Smart contracts exploitation to trust in a world of multi-party remote services

Ada Bagozi
Dept. of Information Engineering, University of Brescia
Brescia, Italy
adabagozi@gmail.com

Valeria De Antonellis
Dept. of Information Engineering, University of Brescia
Brescia, Italy
valeria.deantonellis@unibs.it

ABSTRACT

Servitization has become a trending business strategy among modern enterprises, allowing them to have a strong competitive advantage. As servitization leads to an increasing additional value for all the involved parties in a multi-party business processes, trust among such participants becomes a critical issue. This paper presents an approach where blockchain technology and Smart Contracts are exploited to ensure the required level of trust when implementing remote services.

KEYWORDS

Blockchain, smart contract, trust, trustworthy services

1 INTRODUCTION

With the widespread diffusion of digital transformation in modern factories, organizations are becoming more flexible, being able to introduce servitization as a strategic innovation to shift from selling products to integrated product and service offerings [5]. Servitization leads many business processes, that were confined within a single company, to be multi-party, involving multiple actors belonging to different organizations. Hence, affecting data generated in the process, that could be now manipulated by multiple organizations. Therefore, trust among participants in multi-party processes becomes a critical issue. To ensure the required level of trust in remote services blockchain [3] technology and Smart Contracts (SCs) [4] can be integrated in the processes, ensuring data immutability and irreversibility. Blockchain technology is also proposed for secure data provenance management to ensure a higher level of trust and to avoid authorized users to corrupt the data stored in the provenance system [2]. Indeed, every transaction needs to be digitally signed using public key cryptography which ensures the authenticity of the source of data, as well as non-repudiation of information permanently registered in the blockchain.

2 APPROACH OVERVIEW

For the management and execution of remote services by considering the adoption of blockchain and SCs, in our approach each multi-party process is modeled by specifying: (i) the actor(s) responsible for the service execution; (ii) the tasks of the process; (iii) the data used as tasks I/O; (iv) the actor(s) using the service. In particular, trust-demanding tasks are identified and considered to design the SCs that will be deployed on the blockchain. Each SC is modeled as follows. The functions of the contract represent trust-demanding tasks, with corresponding I/O. The execution of its functions may generate information saved as transactions on the blockchain. The actors invoking the tasks are modeled as users who have access to the contract functions. Finally, in order to let the actors exchange information with the blockchain and invoke the SCs functions, remote services are extended by introducing Blockchain-Clients (BCC), that are responsible for initiating a new transaction, whenever a SC function is called, and for sending a notification to the involved actors when a new transaction is added to the blockchain.

3 APPLICATION IN THE SMART FACTORY

The feasibility of the approach has been demonstrated in a distributed process in the Smart Factory context, extending the IDEAaS framework proposed for interactive data exploration to assist in anomaly detection in remote monitoring services [1]. IDEAaS is composed of: (i) a multi-dimensional model for organization of collected data; (ii) data summarization and relevance evaluation techniques, to identify data of interest for exploration; (iii) anomaly detection techniques based on relevant data. In the considered process, an Original Equipment Manufacturer (OEM) supplies remote monitoring services for anomaly detection, based on data collected from the machines of its clients. Events occurring on monitored machines are stored as transactions in a blockchain-based system, to ensure non repudiation of data that is used to activate remote services. Moreover, trust-demanding tasks in the execution logic of services are implemented as SCs, that guarantee the required level of trustworthiness among participants. For example, a task requiring data trustfulness is the identification of warning/error events and associated fees (i.e., OEM may alter monitoring information to apply higher fees). Trust-demanding tasks identified in the process helps designing a SC that contains the data structures and functions to store and retrieve information about detected warning/error events. When IDEAaS identifies an anomalous event (warning/error), interacts with the BCC in order to add the new detected event to the blockchain by invoking the `writeAnomalousEvent` function implemented. On the other hand, the OEM and the client can retrieve information on the anomalous status by invoking, through the BCC, the `getAnomalousEvent` function in order to verify the reliability of the anomaly detection event.

REFERENCES

[1] A. Bagozi, D. Bianchini, V. De Antonellis, M. Garda, and A. Marini. 2019. A Relevance-based approach for Big Data Exploration. *Future Generation Computer Systems* 101 (2019), 51 – 69. https://doi.org/10.1016/j.future.2019.05.056
[2] I. 2017. A blockchain-based approach for data accountability and provenance tracking. In *Proceedings of the 12th International Conference on Availability, Reliability and Security (ARES '17)*. ACM, 14. http://doi.acm.org/10.1145/3098954.3098958
[3] L. S. Sankar, M. Sindhu, and M. Sethumadhavan. 2017. Survey of consensus protocols on blockchain applications. In *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. 1–5.
[4] N. Szabo. 1997. Formalizing and securing relationships on public networks. *First Monday* 2, 9 (1997).
[5] S. Vandermerwe and J. Rada. 1988. Servitization of business: Adding value by adding services. *European management journal* 6, 4 (1988), 314–324.