# A general domain model of distributed ledger technologies

Simon Curty[1,*], Hans-Georg Fill[1]

[1]Digitalization and Information Systems Group, University of Fribourg, Bd de Pérolles 90, 1700 Fribourg, Switzerland

**Abstract**

Distributed ledger technologies (DLT) comprise a versatile set of concepts whose combination is potentially highly disruptive from a business and technical perspective. However, the underlying technological variety and complexity hinder a more widespread adoption. In particular, it is a major challenge to understand the interplay of business and technical aspects in DLT-based systems. In the paper at hand, we address this challenge by proposing a generally applicable domain model for distributed ledger technologies. Whereas previous approaches predominantly focused on specific aspects of DLT or technological variations, we put a focus on the relations between governance, economic, and technical perspectives in a holistic manner. For illustrating the value of the domain model, we apply it for describing a conceptual representation of the widely used Ethereum blockchain environment.

**Keywords**

Distributed ledger technology, Blockchain, Domain model, Conceptual modeling

## 1. Introduction

The family of distributed ledger technologies (DLT) — often equated with *blockchain* — has seen a growing number of members, offering rich feature sets for various use cases. Thereby, the core properties of DLT, such as tamper-proof distributed storage or authorization mechanisms [1, 2], are often approached in a conceptually similar manner. For example, blockchains share the fundamental data structure of the distributed ledger which enables tamper-proof records. As no single solution can meet the requirements to fit every use case, numerous new concepts, technical approaches, and operation practices have emerged over the years. This conceptual variety, the technological complexity, and the disrupting nature of DLT from a business perspective pose difficulties for the successful adoption of DLT. For facilitating the understanding of these complex systems, we thus propose in the following a general domain model for conceptualizing systems built on distributed ledger technologies. In line with the definition by Guizzardi and Proper, we regard a *domain model* as a social artifact for representing an abstraction of a domain for a particular purpose [3]. Our aim is to increase the shared understanding of the domain of DLT with a particular focus on the interplay between their governance, economic mechanisms, and technical realization.

*ER2023: Companion Proceedings of the 42nd International Conference on Conceptual Modeling: ER Forum, 7th SCME, Project Exhibitions, Posters and Demos, and Doctoral Consortium, November 06-09, 2023, Lisbon, Portugal*
*Corresponding author.
simon.curty@unifr.ch (S. Curty); hans-georg.fill@unifr.ch (H. Fill)
0000-0002-2868-9001 (S. Curty); 0000-0001-5076-5341 (H. Fill)

Previously, a number of conceptualizations of this field have been proposed, e.g., in form of ontologies. Thereby, past approaches focused primarily on selected aspects, such as governance [4], or on specific technologies, e.g., the technologies underlying the Ethereum blockchain [5]. However, for understanding the relations between isolated aspects, such as the impact of a consensus mechanism on a business case, more over-arching, holistic models are required. The challenge in designing such a model lies in capturing the significant concepts on an appropriate abstraction level, such that also relations between seemingly separate perspectives can be expressed. Thus, the main research objective of this work is to provide a *technology-agnostic* conceptualization of distributed ledger technologies, including the relations between *technical concerns*, *governance models* and *economic structures*.

The remainder of this paper is structured as follows. In Section 2 we will introduce foundations on DLT and blockchain-based systems, as well as existing approaches for the conceptualization of DLT. In Section 3 we will present the general domain model of DLT, split into models for representing governance, economic, and technical structures. An application of the domain model will then be illustrated with a conceptualization of the Ethereum ecosystem in Section 4. The paper will conclude with a discussion of the approach in Section 5 and an outlook to further research in Section 6.

## 2. Foundations and related work

In the following we present brief foundations on distributed ledger technologies and the core concepts that are relevant for our approach. Furthermore, we will outline prior work on conceptualizations of the DLT domain.

### 2.1. Distributed ledger technologies

Distributed ledger technologies are characterized by a collection of various technical and socio-technical concepts, which work together to enable properties such as the *decentralized immutable storage of transaction records* in the form of a distributed ledger shared between authorized parties [6]. Thereby, a *consensus mechanism* defines a protocol to ensure the consistency and validity of the ledger. This protocol is enforced by *nodes* that form a peer-to-peer network. Participation in a network may be restricted such that only permissioned parties can access the ledger. Most predominately, the data structure of the ledger is a chain of cryptographically linked data blocks, forming a so-called *blockchain*. Further, so-called *smart contracts* may be added to transactions in some DLT systems for the decentralized execution of algorithms [1]. Decentralized applications (also known as DApps) then rely on distributed algorithms to achieve parts of their function [6].

The computational resources necessary for network operations are provided by the participating nodes. The required infrastructure and operating costs are incurred either by the involved parties themselves or by a crypto-economic approach. In the former case, the parties have some interest in providing resources, e.g., because they are part of a business consortium along with collaborators. In the latter, the node operators are typically rewarded for their efforts in securing the DLT system with payments in the form of cryptocurrencies.

The evolution of a DLT-based system requires decision making on emerging technical developments, adjustments to the consensus mechanism, and changes to the economic structure. The processes and parties involved in making these decisions are subsumed under the term *governance*. This may for example involve community-based voting, or simply a centralized organization with exclusive decision power.

## 2.2. Previous conceptual representations of DLT

A number of conceptual modeling methods for various aspects of DLT have been elaborated in the past. Their goals span, e.g., from providing support for the development of DLT-based applications, the design of business models, over the integration of DLT systems in enterprises, strategic decision making, and to semantic reasoning and technology taxonomies. In the following, we briefly summarize selected works on taxonomies, reference models, and ontological approaches that we deem most closely related to our approach. For a comprehensive overview on modeling methods in the context of DLT, we refer readers to a recent literature survey [7].

