Blockchain ontology fundamentals and related work

Technical Report

Draft version 1.0 – March 28th, 2023

Web3 Foundation Grants Program

Contributors

Rafael Brandão (rafael@mobr.ai)
Marcio Moreno (mmoreno@mobr.ai)

1. Ontology fundamentals

In the Semantic Web Bus concept originally envisioned by Berners-Lee (Fig. 1), the bus would provide a standard communication protocol and a common data format structured by ontologies. It would be used by agents to exchange information and services. It represented a vision for a more intelligent and collaborative web, where agents would work together to solve complex problems and share knowledge and resources. While the original vision has yet to be fully realized, many of the underlying technologies and standards, such as RDF, OWL, and SPARQL, have become crucial components of the Semantic Web.

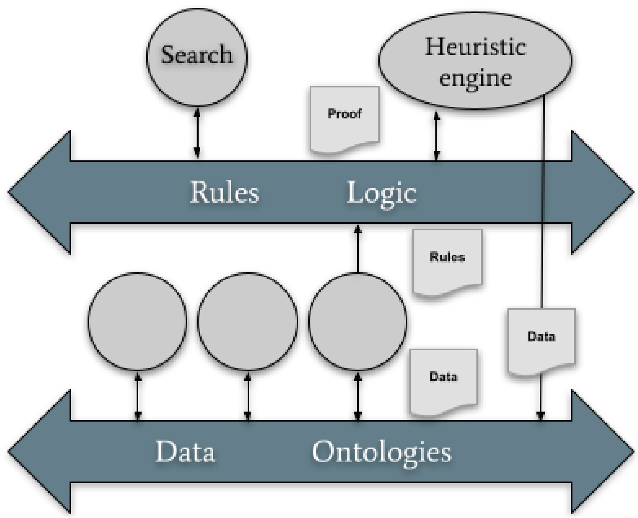

Fig. 1 – The "Semantic Web Bus", as originally envisioned by Berners-Lee[1].

Ontologies have long provided a formal way of organizing information in the Web 2.0 environment. They are essential communication tools, supporting exchange of data among systems, providing querying and reasoning services, publishing reusable knowledge bases, and offering services to facilitate integration and interoperability across multiple, heterogeneous systems and databases (Gruber, 2008).

In practice, they are used to represent and define domain categories (TBoxes) and instances (ABoxes). Along with properties, relationships between concepts, data and entities. Ultimately, ontologies enhance communicability among users.

[1] https://www.w3.org/2000/Talks/1206-xml2k-tbl/slide14-0.html

2. Web3

In Web 3.0, the internet is envisioned as a distributed network of applications and services that operate on top of a decentralized infrastructure (as illustrated in Fig. 2), allowing for a more open, secure, and inclusive online experience. The focus is on decentralization, transparency, and user control.

Distributed ledgers (DLTs) play a key role, allowing for secure and transparent transactions. A common example is blockchain, a distributed ledger with growing lists of records (blocks) that are securely linked together via cryptographic hashes. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a Merkle tree, where data nodes are represented by leaves).

Open-source decentralized applications (DApps) are designed and built to operate on top of distributed infrastructures, accessing DLTs, enabling users to interact in a secure and transparent environment. Smart contracts, distributedly stored code, are automatically executed when conditions are met. This way, guaranteeing DApps to operate in a trustless and decentralized manner without the need for intermediaries.

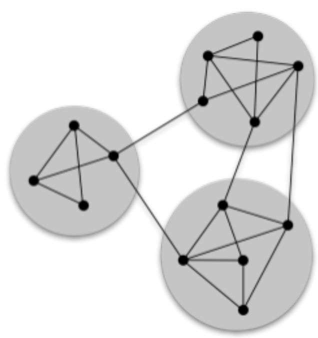

Fig. 2 – Web3's decentralized architecture.

a. Ontologies in Web3

There are various possibilities ontologies can benefit Web3 ecosystems. In essence, by adopting and adhering to W3C standards, web3 developers and service providers can create a more trustworthy and reliable environment for data exchange and interaction, helping to build user trust and confidence in ecosystems.

