Towards the Comparison of Blockchain-based Applications Using Enterprise Modeling*

Simon Curty[0000−0002−2868−9001], Felix Härer[0000−0002−2768−2342], and Hans-Georg Fill[0000−0001−5076−5341]

University of Fribourg, Digitalization and Information Systems Research Group
{simon.curty,felix.haerer,hans-georg.fill}@unifr.ch

Abstract. The design of blockchain-based applications requires today in-depth technical knowledge of the underlying technologies and software frameworks. In order to investigate how enterprise modeling approaches can aid in designing such applications and aligning their structure and behavior with business needs, we conduct a comparison of two types of blockchain platforms using the ArchiMate modeling language. Based on a use case for Non-fungible Tokens for digital image licensing, we derive models for a software application using public and permissioned blockchain platforms. This permits us to gain first insights into the adequacy of ArchiMate for representing blockchain-based applications and for highlighting the architectural differences of public and permissioned blockchain approaches from a conceptual modeling perspective.

Keywords: Blockchain · Enterprise Modeling · Non-fungible Tokens.

1 Introduction

Distributed ledger technologies (DLT) such as blockchains offer technological foundations for the digital transformation of traditional businesses as well as novel opportunities due to a secure, tamper-proof, and decentralized storage [4,6]. Thereby, selecting the appropriate blockchain technology, understanding its effects on business operations and the requirements imposed on the IT infrastructure are fundamental for a successful implementation. For this purpose it can be reverted to enterprise modeling frameworks such as ArchiMate [9,11]. However, typical enterprise modeling languages so far targeted traditional IT systems and might thus not be adequate for accurately capturing the properties of DLT [5,7]. For investigating this adequacy more closely, we describe at first the implementation of a blockchain-based application for so-called Non-fungible Tokens (NFT). The application has been implemented separately on two different, popular blockchain platforms, Ethereum and Hyperledger Fabric. Subsequently, we describe how the application can be represented using the ArchiMate modeling language. This permits us to conduct a first evaluation of the adequacy of ArchiMate for modeling blockchain-based applications and allows for a comparison of different blockchain platforms on a conceptual level.

2 Related Work

Prior work shows the use of enterprise modeling approaches such as ArchiMate in different contexts related to blockchains. For example, ArchiMate has been used previously in combination with the Business Model Canvas approach for modeling blockchain-based business models and according enterprise architectures [4]. Babkin and Komleva applied ArchiMate for modeling an insurance contract on the Hyperledger Fabric blockchain [1]. However, in this work only the business aspects were modeled, without the possibility of comparisons between applications or platforms. Another approach was taken by Ellervee et al. who used ArchiMate for conceptualizing a reference model for distributed ledger technologies [2]. The modeling of smart contracts as one particular feature of DLT has been described via UML and BPMN by various authors, for example in [12,10]. Further, comparisons of visualization approaches for blockchain-based applications including visual models were elaborated in [8]. At present, the modeling of concrete blockchain applications with enterprise models is not well represented in the literature. Prior works employ only rather small models with partial views. Therefore, we discuss and demonstrate in the following the modeling of the business, application, and technology aspects for blockchain applications in an integrated enterprise model.

3 Use Case and Prototypical DLT-Application: Non-fungible Tokens for Digital Image Licensing

As a foundation for the modeling of applications on blockchain platforms, we designed a business use case employing Non-fungible Tokens (NFT) for managing copyrights and licenses for digital images. An NFT is a representation of a digital asset, attesting the uniqueness of the asset. Thus, NFTs of digital assets cannot be interchanged. In the Ethereum ecosystem, ERC721 is the currently accepted standard for implementing NFTs [3]. However, the core concepts can be applied to other blockchain platforms as well, such as Hyperleder Fabric.

The use case considers two parties: a photographer and a licensee. The photographer may create an NFT for an original digital image, identifying the holder as the copyright owner. This 'copyright token' allows the holder to emit an arbitrary number of licenses per image. Other parties may buy such a license for an adjustable unit price, thereby obtaining the right to use the asset. A license is represented as a fungible ERC20 token [13], i.e. it is non-unique and the supply is controlled by the copyright owner. Due to this architecture, the transfer of licenses and the establishment of markets on this basis become possible.