For dealing with the technological complexity and variety, guidelines, and decision models for choosing appropriate DLT systems have been published. Such models may be derived from taxonomies and subsequent classifications of systems. This includes for example architectural characteristics of DLT systems [8], consensus protocols [9], and cryptoeconomic design [10].

Other approaches revert to schematic reference models or ontologies to present aspects of DLT. Thereby, some regard specific systems: EthOn [11] is a detailed ontology formalizing the technical concepts of the Ethereum blockchain. Olivé [5] presented another conceptualization of Ethereum, focusing on system architecture and the data model of the ledger. Of further interest is the ERC-721 standard for the implementation of non-fungible tokens on Ethereum [12], where Ethereum and token contracts are modeled with the OASIS ontology [13] which has been extended with smart contract concepts [14]. In addition, ontological approaches may apply inference mechanisms to derive new insights on modeled systems. This has been leveraged, for example, by Besançon et al. [15] who presented an OWL ontology based on the aforementioned EthON approach for the formalization and development of decentralized applications. In a similar vein, other authors focus on smart contracts and reverted to ontologies in combination with semantic rules for the verification of contracts [16] or the generation thereof [17].

Multiple blockchain-based systems were considered by Ellervee, Matulevicius, and Mayer [18] to derive a reference model using ArchiMate, BPMN and UML of the general architecture and technical processes of blockchains. ArchiMate was also used by another approach to draw a comparison of the architectural differences of Ethereum and Hyperledger Fabric [19]. Non-technical perspectives on DLT may profit especially from considering multiple systems or the domain as a whole. Such system-agnostic conceptualizations were presented, for example, by detailing smart contracts as formalization of legal contracts [20, 21], or for designing decentralized governance [4].

Ontologies stemming from the field of enterprise modeling were also considered for supporting the development of DLT-based systems. For example, Kim et al. [22] extended the Toronto virtual enterprise ontology [23] with traceability concepts for blockchain-based supply chain provenance. The business ontology of the resource-event-agent accounting framework [24] was used in combination with DEMO [25] for the ontological modeling of blockchain transactions and how these relate to economic commitments [26]. Another approach applies the resource-event-agent ontology as part of a development method for smart contracts for multiple blockchains [27].

In summary, existing conceptualizations predominantly regard specific aspects of the DLT domain, e.g., smart contracts on various level of abstraction, or investigate specific systems, most commonly involving Ethereum [7]. Such a focus allows to derive conceptualizations that capture the universe of discourse in detail. Considering isolated concerns in a larger context is still a necessity to understand how these relate to each other. This could be supported by general domain models. Such models for the DLT domain have been sparsely researched so far.

# 3. Domain model of distributed ledger technologies

For the development of the domain model, we pursued an exploratory research approach by searching for generalizations leading to a detailed understanding of the object under study [28]. Thereby, we aim to lay a foundation for subsequent, confirmatory research approaches. Thus, at first, existing conceptualizations, as outlined in Section 2.2 were considered, as well as our own experience of the domain from various projects and as derived from discussions with industry experts. The development was guided by three design objectives: First, the domain model should be *technology-agnostic*, that is, it should represent distributed ledger technologies in general, rather than a specific technical platform. Second, it should capture the domain from a *holistic perspective*, rather than focus on specific concerns such as only technical or only economics aspects. Third, to account for future developments, the model should be easily *extensible*. For ensuring this latter objective, we revert to UML class diagrams that have been extended with reference points to link multiple diagrams together to increase comprehensibility.

The model is split into three parts that each represents a perspective of the DLT domain, focusing on governance, economic, and technical concerns. For clarity, some concepts and relations have been purposely omitted from individual figures. Concepts that appear in more than one perspective are color-coded and tagged with acronyms for each perspective (*gov,eco,tech*).

## 3.1. Governance perspective

DLT systems require a governance concept that describes the decision structure for managing the initial development as well as future evolution of the system. While this is also true for traditional IT systems, the decision structure might be decentralized, using the distributed ledger itself and community-driven. Technology projects, such as Ethereum, individual networks, as well as decentralized applications always define a governance concept, may it be implicitly or explicitly. In Figure 1, a general model of a concept related to governance in DLT is shown, which will be summarized in the following.

**Types of governance structures:** We distinguish between three types of specializations of governance. In the case of *decentralized* governance, the decision makers are participants in the DLT system. This includes, for example, community-driven governance. When reverting to *consortial* governance, the decisions are made by organizations taking part in the system. And when using *centralized* governance, a single party makes the decisions.

