Current Trends for Blockchain and Smart Contracts (In Financial Services)

Pablo Garcia Bringas
University of Deusto, Bilbao, Spain
Iker Pastor-López
University of Deusto, Bilbao, Spain
Giuseppe Psaila
University of Bergamo, Dalmine, Italy

Abstract

The most popular virtual currency (*bitcoin*) has become so popular because it relies on the concept of *Distributed Ledger*, realized by means of a technology known as *BlockChain*. This technology removes the need for third parties (such as authorities) that authorize transactions; in fact, it is a peer-to-peer network where consensus to transactions is given by peers. Furthermore, the distributed nature of BlockChain ensures a high-level of robustness to attacks. However, since its birth (a decade ago) many evolutions have been introduced: now, we distinguish between *permissionless* and *permissioned* BlockChains, the concept of smart contract is now supported, and various platforms are available. The contribution of this paper is to provide novices with the current trends, in particular by discussing the adoption of BlockChain technology in financial services.

Keywords: BlockChain characterization, Smart contracts, Perspective BlockChain in financial services
JEL classification: L86

Introduction

The introduction of *bitcoin*, the virtual currency that has become very popular around the world, has imposed a new technological approach to address the problem of registering in a trustable and immutable way any kind of (possibly financial) transaction. This approach relies on the concept of *distributed ledger*.

The notion of ledger is certainly not new: for a long time, notaries have been registering property changes of houses and terrains, to immutably establish property rights. However, the classical approach to handle ledgers presupposes the existence of a third party (the notary) that guarantees that the registration is correct and immutable.

The novelty of *Bitcoin* (the platform that supports bitcoin, the virtual currency) is the absence of a third party. The idea is that in a peer-to-peer context, nodes participating to the peer-to-peer network maintain a copy of the ledger (for this reason, it is called *distributed ledger*) and cooperate to give consensus to new transactions, by verifying if a transaction is legal.

The technology proposed by Bitcoin to maintain a distributed ledger is named *BlockChain*. The distributed ledger is essentially a chain of blocks, where each block contains a pool of (verified) transactions. A cryptographic mechanism ensures that a block, once inserted into the chain, cannot be changed. In fact, an attempt to change one block would violate the chain of keys generated from that block, including a hash code derived by the content of the block; furthermore, a change in a copy of a block stored within a peer would not correspond to other copies maintained by other peers.

This distributed approach avoids the need for a third party: consensus to make the operation is given by peers participating to the network. The trust about the correctness of the operation is given by the overall network: no trust must be given to one single centralized authority (that is not actually under the control of parties performing transactions).

As it always happens, a new technology opens new scenarios and new developments. As far as BlockChain technology is concerned, various interpretations and/or implementations have been proposed. In particular, it is now clear that BlockChain can live without bitcoin (the virtual currency). Not only, even without money exchange: in fact, it appears to have the potential to support any kind of human activity where the immutability of transactions is the crucial issue.

Furthermore, the idea of *Smart Contract* has further pushed on the possible application of BlockChain. In fact, by borrowing the concepts of object and transformation method (i.e., code that defines data behaviour and transformation) from object-oriented programming, it is possible to introduce very complex transactions, i.e., very complex state changes of data; no third-party system is necessary to validate such transformations, because the transformation code is executed directly within the BlockChain platform.

The goal of this paper is to provide novices (of the BlockChain world) with a high-level and brief overview of currently most popular BlockChain platforms, in order to give hints concerning the current technological trends. We will discuss the difference between *permissionless* and *permissioned* BlockChains, as well as we will present basic ideas concerning smart contracts. Current most popular BlockChain platforms will be presented. After that, we will briefly discuss possible application scenarios for financial services, indicating the platforms that we expect to be suitable for them.

