

# A Feature based Content Analysis of Blockchain Platforms

Suntae Kim  
Dept. of Software Engineering, CAIT  
Chonbuk National University  
Jeonju-Si, Republic of Korea  
[stkim@jbnu.ac.kr](mailto:stkim@jbnu.ac.kr)

Sooyong Park  
Dept. of Computer Science &  
Engineering  
Sogang University  
Seoul, Republic of Korea  
[sypark@jbnu.ac.kr](mailto:sypark@jbnu.ac.kr)

YoungBeom Park  
Dept. of Software  
Dankook University  
Yongin-Si, Republic of Korea  
[ybpark@dankook.ac.kr](mailto:ybpark@dankook.ac.kr)

JeongAh Kim  
Dept. of Computer Education  
Catholic Kwandong University  
Gangneung-Si, Republic of Korea  
[clara@ckd.ac.kr](mailto:clara@ckd.ac.kr)

Young-Hwa Cho, Jae-Young Choi  
Dept. of Computer Engineering  
SungKyunKwan University  
Suwon-si, Republic of Korea  
{choyh2285, jaeychoi}@skku.edu

Chin-Chol Kim  
Dept. of ICT Platform & Services  
National Information Society Agency  
Daegu, Republic of Korea  
[cckim@nia.or.kr](mailto:cckim@nia.or.kr)

**Abstract**— This paper presents a result of feature analysis for building blockchain platforms based on software product line engineering. The result includes a feature model describing major features and constraints among them, and discuss which blockchain platform enables each feature. The result contributes to a developer or an organization who wants to build their own blockchain platform by suggesting commonality and variability of features of the blockchain platform domain, and diverse alternatives to implement the platform.

**Keywords** — *Blockchain platform; A feature model; Software product line*

## I. INTRODUCTION

Recently, blockchain is an emerging technology that protects the data from malicious modification by using a distributed ledger, hash, cryptography and chains of blocks[1]. Thus, it is inherently resistant to modification of the data because an enormous resource is necessary in order to maliciously manipulate the data. Because of the resistance of the modification, diverse areas try to apply this technology in order to resolve their own security issues and enable business models. However, as key technologies of the blockchain are intricately connected each other, it is difficult for developers to have a holistic view of the technologies. This causes a big technical barrier for blockchain starters.

This paper suggests a feature analysis for building blockchain platforms by using a feature model for a developer to comprehend the blockchain platform domain and help them build their own blockchain based services. The feature model is a tool to represent dependencies between major functionalities, so-called *feature*, which is commonly used for analyzing common features and variable features in domain analysis of software product line engineering[2]. In order to build the feature model, we examined diverse documents and internal of the source code of 6 popular blockchain platforms and depicted the result in the feature model.

The rest of the paper is composed as follows: Section 2 summarizes diverse blockchain platforms and notations of a





feature model. Section 3 presents a feature model for blockchain platforms, and Section 4 concludes this paper.

## II. BACKGROUND

### A. Blockchain Platforms

After *Bitcoin*[3] is suggested to support digital currency, diverse variations of blockchain platforms such as *Ethereum*[4], *Tron*[5], *Corda*[6] and have been suggested. Bitcoin initially suggested a blockchain based solution to handles several digital currency issues such as double spending attacks[3]. Ethereum enhanced Bitcoin with adding a smart contract feature written in the Solidity programming language and APIs for building distributed applications[4], which causes diverse Ethereum based applications has appeared. In addition to this, Tron is intended to support entertainment ecosystem with smart contracts and application development platforms written in general programming languages such as Java and C++[5]. The Corda is an open source blockchain platform to build custom platforms[6]. It comes from the R3 CEV projects and supports pluggable consensus algorithm and SQL query. As a popular open source blockchain platform, *Hyperledger Fabric* also supports custom blockchain service. Many developers try to adopt the platform to build private blockchain services[7].