Ontologies provide a standardized schema that can be used to describe content stored on ledgers. This makes it easier to identify and link ledger content to other datasets or content using common concepts and abstractions. For instance, an ontology could define a set of standard terms for describing financial transactions, which could then be used to link ledger content to external financial datasets. Ledger content can be described with RDF in multiple formats (RDF/XML, JSON-LD, TURTLE).

Mapping between different data models in different Web3 networks can also be supported by the use of ontologies, enabling interoperability and creating a more comprehensive view of decentralized data.

Another interesting space to explore is the application of virtual RDF graphs (vRDF graphs). Virtual graphs are RDF graphs local to a Semantic Web application that contains triples from potentially differing, non-local sources. By creating such graphs on-the-fly, it would be possible to develop interesting querying services among knowledge graphs from different networks and data sources.

Blockchain data and meta-data validation can be supported once a chain relies on ontologies and RDF as well. They may benefit from "shapes" used for data validation and reporting errors in the model and the content specification. Ledger consistency validation and reasoning with 3rd-party service could also be provided.

Finally, a promising perspective from the ontologies and web3 integration is to allow searching the ledger data using the SPARQL query language. This would require a third-party service that reads the blockchain and executes the queries, or if the chain itself provided such search service as a native feature.

b. Ontologies and blockchains

Cano-Benito et al. (2019) lists multiple scenarios for integrating ontologies and blockchains. Figure 3 illustrates the first scenario of a chain of blocks with meta-data expressed following an ontology, and the content of the blocks is expressed in a non-RDF format. EthOn (Pfeffer, 2016) is an example of ontology designed with such a purpose.

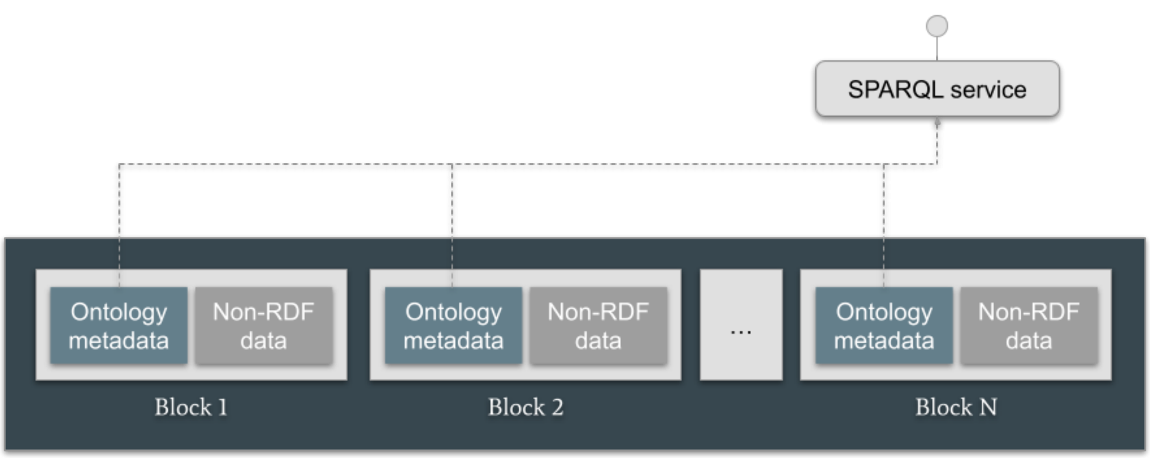

Fig. 3 – Blockchain with metadata referencing ontology.

