Redecentralizing the Web with Distributed Ledgers

Luis-Daniel Ibanez, Elena Simperl, Fabien Gandon, Story Henry

Redecentralizing the Web with Distributed Ledgers

Luis-Daniel Ibáñez and Elena Simperl, *University of Southampton*
Fabien Gandon, *Inria*
Henry Story, *Co-operating Systems*

The World Wide Web was conceived as a global network without barriers, wherein documents stored in remote machines could be instantly available. In the early 1990s, this meant that instead of having to physically walk to the organization's mainframe to consult the phone directory, you could use your terminal and open a connection with the machine storing it. However, the web's creators envisioned far beyond this simple use case. They knew that with Internet connection availability, the concept could achieve planetary scale and enable anyone to share documents—and, eventually, any type of resource—with the rest of the world. To that end, they laid out two fundamental principles. The first was *decentralization*: no permission should be needed from a central authority to post anything on the web; there would be no central controlling node and, therefore, no single point of failure. The second principle was *universality*: for anyone to be able to publish anything on the web, all the computers involved would have to speak the same languages to each other, no matter the differences in users' hardware, location, or cultural and political beliefs.

The web became a mass phenomenon during the late '90s and is now practically ubiquitous, generating a great amount of economic value. Unfortunately, part of the commercial success that many companies had with it came from avoiding the principles of decentralization and universality. A centralized 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. There is an undeniable competitive advantage to having your own API or data format and being able to force others to use it or go through your central server to perform an action. Closed silos of resources became critical assets, with a huge market capitalization, directly proportional to the size and exclusivity of the data assets.

We can observe the effects of a few companies' domination in this respect 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 must surrender to the conditions dictated by a few centralized 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 important. For example, when two parties make a transaction in a marketplace, 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 correctly.

Since this issue was identified, many voices have advocated that decentralization and universality be returned to the web. To achieve this, we need to provide the technical means to make fully decentralized applications efficient, trustworthy, and economically sustainable. The most successful attempt so far to build bridges between data silos was the Linked Data initiative. This initiative laid out a set of principles that were 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 (decentralization). Unfortunately, even if Linked Data has partially solved the efficiency problem, it has not yet satisfactorily found a way to implement decentralized trust and 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 such as accountability, explainability, and traceability, but it is still an open problem. Fortunately, the emergence of distributed ledgers has the potential to change this for good and to propel the redecentralization of the web.

What Are Distributed Ledgers?

A distributed ledger is a linked list of sets of transactions (called *blocks*) between a network's peers, ordered by time, and in which 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 nonrepudiation. To further commit the transaction representing the addition, someone must 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, the network's members share this task by following the result of a voting system; if a certain amount of members consider it valid, it is committed to the ledger. However, an attacker who can 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 protocol introduced an innovative idea based on a socioeconomic argument, dubbed "proof of work."[1] As with real money, where central banks make it difficult to create copies of notes, Bitcoin makes it computationally expensive to cast a vote for committing a transaction, by attaching an algorithmic puzzle that can be solved only 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 puzzle's solution. The more "validating peers" competing for the rewards (called *miners* in Bitcoin's terminology), the more expensive and intractable it is for attackers to take control of the network. An attacker would need to control 51 percent of the network's computational power to be able to inject fraudulent transactions.[2]

> A smart contract is like any other peer in the network in the sense that it can trigger transactions and receive payments.

Researchers and distributed ledger systems designers actively look for alternatives to proof of work (see work by Florian Tschorsch and Björn Scheuermann for a survey[2]). In cases in which 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. Bringing the accountability and decentralization properties of distributed ledgers to program execution is interesting for several domains, because 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 all interested parties can verify it.

Consider an example from the musical industry. Bob, Carol, and Alice are musicians who 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) 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-bitcoin transactions to Bob, Carol, and Alice. As such, the smart contract acts as an automatic custodian of digital assets and enforces contractual clauses in a deterministic, verifiable, and secure way.

