Re-decentralising the Web with Distributed Ledgers

Luis-Daniel Ibáñez Elena Simperl Fabien Gandon Henry Story

Abstract

The Web was originally conceived as decentralised and universal, but during its popularisation, its big value was built upon centralised servers and non universal access. A key element to re-decentralise the Web is to be able to generate trustable, secure and accountable updates among autonomous participants without a central server. We believe that the marriage between Distributed Ledgers and Linked Data can provide this functionality and unlock the true potential of the Web. As a first step towards it, we propose a minimal vocabulary to describe and link Distributed Ledgers.

1 The centralised Web

In its origins, the World Wide Web was conceived as a global network without barriers, where documents stored in remote machines could be instantly available. At the early nineties, this meant that instead of having to physically walk to the organisation's mainframe to consult the phone directory, you could use your terminal and open a connection with the machine storing it. However, the vision of its creators was far beyond this simple use case. They knew that, provided internet connection availability, the concept could achieve planetary scale and enable anyone to share documents, end eventually, any type of *resources* with the rest of the world. For that, two fundamental principles were laid out, first, *decentralisation*, no permission should be needed from a central authority to post anything on the Web, there is no central controlling node, and so no single point of failure. Second, *universality*, for anyone to be able to publish anything on the Web, all the computers involved have to speak the same languages to each other, no matter what different hardware people are using; where they live; or what cultural and political beliefs they have.

The Web became a mass-phenomenon during the late nineties and is now practically ubiquitous, generating great amount of economical value. Unfortunately, part of the commercial success that many companies had with it came thanks to avoiding the principles of decentralisation and universality. A centralised server provides an efficient way of processing data, not only for providing services to clients, but also to derive valuable information or knowledge to open new business opportunities. Having your own API or data format and being able to force others to use it or to go through your central server to perform an action, gives an undeniable competitive advantage. Closed silos of resources became critical assets, with a huge market capitalisation, directly proportional to their size and exclusivity. The effects of the domination of a few companies at this respect can be observed in virtually every aspect of our lives, the economy, and society as a whole, from interpersonal relationships to B2B trading. In e-commerce marketplaces, for example, buyers and sellers have to surrender to the conditions dictated by a few centralised intermediaries, which use their de facto monopolistic position in ways that do not necessarily benefit all marketplace participants. Furthermore, from a data privacy perspective, each of them holds a disproportionate amount of personal information about each individual, threatening their digital sovereignty.

Finally, trust is also an important factor. For example, when two parties make a transaction in a market place, they rely on a trusted central authority to execute the transaction, providing them with guarantees about its validity, successful completion and what to do in case of error. Unfortunately, if this central figure fails or gets compromised, the transaction cannot proceed or will do it wrongly.

Since this issue was identified, many voices have been raised advocating that the Web requires to put back de-centralisation and universality in the Web. To achieve this, we need to provide the technical means to make fully decentralised applications efficient, trustworthy and economically sustainable. The most successful attempt so far to tend bridges among data silos was the Linked Data initiative. The Linked Data initiative laid out a set of principles very similar to those of the original Web, but oriented to data instead of documents, providing a set of standards (universality) and the technological means to integrate and process data stored in remote machines (decentralisation). Unfortunately, even if Linked Data has solved to a certain extent the efficiency problem, it has not yet satisfactorily found a way to implement decentralised trust and to support economically sustainable use cases. This need for a *trust layer* is recognized in the Semantic Web roadmap as a way to achieve desirable properties like accountability, explainability and traceability, but is still an open problem. Fortunately, the irruption of Distributed Ledgers has the potential to change this for good, and propel the re-decentralisation of the Web.

2 What are Distributed Ledgers?

A Distributed Ledger is a linked list of sets of transactions (called *blocks*) between the peers of a network, ordered by time, and where each peer holds a local copy. To add a register to the ledger, a peer needs to sign it using a cryptographic key, guaranteeing integrity and non-repudiation. To further commit the *transaction* representing the addition, someone needs to check that it abides to the particular business rules of the system. The simplest way is to assign the responsibility to a trusted central party, but this opens questions about what happens if peers suddenly lose trust in the central authority, or in case of reaching a bottleneck. In distributed Ledgers, this task is shared among the members of the network by following the result of a voting system, if a certain amount of members consider it valid, then, it is committed to the ledger. However, an attacker able to create several puppet identities in the network, or a subset of members in criminal association, can commit fraudulent transactions in the ledger.

