bef0c51350699afd93b05757350f8c20de71bc75e18`. We observe that its first 4 bytes are `0xb20a9aa9` too. We thereby recognise that the reported execution corresponds to task Receive order of process instance 1.

3.5 Limitations

The initial results of our approach demonstrate the applicability of a fully automated technique to enable traceability of inter-organisational processes deployed on the blockchain. However, we acknowledge limitations which draw future research endeavours. First and foremost, we remark that the hash-code comparison to identify process and task instances is based on Caterpillar. However, the rationale is rooted in a reasonable design choice which other blockchain-based BPMSs might adopt as well. Indeed, the underlying idea follows the paradigm of self-identification which is inherent to the concept of blockchain, according to which transactions must be signed by the transactor – in this case, contracts enabling the process execution.

Our approach to identify process and task instances requires that interested parties share knowledge on the hash key and hashing function in use. This calls for further investigations, because the fully public access to that information might be undesirable as knowledge on the processes could be disclosed. Those studies would build upon *(i)* methodologies for the specification of interfacing (visible) and internal (hidden) fragments of cooperating processes [31,1] and *(ii)* infrastructural solutions based on permissioned access grants enforced by cryptography of undisclosed information [18].

At the time of writing, the capability of handling BPMN pools in Caterpillar is under ongoing implementation, therefore unique process instance identifiers are used to indicate all operations pertaining to the inter-organisational processes. In a distributed scenario, diverse process instances with different identifiers could run, e.g., one per pool, even on different platforms accessing the shared blockchain. This suggests the implementation of novel techniques based upon record linkage and object matching [11,34] that allow for an automated recognition of matching instances.

4 Conclusions

In this paper we presented a technical solution to ensure traceability through blockchain of inter-organisational business process. Our case study pertains to the pharmaceutical domain, and is based upon the model defined by the GS1 standard.

The results gathered through the investigation via our implemented prototype shows feasibility and correctness of the approach, thus representing a first step towards a comprehensive framework. Our future work to extend the approach includes the overcoming of discussed limitations, as well as the following points. Firstly, it is in our plans to investigate the integration of other BPMSs and blockchain technologies than Caterpillar with Ethereum, e.g., IBM Blueworks Live[3] with Hyperledger[4], or Bonitasoft Bonita BPM[5] with Chain Core[6]. To that extent, we will create dedicated programming interfaces that act as a façade towards the implementation layer, thus allowing for higher-level BPM functionalities. Furthermore, we argue that the exchange of information between the blockchain and the real-world is crucial in a supply chain. Therefore we aim at extending our approach to include agents connecting with the real world from within the blockchain, also known as *oracles*, also considering possible security

[3] `https://www.ibm.com/cloud/automation-software/business-process-management`

[4] `https://hyperledger.org`

[5] `https://www.bonitasoft.com`

[6] `https://chain.com/technology`

Pre-print copy of the manuscript published by Springer (available at `link.springer.com`)
identified by DOI: `10.1007/978-3-319-94214-8_4`

threats arising from them, e.g., trustfulness of those agents. This would enable the triggering of activity completions via scanning of QR-codes, e.g., as suggested by GS1 [10]. Finally, we aim at the creation of rich event logs to allow for process mining on the blockchain [16].

Acknowledgements. The work of Claudio Di Ciccio was partially funded by the Austrian Research Promotion Agency (FFG) under grant 862950 (Business Process Optimization Toolkit). The work of Alessio Cecconi was funded by the Austrian Research Promotion Agency (FFG) under grant 861213 (CitySPIN). The authors want to thank Orlenys López-Pintado, Luciano García Bañuelos, and Marlon Dumas, for the valuable advice and precious technical help provided with Caterpillar.