### B. Software Product Line Engineering

Based on our analysis of the diverse blockchain platforms, we found feature modeling[2] suitable for representing the relationships among major components of each blockchain platform. Feature modeling is a design methodology for modeling the commonalities and variations of an application family and their interdependencies. A feature model consists of mandatory features (expressed in ) capturing commonalities, optional features (expressed in ) capturing variations, feature relationships representing logical groupings of features and configuration constraints. There are two types of groups, which are exclusive-or() and inclusive-or()

groups. An exclusive-or group specifies that only one feature can be selected from the group, while an inclusive-or group denotes that one or more features can be selected from the group. Configuration constraints are feature relationships constraining feature selection. There are two types of relationships, requires and mutually exclusive. A requires relationship constrains that selection of one feature in the relationship requires the other feature. A mutually exclusive relationship specifies that the two features in the relationship cannot co-exist.

### III. A FEATURE MODEL OF BLOCKCHAIN PLATFORMS

This section describes the result of feature modeling with a feature model analysis. Figure 1 shows the feature model for building blockchain platforms. The following describes each feature in more detail.

- *Blockchain Platform* indicates a platform generally composing of its sub features such as *Consensus*, *Communication*, *Key*, *Programming*, *Transaction*, *Block Storage*, each of which is the core technology to build a blockchain platform. Except the *Programming* feature, all blockchain platforms contains most of features.
- The *Consensus* feature is one of the core technologies to resolve the Byzantine general problem [8] in a distributed network environment where diverse malicious attackers can exist. In this feature, several consensus algorithms exist such as PoW(Proof of Work)[9], PoS(Proof of Stake) [10] and PBFT (Practical Byzantine Fault Tolerance) [11] and so forth. In the blockchain platform, generally only one consensus algorithm is selected from the beginning because it broadly affects to diverse interactions in the blockchain platform. Thus, open source platform such as *Corda* and *Hyperledger Febric* highlight the pluggable consensus feature as one of the best features.
- The *Communication* feature indicates how to share transactions, blocks or intermediate data between blockchain nodes. Bitcoin and Ethereum adopted the *P2P (Peer2Peer)* communication protocol. However, the protocol causes the communication delay and data disordering problems, which is one of the major causes of the performance issue. Thus, some of the blockchain platforms adapt *Messaging* that supports reliable data transmission. For example, Corda uses Apache ActiveMQ Artemis[12] which is an open source JMS (Java Messaging Service) implementation. Also, Tron and Hyperledger Febric include Apache kafka[13] that is a distributed streaming platform.
- The *Key* feature indicates ways to create keys. Generally, the keys are composed of private key and public key, which can be generated by *RSA*[14] and *ECDSA* [15]. Most of the blockchain platforms adopt ECDSA rather than RSA because ECDSA can achieve safety similar to that of RSA with the smaller number of bits.
- The *Transaction* feature denotes which type of data can be managed as a transaction. Most of the blockchain platform supporting a digital currency that is expressed in the *Digital Currency* feature in the figure. The transaction data generally contains the change of ownership information with proof of ownership mechanism. Also, the diverse transaction data can be included such as commands and attachments expressed in the *Domain Specific Tx* feature in the figure.
- The *Programming* feature indicates programming tool support for the transactions. As the initial blockchain platforms such as Bitcoin does not include this feature, it is expressed in an optional feature. This feature shows from Ethereum named with *Smart Contract* to programmatically manipulates the digital currency transactions without a middleman.