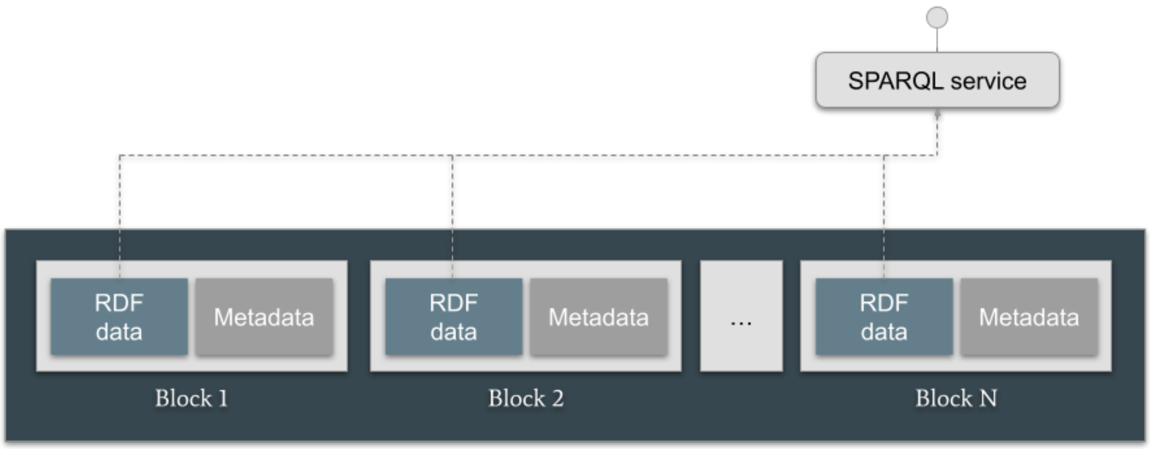

Fig. 4 – Blockchain with RDF content.

Figure 4 illustrates a variation of the first scenario where content of the blocks may be expressed in any format that RDF supports, e.g. Turtle, JSON-LD or XML/RDF.

Virtual RDF services such the one illustrated in Fig. 5 can be used for linking data and combining several data on-the-fly. They consume a data source (e.g, the blockchain or other data structure) and output RDF. Services can publish the data as a dataset and provide a SPARQL query endpoint, or only generate an RDF dump to be stored in a triple store in order to query the data.

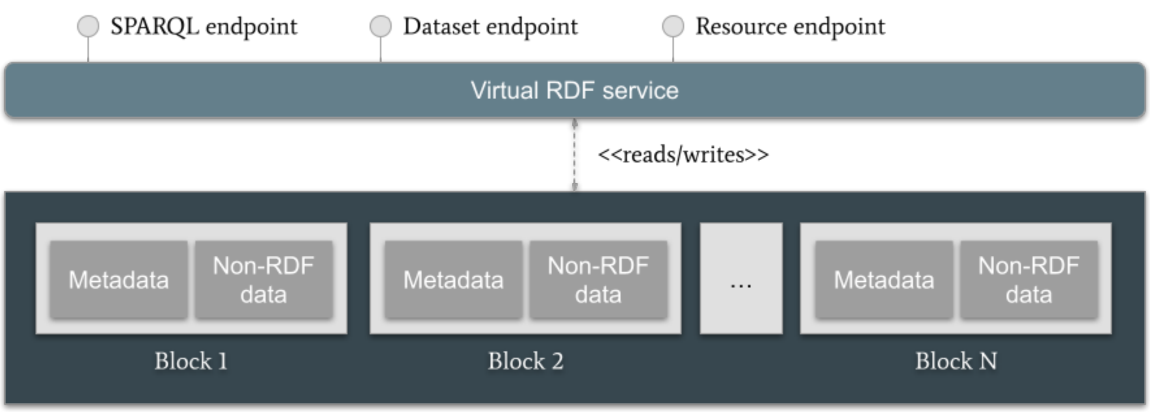

Fig. 5 – Blockchain and Virtual RDF.

According to authors, the possible implementation depicted in Fig. 6 would benefit the semantic web stack, since a blockchain could be used to maintain immutable and trustworthy references to external RDF datasets. In a way, in this scenario the blockchain could act as a DNS to identify RDF fragments, even in case DNS servers change and no longer identify RDF elements URI.

