A Blockchain-based approach for Trust Management in Collaborative Business Processes

Ada Bagozi[1], Devis Bianchini[1], Valeria De Antonellis[1], Massimiliano Garda[1], and Michele Melchiori[1]

University of Brescia, Dept. of Information Engineering
Via Branze 38, 25123 - Brescia (Italy)
a.bagozi@unibs.it, devis.bianchini@unibs.it,
valeria.deantonellis@unibs.it, m.garda001@unibs.it,
michele.melchiori@unibs.it

Abstract. Blockchain is becoming a powerful technology for re-engineering collaborative business processes implemented on Web-based distributed systems, spanning across enterprises. On the blockchain, cross-organisation Web services orchestrated to form collaborative business processes can be transparently deployed as smart contracts. However, proper methods and tools are required to guide the process designer for exploiting the blockchain technology. To preserve data and business logics ownership and to ensure performance/cost tradeoff, only data and process activities requiring transparency and trust among the distributed process actors should be stored as transactions on the blockchain and deployed as smart contracts. In this paper, we propose a methodology and a tool that rely on methodological steps to support blockchain-based trust management in Web-based collaborative business processes originally designed according to a centralised BPM strategy. The methodology and the tool are grounded on a set of criteria, properly enforced with metrics, to identify trust-demanding elements to be considered for their deployment on the blockchain. The approach has been validated on a real case study of food quality certification in the biological domain.

Keywords: blockchain, smart contract, collaborative processes, BPM

1 Introduction

Collaborative processes implemented on Web-based distributed systems are widely used to model cooperation between (potentially untrustworthy) organisations, that cooperate in order to increase their value. In recent years, researchers proposed the adoption of blockchain and smart contracts for implementing collaborative business processes going beyond a centralised Business Process Management (BPM) perspective [8, 9]. Blockchain and BPM have been jointly investigated in several real case scenarios, such as supply chain management, logistics, manufacturing and material industry [7]. Existing approaches investigated mainly how smart contracts can be generated from BPMN models,

further proposing cost optimisation strategies [4]. In general, they are platform-dependent, referring to specific blockchain technologies. Recently, the need for proper methods and tools, to guide the process designer for exploiting the blockchain technology, is emerging. To preserve data and business logics ownership and to ensure performance/cost tradeoff, only data and process activities requiring transparency and trust among the distributed process actors should be stored as transactions on the blockchain and deployed as smart contracts. In [1] we defined such elements as trust-demanding objects and trust-demanding activities, respectively. In this paper, we propose a methodology and a tool to support blockchain-based trust management in Web-based collaborative business processes originally designed according to a centralised BPM strategy. The methodology and the tool are grounded on a set of criteria, properly enforced with metrics, to identify trust-demanding elements to be considered for their deployment on the blockchain. This paper further extends our previous research [1] with the introduction of quantitative metrics in the methodological steps and the Web-based tool that supports the process re-engineering.

The paper is organised as follows: Section 2 discusses the cutting-edge features of the approach with respect to related work; Section 3 describes a real world case study for food quality certification in the biological domain; Section 4 presents the methodology; Section 5 describes implementation and validation issues; finally, Section 6 closes the paper and sketches future research directions.

2 Related Work

The exploitation of blockchain technology in BPM lifecycle has been fruitfully investigated in recent work. In particular, Model-Driven Engineering solutions [5,7] have been proven to be effective for modelling blockchain-based collaborative business processes. In [2] models at various levels of abstraction have been conceived to produce smart contracts code for implementing, either totally or partially, the collaborative business process. The Caterpillar approach [8] introduces an abstraction layer over the Ethereum blockchain, recording states of each process instance on the blockchain and deploying the control flow logic as smart contracts. The Lorikeet approach [9] provides an extension of BPMN to represent asset registries as list of information recorded by a trusted authority. The work in [6] focuses on the importance of conceptual modelling to demonstrate how business artifacts leverage the data-centric nature of blockchain. In the latter contributions, proper policies to select which data should be stored on-chain are also considered. Distinction between off-chain and on-chain elements is extensively investigated in [3], where on-chain and off-chain design patterns are described.

