



HAL
open science

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

Julien Hatin, Emmanuel Bertin, Baptiste Hemery, Nour El Madhoun

► To cite this version:

Julien Hatin, Emmanuel Bertin, Baptiste Hemery, Nour El Madhoun. Welcome to the jungle: A Reference Model for Blockchain, DLT and Smart-Contracts. *Tokenomics 2020 on Blockchain Economics, Security & Protocols (2nd International Conference)*, Oct 2020, Toulouse, France. hal-03014631

HAL Id: hal-03014631

<https://hal.science/hal-03014631>

Submitted on 19 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

1 Welcome to the jungle: A Reference Model for 2 Blockchain, DLT and Smart-Contracts

3 **Julien Hatin**

4 Orange Labs, 42 rue des Coutures, Caen, France

5 **Emmanuel Bertin**

6 Orange Labs, 42 rue des Coutures, Caen, France

7 **Baptiste Hemery** 

8 Orange Labs, 42 rue des Coutures, Caen, France

9 **Nour El Madhoun**

10 ISEP, 28 rue Notre Dame des Champs, Paris, France

11 — Abstract —

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

19 **2012 ACM Subject Classification** General and reference → Surveys and overviews; Computer
20 systems organization → Distributed architectures

21 **Keywords and phrases** Blockchain, DLT, Smart-Contracts

22 **Digital Object Identifier** 10.4230/OASICS.CVIT.2016.23

23 **1** Introduction

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

34 **2** State of the Art and Methodology

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



© Julien Hatin, Emmanuel Bertin, Baptiste Hemery, and Nour El Madhoun;
licensed under Creative Commons License CC-BY

42nd Conference on Very Important Topics (CVIT 2016).

Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:5

OpenAccess Series in Informatics



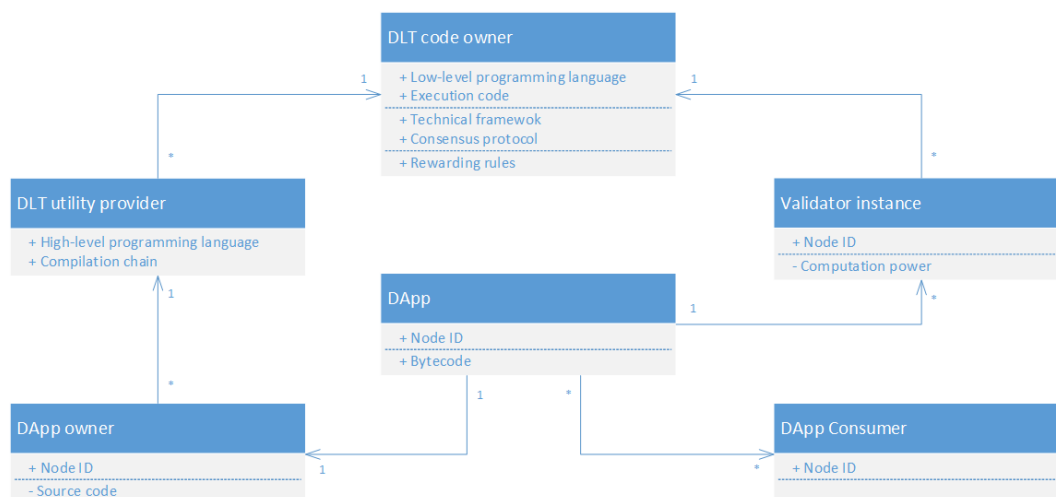
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

(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.

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

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

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

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

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

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

95 **4 Discussion and Application to existing DLT**

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

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

111 We can also distinguish solutions by looking at their openness strategy. Two approaches
112 exist here: the consortium approach and the free access approach. The main public blockchain
113 that can run *DApps* as overviewed in this paper are Ethereum, Eos, Counterparty and Lisk.
114 Those blockchains are open to any *DApp owner*. Oppositely, some actors have chosen to be
115 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 CounterpartyXCP. Counterparty 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.

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
Ethereum[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	Counterparty	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	Any actor within the system	Any actor

Table 1 Comparison of studied DLT