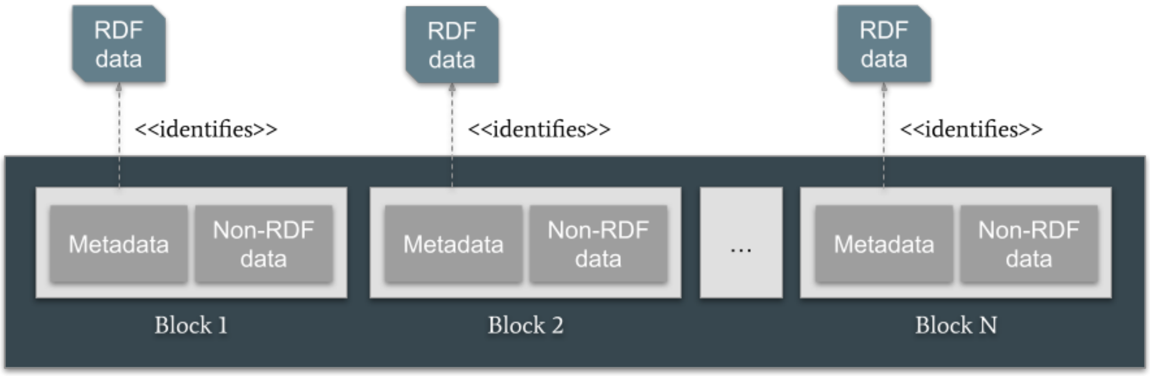

Fig. 6 – Blockchain with external pointers.

Figure 7 shows a variation with cross-chain references. A first chain is used to identify RDF resources stored in a second chain implemented through any of the previous approaches. RDF data would be immutable, transparent, and double identified (by the URIs and the hashes of the first blockchain).

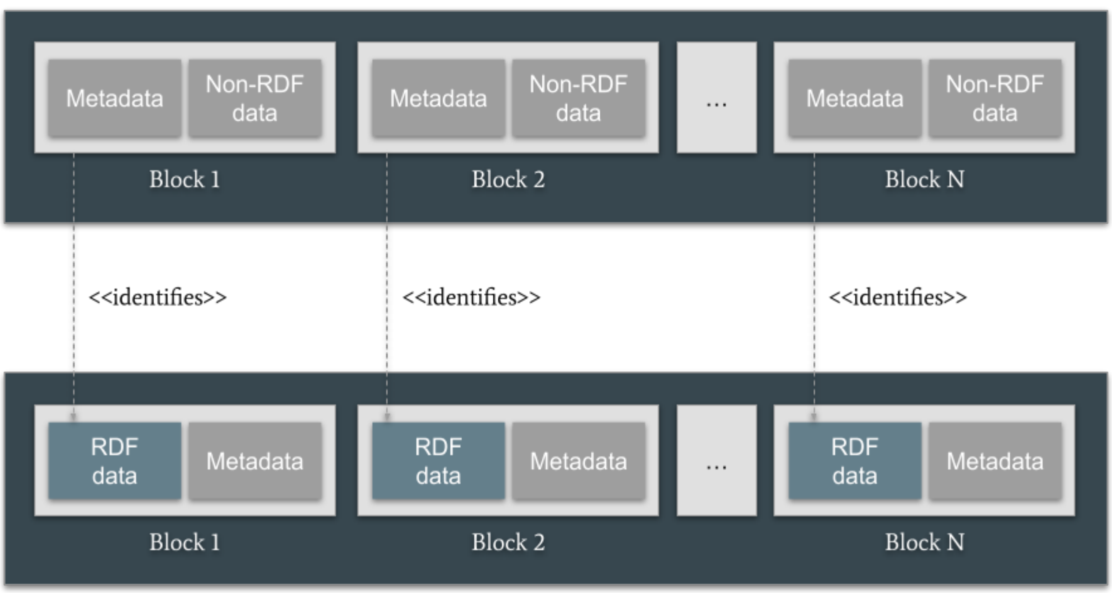

Fig. 7 – Blockchain referencing another blockchain.