Following the lesson learned in [3], we propose a methodology that guides, with the help of proper criteria, the identification of collaborative business process elements to be moved on-chain and elements to be left off-chain. The criteria are based on the notion of trust-demanding objects and activities as defined in [1] and are enforced through quantitative metrics. With respect to [3], the proposed

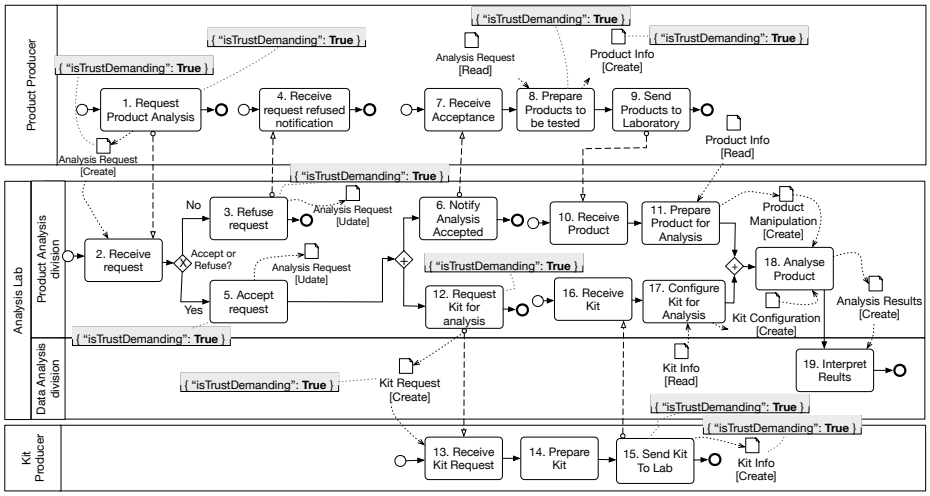

Fig. 1. BPMN diagram of the food quality certification process.

methodology produces a blockchain-based implementation of the business process. The methodology has been implemented in a Web-based tool, which like Caterpillar [8] guides the process designer from business process design to its deployment on-chain. Furthermore, with respect to the other approaches, the methodology supports the generation of Abstract Smart Contracts that are independent of any specific blockchain technology.

3 Motivating Example

Figure 1 reports the BPMN diagram representing the orchestration of Web-based services in a real case study of food quality certification. In the considered process, a product producer issues a request to an analysis laboratory (organised into a Product Analysis and a Data Analysis division) to obtain a quality certification in order to trade the product, in compliance with current regulations. The product to be tested is prepared by product producer and sent to the Product Analysis division, which prepares the product for the analysis. In parallel, the kit producer prepares the kit and sends it to the Product Analysis division, which configures the kit. Once both the product and the kit have been prepared, the Product Analysis division analyses the manipulated product using the prepared kit. Finally, the analysis results are sent to the Data Analysis division to be analysed.

In this scenario, analysis results and certificate could be potentially sources of trust problems between process actors and must not be repudiable. Similarly, the analysis procedure should be transparently shared among all involved participants. Moreover, finding a central authority that ensures trust among participants may be difficult if participants change over time (e.g., if the product producer decides to rely on a different analysis lab). To cope with the former issues, blockchain technology comes to the rescue. However, when deploying the

business process on-chain, there is the need of ensuring performance and cost tradeoff. According to these considerations, the methodology proposed in this paper aims at supporting the identification of candidate data to be stored as transactions on the blockchain and activities to be deployed as smart contracts.

Trust-demanding objects and activities. In the BPMN diagram of Figure 1, annotations are used to highlight elements to be stored as transactions on the blockchain, and deployed as smart contracts. In [1] we defined such elements as *trust-demanding objects* and *trust-demanding activities*, respectively. A trust-demanding object is created/updated/deleted by a participant p_i and read as input of another activity associated with $p_j \neq p_i$, where p_i and p_j belong to different pools (e.g., `Product Info` data object is created by the *Product Producer* and read by the *Product Analysis* division). Conversely, participants corresponding to different lanes within the same pool are conceived as trusted actors, since they represent distinct divisions within the same organisation. Trust-demanding activities create/update/delete trust-demanding objects (e.g., `Product Info` data object is generated by Activity 8) and thus the logic behind such manipulations should be transparently shared among potentially untrusting participants.

4 Methodology

The proposed methodology is conceived as a set of (possibly iterative) steps and starts from an AS-IS collaborative process represented in BPMN and supports the process designer for preparing the implementation of the TO-BE blockchain-based process. We remark that the first three steps are independent of any specific blockchain technology.

1) Candidate on-chain objects and activities identification. The input of this step is the BPMN diagram representing the AS-IS collaborative process. Herein, the identification criteria exposed in Section 3 are used to automatically highlight candidate objects and activities to be deployed on-chain.

