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

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-
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.

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 enables 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

<table>
<thead>
<tr>
<th>Blockchain</th>
<th>DLT code owner</th>
<th>DLT utility provider</th>
<th>Developing language</th>
<th>Script</th>
<th>DApp owner</th>
<th>DApp consumer</th>
<th>Validator Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bitcoin[1]</td>
<td>Bitcoin Foundation</td>
<td>Bitcoin Foundation</td>
<td>anyone</td>
<td>Script</td>
<td>anyone</td>
<td>anyone</td>
<td>Bitcoin miners</td>
</tr>
<tr>
<td>Quorum[10]</td>
<td>Ethereum Foundation, JP Morgan</td>
<td>Counterparty</td>
<td>anyone</td>
<td>solidity</td>
<td>anyone with access to a permissioned node</td>
<td>anyone</td>
<td>Ethereum miners</td>
</tr>
<tr>
<td>Counterparty[4]</td>
<td>Counterparty</td>
<td>Counterparty</td>
<td>anyone</td>
<td>solidity</td>
<td>anyone</td>
<td>anyone</td>
<td>any other holder</td>
</tr>
<tr>
<td>EOS[1]</td>
<td>Block.One</td>
<td>Block.One</td>
<td>anyone</td>
<td>C++</td>
<td>anyone</td>
<td>token holder</td>
<td>any other holder</td>
</tr>
<tr>
<td>Libra[8]</td>
<td>Libra association</td>
<td>Libra association</td>
<td>undefined</td>
<td>Move</td>
<td>Association member</td>
<td>Association member</td>
<td>any other holder</td>
</tr>
<tr>
<td>Monax[9]</td>
<td>The Linux Foundation</td>
<td>The Linux Foundation</td>
<td>solidity, Graphical flow</td>
<td>Javascript</td>
<td>undefined</td>
<td>undefined</td>
<td>any other holder</td>
</tr>
<tr>
<td>Lisk[6]</td>
<td>Lisk Foundation</td>
<td>Lisk Foundation</td>
<td>undefined</td>
<td>Javascript</td>
<td>undefined</td>
<td>undefined</td>
<td>any other holder</td>
</tr>
<tr>
<td>Hyperledger[7]</td>
<td>The Linux Foundation</td>
<td>The Linux Foundation</td>
<td>undefined</td>
<td>Javascript</td>
<td>undefined</td>
<td>undefined</td>
<td>any other holder</td>
</tr>
</tbody>
</table>