Methodology

Brief history of BlockChain

Haber & Stornetta (1990) developed a cryptographically protected chain of blocks in which no one could manipulate the timestamps of the documents. But it was only in 2008 that Satoshi Nakamoto described the first BlockChain (Nakamoto, 2008) named *Bitcoin*, that supports the virtual currency named *bitcoin*. In Bitcoin, the BlockChain constitutes the underlying protocol of any crypto-currency and is a novel peer-to-peer methodology to link a sequence of transactions or events that ensures their immutability.

A few months later, a new open source application implementing the Bitcoin protocol was released and the first block of the chain, called Genesis, was generated. By installing this application, anyone can become a part of the Bitcoin peer-to-peer network.

Even if bitcoin is the most famous application of BlockChain technology, many different applications could significantly benefit by its adoption. To this end, in 2013 V. Buterin started working on a BlockChain capable of providing advanced functionalities, such as smart contracts, executing them directly within the peer-to-peer network (Buterin, 2014).

Ethereum, presented in Wood (2014) and Dannen (2017), was presented as a new public BlockChain platform which overtakes the simple support to a cryptocurrency (named *Ether*), by evolving into a platform to develop decentralized applications as well. This is made possible by introducing the concept of smart contract, (thus, actually it has implemented the ideas in Buterin (2014).

The concept of smart contract was introduced by Szabo (1997): it combines computer protocols with user interfaces to execute the terms of a contract. Furthermore, in 2014, when BlockChain was clearly emerging, Fairfield (2014) proposed the use of smart contracts to carry out with the transaction processes by automatically executing contracts in a cost-effective, transparent and secure manner.

However, this does not end the history of BlockChain: this is just the beginning of its evolution. In fact, the history continues with the *HyperLedger* project (Dhillon et al., 2017), by Linux Foundation. It is aimed at developing a family of BlockChain platforms based on the same basic architecture, whose goal is to support information systems. The most famous platform belonging to this family is *HyperLedger Fabric* (Sousa et al., 2018).

Classical vs permissioned BlockChain

What is the level of trust that each participant to the BlockChain has in relation to other participants? It depends on the application context. To understand, we classify possible application scenarios.

- *No trust*. When no trust is possible, i.e., *nobody trusts anybody*, the best warranty that transactions can be performed is given by a multitude of nodes. In fact, a large number of nodes involved both to validate transactions and to store blocks makes very difficult to attack and corrupt the system.
- *Partial trust*. In a controlled environment, where many parties co-operate to get a common goal (such as an integrated supply chain - Korpela et al., 2017), in principle each party trusts other parties a little bit. However, they do not fully trust each other, for several reasons: a centralized approach, where a central entity provides the IT support for everybody, could be prone to system faults, programming errors and external attacks. In contrast, having several nodes that provide consensus to transactions as well as that store multiple copies of the ledger, significantly increases the reliability of the system.
- *Full trust*. This scenario is the classical approach to the development of information systems to provide a service to many parties. A central authority is (or must be) fully trusted by other parties (they subscribe a service contract, or they are forced by laws).

Clearly, the third scenario motivated the original design of BlockChain. However, current BlockChain platforms can be divided in two families:

- *Permissionless BlockChain platforms*. This family encompasses classical BlockChain technology, devoted to support virtual currencies. A new node is totally free to enter the network, provided that its behaviour is compliant with general rules of the platform.
- *Permissioned BlockChain platforms*. This family encompasses platforms such that new nodes cannot freely enter the network; they must be authorized by an administrator. Furthermore, they must agree with the business logic supported by the system.

Smart contracts

Originally, BlockChain technology was thought to support money exchange based on a virtual currency. However, a ledger can be used for many application contexts, not only for money exchange. Consequently, the idea of using BlockChain for application contexts without a simple exchange of (possibly virtual) money is straightforward.

How to foster the adoption of BlockChain? If transactions are not totally-free money transfers, but ruled operations that can be performed on the basis of an agreed contract, a new and immense scenario opens. This is the concept of *smart contract*; originally, it was introduced in Szabo (1997), "to describe agreements between two or more parties, that can be automatically enforced without a trusted intermediary." (from Lande & Zunino, 2018). The idea was ignored for a few years; then, the advent of BlockChain has made it actually applicable.