References

1. van der Aalst, W.M.P., Mooij, A.J., Stahl, C., Wolf, K.: Service interaction: Patterns, formalization, and analysis. In: SFM. pp. 42–88. Springer (2009)
2. Bentov, I., Gabizon, A., Mizrahi, A.: Cryptocurrencies without proof of work. In: FC Workshops. pp. 142–157 (2016)
3. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: The making of KECCAK. Cryptologia 38(1), 26–60 (2014), https://doi.org/10.1080/01611194.2013.856818
4. Dannen, C.: Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners. Apress (2017)
5. Diedrich, H.: Ethereum: Blockchains, Digital Assets, Smart Contracts, Decentralized Autonomous Organizations. Wildfire Publishing (2016)
6. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management, Second Edition. Springer (2018)
7. Egelund-Müller, B., Elsman, M., Henglein, F., Ross, O.: Automated execution of financial contracts on blockchains. Business & Information Systems Engineering 59(6), 457–467 (2017)
8. García-Bañuelos, L., Ponomarev, A., Dumas, M., Weber, I.: Optimized Execution of Business Processes on Blockchain. In: Carmona, J., Engels, G., Kumar, A. (eds.) Business Process Management. pp. 130–146. Springer International Publishing (2017)
9. Governatori, G., Idelberger, F., Milosevic, Z., Riveret, R., Sartor, G., Xu, X.: On legal contracts, imperative and declarative smart contracts, and blockchain systems. Artificial Intelligence and Law (2018)
10. GS1: Global Traceability Standard for Healthcare (2013), https://www.gs1.org/sites/default/files/docs/traceability/Global_Traceability_Standard_Healthcare.pdf
11. Inan, A., Kantarcioglu, M., Bertino, E., Scannapieco, M.: A hybrid approach to private record linkage. In: ICDE. pp. 496–505. IEEE Computer Society (2008)
12. Lu, D.: Fundamentals of Supply Chain Management. Ventus Publishing ApS (2011)
13. López-Pintado, O., García-Bañuelos, L., Dumas, M., Weber, I.: Caterpillar: A Blockchain-Based Business Process Management System. In: BPM Demos. vol. 1920. CEUR-WS.org (2017)
14. Magazzeni, D., McBurney, P., Nash, W.: Validation and verification of smart contracts: A research agenda. IEEE Computer 50(9), 50–57 (2017)

Pre-print copy of the manuscript published by Springer (available at link.springer.com) identified by DOI: 10.1007/978-3-319-94214-8_4