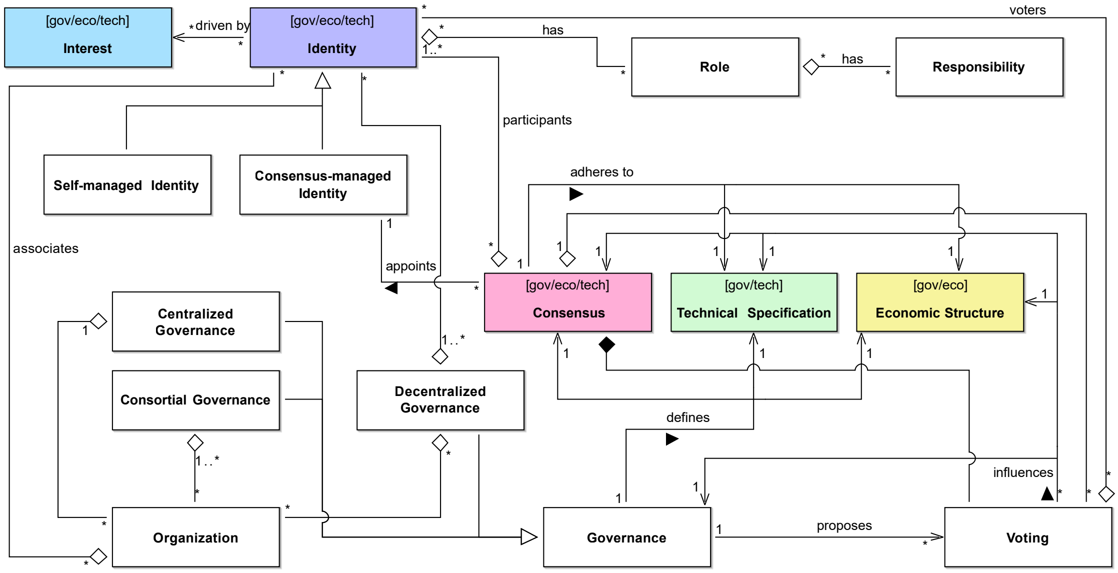


**Figure 1:** Governance perspective on the DLT domain.

This is, for example, the case for private blockchains or government operated networks such as the European Blockchain Services Infrastructure (EBSI[1]). In accordance with governance decision specifications, economic structures and consensus processes are defined. To decide on changes to the system, voting sessions may be proposed by the governance structure. The outcome of such voting may then impact the governance itself.

**Identity and decision drivers:** An *identity* in a DLT system is held by some entity, partaking in the system. Within the boundaries of the system, an identity is unique and represents what interests and roles the holder has. Identities may participate in the consensus process or in voting sessions and can be associated to organizations. The model introduces two specialized identities: a *self-managed* identity, which is created and managed by the holder entity. An example for self-managed identities are accounts in Ethereum [1]. In contrast, a *consensus managed* identity is issued to an entity upon some consensus. For example, in Hyperledger Fabric networks, identities can be issued by certificate authorities that have the required authorization. Identities can have roles with associated responsibilities and are driven by some interest. This can be anything from monetary compensation to the fulfillment of a mission.

**Establishing consensus:** The *consensus* concept is a representation of all processes in the system that have been established for reaching consensus on some matter relevant for the system's correct operation. This includes, for example, a consensus protocol, which will be added through the economic and technical perspectives. These processes adhere to the specification of the system and its economic structure. Identities may partake in such a consensus process in a manner that is dictated by their interests and roles.

[1]https://ec.europa.eu/digital-building-blocks/wikis/display/EBSI/Home

## 3.2. Economic perspective

An important part of a DLT system is its economic structure. This may include operational costs, as well as incentives for the participants for correct behaviors. In particular, public blockchains require monetary incentives, so that node operators provide the necessary computational resources for securing the ledger. For this service, node operators are typically rewarded in some cryptocurrency, which is an integral part of the system. In private and consortium DLT systems, the effort of operating the infrastructure is provided by the authorized participants, usually without any involvement of cryptocurrency. The fundamental concepts of the economics of a DLT system are illustrated in Figure 2.

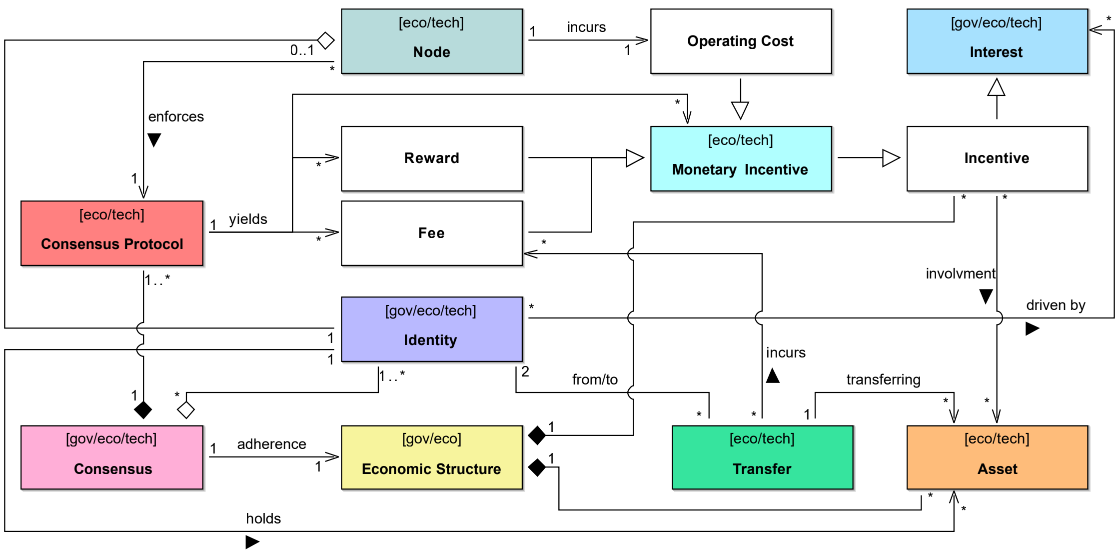


**Figure 2:** Economic perspective on the DLT domain.

**Incentives:** An integral part of the economic structure of a DLT-based system is the provision of *incentives*. This includes *monetary incentives* in the form of *rewards* and *fees* that result from the execution of the consensus protocol. That is, the consensus protocol may allot rewards for correct behavior — for example, nodes that contribute to securing the ledger by adhering to the consensus protocol. Participants that do act maliciously towards the network may be punished.