Cryptocurrencies and smart contracts are only the first step toward a

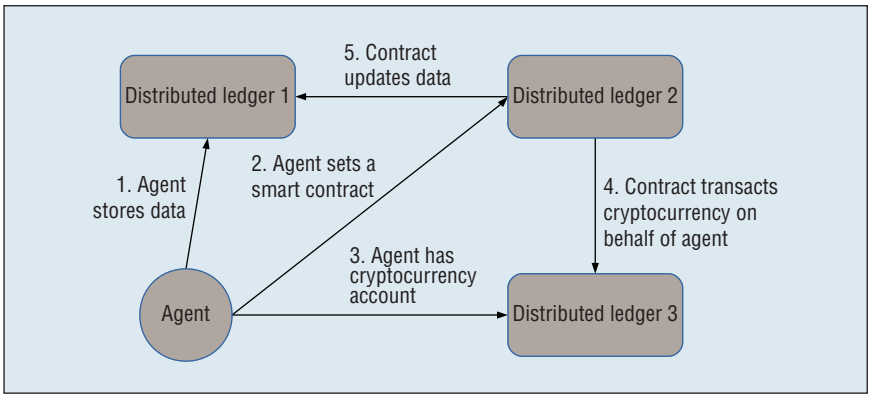

Figure 1. Possibilities of interlinked distributed ledgers. An agent can store a data asset in one ledger, set up a smart contract in a second one, and have cryptocurrency managed by a third one. Enabling links among the agent, data, smart contract, and cryptocurrency enables the smart contract to update data and transact cryptocurrency on the agent's behalf.

much more ambitious goal. Currently, the platforms supporting them are restricted to communities looking to solve a specific transaction register use case—what if we could make available their power to every agent in the web?

Decentralizing the Web with Distributed Ledgers

The web thrives for two axes of decentralization: *architectural decentralization*, as mandated by its core principles and achieved through current standards, and *application decentralization*, which requires the decentralization of higher-order functionalities. Distributed ledgers have achieved application decentralization but only in small to medium communities. How can we marry both worlds to turn the web into the ultimate decentralized autonomous system?

Marrying Two Different Architectures

We expect that distributed ledgers will continue to appear organically to support different communities with different privacy, verifiability, and trust needs, while other communities in the web will stick to traditional tools. This situation makes the design and acceptance of a universal distributed ledger utopic. We believe that the most straightforward path is to use the web's proven success as an open platform for interoperability, and Linked Data's advances on the web,[3] 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,[4] 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 web's *trust layer*.[5]

Figure 1 illustrates the possibilities of such a framework. An agent can have its data, contracts, and cryptocurrency in different distributed ledger platforms; by declaring the links between an agent's assets in each platform, the contract can execute actions that update the agent's data and cryptocurrency balance.

A Minimal Vocabulary for Linking Ledgers

The first step toward linking distributed ledgers is agreeing on a vocabulary to describe the ledgers themselves. To allow maximum flexibility, we propose a vocabulary that aims to describe the basic classes and properties common to all ledgers (see Figure 2). There are already vocabularies to describe facts about cryptocurrencies (see http://doacc.github.io), and we expect that many other existing vocabularies will be reused to describe domain-specific relations.

Our proposed vocabulary comprises several terms. *Member* is an identity authorized to have digital assets under its name and trigger transactions in a ledger. It can be an individual, an organization, or an automated agent. Members can choose to link their identities in several ledgers or use the same identifier among several ledgers, or keep them separated. In certain use cases, it can be necessary for a member to be linked to a legal identity. A *smart contract* is executable code that resides in a ledger. A smart contract has, minimally, a set of *signatories*, which are members of the ledger. In the example given earlier, Bob, Carol, and Alice are signatories of the contract. Smart contracts also have a *definition* (that is, the code) and a *validity* (that is, the time on which they are valid). *Transactions* are triggered by a member or a smart contract. We leave open to each specific use case the definition of further relationships between members and smart contracts. Finally, *blocks*,