15. McAdam, R., McCormack, D.: Integrating business processes for global alignment and supply chain management. Business Proc. Manag. Journal 7(2), 113–130 (2001)
16. Mendling, J., Weber, I., van der Aalst, W.M.P., vom Brocke, J., Cabanillas, C., Daniel, F., Debois, S., Di Ciccio, C., Dumas, M., Dustdar, S., Gal, A., García-Bañuelos, L., Governatori, G., Hull, R., Rosa, M.L., Leopold, H., Leymann, F., Recker, J., Reichert, M., Reijers, H.A., Rinderle-Ma, S., Solti, A., Rosemann, M., Schulte, S., Singh, M.P., Slaats, T., Staples, M., Weber, B., Weidlich, M., Weske, M., Xu, X., Zhu, L.: Blockchains for business process management - challenges and opportunities. ACM Trans. Management Inf. Syst. 9(1), 4:1–4:16 (2018)
17. Merminod, N., Paché, G.: Supply management and corporate social responsibility: the challenge of global chain traceability. Journal on Chain and Network Science (11), 213–222 (2011)
18. Miklau, G., Suciu, D.: Controlling access to published data using cryptography. In: VLDB. pp. 898–909. Morgan Kaufmann (2003)
19. Mougayar, W.: The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology. Wiley (2016)
20. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008), https://bitcoin.org/bitcoin.pdf
21. OMG: Business Process Model And Notation Specification Version 2.0 (2011), https://www.omg.org/spec/BPMN/2.0
22. Risius, M., Spohrer, K.: A blockchain research framework - what we (don't) know, where we go from here, and how we will get there. Business & Information Systems Engineering 59(6), 385–409 (2017)
23. Rosemann, M., vom Brocke, J.: The Six Core Elements of Business Process Management, pp. 105–122. Springer (2015)
24. Saveen, A., Monfared, R.P.: Blockchain ready manufacturing supply chain using distributed ledger. International Journal of Research in Engineering and Technology 5(9), 1–10 (2016)
25. Shah, N.: Pharmaceutical supply chains: Key issues and strategies for optimisation. Computers & Chemical Engineering 28, 929–941 (2004)
26. Snyder, L.V., Shen, Z.J.M.: Fundamentals of supply chain theory. Wiley (2011)
27. Sweeney, E.: Towards a unified definition of supply chain management: The four fundamentals. International Journal of Applied Logistics, IJAL 2(3), 30–48 (2011)
28. Szabo, N.: Formalizing and securing relationships on public networks. First Monday 2(9) (1997), http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/548
29. Tian, F.: A supply chain traceability system for food safety based on haccp, blockchain internet of things. In: ICSSSM. pp. 1–6 (2017)
30. Weber, I., Xu, X., Riveret, R., Governatori, G., Ponomarev, A., Mendling, J.: Untrusted Business Process Monitoring and Execution Using Blockchain. In: BPM. pp. 329–347. Springer, Cham (2016)
31. Wolf, K.: Does my service have partners? Trans. Petri Nets and Other Models of Concurrency 2, 152–171 (2009)
32. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger (2018), https://ethereum.github.io/yellowpaper/paper.pdf
33. Wüst, K., Gervais, A.: Do you need a Blockchain? Tech. Rep. 375 (2017), http://eprint.iacr.org/2017/375
34. Zardetto, D., Scannapieco, M., Catarci, T.: Effective automated object matching. In: ICDE. pp. 757–768. IEEE Computer Society (2010)

Pre-print copy of the manuscript published by Springer (available at link.springer.com) identified by DOI: 10.1007/978-3-319-94214-8_4

This document is a pre-print copy of the manuscript
(Di Ciccio et al. 2018)
published by Springer (available at link.springer.com).

The final version of the paper is identified by DOI: 10.1007/978-3-319-94214-8_4

References

Di Ciccio, Claudio, Alessio Cecconi, Jan Mendling, Dominik Felix, Dominik Haas, Daniel Lilek, Florian Riel, Andreas Rumpl, and Philipp Uhlig (2018). "Blockchain-Based Traceability of Inter-organisational Business Processes". In: *BMSD*. Ed. by Boris Shishkov. Vol. 319. Lecture Notes in Business Information Processing. Springer, pp. 56–68. ISBN: 978-3-319-94214-8. DOI: 10.1007/978-3-319-94214-8_4.

BibTeX

```
@InProceedings{   DiCiccio.etal/BMSD2018:BlockchainBasedTraceabilityBusinessProcesses,
  author        = {Di Ciccio, Claudio and Cecconi, Alessio and Mendling, Jan
                  and Felix, Dominik and Haas, Dominik and Lilek, Daniel and
                  Riel, Florian and Rumpl, Andreas and Uhlig, Philipp},
  title         = {Blockchain-Based Traceability of Inter-organisational
                  Business Processes},
  booktitle     = {BMSD},
  year          = {2018},
  pages         = {56--68},
  crossref      = {BMSD2018},
  doi           = {10.1007/978-3-319-94214-8_4},
  keywords      = {Blockchain; Ethereum; Smart Contracts; Supply chain;
                  Business Process Management}
}
@Proceedings{     BMSD2018,
  title         = {Business Modeling and Software Design, BMSD 2018, Vienna,
                  Austria, July 2-4, 2018, Proceedings},
  year          = {2018},
  editor        = {Boris Shishkov},
  volume        = {319},
  series        = {Lecture Notes in Business Information Processing},
  publisher     = {Springer},
  isbn          = {978-3-319-94214-8}
}
```