**Operating costs:** Running a node that participates in the consensus protocol incurs *operating costs* paid by the node's operator. These include, for example, hardware provisioning, maintenance, energy consumption, and facilities. In systems, where there are no rewards directly tied to the execution of the consensus protocol, providing computing resources is motivated by some other intent. In the model, this is expressed as an *interest* that serves as motivation for the identity held by a node.

**Assets and transfers:** *Assets* represent something of value that is held by an identity and can be *transferred* to another. Both fungible tokens, e.g., cryptocurrencies, and non-fungible tokens are represented by the asset concept. Assets may be awarded for participating in the consensus protocol. In the model, this is expressed as an incentive, namely the payment of some assets.

### 3.3. Technology perspective

From a computer science perspective, DLT-based systems are inherently state machines, where the current and previous states are recorded in a ledger that is distributed among interconnected nodes. Thereby, a state change is caused, for example, by transferring tokens between participants. To preserve the integrity of the ledger, nodes must reach consensus on the contained data. Figure 3 represents these ideas from a technical perspective.

**Specification:** DLT systems are developed according to a *technical specification*. This may be in form of a document, or a reference implementation. The specification may define the *consensus* and *networking protocols*, how *assets* are represented, *execution environments* for distributed algorithms, and languages.

**Consensus protocols:** The consensus protocols are part of the general *consensus* concept, encompassing all processes for reaching some consensus. These protocols are enforced by *nodes* to ensure that a state-changing result produced by a node is a valid next state. Further, consensus can be reached on software *artifacts*, for example, a smart contract. An artifact is written in some *language*, e.g., in a smart contract language such as Solidity, with a defined *syntax* and expresses a meaning, i.e., *semantics*. The artifact concept is not commonly present in popular blockchains, e.g., Ethereum. However, in academia this has been demonstrated, e.g., for reaching consensus on collaboratively constructed process models and their execution [29].

**Architecture of nodes:** *Nodes* are the fundamental infrastructure building blocks of DLT systems and the peer-to-peer network. Here, we do not conceptually distinguish between the node as software and the device that runs it. Nodes may however have a number of *components*, each contributing to the overall function of the node. Not all nodes may have the same function or purpose. For example, in Ethereum several types of nodes exist that store the entire ledger or only parts of it. The purpose of a node is given by the identity holding the node, attached roles and its responsibilities (see Figure 1).

**Networking:** Usually DLT systems employ a peer-to-peer *networking* model. While peer connections are not directly represented in the domain model, interconnected nodes form *network boundaries*. Within a boundary, nodes access the same distributed ledger. It is possible for a DLT system to have multiple networks or sub-networks, represented by hierarchical network boundaries. This is similar to the channel concept of Hyperledger Fabric, where the network may be divided into so-called channels that have their own ledger and access policies[2]. Authorization to networks is represented by giving *access* to identities.

[2]https://hyperledger-fabric.readthedocs.io/en/release-2.5/glossary.html#channel