2) On-chain objects and activities selection. The process designer manually selects and annotates objects and activities to be deployed on-chain with the support of proper metrics providing a quantitative feedback regarding her selection strategy (i.e., either to foster trustworthiness or costs containment). Indeed, identification does not automatically entail selection. The process designer may decide to keep off-chain candidate elements, based on her knowledge of the process (e.g., two participants that, albeit modelled with different pools in the BPMN, belong to the same consortium).

3) Abstract Smart Contracts generation. Once on-chain objects and activities have been selected, a set of Abstract Smart Contract (ASC) descriptors is generated, to provide high level description, independent of the blockchain technology adopted, of: (a) each activity selected for on-chain migration, because its business logic must be stored on the blockchain as shared code; (b) each selected on-chain object, in which case the ASC includes as functions the CRUD actions performed on the object when stored as transaction on the blockchain.

4) Concrete Smart Contracts deployment. Starting from the set of ASC descriptors generated in the previous step, the developer implements the set of Concrete Smart Contracts (CSCs) on a specific blockchain platform. Developers' skills are clearly distinguished from the ones of process designers, who are in charge of supervising the blockchain-based re-engineering of the collaborative process.

4.1 Metrics to support on-chain elements selection

Deployment metric. The first metric aims at supporting the process designer during the selection of activities (resp., data objects) to be deployed on-chain. On the one hand, the metric can be defined in order to suggest to deploy on-chain as much activities as possible. We refer to this strategy as *trust-oriented*, that may be typical of collaborative business processes that are characterised by a low degree of trustworthiness between process participants. On the other hand, the metric can pursue the improvement of the performance/cost ratio. We refer to this strategy as *performance/cost-oriented*, since it gives more importance to the containment of costs and performance. Focusing on the set of activities A in the AS-IS process, we denote with $A^c_{on} \subseteq A$ the set of candidate on-chain activities, with $A^c_{off} = A \setminus A^c_{on}$ the set of candidate off-chain activities, with $A^s_{on} \subseteq A$ the set of activities that have been selected by the designer for their migration on-chain and with $A^s_{off} = A \setminus A^s_{on}$ the set of not selected activities:

$$m_A = \alpha \cdot \frac{|A^c_{on} \cap A^s_{on}|}{|A^c_{on}|} + \beta \cdot \left(1 - \frac{|A^s_{on} \cap A^c_{off}|}{|A^s_{on}|}\right) \in [0,1] \tag{1}$$

where α and β weights balance the impact of the terms in Equation (1) ($\alpha + \beta = 1$, with $\alpha = \beta = \frac{1}{2}$ both strategies are equally considered). In particular, $|A^c_{on} \cap A^s_{on}|/|A^c_{on}|$ is maximised when $A^c_{on} \equiv A^s_{on}$ (trust-oriented strategy). On the other hand, $|A^s_{on} \cap A^c_{off}|/|A^s_{on}|$ is minimised when candidate off-chain activities are not selected to be deployed on chain (performance/cost-oriented strategy). The deployment metric m_O for data objects is defined in a similar way.

Context switch metric. The deployment metric m_A does not consider the cost of a deployment strategy in terms of context switches between on-chain and off-chain activities. Indeed, switches in the process execution between on-chain and off-chain parts of the process typically affect negatively its execution cost and performances (e.g., to deploy and execute smart contracts functions). To limit the negative effects of context switch, a proper metric is proposed, aimed at quantifying the average number of switching from an activity executed off-chain to an activity executed on-chain, and vice versa. Let $p_{\langle a_i, a_j \rangle}$ be a path in the BPMN process, i.e. a sequence of activities from a_i to a_j (where $a_i, a_j \in A$). The following metric $\mu p_{\langle a_i, a_j \rangle}$ is used to count the average number of context switches along $p_{\langle a_i, a_j \rangle}$:

$$\mu p_{\langle a_i, a_j \rangle} = \frac{\sum_{l=i}^{j-1} (isSel(a_l) - isSel(a_{l+1}))^2}{|p_{\langle a_i, a_j \rangle}| - 1} \in [0,1] \tag{2}$$

where a_l and a_{l+1} represent two consecutive activities along $p_{\langle a_i,a_j\rangle}$ and $isSel(a_l) \in \{0,1\}$ represents a boolean function that checks whether a_l is selected to be on-chain or not. Considering that there may be multiple paths from a_i to a_j, an overall *context switch* metric $\Phi_{\langle a_i,a_j\rangle}$ is proposed:

$$\Phi_{\langle a_i,a_j\rangle} = \frac{\sum_{k=1}^{|P_{\langle a_i,a_j\rangle}|} \mu p^k_{\langle a_i,a_j\rangle}}{|P_{\langle a_i,a_j\rangle}|} \in [0,1] \tag{3}$$

where $P_{\langle a_i,a_j\rangle}$ is the set of all the possible paths from a_i to a_j. If there is a loop from a_i to a_j, only one iteration is considered in Equation (3).