Traditionally, a photographer would submit the photo to an agency, managing and selling licenses on their behalf. Introducing a blockchain-based solution yields multiple benefits:

– No need for a central authority: Licensees buy directly from the copyright holder, eliminating costs associated with an intermediary party.

- Traceable licenses: It is apparent who has bought a license token at what point in time and thus who holds a license.
- Guaranteed uniqueness of the digital asset: Copyright tokens in the form of NFTs are unique and thus certify the uniqueness of the digital asset. No two parties can hold an NFT for an asset.

For verifying that an application for this use case is technically feasible using the ERC721 and ERC20 token standards, we implemented two prototypes for Ethereum and Hyperledger Fabric[1].

4 Model-based Comparison Using ArchiMate

From the prototype we derived the necessary application structure and infrastructure components. We then approached the creation of the models and the comparative study using an exploratory research approach. To this end, we found ArchiMate to be a promising candidate among the popular enterprise modeling languages. Thus, the NFT use case was represented using the ArchiMate business layer, while the application and technology layers are based on insights gained from the prototype development. Upon several iterative revisions and discussions between the authors, the model shown in Figure 1 finally emerged. We chose two popular blockchain platforms, the permissionless Ethereum and the permissioned Hyperleder Fabric platform, as represented in the ArchiMate technology layer. This choice was motivated by the significant differences between these platforms as we will briefly outline in the following.

4.1 Ethereum

Ethereum is a popular programmable blockchain platform powering the cryptocurrency Ether. Ethereum's ability to execute smart contracts, i.e. pieces of code, enables versatile business applications and use cases. The public Ethereum network is permissionless. Anyone can participate by running a node contributing computing resources to the peer-to-peer network. Such participation is awarded by block rewards and charging fees on smart contract execution and transactions.

4.2 Hyperledger Fabric

Hyperleder Fabric was developed for business blockchain applications. Ethereum and Hyperledger Fabric share fundamental blockchain concepts, such as executable code in the form of smart contracts - known as *chaincode* in Hyperledger Fabric, decentralized networking, transaction ledgers and a consensus mechanism. Contrary to Ethereum, the network is permissioned, i.e., all participants are authorized and know of each other. Participating roles are assigned roles by an authority, based on a network definition specific to the industry use case.

[1] The prototypes are available via Zenodo: `https://doi.org/10.5281/zenodo.5211569`