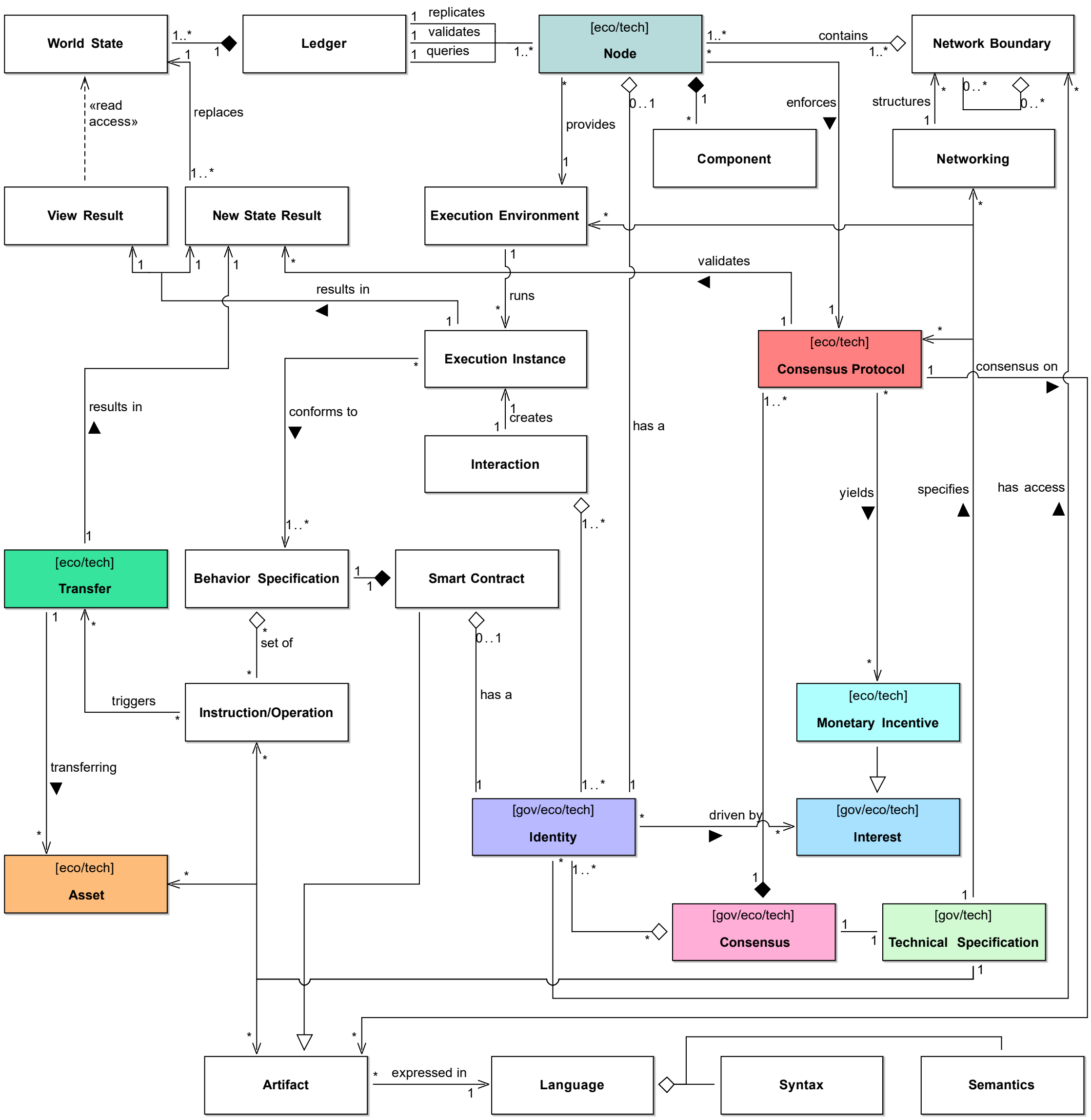


**Figure 3:** Technical perspective on the DLT domain.

**State results:** Two types of actions may result in a *new state*, leading to changes to the ledger. First, a *transfer* of assets from one identity to another (see Figure 2). Second, an *interaction* that triggers a state changing execution of a smart contract. The latter case is represented as an interaction with a contract identity, creating an *execution instance* that is run by an *execution environment* provided by a node. The execution instance conforms to the behavior specifications of invovled smart contracts. Not every interaction is state changing. For example, read-only access to data of smart contracts requires querying the current *world state*. This is represented as *view results* in the model. If a state transition is necessary, the new state needs to be valid in accordance with the consensus protocol. If that is the case, the new state replaces the current and is included in the *ledger*.

## 4. Illustrative application of the domain model

As an illustrative application example, we present a conceptualization of various parts of Ethereum by reverting to the domain model. In the individual models, some elements and relations have been omitted for brevity or due to redundancy. In particular, we regard the public Ethereum mainnet in its current stage of development.

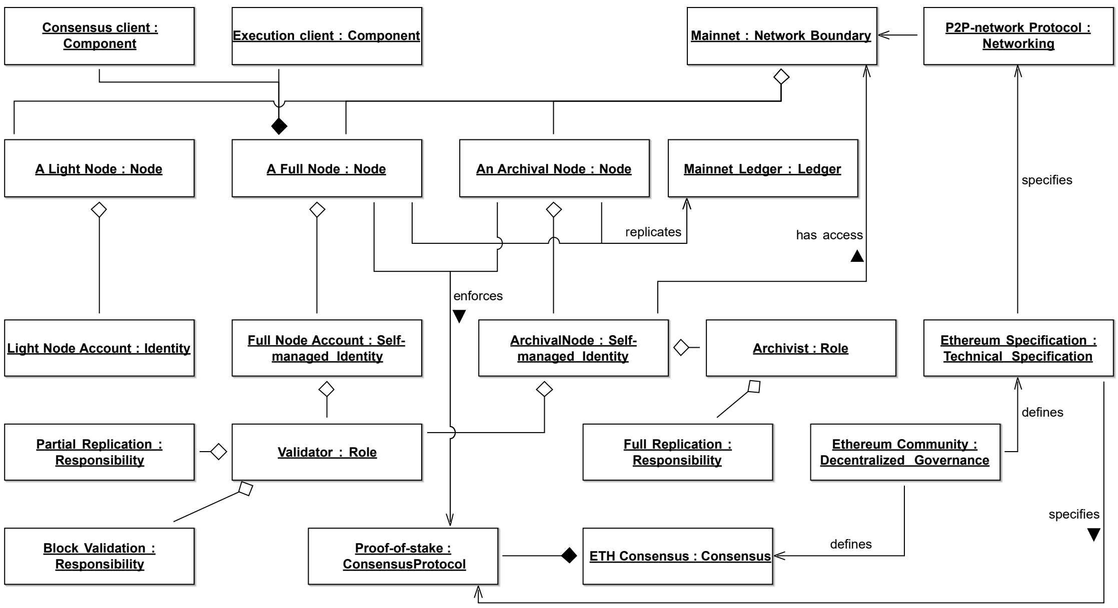


**Figure 4:** Basic network structure of the mainnet and the three node types with their roles and responsibilities.

Figure 4 shows an overview of the network and infrastructure of the Ethereum mainnet. In Ethereum, there are three types of nodes that differ mainly by the synchronization mechanisms for the ledger, specifying to what extent the ledger is replicated. For example, archival nodes hold a full copy of the ledger back to the very first transaction, but full nodes prune the ledger data to preserve storage space[3]. In contrast, light nodes only store a summary of the ledger in form of block headers and rely on full or archival nodes to access further information. These nodes cannot participate in consensus. All identities are self-managed, that is, users and node operators must create and manage their identities themselves. However, an identity is in principle transferable. The capabilities of the node types are expressed by their assigned roles and responsibilities. For, example, an archival node has the responsibility to fully replicate the ledger and to validate new blocks to be appended to the chain. Further, the model expresses that a node must operate, or have access to, two pieces of software, namely a consensus client