The scenario illustrated in Fig. 8 consists of a blockchain implementation which would take advantage of all the semantic web technologies from scratch. That is, a knowledge-oriented blockchain implementation. No such implementation is currently available.

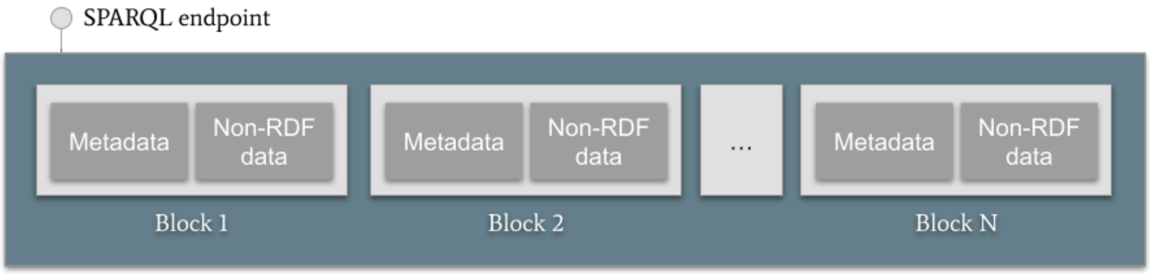

Fig. 8 – Blockchain implementation relaying on Semantic Web.

3. Related work

A number of ontologies have been conceived and applied in the context of Web3. However, to the best of our knowledge no ontology has been specifically proposed to model the Polkadot ecosystem. In the following subsections, we briefly discuss three different ontologies observed during the literature review we conducted previously.

a. EthOn

The Ethon Ontology is a formal representation of the Ethereum ecosystem that aims to provide a standardized vocabulary and ontology for Ethereum-related concepts and entities. It is modeled using Web Ontology Language (OWL). It is closely aligned with the Ethereum yellow paper[2]. Use cases include allowing semantically annotating content provided by Ethereum based tools and DApps.

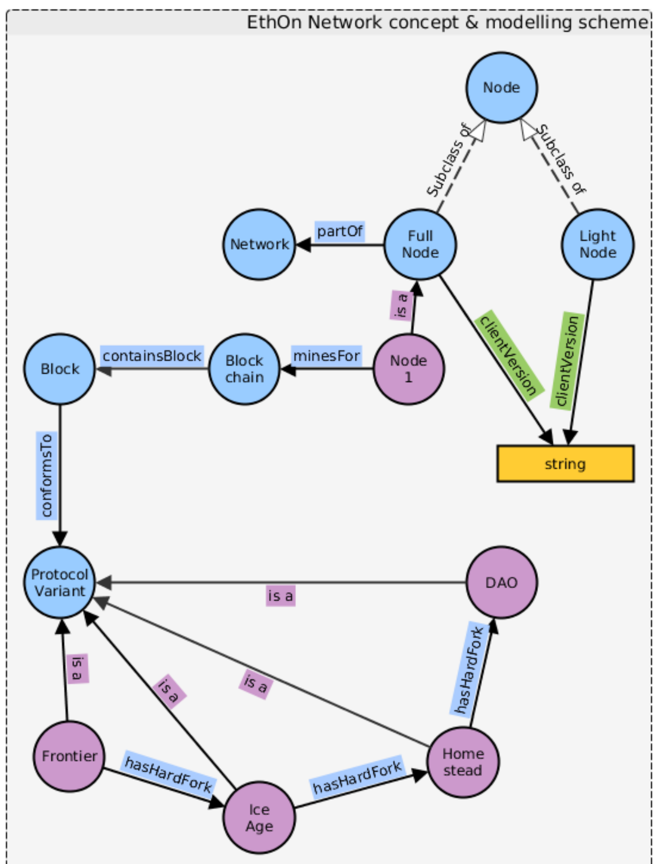

Fig. 9 – EthOn's network concept and modeling scheme.

