Poster: Architecture Reconstruction and Evaluation of Blockchain Open Source Platform

Jungho Kim, Sungwon Kang, Hwi Ahn
Korea Advanced Institute of Science and Technology
373-1, Guseong-dong, Daejeon, Republic of Korea
architect, sungwon.kang, ahnhwi@kaist.ac.kr

Changsup Keum
Electronics and Telecommunications Research Institute
218 Gajeong-ro, Yuseong-gu, Daejeon, Republic of Korea
cskeum@etri.re.kr

Chan-Gun Lee
Chung-Ang University
Huekseok 221, Seoul, Republic of Korea
cglee@cau.ac.kr

ABSTRACT

Recently, public interest in the blockchain technology has surged and various applications based on the technology have emerged. However, there has been little study on architectural evaluations of popular block chain platforms that can help the developers choose an appropriate architecture matching their needs. In this paper, we reconstruct and evaluate the architecture of Hyperledger and Ethereum, which are representative open source platforms for blockchain. The evaluation results indicate that Hyperledger is strong in modifiability and performance whereas Ethereum is better in security.

KEYWORDS

Blockchain, Architecture reconstruction, Architecture evaluation

1 INTRODUCTION

Recently, there has been increasing interest in blockchain platforms in various domains. However, evaluation of blockchain platforms has rarely been conducted although analyzing the pros and cons of blockchain platforms can help developers better choose appropriate architecture matching their needs and maintain the systems based on the platforms. Evaluating a platform typically entails an evaluation of its architecture since the system qualities of the platform are directly influenced by the architecture [1].

In this paper, we evaluate the architectures of the two well-known blockchain open source platforms, Hyperledger (V.1.0) [2] and Go-Ethereum (V.1.7.3) [3]. A big challenge in this endeavor is that architecture document are not available. So we first reconstruct architectures by employing the state-of the art architecture reconstruction techniques. Then we evaluate the architectures from modifiability, security, and performance, which are considered the most important quality attributes by the developers of blockchain platforms. To the best of our knowledge, our study is the first attempt to evaluate the architectures of blockchain platforms with the reconstruction efforts.

2 Architecture Reconstruction

For architecture reconstruction, we focus on module view and C&C view [1]. Module view is particularly related to modifiability, and C&C view is related to security and performance.

To reconstruct module view, we apply the view fusion and analysis method of [4]. To that end, we first perform a reverse engineering at the package (subdirectory) level by using the *go list* tool included in the Go language (V.1.9.2). The external libraries are excluded during this process. We populate hypothesized layers by reflecting package use relationships (import relations) and consolidate the modules between which many use relationships exist into the same layer. By rendering the source and the target modules into upper and lower layers, respectively, we finally produce the module views with the layered style as in Figure 1.

To reconstruct C&C view [5], we apply the execution view reconstruction method of. To that end, first, we collect the *Import* relationships to the external libraries corresponding to the architectural elements such as gRPC, REST API, and inter-process communication (IPC) through static analysis. Then we derive the implementation mechanisms of the gRPC, REST API, and IPC connectors. The processes for the source and target of each connector instance are mapped to components. The components are further refined by utilizing the official project documents and the elements of the module view. Figure 2 shows the C&C view constructed by deriving sub-processes from the execution sequences found in the official documents of the platforms and identifying instances of the elements of the module view.

3 Architecture Evaluation

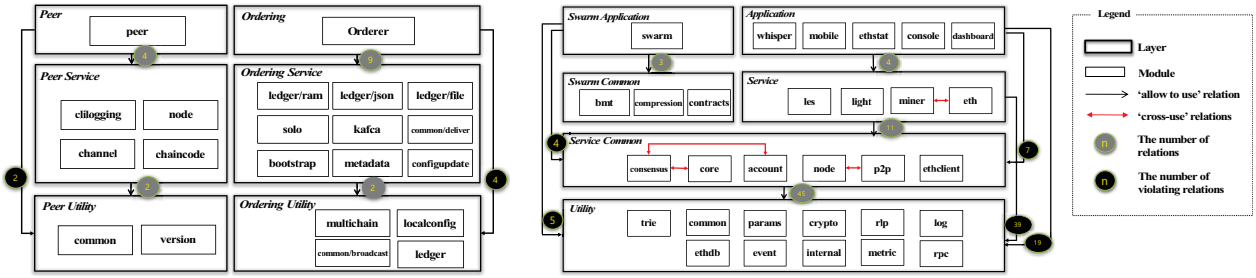
The main quality attribute scenarios for the blockchain platforms are as follows (priorities are not considered in this study):

- Modifiability: New features can be added to the platform within one week.
- Security: The consensus guarantee process should not fail even in the presence of hacking attempts.
- Performance: The consensus latency for one million nodes should be less than 5 seconds.

