

Abstract

GraphChain is a framework for on-chain data management for Blockchains is presented. The framework forms the foundational technology for the Ontochain project offering the synergy between ontologies and the Blockchain mechanisms. The use of Ethereum based Layer-2 mechanisms helped create the idea of Ontospace, which designates an ecosystem for trusted ontologies and trusted processing of smart contracts that can directly use the semantic data.

GraphChain 2.0 architecture

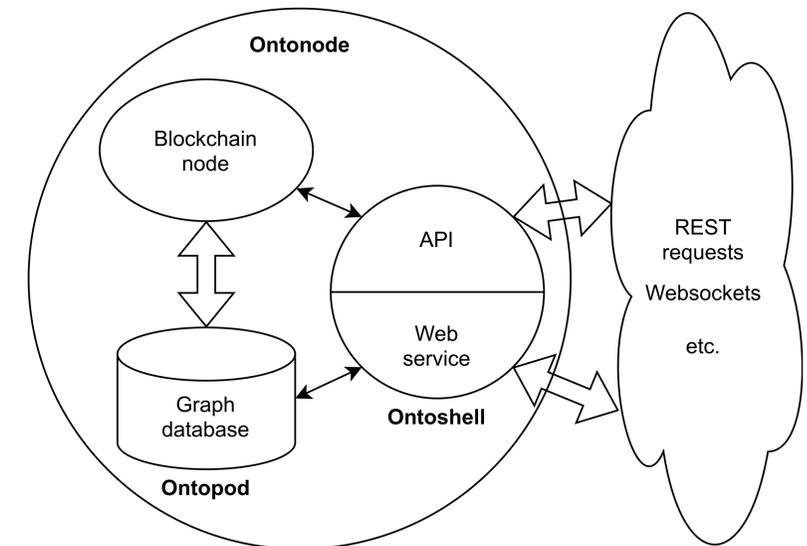
In the big picture, we envision Ontochain as an ecosystem of several different blockchains all linked and pegged to the main chain of the system. We call that ecosystem *ontospace*. In our ecosystem, the core element is an *ontonode*, which is a single *ontoSidechain* node. The general idea of *ontonode* operation is similar to every blockchain network. The blockchain nodes process transactions and achieve consensus over data that represent them. The difference is that in parallel to the creation of new blocks in the Blockchain's chain, the chain of named RDF-star graphs [1] is created according to the GraphChain 1.0 specification [2].

Ontopod

Ontopod is one of the most important sub-elements of Ontonode. It is an RDF-star compliant graph database that stores all named graphs protected and distributed by blockchain network. GraphChain is neutral to the choice of a graph database engine. There is nothing in its architecture that requires a specific triplestore for its operations. The key cryptographic algorithms are, by design, independent from the choice of the triplestore. Our choice as an RDF-star triplestore is the Blazegraph database because of RDF-star support, internal and external full-text search, Linked Data support, and the open-source Java code, which is important from the perspective of the choice of Besu (also written in Java) as an Ethereum client.

Ontoshell

Another important sub-element of ontonode is Ontoshell that is a set of endpoints and interfaces. It is a crucial component because all blockchain interactions, which are not internal, work on this layer. Ontoshell literally functions as a shell hiding both blockchain node and Ontopod from external access. It processes all requests and is responsible for dispatching queries to Ontopod and executing methods in deployed smart contracts. The most standard way of interaction with Ontonode is through REST API. In this particular case, we decided to model our interface on SPARQL 1.1 Graph Store HTTP Protocol [3].



Conclusions

We have presented the idea of a distributed graph data storage for the Ethereum ecosystem. By applying the GraphChain architecture in synergy with Ethereum based Layer-2 architectures, we have designed a 3rd generation Blockchain system focused on storing and processing semantic data. We have demonstrated how such a design forms a foundation for the entire ecosystem of trusted, intelligent data, which we called Ontospace. By allowing to use the latest variants of RDF format (RDF-star), we opened the path to integrate Property Graphs into the ecosystem.

References

- [1] <https://w3c.github.io/rdf-star/cg-spec>
- [2] Sopek, Mirek, et al. "GraphChain: a distributed database with explicit semantics and chained RDF graphs." Companion Proceedings of the The Web Conference 2018. 2018.
- [3] <https://www.w3.org/TR/sparql11-http-rdf-update/>