[3]https://ethereum.org/en/developers/docs/nodes-and-clients/#node-types

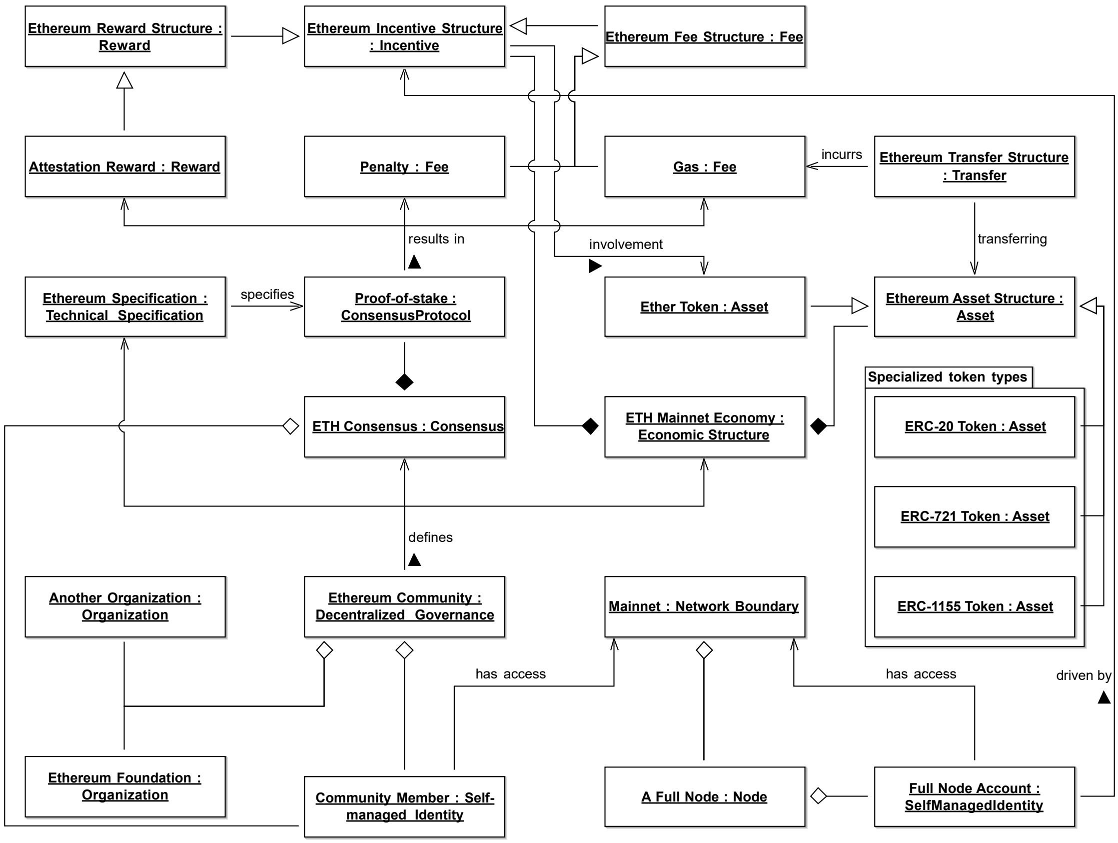


**Figure 5:** Governance and economic structure of the Ethereum mainnet, including some popular tokens.

and an execution client[4]. The Ethereum mainnet utilizes a proof-of-stake consensus protocol[5] enforced by validator nodes.

In Figure 5, another perspective of the Ethereum ecosystem is shown, focusing on governance and economic aspects. Ethereum applies a decentralized governance concept, driven by its community. In addition, various organizations are invested in Ethereum, for example the non-profit Ethereum Foundation [6]. Private or consortial networks based on Ethereum might deviate from official developments and governance decisions and instead implement their own governance concept. The mainnet is public, that is, everyone can access the network and the ledger. This is simply expressed by an association between the network boundary and the identities of the community members — here we assume that everyone who interacts with the network is considered a community member. Ethereum defines several token standards[7]. Some popular tokens are depicted as assets, namely Ether (ETH) the main cryptocurrency of Ethereum, other fungible tokens (e.g., ERC-20), and non-fungible tokens (e.g., ERC-721). These

[4]https://ethereum.org/en/developers/docs/nodes-and-clients/#what-are-nodes-and-clients
[5]https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/
[6]https://ethereum.org/en/foundation/
[7]https://ethereum.org/en/developers/docs/standards/tokens/

tokens are a fundamental part of the crypto-economic system of the mainnet, condensed in the economic structure concept. The token assets can be transferred for a so-called gas fee paid in Ether. Depicted is another fee that is paid by misbehaving nodes as penalty — slashing, i.e., removing a node entirely from the network, is omitted here. Validating nodes that behave correctly get paid for their effort with an attestation reward. These rewards and fees are the motivational drivers for proper conduct and participation in the network.