To tackle this problem, and in the scope of Distributed Ledgers for digital currencies, the Bitcoin[4] protocol introduced an innovative idea based on a socio-economic argument, dubbed *Proof-of-Work*: As with real money, where central banks make it difficult to create copies of notes. Bitcoin makes computationally expensive to cast a vote for committing a transaction by attaching an algorithmic puzzle that can only be solved with intensive CPU processing, to encourage peers to invest their CPU resources in validating transactions, every validated transaction is rewarded with a certain amount of bitcoins in exchange for the work done, proved by the solution of the puzzle. The more "validating peers" competing for the rewards (called *miners* in Bitcoin terminology), the more expensive and intractable for attackers to take control of the network. It has been proved that an attacker would require to control 51% of the computational power of the network to be able to inject fraudulent transactions. Researchers and Distributed Ledger systems designers actively look for alternatives to Proof-of-Work, for a survey, see [1]. Note that in the case the network is private, that is, there is certainty that no multiple identities can be forged, other less expensive *consensus algorithms*, extensively studied in the Distributed Systems field, can be used.

Distributed Ledgers can also be used to store executable code, in a somewhat similar fashion to stored procedures in relational database systems or any descendent of procedural attachments in knowledge representation systems. The inheritance of the properties of accountability and decentralisation of ledgers to code execution is a very interesting use case for several domains, as it allows the implementation of *Smart Contracts*. A Smart Contract is a program that encodes contractual clauses and executes them automatically, leaving the trace of its activity in the ledger itself, where it can be verified by all interested parties.

An example from the musical industry is as follows: Bob, Carol and Alice are musicians that recorded a single, they agree that each one will receive a third of the earnings derived from the reproduction of the single in music platforms, they set up a smart contract stating that for every 300 bitcoins (or the cryptocurrency of choice) value of reproductions received, each of them will receive 100. A smart contract is like any other peer in the network in the sense that it can trigger transactions and receive payments. Musical platforms transfer the earnings of the reproductions to the smart contract in cryptocurrency and whenever the received amount reaches 300 bitcoins, the contract triggers three 100 bitcoins transactions to Bob, Carol and Alice. As such, the Smart Contract acts as an automatic custodian of digital assets that enforces contractual clauses in a deterministic, verifiable and secure way.

Cryptocurrencies and Smart Contracts are only the first step towards a much more ambitious goal. Currently, platforms supporting them are restricted to communities looking to solve a specific use case of transaction register, what if we could make available their power to every agent in the Web?

3 Decentralizing the Web with Distributed Ledgers

3.1 Marrying two Different Architectures

The Web thrives for two axis of decentralisation: architectural decentralisation, as mandated by its core principles and achieved through current standards; and application decentralisation, that requires the decentralisation of higher order functionalities. Distributed Ledgers have achieved application decentralisation, but restricted to small to medium communities. How could we marry both worlds to turn the Web into the ultimate Decentralised Autonomous System? We expect that Distributed Ledgers will continue to appear organically to support different communities with different needs of privacy, verifiability and trust, while other communities in the Web will stick to traditional tools. This situation makes the design an acceptance of an universal Distributed Ledger an Utopia. We believe that the most straight-forward path is to use the Web's proven success as an open platform for interoperability, and Linked Data's advancing on the Web to bring together heterogeneous information sources through modular, mixable, and shareable vocabularies to integrate Distributed Ledgers and make them interoperable with themselves and with the Web.

By enabling seamless integration of ledgers via *linking*, agents will be able to choose different distributed ledger platforms based on their affordances and *compose* them. This composition enables complex use cases that are backed up by the composed trust enabled by the underlying distributed ledgers. Data not backed up by other platforms will have a trust score based on state of the art frameworks like (cite), the key difference with distributed ledgers is that they provide mathematical guarantees over their contents, backed up by a whole community instead of a central node. Distributed Ledgers could provide a formal keystone to build the *trust layer* of the Web [3]. Figure 1 illustrates the possibili-

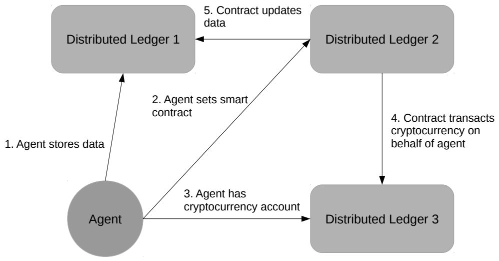
5. Contract updates data
Distributed Ledger 1
Distributed Ledger 2
2. Agent sets smart contract
4. Contract transacts cryptocurrency on behalf of agent
1. Agent stores data
3. Agent has cryptocurrency account
Agent
Distributed Ledger 3

Figure 1: Possibilities of Interlinked Distributed Ledgers

ties of such a framework. An agent can have its data, contracts and cryptocurrency in different Distributed Ledger platforms, by declaring the links between agent's assets in each platform, the contract can execute actions that update agent's data and cryptocurrency balance.

3.2 A Minimal Vocabulary for Linking Ledgers

The first step towards linking Distributed Ledgers is a vocabulary to describe the ledgers themselves. To allow maximum flexibility, we propose the vocabulary depicted in figure 2. It aims at describing the basic classes and properties common to all Ledgers. There are already vocabularies to describe facts about cryptocurrencies like [6] and we expect that many other existing vocabularies will be reused to describe domain-specific relations.

