Using Triples as the Data Model for Blockchain Systems

Dennis Przytarski

University of Stuttgart, IPVS, 70569 Stuttgart, Germany
Dennis.Przytarski@ipvs.uni-stuttgart.de

Abstract. Current permissioned blockchain systems utilize the key-value data model to store and query the ledger. As the key-value pairs are not sufficiently expressive to represent relationships between data, we present a proposal for the utilization of triples as the data model for blockchain systems. This approach enables a powerful query engine and reduces the number of data stores that have to be maintained.

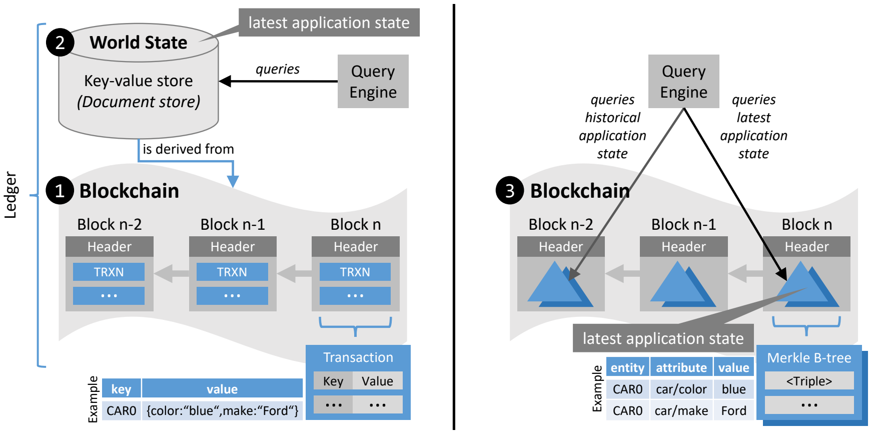
Keywords: blockchain · data model · Merkle B-tree · query · triple

Blockchain systems were initially designed for cryptocurrencies, but applications have started using them as immutable data stores to share business data among many untrusted participants without a central authority.

Current permissioned blockchain systems such as Hyperledger Fabric [1] store the blockchain (❶ in Figure 1) serialized on the file system. The blockchain is a chained list of blocks where each block contains a sequence of transactions. A transaction contains important application data represented as key-value pairs. The latest application state is computed by traversing the entire blockchain from the first to the last block while considering every transaction within a block. Efficient access to the latest application state is achieved through the maintenance of a database called world state (❷ in Figure 1).

This approach has several disadvantages. First, the absence of an explicit history means that if analytical queries on historical blockchain data are needed, the blockchain data (fully or partially) either have to be analyzed manually or exported to an additional separate analytics data store. Secondly, important relationships between the transaction data are embedded in the key-value representation and require their reconstruction at a higher layer (i. e., in the application). Thirdly, there are increased maintenance costs due to the additional data stores. Finally, the tamper-resistance property, which is a fundamental property of blockchain technology, is not ensured for the world state and the analytics data store without the implementation of additional data integrity measures.

Our approach seeks to eliminate these disadvantages by replacing the key-value data model with a generic but flexible data model. We propose to store the blockchain data in a triple data model using `<entity, attribute, value>` triples. This structured data representation enables the generic modeling of relationships, flexible schemas, and an ad-hoc query facility. Complex queries execute



key	value
CAR0	{color:"blue",make:"Ford"}

entity	attribute	value
CAR0	car/color	blue
CAR0	car/make	Ford

Fig. 1. The blockchain storage architecture of Hyperledger Fabric [1] (left side) in comparison with our approach (right side).

directly on the blockchain data, eliminating the maintenance of additional data stores. Our approach addresses the key issues of preserving the integrity of the blockchain's data structure, maintaining an efficient data representation, and supporting a powerful query engine.

Our blockchain (❸ in Figure 1) implementation utilizes Merkle B-trees [2]. A Merkle B-tree contains all triples that reflect the respective application state. Each block is represented by two Merkle B-trees where the first one is sorted by `<entity, attribute>` and the second one by `<attribute, entity>`. The latest application state is always stored in the last block, any historical application state in one of the preceding blocks. A query engine supporting a SPARQL-like query language uses these two Merkle B-trees within a block to efficiently compute the result of a query.

This implementation, however, has high storage requirements and requires several optimization mechanisms. The use of techniques such as data compression, data deduplication, and data encoding as well as the reuse of already stored data structures contribute to reducing storage usage.

Future work will entail the research of further optimization mechanisms, the mechanism and format for exposing the triple data model to smart contracts as well as the development and evaluation of a prototype implementation.

References

1. Androulaki, E., et al.: Hyperledger fabric: A distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference. pp. 30:1–30:15. EuroSys '18, ACM, New York, NY, USA (2018)
2. Li, F., et al.: Dynamic authenticated index structures for outsourced databases. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data. pp. 121–132. ACM (2006)