A Methodology for Extracting and Decoding Smart Contracts Data
(Lightning Talk Abstracts)

Flavio Corradini[1], Alessandro Marcelletti[1], Andrea Morichetta[1], and Barbara Re[1]

University of Camerino, Camerino, Italy,
name.surname@unicam.it

Blockchain is gaining considerable attention to guarantee security and decentralisation in large-scale pervasive systems. Thanks to blockchain's security, it is possible to mitigate Distributed Denial-of-Service attacks [2] and secure communications between devices and users [1]. Its decentralised nature helps eliminate single points of failure associated with traditional single-server or third-party services [5]. Moreover, adopting smart contracts enables automated actions and encodes immutable conditions directly within the blockchain. In such a context, data generated from smart contracts can be used for certified auditing and monitoring activities [3], as well as new data analytics perspectives [8]. Indeed, smart contract analysis permits a better understanding of the system's behaviour by detecting anomalies, fraud, and vulnerabilities. Extracting data from executed smart contracts is crucial in supporting the analysis and the continuous improvement of blockchain applications. Data extraction is the most time- and effort-consuming task in an analysis project, typically requiring more than 80% of the resources [4]. For this reason, in this work we do not consider analysis activities but focus only on the extraction.

Data extraction from blockchain presents several challenges. The execution of a smart contract generates data stored in blocks (e.g., timestamps), transactions (e.g., sender, inputs, gas, and more), events, and storage (i.e., the memory containing the smart contract state) [6]. Additional effort is required to decode information that cannot be easily interpreted in its original form on the blockchain. Thus, the extraction activity must address the heterogeneity of storage and decoding factors. Moreover, catching a contract's state changes permits a comprehensive understanding of the application and enables detailed analysis of the contract's evolution over time. Differently from a transaction or a block, a state change does not generate a clear and accessible track, requiring a deep investigation of the low-level data structure [4, 9]. In Ethereum-based blockchains, each variable influencing the state of a smart contract is permanently stored and encoded in the storage memory based on a specific slot. This slot is statically assigned for simple variables, while for complex types (e.g., mappings and structs), it is dynamically combined with a key generated during the execution. In the last few years, some approaches were proposed to extract data stored in different blockchain sources [7]. However, these approaches mainly extract information related to the execution of smart contract functions (e.g., events, inputs, senders) without considering the evolution of its state.

For these reasons, we propose **a data extraction methodology to extract data from smart contracts, including execution-related data and state changes**. To this aim, our methodology first captures knowledge about the contract transactions and extracts the related state changes for each of them. This is possible by replaying transactions inside the Ethereum Virtual Machine (EVM) and obtaining the traces generated to reconstruct the history of changes in smart contract variables. Usually, this leads to exploiting archive nodes requiring a size of several TB of memory, depending on the client being used, and to define ad-hoc solutions with strong domain knowledge [4, 9]. Our methodology relies on a resource-efficient solution in terms of used technologies (e.g., massive data storage) and information. Our proposal permits the extraction of traces without needing an archival node or other heavy data sources. To demonstrate the feasibility of the proposed solution, the methodology was implemented as a web application that extracts smart contract data according to user inputs. To enhance the efficiency of the methodology, data is stored in a database for faster retrieval in case of subsequent extractions. Furthermore, a dedicated query interface allows for defining complex queries leveraging a traditional Database Management System. The proposed methodology can be generally applied to any EVM-based smart contract, providing high compatibility with other blockchain networks. However, given the unique and different implementations of non-EVM solutions, the methodology currently does not aim to support these scenarios. To show the application in practice, we extracted data from three real-world projects on the Ethereum, Polygon and Fantom blockchains, measuring the related performance.

References

1. Aggarwal, S., Chaudhary, R., Aujla, G.S., Kumar, N., Choo, K.R., Zomaya, A.Y.: Blockchain for smart communities: Applications, challenges and opportunities. Journal of Network and Computer Applications **144**, 13–48 (2019). https://doi.org/10.1016/J.JNCA.2019.06.018
2. Chaganti, R., Bhushan, B., Ravi, V.: A survey on blockchain solutions in ddos attacks mitigation: Techniques, open challenges and future directions. Computer Communications **197**, 96–112 (2023). https://doi.org/10.1016/J.COMCOM.2022.10.026, https://doi.org/10.1016/j.comcom.2022.10.026
3. Ciccio, C.D., Meroni, G., Plebani, P.: Business process monitoring on blockchains: Potentials and challenges. In: Enterprise, Business-Process and Information Systems Modeling. LNBIP, vol. 387, pp. 36–51. Springer (2020)
4. Diba, K., Batoulis, K., Weidlich, M., Weske, M.: Extraction, correlation, and abstraction of event data for process mining. WIREs Data Mining and Knowledge Discovery **10**(3), e1346 (2020). https://doi.org/10.1002/WIDM.1346
5. Kara, M., Merzeh, H.R.J., Aydin, M.A., Balik, H.H.: Voipchain: A decentralized identity authentication in voice over IP using blockchain. Computer Communications **198**, 247–261 (2023). https://doi.org/10.1016/J.COMCOM.2022.11.019
6. Moctar-M'Baba, L., Assy, N., Sellami, M., Gaaloul, W., Nanne, M.F.: Process mining for artifact-centric blockchain applications. Simula-

tion Modelling Practice and Theory | Journal **127**, 102779 (2023). https://doi.org/https://doi.org/10.1016/j.simpat.2023.102779
7. Moctar-M'Baba, L., Sellami, M., Gaaloul, W., Nanne, M.F.: Blockchain logging for process mining: a systematic review. In: International Conference on System Sciences. pp. 1–10. ScholarSpace (2022), http://hdl.handle.net/10125/80091
8. Sanka, A.I., Irfan, M., Huang, I., Cheung, R.C.C.: A survey of breakthrough in blockchain technology: Adoptions, applications, challenges and future research. Computer Communications **169**, 179–201 (2021). https://doi.org/10.1016/J.COMCOM.2020.12.028
9. Weerdt, J.D., Wynn, M.T.: Foundations of process event data. In: Process Mining Handbook, LNBIP, vol. 448, pp. 193–211. Springer (2022). https://doi.org/10.1007/978-3-031-08848-3_6