Metrics-driven selection of on-chain/off-chain activities. The knowledge provided by the metrics is leveraged by the process designer to correct/revise her design policy. In fact, Equation (1) does not consider context switch. Thus, to limit the latter, the designer has to move off-chain activities that are currently on-chain (performance/cost-oriented strategy) or vice versa (trust-oriented strategy). In this respect, if the strategy adopted is trust-oriented, the activities lying on the path between two candidate on-chain activities can be in turn selected to be moved on-chain, to reduce the context switch metric value. Otherwise, if the strategy adopted is performance/cost-oriented, the designer may choose to move off-chain several candidate activities initially conceived to be on-chain.

4.2 Abstract Smart Contracts generation

Once the sets of selected on-chain objects and activities have been chosen, a set of Abstract Smart Contract (ASC) descriptors can be generated. ASC descriptors are independent of the adopted blockchain technology and are used by the developer, after selecting a target blockchain technology, to automatically generate the skeleton of Concrete Smart Contracts. According to the definition given in [1], an ASC *asc* is modelled as a tuple containing: (i) the name n_{asc} of the ASC; (ii) the set of state variables VAR_{asc} of the ASC (i.e., primitive data types or objects) on which contract functions operate; (iii) the set of participants P_{asc} registered on the blockchain allowed to interact with the ASC; (iv) the set F_{asc} of the signatures of the functions of the ASC.

Example 1. If the trust-demanding activity 8 in Figure 1 is selected to be deployed on-chain, the following ASC asc^w is generated:
n_{asc^w} = "$PrepareProductsToBeTestedSC$"
$VAR_{asc^w} = \{AnalysisRequest, ProductInfo\}$
$P_{asc^w} = \{ProductProducer, BPMNengine\}$
$F_{asc^w} = \{prepareProducts([AnalysisRequest]) : [AnalysisRequest]\}$

5 Implementation and validation issues

Tool for blockchain-based process re-engineering. The Web-based tool[1] conceived to support process designers has been implemented on top of the archi-

[1] Screencast of the tool is available at: `https://tinyurl.com/wise-screencast`

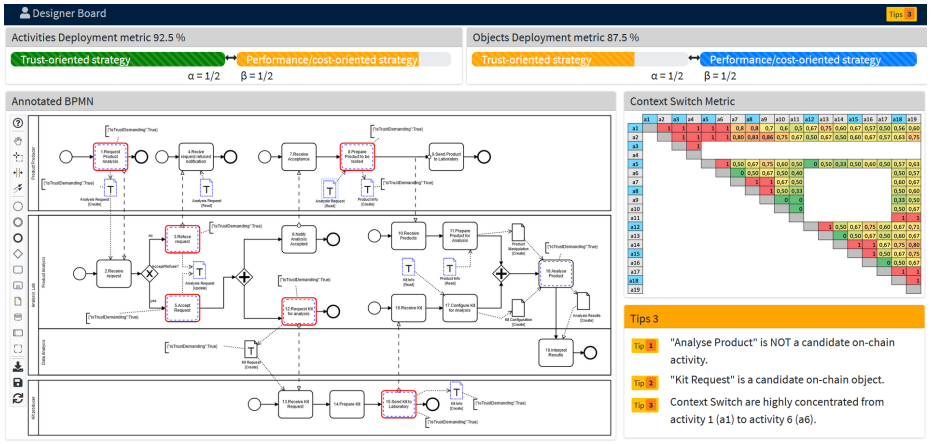

Fig. 2. Dashboard of the tool for blockchain-based business process re-engineering.