## 5. Discussion

The technical complexity and variety of DLT leads to a domain that is challenging to grasp, even for experts. This hinders the adoption of DLT and its integration in existing enterprise architectures and the design of viable business cases. Conceptual models of DLT help to address this by representing knowledge on various aspects of this technology, its economic ecosystem, and strategic implications. However, these aspects should not be regarded as separately, but instead relations and dependencies need to be investigated. In this regard, the proposed domain model offers an overview of DLT by giving perspectives on governance, economic, and technical characteristics. A successful adoption of DLT real-world scenarios requires to consider all layers from strategic and business considerations down to the technical infrastructure. This is due to multiple factors, including the low maturity of the technology and its surrounding ecosystem [30], but organizational barriers may also inhibit adoption in practice [31]. In alignment with the model types and goal taxonomy proposed by Guizzardi and Proper [3], our domain model can be best described as a *descriptive* model. Its main goals are to facilitate a better understanding of the domain and to support communication. Our holistic approach for representing the DLT domain allows to make the relationships between the different concepts and layers explicit. Consequently, it permits to trace dependencies between different layers. Further, the use of UML notation allows to instantiate and easily extend the concepts, as illustrated with the example of Ethereum, and facilitates the exploration of varied situations concerning DLT systems, e.g., specific decentralized applications.

The domain model is of considerable complexity, despite a number of purposely omitted or abstractly represented concepts. However, DLT constitute a very complex domain per se. As such, the model reflects the domain's complexity. Some omitted concepts commonly appear in other conceptualizations of DLT. This includes for example, *wallets* that hold tokens and cryptocurrencies and are associated with an identity. This has been condensed in our model into the *identity* concept. Since token ownership can still be expressed, we consider this an acceptable simplification. For similar reasons, we did not include an *actor* concept. The relation between an actor and identities seems quite intuitive — an actor may have multiple identities at the same time. Further, concepts for representing the *data structure* of the ledger are not available. A concrete data structure is rather project-specific and we were so far not able to derive a coherent, general conceptual representation in line with the level of abstraction that we aimed for. This may however change in future: next steps will include to validate the domain model with further instances — thereby refining the model in an iterative fashion.

## 6. Conclusion and future research

This work presents a general domain model of distributed ledger technologies, including system governance concerns, as well as the economic and technological structures. Further, we demonstrated the application of the domain model through a conceptual representation of the public Ethereum blockchain mainnet. The main benefits of such a general model as a holistic conceptualization of the DLT domain are on one hand the capability to represent multiple DLT scenarios and environments and on the other hand to express relations between concepts across perspectives. Thereby, an inherent downside is a loss in specificity of the contained concepts.

In the future we plan to further refine the domain model. Additionally, a dedicated graphical notation for the domain model could ease interaction and use. This could be done, for example, along previous work [32], where a combination of established modeling languages was used to facilitate the holistic development of decentralized applications, from the business case to the technical realization. Alternatively, future work may investigate domain-specific notations.

## Acknowledgments

This work was supported by the Swiss National Science Foundation project Domain-Specific Conceptual Modeling for Distributed Ledger Technologies [196889].

## References

[1] A. M. Antonopoulos, G. Wood, Mastering Ethereum: Building smart contracts and DApps, O'reilly Media, 2018.

[2] H.-G. Fill, A. Meier (Eds.), Blockchain: Grundlagen, Anwendungsszenarien und Nutzungspotenziale, Edition HMD, Springer Fachmedien, Wiesbaden, 2020. doi:10.1007/978-3-658-28006-2.

[3] G. Guizzardi, H. A. Proper, On understanding the value of domain modeling (regular paper), in: Proceedings of the International Workshop on Value Modelling and Business Ontologies, volume 2835 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 51–62.

[4] F. Kaya, F. Perez, J. Dekker, J. Gordijn, DECENT: A domain specific language to design governance decisions, in: Research Challenges in Information Science: Information Science and the Connected World, Lecture Notes in Business Information Processing, Springer Nature Switzerland, Cham, 2023, pp. 603–610. doi:10.1007/978-3-031-33080-3_43.

[5] A. Olivé, The conceptual schema of Ethereum, in: Conceptual Modeling - 39th International Conference, ER 2020, Vienna, Austria, November 3-6, 2020, Proceedings, volume 12400 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 418–428. doi:10.1007/978-3-030-62522-1_31.

[6] H.-G. Fill, A. Meier, Blockchain kompakt: Grundlagen, Anwendungsoptionen und kritische Bewertung, IT kompakt, Springer Fachmedien, Wiesbaden, 2020. doi:10.1007/978-3-658-27461-0.

[7] S. Curty, F. Härer, H.-G. Fill, Design of blockchain-based applications using model-driven

engineering and low-code/no-code platforms: A structured literature review, Software and Systems Modeling (2023). doi:10.1007/s10270-023-01109-1.

[8] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, P. Rimba, A taxonomy of blockchain-based systems for architecture design, in: 2017 IEEE International Conference on Software Architecture (ICSA), 2017, pp. 243–252. doi:10.1109/ICSA.2017.33.

[9] S. Bouraga, A taxonomy of blockchain consensus protocols: A survey and classification framework, Expert Systems with Applications 168 (2021) 114384. doi:10.1016/j.eswa.2020.114384.

[10] M. C. Ballandies, M. M. Dapp, E. Pournaras, Decrypting distributed ledger design—taxonomy, classification and blockchain community evaluation, Cluster Computing 25 (2022) 1817–1838. doi:10.1007/s10586-021-03256-w.

[11] J. Pfeffer, EthOn - An Ethereum ontology, 2021. URL: https://consensys.github.io/EthOn/EthOn_spec.html, accessed 2023-08-21.