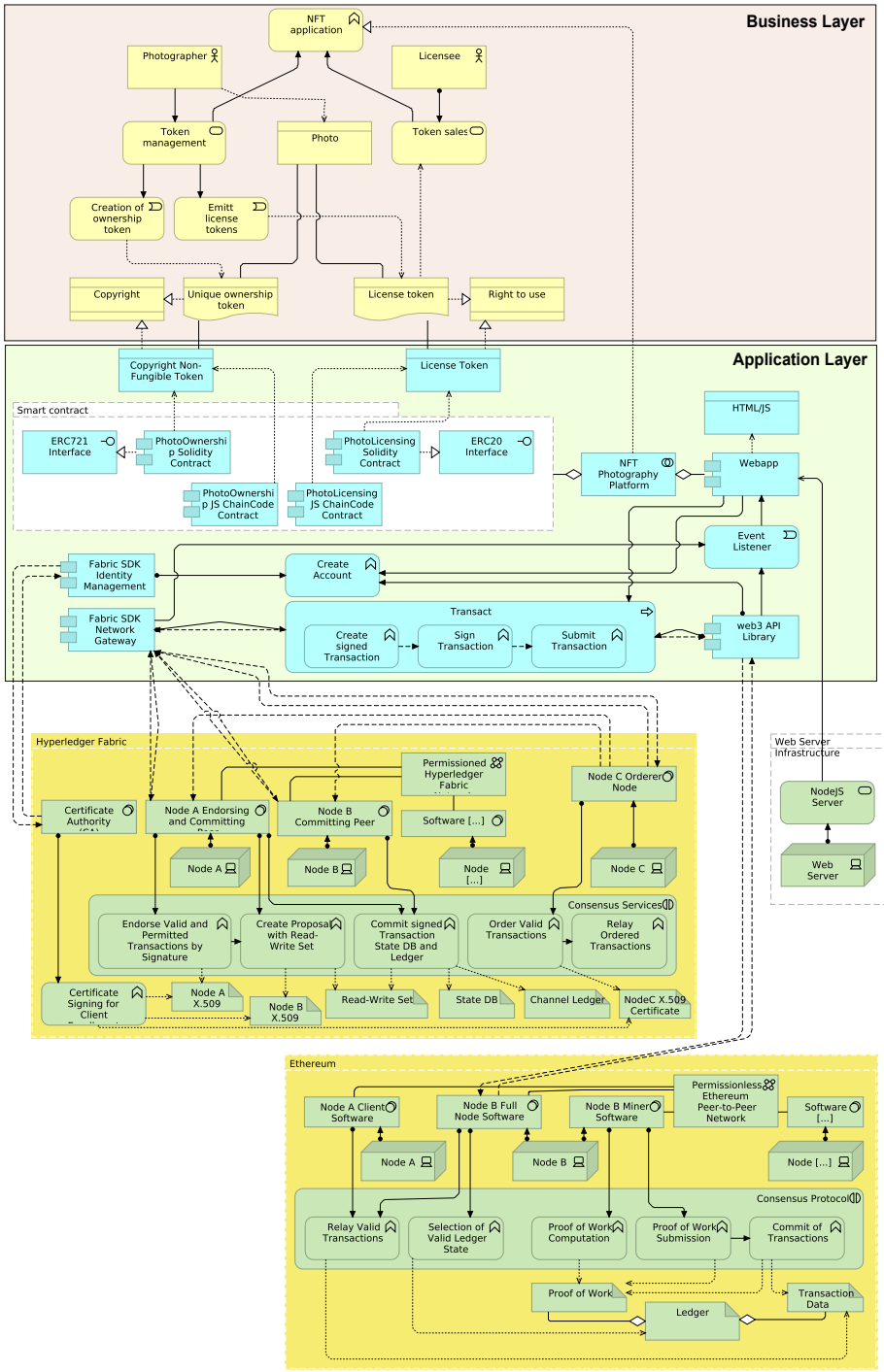

Fig. 1. ArchiMate model showing business, application and technology layer of the NFT application, realized on both Ethereum and Hyperledger Fabric.

5 Discussion

The ArchiMate language provides a multitude of options for modeling a layered enterprise architecture. By aligning business and IT layers, an understanding of the dependencies between hardware, software and business services is established. In principle, a blockchain application can be integrated in an existing IT landscape with ArchiMate. However, we discovered multiple shortcomings:

1. The user is not guided through the modeling process, i.e., a formal modeling procedure as for example used in [14] for creating business plans is missing. Similarly, there is not guidance on how to create business services and functions, application components or technology artifacts. ArchiMate allows the user to make many individual modeling decisions. This generic approach offers great flexibility and can be an advantage for ad-hoc modeling tasks. However, it also introduces ambiguity and thus is not suited for guaranteeing certain system properties, e.g. as required for secure systems.
2. ArchiMate lacks concepts for accurately modeling components in blockchain systems, such as the representation of consensus mechanisms and arbitrary numbers of nodes, which are essential properties of blockchain-based environments.
3. There is no differentiation between model elements for types and instances. For example, it may not be clear if a node element represents a specific node of the network or a type of node. While it is not practical to represent all instances of nodes in the public Ethereum peer-to-peer network, the participating nodes for the particular application need to be modeled in Ethereum as well as in Hyperledger Fabric, where the network is managed.
4. The ArchiMate language lacks concepts specific to DLT and therefore presents ambiguities in the available relationships and elements. For example, flow and trigger relations may or may not imply an ordering of process steps in the way shown for DLT consensus.
5. The structure of an application is not clearly separated from its behavior. For example, application components and their interactions as an order of calling specific functions are shown in one model. It might be desirable to specify the behavior based on the structure of the specified application.

