Welcome to the Jungle: A Reference Model for Blockchain, DLT and Smart-Contracts

Julien Hatin
Orange Labs, 42 rue des Coutures, Caen, France

Emmanuel Bertin
Orange Labs, 42 rue des Coutures, Caen, France

Baptiste Hemery
Orange Labs, 42 rue des Coutures, Caen, France

Nour El Madhoun
ISEP, 28 rue Notre Dame des Champs, Paris, France

Abstract

Blockchain technology has gained increasing attention from research and industry over the recent years. This interest is mainly due to its core property that allows users to perform transactions without a Trusted Third Party (TTP), while offering a transparent and fully protected tracking of these transactions. However, there is a lack of reference models to describe and compare various Blockchain technologies, leading to some confusion between different kinds of solutions. We propose in this paper a reference model aiming to assess and compare different kind of Blockchain-based ecosystems, including Decentralized Applications (DApp).

2012 ACM Subject Classification General and reference → Surveys and overviews; Computer systems organization → Distributed architectures

Keywords and phrases Blockchain, DLT, Smart-Contracts

Digital Object Identifier 10.4230/OASIcs.Tokenomics.2020.13

Category Short Paper

1 Introduction

Starting from the Bitcoin application ten years ago, Blockchain and Distributed Ledger Technologies (DLT) have since considerably expanded in both industry and academic communities, leading to a very fragmented and somehow puzzling landscape. In this context, the aim of this paper is to define a reference model for Blockchain and DLT stakeholders, to properly characterize various ecosystems and use-cases, especially in the case of distributed applications. While existing modeling works have focused either on the engineering of DLT solutions, or on the business relationships between the stakeholders, we intend to propose a model addressing the interactions between these both levels. This model may then be used as a kind of "traveler's guide" for a Blockchain journey, enabling to better model needs and possible solutions.

2 State of the Art and Methodology

In the state of the art, few papers address the question of modelling DLT. Most research works in this field survey the Blockchain technology and its application use cases while introducing some high-level modelling on how transactions are performed [3, 12, 13, 14]. [12] is a typical example of this category of articles, providing a synthesis on Blockchain properties and a typology of application domains. However, some papers address more explicitly the question of modeling Blockchain and DLT. [5] introduces for example a comprehensive UML-

2nd International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2020).
Editors: Emmanuelle Anceaume, Christophe Bisière, Matthieu Bouvard, Quentin Bramas, and Catherine Casamatta; Article No. 13; pp. 13:1–13:5
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

based Blockchain ontology, based on a study on how the Blockchain technology operates (transactions, blocks, etc.). However, such work remains focused on the theoretical Blockchain operations, and not on existing Blockchain or DLT solutions.

To fill this research gap, we have chosen to apply an empirical approach rather than starting from a theoretical study on the way Blockchain technology operates. To reach this target, we selected nine typical DLT solutions and studied their architecture, focusing on both similarities and differences. This choice of these nine DLT solutions is based on the reputation and usage, combined with a will to consider solutions with different architectures. We also focused not only on fundamentals of Blockchain (i.e. mining blocks to build a distributed ledger), but also on their usages with dApps (distributed applications) and smart contracts. However, we acknowledge that this choice of nine solutions might include some bias, due to our knowledge and practice of these technologies. This constitutes a limit of our work that we intend to address in our future work by extending the scope of the surveyed solutions.

3 Reference Model

DLT are not only bringing technical changes, but also changes in the actor model and in the value chain. This is precisely what we intend to capture in this article. Figure 1 details our reference model of the actors building a Blockchain ecosystem.