A smart contract can be seen as a contract state, i.e., the set of properties that characterize it, provided with some transformation methods, i.e., procedures that determine how the contract state can change. A transaction consists in asking to change the contract state by invoking a transformation method. When a transaction is issued, the contract state is changed, according to the invoked transformation method.

The potentiality of this concept is incredible: once two parties have agreed to start the contract, its behaviour can become automatic, no third party that handles the contract is necessary.

Anyway, smart contracts are supported by different BlockChain platforms in different ways.

- *In-platform code*. This approach to smart contracts is characterized by the fact that transformation methods are shared within the BlockChain platform, ready to be used when necessary. A transaction is a triple ‹*os*, *ns*, *r*›, where *os* is the old state, *ns* is the new state and r is the request that generated the new state. As far as the possibility of using transformation methods by parties is concerned, three different scenarios are available: (i) *Contract-Specific Code*, i.e., the code of transformation methods is associated to one specific contract; (ii) *Contract-family code*, i.e., the code of transformation methods is shared among contracts belonging to the same family; (iii) *Global code*, i.e., transformation methods are global, in the sense they can affect many objects stored within the ledger.
- *External Code*. This category encompasses smart contracts whose business logic is not within the platform. This is the case of Bitcoin: the first BlockChain platform in the world has not been designed to host smart contracts; nevertheless, it is used for many smart contract-based applications (see Lande & Zunino, 2018), This is made possible by implementing protocols based on cryptographic-message exchange: transactions are registrations of messages; involved parties receive messages (only involved parties can decipher them) and, consequently, act (see the description of Bitcoin in further sections). It is clear that, in this scenario, the business logic of smart contracts is handled outside the BlockChain platform: each party must implement it, hoping to be conformant with specifications.

Results

Principal BlockChain platforms

Based on the considerations made in previous sections, we briefly introduce the main BlockChain platforms that currently are most popular. In addition, *Table I* resumes this classification.

- *Bitcoin* has been the first BlockChain platform. Born to support the bitcoin virtual currency, in fact, it validated the approach, proving the effectiveness of the idea. It is a typical *permissionless* platform: a node can freely enter the network; in that virtual-money transfer is the typical context where *nobody trusts anybody*.

  No explicit support to smart contracts is provided. However, it is possible to realize something similar by means of complex workarounds.
- *Ethereum* (Wood, 2014) is the BlockChain of the *Ether* virtual currency. Since it is designed to support virtual money exchange, it is still a *permissionless* platform.

  However, if compared with Bitcoin, it natively supports smart contracts: they are designed to deal with exchange of money, even though contracts that do not exchange money could be developed.
- *HyperLedger Fabric* (Sousa et al., 2018) is a *permissioned* platform that gives a different perspective to the adoption of BlockChain technology: a BlockChain is seen as a database that immutably logs all changes (transactions) performed on the database; this approach ensures that the current state of the database can be rebuilt, by re-executing change requests stored within the BlockChain. The first effect of this database view is that not everybody can enter the network: only authorized parties are admitted (in fact, it is a permissioned platform).

  The second consequence is that smart contracts are global procedures, called *chain code*, that actually perform changes on data; a transaction is the invocation of a procedure by a party.
- *Corda* is "a distributed ledger platform for recording and processing financial agreements" (from Brown et al., 2016). It is a *permissioned* platform, thus only authorized parties can enter the network.

  It is designed to naively support legal aspects concerning smart contracts: this means that contracts are accompanied by a legal-prose description of the contract itself; not only, when a transaction happens, a legal prose version is generated.

  A smart contract (or *smart agreement*) has a state (accessible only by involved parties) as well as transformation code and validity rules. An interesting concept provided by Corda is the notion of *Contract Template* (Clack et al., 2016): parties pre-load templates of contracts, where details (e.g., interest rate and duration) are not specified; when two or more parties agree, the actual contract is derived from the template, by specifying missing details; this way, all contracts derived from the same template share the same code.