6 Conclusion and Outlook

In this paper we conducted a preliminary evaluation of the application of an enterprise modeling language to the domain of blockchains. For this purpose we created an ArchiMate model of a use case for NFT tokens that has been prototypically realized using Ethereum and Hyperledger. The preliminary analysis showed that the concepts currently contained in ArchiMate do not permit a fully adequate representation of blockchain-based applications. Therefore, it will be investigated in future work, which specific extensions are necessary to enhance the adequacy and offer better modeling support for blockchain-based systems. In particular, we will consider extensions and profiles for ArchiMate for extending the scope of the language as well as the addition of formal modeling procedures.

Acknowledgement

The research on this paper has been partially financed by the Swiss National Science Fund grant number 196889.

References

1. Babkin, E., Komleva, N.: Model-Driven Liaison of Organization Modeling Approaches and Blockchain Platforms. In: Advances in Enterprise Engineering XIII, vol. 374, pp. 167–186. Springer (2020)
2. Ellervee, A., Matulevicius, R., Mayer, N.: A comprehensive reference model for blockchain-based distributed ledger technology. In: ER Forum 2017 and the ER 2017 Demo Track. CEUR, vol. 1979, pp. 306–319 (2017)
3. Entriken, W., Shirley, D., Evans, J., Sachs, N.: EIP-721: ERC-721 Non-Fungible Token Standard. Ethereum Improvement Proposals **no. 721** (January 2018), `https://eips.ethereum.org/EIPS/eip-721`
4. Fill, H.: Enterprise Modeling: From Digital Transformation to Digital Ubiquity. In: 2020 FedCSIS Conference. Annals of Computer Science and Information Systems, vol. 21, pp. 1–4 (2020)
5. Fill, H., Fettke, P., Rinderle-Ma, S.: Catchword: Blockchains and enterprise modeling. Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model. **15**, 16:1–16:8 (2020). https://doi.org/10.18417/emisa.15.16, `https://doi.org/10.18417/emisa.15.16`
6. Fill, H., Härer, F.: Knowledge blockchains: Applying blockchain technologies to enterprise modeling. In: Bui, T. (ed.) 51st Hawaii International Conference on System Sciences. pp. 1–10. ScholarSpace / AIS (2018)
7. Fill, H., Köpke, J.: 1st International Workshop on Conceptual Modeling for Distributed Ledger Technologies (preface). vol. 2542, pp. 40–41. CEUR-WS.org
8. Härer, F., Fill, H.G.: A Comparison of Approaches for Visualizing Blockchains and Smart Contracts. Jusletter IT Weblaw, ISSN 1664-848X **February 2019** (2019). https://doi.org/10.5281/zenodo.2585575
9. Lankhorst, M.M., Proper, H.A., Jonkers, H.: The anatomy of the archimate language. Int. J. Inf. Syst. Model. Des. **1**(1), 1–32 (2010)
10. Lu, Q., Tran, A.B., Weber, I., O'Connor, H., Rimba, P., Xu, X., Staples, M., Zhu, L., Jeffery, R.: Integrated model-driven engineering of blockchain applications for business processes and asset management. Softw. Pract. Exp. **51**(5), 1059–1079 (2021)
11. Pittl, B., Bork, D.: Modeling digital enterprise ecosystems with archimate: A mobility provision case study. In: ICServ Conference 2017. vol. 10371, pp. 178–189. Springer (2017)
12. Rocha, H., Ducasse, S.: Preliminary steps towards modeling blockchain oriented software. In: Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain. pp. 52–57. ACM (May 2018)
13. Vogelsteller, F., Buterin, V.: EIP-20: ERC-20 Token Standard. Ethereum Improvement Proposals **no. 20** (November 2015), `https://eips.ethereum.org/EIPS/eip-20`
14. Wieland, M., Fill, H.: A domain-specific modeling method for supporting the generation of business plans. In: Modellierung 2020. LNI, vol. P-302, pp. 45–60. GI (2020)