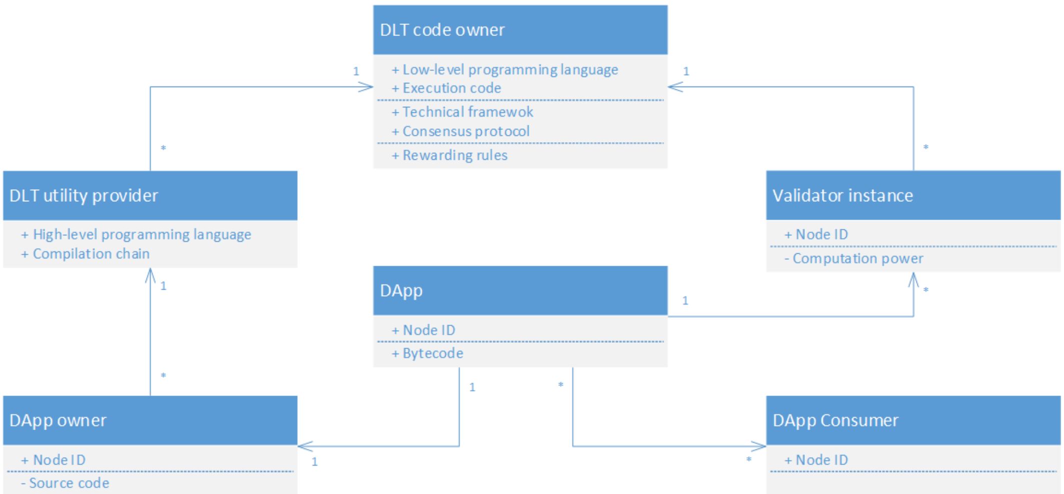

Figure 1 Actor model of DLT.

More precisely, we identified 6 different roles. The *DLT Code Owner* role is assigned to the company or organization that develops and maintains the original source code. Trust in the Core Code is essential, as the whole security of the blockchain network depends on it. Three elements are produced by the actor playing this role:

- The protocol that enable to issue, exchange and validates blocks and executes the various decentralized application.
- The low-level assembly-like language of dApp.
- The virtual machine that is able to execute it.

To run a decentralized application, a network of peers is needed. The members of this network are the blockchain nodes. A node has access to the ledger, can execute transaction and DApp on the blockchain.

To build concrete application, a set of tools is mandatory. They are provided by the *DLT Utility Provider*. Those tools consist of a human usable high-level language (i.e. C++ for EOS or solidity for Ethereum) and of the tools to compile it. Those mandatory tools are needed to build concrete dApp.

Using tools from the *DLT Utility Provider*, a *DApp Owner* can develop a new decentralized application. The compiled code is then published on the DLT by the *DApp Owner*. In blockchain architecture, the code is indeed not hosted by a central server. Instead the code is published in a network by the *DApp Owner*, and the storage is shared by every nodes of this network. The publication and the deployment of a decentralized application in this network can be done freely (i.e. EOS) or with fees (i.e. Ethereum or NEO).

A *DApp Consumer* is the end user of the decentralized application. It can execute function and read the result. Moreover, a *DApp consumer* is not necessarily a person or an organization but can also be another DApp. The function execution can be free, or with fees paid either by the *DApp Consumer* or the *DApp Owner*.

A *Validator Instance* is a key role in the blockchain systems: it validates the transaction. Depending on the platform this could be using a proof of stakes or a proof of work or even another consensus algorithm. The *DLT Code Owner* defines the consensus protocol and technical framework used by the validators, as well as potential incentives and rewarding rules. A *Validator Instance* is characterized by its computing power, in the case of Proof of Work mechanisms.

Finally, the *DApp* is also a part of the blockchain, holding the bytecode of the application. It is depending on the existence of *Validator Instances* to be published in the blockchain and to able to perform transactions.

At the end of this article, table 1 provides an extensive comparison between the surveyed DLT solutions according to the proposed role model.

4 Discussion and Application to existing DLT

The *DLT code owner* of any DApps-enabling blockchain is always endorsed by a single company for permissioned blockchains, or by a foundation for permissionless blockchains (cf Table 1). However, this entity is usually working with agile management process to include new features in the core code, e.g. the Lisk Improvement Proposition and the Ethereum Improvement Proposition. In addition, the core code is usually open source. Trusting the protocol and the low-level features is indeed required to enable trust between actors. An exception is Libra, which is permissioned, but managed by an association.

Another difference between the various *DLT Code Owners* lies in their level of control over their technical assets. We can identify here a first organizational strategy that aims at controlling the network protocol and leaving the application development and deployment to the *DApp owners* in association with their partners. In this category we find Libra, Ethereum, EOS, Lisk, and Hyperledger. Alternatively, other companies rely on existing open-source DLT (that they do not control) to build their solution on top of it. We can cite here Quorum, Monax, Counterparty. This model is usually associated with a tighter control on *DApps owners*.