1. A *member* is an identity authorized to have digital assets under its name and trigger transactions in a Ledger. It can be an individual, an organisation or an automated agent. Members have the choice of linking their identities in several Ledgers or using the same identifier among several Ledgers, or to keep them separated. In certain use cases, it can be necessary a link to a legal identity.
2. A *Smart Contract*, executable code that resides in a Ledger. Smart Contract has, minimally, a set of *signatories*, members of the Ledger. In the example of section 2, Bob, Carol, Alice are signatories of the contract. Smart Contracts also have a *definition*, *i.e.*, its code and a *validity*, *i.e.* the time on which they are valid.
3. *Transactions* are *triggered* by a member or a Smart Contract. We leave open to each specific use case the definition of further relationships between .
4. *Blocks*, to which transactions belong to, and are related to one and only one other block through the *previousBlock* relationship.

4 Challenges ahead

In this article, we described a first attempt to use Distributed Ledgers as a mean to re-decentralise the Web. Distributed Ledgers provide a trustable, secure and accountable way of tracking transactions without the need of a central validating authority and could provide the cornerstone to make the Web a truly Decentralised Autonomous System. However, there are still scientific challenges, for example: How to evolve the vocabularies that govern individual Distributed Ledgers with the same desirable property of independence of central authority and protection against multiple identities. How to manage the different approaches that different Distributed Ledgers have concerning levels of privacy, trust and performance from the point of view of an agent or Smart Contract that wants to execute one or more transactions across all of them? Conversely, how to take advantage of this diversity to choose the best combination of Distributed Ledgers for a given use case? We make a call

References

[1] Florian Tschorsch and Björn Scheuermann, *Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies* IEEE Communications Surveys & Tutorials (Volume: 18, Issue: 3, thirdquarter 2016)

[2] Konstantinos Christidis and Michael Devetsikiotis *Blockchains and Smart Contracts for the Internet of Things* IEEE Access (Volume 4, 2016)

[3] Ian Horrocks, Peter F. Patel-Schneider, Sean Bechhofer, and Dmitry Tsarkov. *OWL rules: A proposal and prototype implementation* Web Semantics: Science, Services and Agents on the World Wide Web 3, no. 1 (2005)

[4] Satoshi Nakamoto *Bitcoin: A Peer-to-Peer Electronic Cash System* Technical Report, 2008

[5] Tom Heath and Christian Bizer *Linked Data: Evolving the Web into a Global Data Space (1st edition)*. Synthesis Lectures on the Semantic Web: Theory and Technology, 1:1, 1-136. Morgan & Claypool. 2011

[6] *Description of a Cryptocurrency.* Available at `https://doacc.github.io/`

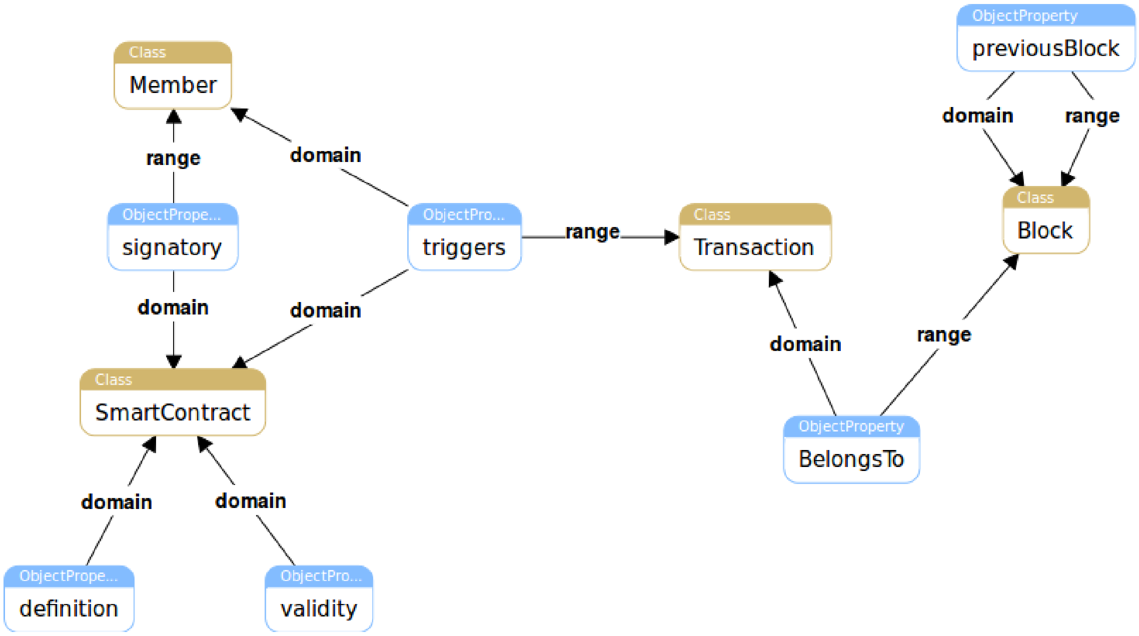

Figure 2: A minimal vocabulary for Distributed Ledgers