[2] https://ethereum.github.io/yellowpaper/paper.pdf

EthOn is organized into several modules, each of which corresponds to a specific aspect of the Ethereum ecosystem. These modules include: Core (fundamental classes and properties), Blockchain (blocks, transactions, and mining, etc.), Contracts (smart contracts, their state variables, and functions), Tokens (including ERC20, ERC721, and ERC1155 tokens), and DeFi (decentralized exchanges, liquidity pools, lending protocols, and others).

b. BLONDIE

BLONDIE (BLockchain ONtology with Dynamic Extensibility) is an ontology expressed in OWL language aiming to support interoperability among different blockchain systems, in its current version: Bitcoin, Ethereum and IBM Hyperledger. It provides a common vocabulary for describing chain-specific concepts and relationships. This can be useful for developing applications that need to interact with multiple blockchains or for comparing and contrasting different blockchain systems.

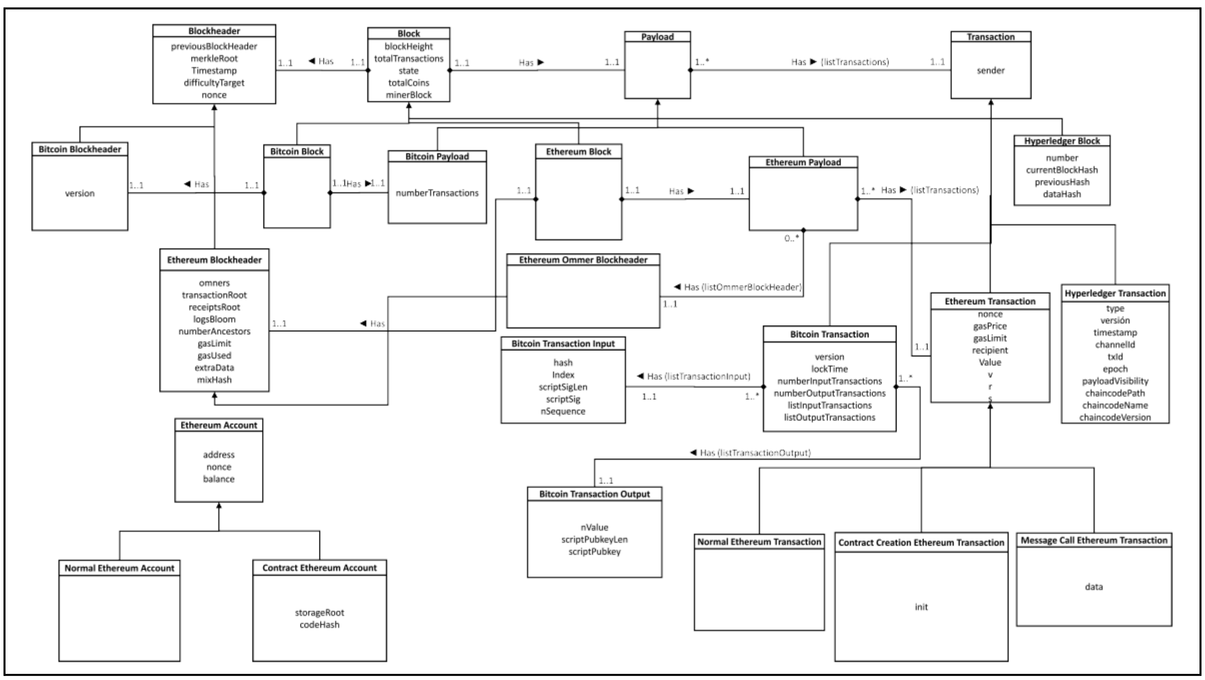

Fig. 10 – Entity-Relationship diagram of BLONDiE.