We can also distinguish solutions by looking at their openness strategy. Two approaches exist here: the consortium approach and the free access approach. The main public blockchain that can run *DApps* as overviewed in this paper are Ethereum, Eos, Counterparty and Lisk. Those blockchains are open to any *DApp owner*. Oppositely, some actors have chosen to be selective on who can participate as they address a specific vertical such as Monax or Quorum.

Hyperledger have chosen a different approach by offering to its customers a solution to deploy their own consortium with their own partners. This can be a well-suited solution to construct quickly a blockchain environment. But this does not enable to open this network to anybody or to switch to a free access model. Facebook have chosen here to develop its own solution, which is very specific, because it starts as a permissioned network but aims to become a free access network.

5 Conclusion and perspectives

The proposed reference model is designed as a tool for helping practitioners (e.g., business managers and architects) to assess their choices in terms of roles and business models for designing DApps. It is also designed as a tool for researchers to ground studies on the value-chain of DLT and DApps, e.g. by simulating the behavior of the various actors according to different incentives

Our perspective is to use a multi-agent model, relying on the roles described in the model, to have a better comprehension of these actors' behaviors – in a context where *DApps owners*, *validator instances* and *DApps consumers* are all evolving in the jungle of competitive and often incompatible DLT solutions.

References

1 Block.one. Eos white paper, 2018. URL: https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md.
2 Vitalik Buterin et al. Ethereum white paper, 2013. URL: https://ethereum.org/whitepaper/.
3 Konstantinos Christidis and Michael Devetsikiotis. Blockchains and smart contracts for the internet of things. *Ieee Access*, 4:2292–2303, 2016. doi:10.1109/ACCESS.2016.2566339.
4 CounterpatyXCP. Counterpary documentation. URL: https://counterparty.io/docs/.
5 Joost de Kruijff and Hans Weigand. Understanding the blockchain using enterprise ontology. In *International Conference on Advanced Information Systems Engineering*, pages 29–43. Springer, 2017. doi:10.1007/978-3-319-59536-8_3.
6 Lisk Foundation. Lisk documentation. URL: https://lisk.io/documentation/lisk-sdk/index.html.
7 The Hyperledger White Paper Working Group. Hyperledger white papers. URL: https://www.hyperledger.org/learn/white-papers.
8 Libra Association Members. Libra white paper v2.0, 2020. URL: https://libra.org/en-US/white-paper/.
9 Monax.io. Monax documentation. URL: https://docs.monax.io/.
10 JP Morgan. Quorum wiki. URL: https://github.com/jpmorganchase/quorum/wiki.
11 Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Technical report, Manubot, 2019.
12 Deepak Puthal, Nisha Malik, Saraju P Mohanty, Elias Kougianos, and Chi Yang. The blockchain as a decentralized security framework [future directions]. *IEEE Consumer Electronics Magazine*, 7(2):18–21, 2018. doi:10.1109/MCE.2017.2776459.
13 Horst Treiblmaier and Roman Beck. *Business Transformation through Blockchain*. Springer, 2019.
14 Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, 14(4):352–375, 2018. doi:10.1504/IJWGS.2018.10016848.

Table 1 Comparison of studied DLT.

Blockchain	DLT code owner	DLT utility provider	Developing language	DApp owner	DApp consumer	Validator Instance
Bitcoin[11]	Bitcoin Foundation	Bitcoin Foundation	Script	anyone	anyone	Bitcoin miners
Ehtereum[2]	Ethereum Foundation	anyone	solidity	ether holder	ether holder	Ethereum miners
Quorum[10]	Ethereum Foundation, JP Morgan	anyone	solidity	anyone with access to a permissioned node	anyone with access to a permissioned node	permissioned node
Counterparty[4]	Bitcoin Foundation	Counterpary	solidity	anyone	anyone	Bitcoin miners
EOS[1]	Block One	anyone	C++	token holder	anyone	approved block producer, elected by token holder
Libra[8]	Libra association	Libra association	Move	Association member	Libra holder	Association member
Monax[9]	The Linux Foundation	The Linux Foundation	solidity, Graphical flow	Any actor within the Monax system	Any actor within the Monax system	Monax affiliates
Lisk[6]	Lisk Foundation	Lisk Foundation	Javascript	anyone	anyone	101 active delegates
Hyperledger[7]	The Linux Foundation	The Linux Foundation	C++, solidity	Any actor within the system	Anyone within the system	Any actor