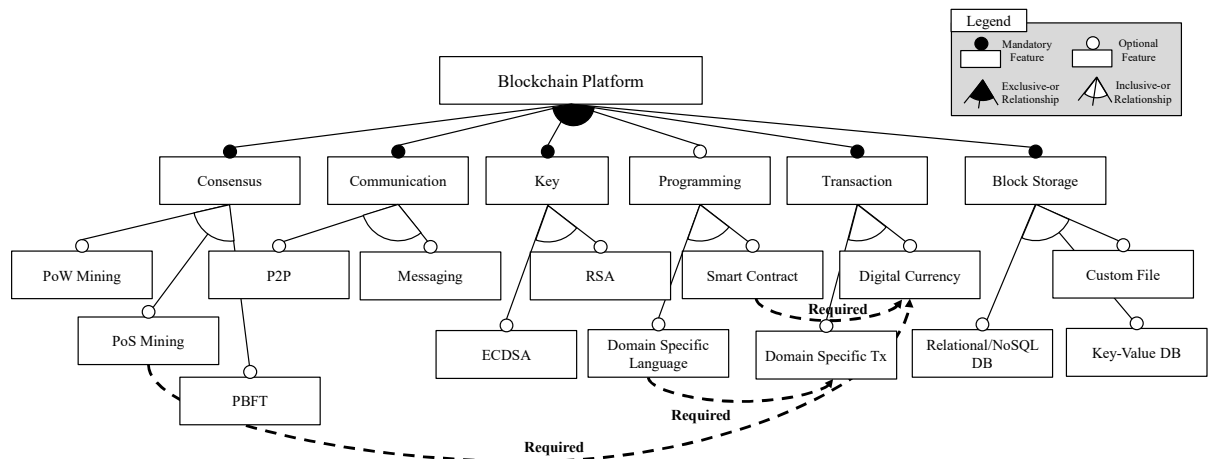


Figure 1. Feature Model for Blockchain Platform

The programming feature depends on the data types of transactions, the programming language should be associated with the data. Thus we expressed it in the *Domain Specific Language* in the figure and denoted the relationship with the *Required* constraint.

- The *Block Storage* feature handles how to maintain the transferred blocks in the storage. As the storage should be fast as possible and manage a massive data, many of the blockchain platform store the blockchain data into the *custom file*. Also, the key-value type database (e.g., Level DB[Level DB] in Tron) for handling the performance issues. In addition, some of the blockchain platform use the *Relational/No-SQL* database to support a query to the blockchain data.

#### IV. CONCLUSION

This paper presents the feature model of blockchain platforms and discuss each feature with exemplifying existing blockchain platforms. Generally, the blockchain platforms focuses on its security integrity, safety and performance. However, the relationships between the features are less focused. This paper intends to conceptually visualize the relationships between features for a blockchain developers by using a feature model of software productline engineering. It should be noted that the feature and feature binding analysis cannot be finalized because the blockchain platforms have been continuously growing fast.

#### ACKNOWLEDGMENT

This research was supported by the Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the

Ministry of Science, ICT and Future Planning (NRF-2014M3C4A7030503).

#### REFERENCES

- [1] A. Antonopoulos, "Mastering Bitcoin: Programming the Open Blockchain", O'Reilly Media 2nd Edition, 2017
- [2] K. Czarnecki, U. Eisenecker Generative Programming: Methods, Tools, and Applications Addison Wesley, 2000
- [3] Bitcoin Homepage available at <https://bitcoin.org/en/>, 2018
- [4] Ethereum Homepage available at <https://www.ethereum.org/>, 2018
- [5] Tron Homepage available at <https://tron.network/en.html>, 2018
- [6] Corda Homepage available at <https://www.r3.com/>, 2018
- [7] Hyperledger Fabric Homepage available at <https://www.hyperledger.org/projects/fabric>, 2018
- [8] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. ACM Trans. Program. Lang. Syst., 4(3):382–401, July 1982
- [9] A. Gervais, G. O. Karame, K. Wust, V. Glykantzis, H. Ritzdorf, and Srdjan, "Capkun. On the security and performance of proof of work blockchains". In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS), pages 3–16, 2016.
- [10] S. King and S. Nadal, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake", available at <https://archive.org/details/PPCoinPaper>, 2012
- [11] M. Castro and B. Liskov. "Practical byzantine fault tolerance and proactive recovery". ACM Trans. Comput. Syst., 20(4):398–461, 2002.
- [12] Apache Artemis ActiveMQ Homepage available at <https://activemq.apache.org/artemis/>, 2018
- [13] Apache Kafka Homepage available at <https://kafka.apache.org/>, 2018
- [14] R. L. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems". Commun. ACM, 21 (2): pp 120–126. 1978
- [15] X9.62 : Public key cryptography for the financial services industry: The elliptic curve digital signature algorithm (ECDSA), ANSI, X9.62-1998, 1999.
- [16] Level DB Homepage available at <https://github.com/google/leveldb>, 2018