A potential limitation of the ontology is that it may not capture all the nuances and details of other blockchain systems, as it does not aim to provide a general and abstract model of blockchain technologies. Nonetheless, BLONDIE ontology represents an important contribution to the development of a standardized vocabulary for blockchain technology and has the potential to facilitate the development of interoperable blockchain applications.

c. DLT Ontology

The DLT Ontology[3] defines a set of concepts and relationships that capture the essential characteristics of DLT systems, including their architecture, consensus mechanisms, transaction models, and data structures. The ontology is designed to be modular and extensible, allowing developers and researchers to tailor it to their specific needs. It includes concepts to model security aspects such as technical threats and vulnerabilities of distributed ledger systems, application domains, as well as relevant standards and regulations.

The ontology was developed based on collection of information and competency questions. It covers technical setup and components of DLT systems, security aspects, as well as applications and use cases. It covers 115 classes and 15 properties, and consists of a total of 571 triples. Figure 10 shows the core concepts of the ontology.

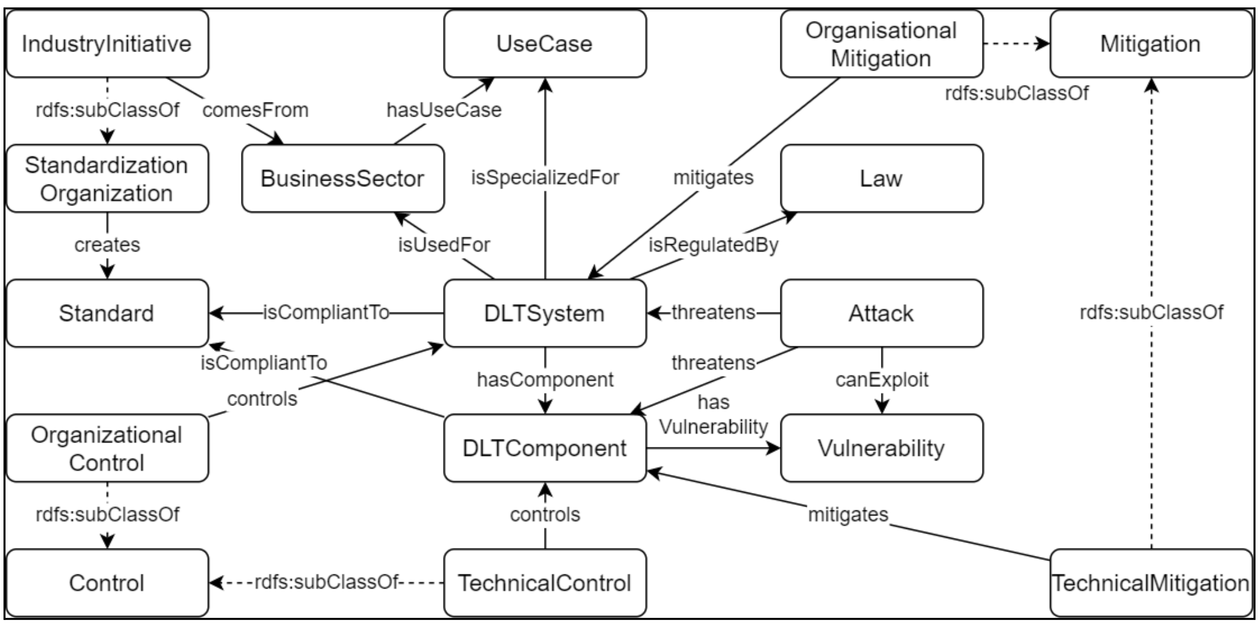

Fig. 11 – Core concepts of the DLT Ontology.

[3] https://dlt-ontology.github.io/

4. References

Cano-Benito, J., Cimmino, A., & García-Castro, R. (2019). Towards blockchain and semantic web. In Business Information Systems Workshops: BIS 2019 International Workshops, Seville, Spain, June 26–28, 2019, Revised Papers 22 (pp. 220-231). Springer International Publishing.

Pfeffer, J., Beregszazi, A., & Li, S. (2016). EthOn-an Ethereum ontology.

Gruber, T. (2008). Ontology. Encyclopedia of database systems.