[12] G. Bella, D. Cantone, C. Longo, M. Nicolosi-Asmundo, D. F. Santamaria, Blockchains through ontologies: The case study of the Ethereum ERC721 standard in OASIS (extended version), arXiv (2021). doi:10.48550/ARXIV.2109.02899.

[13] G. Bella, D. Cantone, C. F. Longo, M. Nicolosi-Asmundo, D. F. Santamaria, The ontology for agents, systems and integration of services: OASIS version 2, Intelligenza Artificiale 17 (2023) 51–62. doi:10.3233/IA-230002.

[14] D. Cantone, C. Longo, M. Nicolosi Asmundo, D. Santamaria, C. Santoro, Ontological smart contracts in OASIS: Ontology for agents, systems, and integration of services, in: Intelligent Distributed Computing XIV, 2022, pp. 237–247. doi:10.1007/978-3-030-96627-0_22.

[15] L. Besançon, C. F. da Silva, P. Ghodous, J.-P. Gelas, A blockchain ontology for DApps development, IEEE Access 10 (2022) 49905–49933. doi:10.1109/ACCESS.2022.3173313.

[16] N. Petrović, M. Tošić, Semantic approach to smart contract verification, Facta Universitatis, Series: Automatic Control and Robotics 19 (2020) 021–037. doi:10.22190/FUACR2001021P.

[17] O. Choudhury, N. Rudolph, I. Sylla, N. Fairoza, A. Das, Auto-generation of smart contracts from domain-specific ontologies and semantic rules, in: IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, July 30 - August 3, IEEE, 2018, pp. 963–970. doi:10.1109/Cybermatics_2018.2018.00183.

[18] A. Ellervee, R. Matulevicius, N. Mayer, A comprehensive reference model for blockchain-based distributed ledger technology, in: Proceedings of the ER Forum 2017 and the ER 2017 Demo Track Co-Located with the 36th International Conference on Conceptual Modelling (ER 2017), Valencia, Spain, November 6-9, 2017, volume 1979 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2017, pp. 306–319.

[19] S. Curty, F. Härer, H.-G. Fill, Towards the comparison of blockchain-based applications using enterprise modeling, in: Proceedings of the ER Demos and Posters 2021 Co-Located with 40th International Conference on Conceptual Modeling (ER 2021), St. John's, NL, Canada, October 18-21, 2021, volume 2958 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 31–36.

[20] J. Ladleif, M. Weske, A unifying model of legal smart contracts, in: Conceptual Modeling - 38th International Conference, ER 2019, Salvador, Brazil, November 4-7, 2019, Proceedings,

volume 11788 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 323–337. doi:10.1007/978-3-030-33223-5_27.

[21] V. Dwivedi, A. Norta, A legal-relationship establishment in smart contracts: Ontological semantics for programming-language development, in: Advances in Computing and Data Sciences, volume 1440, Springer International Publishing, Cham, 2021, pp. 660–676. doi:10.1007/978-3-030-81462-5_58.

[22] H. M. Kim, M. Laskowski, Toward an ontology-driven blockchain design for supply-chain provenance, Intell. Syst. Account. Finance Manag. 25 (2018) 18–27. doi:10.1002/isaf.1424.

[23] M. S. Fox, The TOVE project: Towards a common-sense model of the enterprise, in: Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 1992, pp. 25–34. doi:10.1007/BFb0024952.

[24] W. E. McCarthy, The REA accounting model: A generalized framework for accounting systems in a shared data environment, The Accounting Review 57 (1982) 554–578.

[25] J. L. G. Dietz, Enterprise ontology: Theory and methodology, Springer, 2006. doi:10.1007/3-540-33149-2.

[26] J. T. de Kruijff, H. Weigand, Understanding the blockchain using enterprise ontology, in: Advanced Information Systems Engineering - 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings, volume 10253 of *Lecture Notes in Computer Science*, Springer, 2017, pp. 29–43. doi:10.1007/978-3-319-59536-8_3.

[27] H. Syahputra, H. Weigand, The development of smart contracts for heterogeneous blockchains, in: Enterprise interoperability VIII: Smart services and business impact of enterprise interoperability, volume 9 of *Proceedings of the I-ESA Conferences*, Springer, Berlin, Germany, 2018, pp. 229–238. doi:10.1007/978-3-030-13693-2_19.

[28] R. A. Stebbins, Exploratory research in the Social Sciences, volume 48 of *Qualitative Research Methods*, Sage Publications, 2001.

[29] F. Härer, Process modeling in decentralized organizations utilizing blockchain consensus, Enterprise Modelling and Information Systems Architectures (EMISAJ) International Journal of Conceptual Modeling 15 (2020) 13:1–13:17. doi:10.18417/emisa.15.13.

[30] S. Flovik, R. A. Moudnib, P. Vassilakopoulou, Determinants of blockchain technology introduction in organizations: An empirical study among experienced practitioners, Procedia Computer Science 181 (2021) 664–670. doi:10.1016/j.procs.2021.01.216.

[31] T. Clohessy, T. Acton, N. Rogers, Blockchain adoption: Technological, organisational and environmental considerations, in: H. Treiblmaier, R. Beck (Eds.), Business Transformation through Blockchain, Springer International Publishing, Cham, 2019, pp. 47–76. doi:10.1007/978-3-319-98911-2_2.

[32] S. Curty, H.-G. Fill, Exploring the systematic design of blockchain-based applications using integrated modeling standards, in: Proceedings of the PoEM 2022 Workshops and Models at Work Co-Located with Practice of Enterprise Modelling 2022, London, United Kingdom, November 23-25, 2022, volume 3298 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022.