Hyperledger has a separate component called Membership Service Provider (MSP) to guarantee the identity of the network peers. The ordering service nodes only needs to arrange the sequence of transactions in the block construction process (Step 4 of Fig. 2 (a)). However, each node in Ethereum should perform mining to guarantee the identity through the consensus during the block creation process (Step 3 of Fig. 2 (b)), which may impact performance.

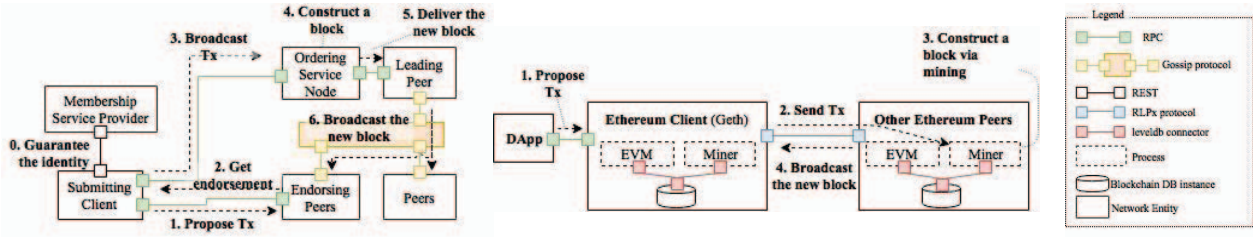
ICSE '18 Companion, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5663-3/18/05.
https://doi.org/10.1145/3183440.3195088



(a) Hyperledger (b) Ethereum

Figure 1. Module view of Hyperledger and Ethereum



(a) Hyperledger (b) Ethereum

Figure 2. C&C view of Hyperledger and Ethereum

Even worse, Ethereum should transmit the generated blocks to all the nodes (Step 4 of Fig. 2 (b)), hence the propagation speed is low. In contrast, Hyperledger transmits the generated blocks to the leader node only in a batch (Step 5 of Fig. 2 (a)) and the leader node propagates them to the remaining nodes (Step 6 of Fig. 2 (a)). Therefore, we can conclude that Hyperledger is superior to Ethereum in performance.

From the perspective of modifiability, we could not find any cyclic-references between the modules in Hyperledger although there were six use relationships violating the layering principle [1]. On the other hand, Ethereum had four cyclic references between the modules and 126 cases of use relationships violating the layering principle. The information entropy [6], which measures the architecture complexity, for Hyperledger and Ethereum were, respectively, 8.796 and 14.251. Overall, the analysis results indicate better modifiability of Hyperledger over Ethereum.

From the perspective of security, Hyperledger has the disadvantage that block synchronization may fail in case the MSP guaranteeing the network peers or the ordering service node guaranteeing the sequence of blocks is compromised. In contrast, Ethereum synchronizes the blocks of all the nodes, hence it provides better security against hacking attempts.

In summary, architecture evaluation shows that Hyperledger is strong in modifiability and performance whereas Ethereum exhibits better security as shown in Table 1.

Table 1: Evaluation of Hyperledger and Ethereum

	Hyperledger	Ethereum
Modifiability	+	-
Security	-	+
Performance	+	-

ICSE '18 Companion, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5663-3/18/05.
https://doi.org/10.1145/3183440.3195088

4 CONCLUSIONS

During the reconstruction process, we encountered various difficulties including determination of the elements for module and C&C views. We found that the view fusion and analysis method [4] and the execution view reconstruction method [5] are very effective for architecture reconstruction in the absence of architecture documents. For example, it was difficult to cluster semantically common modules into the same layer but, with the first method, we were able to infer semantic common modules by referencing the steps of C&C views and hypothesize layers to specify least relationships violating the layering principle. While reconstructing the C&C view, we have decided to derive the connectors and components based on the execution sequences of the main scenarios presented in the official documents and, with the second method, the subcomponents of the C&C view were identified by relating core elements explained in the official documents to the instances of the elements of the module view.

REFERENCES

[1] Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Merson, P., Nord, R., Stafford, J., *Documenting Software Architectures: Views and Beyond, 3rd Edition*, Addison-Wesley Professional, 2011.
[2] Hyperledger/fabric. [Online] Available at: https://github.com/hyperledger/fabric
[3] Ethereum. [Online] Available at: https://github.com/ethereum/go-ethereum
[4] Bass, L., Clements, P., Kazman, R., *Software Architecture in Practice, 3rd Edition*, Addison Wesley Longman, 2012.
[5] Hwi A., Sungwon K., Seonah L., "Reconstruction of Execution Architecture View Using Dependency Relationships and Execution Traces", The 33rd ACM Symposium on Applied Computing (SAC 2018), Pau, France, 2018.
[6] Anan, M., Saiedian, H., Ryoo, J., "An architecture-centric software maintainability assessment using information theory", *Journal of Software Maintenance and Evolution: Research and Practice*, 21(1), 2009.