The interested reader can find a detailed comparison of Ethereum, HyperLedger Fabric and Corda in Valenta & Sandner (2017).

Table 1
Features of Popular BlockChain Platforms and Financial Service Type for Which Adopt Them

	Bitcoin	Ethereum	HyperLedger F.	Corda
Features				
Permissionless	X	X		
Permissioned			X	X
Financial Service types				
Virtual currency	X	X		
Asset property			X	
Contract between Financial Operators				X

Source: Authors' work

Discussion

In the classical centralized scheme, parties performing transactions put their trust in the intermediary. However, this starting hypothesis of unwavering confidence in central entities (and their information systems) is not always clear: can we really trust central entities?

This is the key point that has made BlockChain technology disruptive: it eliminates intermediaries, ensuring the maintenance of trust and even increasing it (Brezo & Bringas, 2012), by making possible to build networks with decentralized validation mechanism in untrusted contexts.

In this section, we want to discuss potential applications of BlockChain platforms for financial services.

We begin with a question: what is the best platform for financial services? The answer is: it depends on the type of service. Hereafter, we discuss some possibilities, summarized in section *Financial Service types* of *Table I*.

- *Virtual Currency*. If the service to provide is based on a virtual currency, like *bitcoin* or *Ether*, the choice is mandatory: they are *Bitcoin* or *Ethereum*, respectively.
- *Asset Property*. Financial institutions exchange assets of various types, such as equities, bonds, and so on. In this context, transparency is a key issue: a platform like *Hyper-Ledger Fabric* could be the right solution.
- *Contract between Financial Operators*. When two or more financial operators sign a contract, this could be managed as a smart contract, in order to ensure transparency. The adoption of a smart contract executed within a BlockChain platform removes duplications and possible inconsistencies of data. In this case, *Corda* could be the best solution, because, among other features, it provides the legal-prose version of agreements and transactions.

Conclusion

In this paper, we presented a brief overview of most popular BlockChain platforms, by introducing their main features as well as by introducing the different ways they support the concept of smart contract. In particular, we discussed their application to realize financial services.

The reader should notice that our contribution is not the same contribution provided by Bouri et al. (2018) and Corbet et al. (2018). In those papers, authors analyse the dynamics of crypto-currencies, by performing an analysis from the perspective of the financial market. In the present work, we focus our attention on technology and platforms, trying to identify a possible match between platforms (on the basis of their characteristics) and some financial applications, so as to help novices. A similar approach is followed by Glaser (2017): platform features are analysed by means of a technological ontology; however, no specific platforms are described, and the analysis remains (in our opinion) too generic.

Furthermore, readers interested in understanding how HyperLedger Fabric works, can refer to Garcia-Bringas et al. (2019).