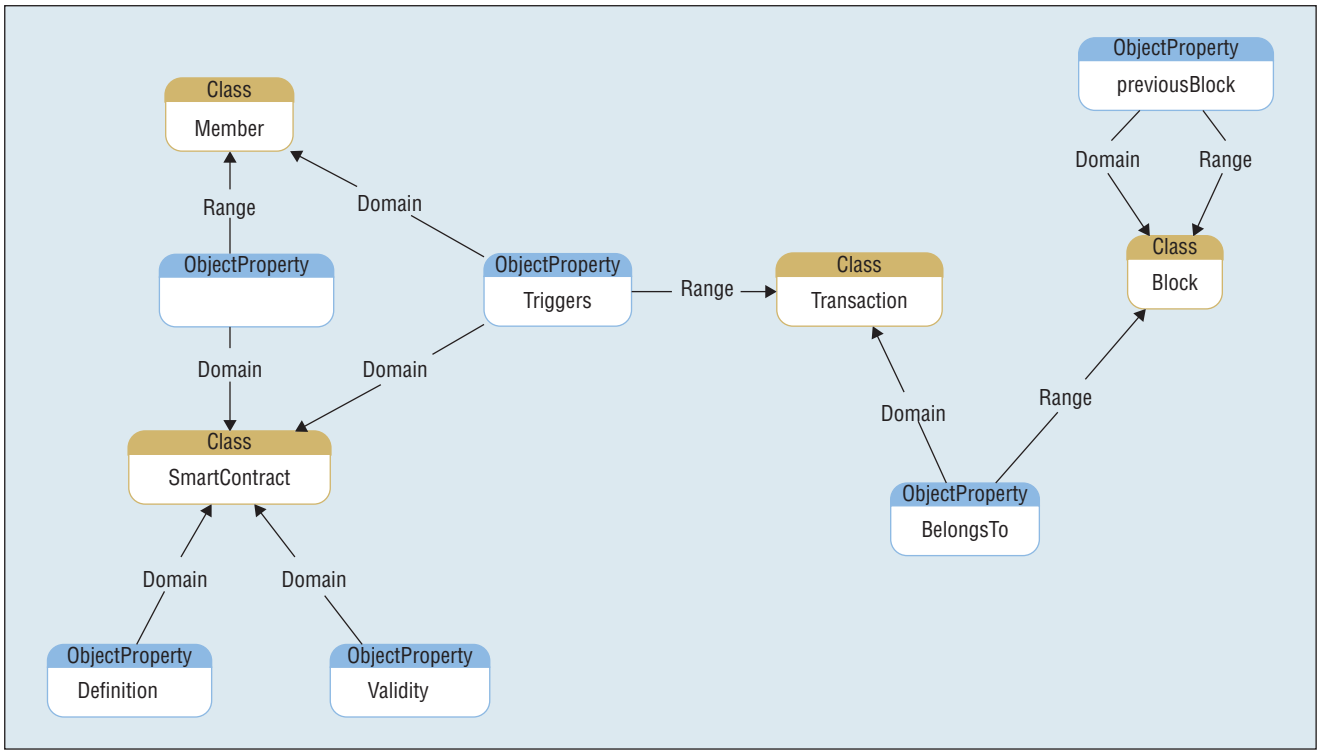

Figure 2. A minimal vocabulary for distributed ledgers. It allows the description of members of distributed ledgers that can trigger transactions and be signatories of smart contracts. Transactions belong to blocks that have an order relationship. Smart contracts have definition (their code) and a validity.

to which transactions belong, are related to one and only one other block through the *previousBlock* relationship.

Distributed ledgers provide a trustworthy, secure, and accountable way of tracking transactions without the need for a central validating authority, and they could provide the cornerstone to make the web a truly decentralized autonomous system. However, scientific challenges still exist—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. Another challenge is how to manage the different distributed ledgers' approaches to 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, another challenge would be how to exploit this diversity to choose the best combination of distributed ledgers for a given use case. ■

References

1. S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, tech. report, 2008.
2. F. Tschorsch and B. Scheuermann, "Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies," *IEEE Comm.*, vol. 18, no. 3, 2016, pp. 2084–2123.
3. T. Heath and C. Bizer, *Linked Data: Evolving the Web into a Global Data Space*, 1st ed., Morgan and Claypool, 2011.
4. J. Golbeck, "Trust on the World Wide Web: A Survey," *Foundations and Trends in Web Science*, vol. 1, no. 2, 2006, pp. 131–197.
5. I. Horrocks et al., "OWL Rules: A Proposal and Prototype Implementation," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 3, no. 1, 2005, pp. 23–40.

Luis-Daniel Ibáñez is a research fellow at the University of Southampton. Contact him at l.d.ibanez@soton.ac.uk.

Elena Simperl is a professor of computer science at the University of Southampton. Contact her at e.simperl@soton.ac.uk.

Fabien Gandon is the research director in Informatics and Computer Science at Inria. Contact him at fabien.gandon@inria.fr.

Henry Story is a cofounder of Co-operating Systems. Contact him at hjs@bblfish.net.