tecture presented in [1]. Through the dashboard (Figure 2), the process designer invokes the routine apt to automatically highlight candidate on-chain objects (indicated by letter "T" in the symbol) and activities (marked with a red border). In the process of Figure 1, the tool identifies 6 activities and 4 data objects as candidates to be deployed on-chain. Focusing on activities, the process designer selected for on-chain deployment 7 activities, but one of them has not been identified as on-chain candidate (i.e., *Analyse Product*), leading to a $m_A = 92.5\%$. In this case, the designer's choice affects mainly the performance/cost-oriented strategy (the selected activity is not candidate to be on-chain). To fulfil the performance/cost-oriented strategy, the tool suggests to deselect the "Analyse Product" activity (`Tip 1`). Regarding the context switch metric $\Phi_{\langle a_i, a_j \rangle}$, it is represented as a heatmap wherein red sections represent paths where the context is switched for every activity. For instance, `Tip 3` states that from activity 1 to activity 6, the context has to be changed each step (activities 1 and 5 are selected to be on-chain).

Preliminary cost analysis. We demonstrated that the proposed metrics-based selection of elements to be migrated on-chain enables mitigation of deployment and execution costs. Based on the process in Figure 1, we considered three different configurations on the Ethereum permissionless blockchain: (1) deployment of all the activities and data objects on-chain; (2) deployment of all the activities and of on-chain candidate data objects only; (3) deployment of on-chain candidate activities and candidate data objects. Practically, we generated CSC skeletons in Solidity language starting from the related ASC descriptors, and calculated both smart contracts deployment and functions execution costs. As expected, configuration (1) yields the highest costs ($\approx$ 2.5 times the ones of (3)), which indirectly affect the average confirmation time for transaction on the blockchain (please refer to [1] for further details).

6 Concluding Remarks and Future Research

In this paper, a methodology to support blockchain-based re-engineering of collaborative business processes has been proposed, originally designed according to a centralised BPM strategy. Furthermore, a Web-based tool implementing methodological steps has been developed as well. The methodology and the tool are grounded on a set of criteria, properly enforced with metrics, to identify on-chain elements to be considered for their deployment on the blockchain. Future efforts will consider further usability experiments and the validation of the Web-based tool in specific application domains, namely energy distribution on smart grids and food quality certification. Furthermore, the set of criteria used to distinguish between on-chain and off-chain elements will be enriched, for example based on other experiences like the ones described in [3], in order to make more accurate the performance/costs tradeoff obtained by applying our methodology.

References

1. Bagozi, A., Bianchini, D., De Antonellis, V., Garda, M., Melchiori, M.: A three-layered approach for designing smart contracts in collaborative processes. In: Proc. of 27th Int. Conf. on Cooperative Information Systems (CoopIS 2019). pp. 440–457. Rhodes; Greece (2019)
2. Di Ciccio, C., Cecconi, A., Dumas, M., García-Bañuelos, L., López-Pintado, O., Lu, Q., Mendling, J., Ponomarev, A., Tran, A.B., Weber, I.: Blockchain support for collaborative business processes. Informatik Spektrum **42**(3), 182–190 (2019)
3. Eberhardt, J., Tai, S.: On or off the blockchain? insights on off-chaining computation and data. In: European Conference on Service-Oriented and Cloud Computing. pp. 3–15. Springer (2017)
4. Garcia-Banuelos, L., Ponomarev, A., Dumas, M., Weber, I.: Optimized Execution of Business Processes on Blockchain. In: Proc. of the 15th Int. Conference on Business Process Management (BPM) (2017)
5. Garcia-Garcia, J., Sanchez-Gomez, N., Lizcano, D., Escalona, M., Wojdynski, T.: Using Blockchain to Improve Collaborative Business Process Management: Systematic Literature Review. IEEE Access **8** (2020)
6. Hull, R., Batra, V., Chee, Y., Deutsch, A., Health, F., Vianu, V.: Towards a shared ledger business collaboration language based on data-aware processes. In: Proc. of Int. Conf. on Service Oriented Computing (ICSOC 2016). pp. 18–36. Banff, AB, Canada (2016)
7. J. Mendling, e.a.: Blockchains for Business Process Management - Challenges and Opportunities. ACM Trans. on Management Information Systems **9**(0), 1–16 (2018)
8. López-Pintado, O., García-Bañuelos, L., Dumas, M., Weber, I., Ponomarev, A.: Caterpillar: A business process execution engine on the ethereum blockchain. Software: Practice and Experience **49**(7), pp. 1162–1193 (2019)
9. Tran, A.B., Lu, Q., Weber, I.: Lorikeet: A model-driven engineering tool for blockchain-based business process execution and asset management. In: BPM (Dissertation/Demos/Industry). pp. 56–60. Sydney, Australia (2018)