References

1. Bouri, E., Gupta, R., Roubaud, D. (2019), "Herding behaviour in cryptocurrencies", Finance Research Letters, Vol. 29, pp. 216-221.
2. Brezo, F., Bringas, P. G. (2012), "Issues and risks associated with cryptocurrencies such as bitcoin", in the Proceedings of the Second International Conference on Social Eco-Informatics, Venice, Italy, pp. 20-26.
3. Brown, R. G., Carlyle, J., Grigg, I., Hearn, M. (2016), "Corda: an introduction", R3 CEV.
4. Buterin, V. (2014), "A next-generation smart contract and decentralized application platform", Ethereum White Paper.
5. Clack, C. D., Bakshi, V. A., Braine, L. (2016), "Smart contract templates: foundations, design landscape and research directions".
6. Corbet, S., Meegan, A., Larkin, C., Lucey, B., Yarovaya, L. (2018), "Exploring the dynamic relationships between cryptocurrencies and other financial assets", Economics Letters, Vol. 165, pp. 28-34.
7. Dannen, C. (2017), Introducing Ethereum and Solidity, Apress, Berkeley.
8. Dhillon, V., Metcalf, D., Hooper, M. (2017), "The hyperledger project", in Dhillon, V., Metcalf, D., Hooper, M. (Eds.), Blockchain enabled applications, Springer, pp. 139-149.
9. Fairfield, J. A. (2014), Smart contracts, bitcoin bots, and consumer protection. Washington and Lee Law Review Online, Vol. 71.
10. Garcia-Bringas, P., Pastor, I., Psaila, G. (2019), "Can BlockChain Technology Provide Information Systems with Trusted Database? The case of HyperLedger Fabric", in the Proceedings of International Conference on Flexible Querying Answering Systems FQAS-2019, Amantea, Italy, Springer.
11. Glaser, F. (2017), "Pervasive decentralisation of digital infrastructures: a framework for blockchain enabled system and use case analysis", in the Proceedings of the 50th Hawaii International Conference on System Sciences, Hilton Waikoloa Village, Hawaii, USA, AIS Electronic Library, pp. 1543-1552.
12. Haber, S., Stornetta, W. S. (1990), "How to time-stamp a digital document", in the Proceedings of the Conference on the Theory and Application of Cryptography, Santa Barbara, CA, USA, Springer, pp. 437-455.
13. Korpela, K., Hallikas, J., Dahlberg, T. (2017), "Digital supply chain transformation toward blockchain integration", in the Proceedings of the 50th Hawaii International Conference on System Sciences, Hilton Waikoloa Village, Hawaii, USA, AIS Electronic Library, pp. 4182-4191.
14. Lande, S., Zunino, R. (2018), "SoK: unraveling bitcoin smart contracts", Principles of Security and Trust LNCS 10804.
15. Nakamoto, S. (2008), "Bitcoin: A peer-to-peer electronic cash system".
16. Sousa, J., Bessani, A., Vukolic, M. (2018), "A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform", in the Proceedings of the 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Luxembourg City, Luxembourg, IEEE, pp. 51-58.
17. Szabo, N. (1997), "Formalizing and securing relationships on public networks", First Monday, Vol. 2, No. 9.
18. Valenta, M., Sandner, P. (2017), "Comparison of ethereum, hyperledger fabric and corda", Working Paper, Frankfurt School, Blockchain Center.
19. Wood, G. (2014), "Ethereum: A secure decentralised generalised transaction ledger", Ethereum project yellow paper.

About the authors

Pablo García Bringas is actually working as Vice-Dean of Deusto Engineering, at University of Deusto. He's been dedicated to Research, Technology and Innovation for 20 years, from positions as Head Researcher of DeustoTech Computing - S3Lab, and also as General Director of DeustoTech - Deusto Institute of Technology. García-Bringas has combined teaching (in several BSc, MSc, PhD, and in-company courses) with research since 1998, mainly focused in Data Mining applied to the fields of (a) Information Security, (b) Industrial Processes, and (c) Genomics and Proteomics. He has over 15 years of experience in R&D management, with many projects, journal papers, and PhD supervised dissertations. The author can be contacted at **pablo.garcia.bringas@deusto.es.**

Iker Pastor-Lopez holds a PhD in Computer Science with the highest qualification and with Cum Laude Mention, since 2013. Master in Information Security, since 2010. Computer Engineer, since 2007. Program in Big Data and Business Intelligence, since 2016. He works in the Faculty of Engineering of the University of Deusto, and focuses him scientific interests in the areas of Big Data Analytics, Opinion Mining and Computer Vision. He is the author of several scientific articles in conferences and indexed journals. In addition, he is a member of the scientific committee of several congresses and reviewer of JCR journals. The author can be contacted at **iker.pastor@deusto.es.**

Giuseppe Psaila received his PhD at Politecnico di Torino (Italy). He is researcher and professor at University of Bergamo (Italy). He works on many topics concerning data management, such as data mining, XML processing, query languages and soft computing, information retrieval, Big Data and Open Data. The author can be contacted at **giuseppe.psaila@unibg.it.**