Neuro-Ledger™

Executive Summary

Peter Waher

Waher Data AB, Sockenvägen 42, Saltsjö-Boo, Sweden
peterwaher@hotmail.com

Abstract. This paper describes a new type of Distributed Ledger named the Neuro-Ledger™, that provides a mechanism for resilient and distributed storage without many of the negative aspects associated with more traditional blockchain-based ledger technologies. The Neuro-Ledger™ also provides a mechanism to protect privacy, compliant with the requirements defined in the General Data Protection Regulation.

Keywords: Distributed Ledger Technology, DLT, Persistence, Resilience.

1 Introduction

It has long been known that blockchain technologies raise serious concerns with regards to privacy. The problem has been widely analyzed. The European Parliament has completed a study where they've analyzed if blockchain even can be squared with the GDPR at all, and if so, for what purposes. In their final report [1], they present several aspects of the problem, and severely restricts the use cases in which blockchain technologies can be used, and how. This executive summary will list some of the problems with blockchain, and also how the Neuro-Ledger™ solves these issues.

2 Brief introduction to Blockchain technologies

The goal of a blockchain is to provide a means for resilient and immutable persistence of data and information in an environment of untrusted participants. Information, or data, is packed into blocks that are digitally signed, and added to a repository of blocks. The blocks are identified by a hash digest of the block contents. Each block also points to a previous block, using its hash digest, creating a long chain of blocks, from the latest block, all the way back to the first block. Blocks are then distributed between the machines (called nodes) in an untrusted network using some form of communication protocol.

Since blocks can be generated by many different nodes in the network, it may occur that two nodes create blocks more or less at the same time. This may cause a fork

in the chain. Depending on which of the two blocks other nodes choose to link to, one of the branches "wins" over the other. This selection of branch to build on, is sometimes referred to as a "consensus" algorithm. Simply put, it means the longest of the two branches, as seen by a node, is chosen to continue to build on. The "information" stored in a blockchain, is the information available in the longest chain, from the last block, back to the first block.

Since an untrusted network may contain malicious nodes, a method to dissuade creators of malicious, false or fake blocks is needed. This method typically consists of some form of mathematical problem that requires computational power, and therefore costs money, to solve. This method is called a "proof of work". A common type of proof of work, is for the node creating a new block to be required to add seemingly random data to the block, in such a way that a hash digest of the block, including the random data (called salt), generates a certain number of zeroes. This problem is solved by trying random data until such a digest is found, which statistically requires an expected amount of work to be performed.

3 Desired effects of Blockchain design choices

Two of the most important properties of a blockchain, is resilience and immutability. Since all blocks are identified by a hash digest, any change to the contents of the block will (for any practical purposes) also change the hash digest, and therefore also the identity of the block. To change the contents of a block in a way that retains the identity of the block, is equivalent to generating a hash collision, and is considered practically unfeasable, unless the hash algorithm is cryptographically broken. As long as the hash algorithm is not broken, blocks can be considered immutable, and not possible to manipulate in any practical sense.

Once blocks are created, they are then distributed in the network. Each node receiving a block can validate that the block is valid, according to the principles defined for the network. Once accepted, the node in turn can distribute it on to new nodes, etc. Since the block is distributed across many nodes, the information is resilient to attacks on individual nodes in the network.

As a blockchain is constructed, there is no way to delete a block, as this would destroy the chain, and all information stored prior to the deleted block would be lost. Once created, a block can be considered persisted (at least until the corresponding hash algorithm is broken). Both the resilience of the blocks in the chain, and the immutability of the contents, are desirable properties for ledgers.

4 Side effects of Blockchain design choices

While ledgers based on traditional blockchain technologies are resilient and immutable, at least in the short term, they also have serious side effects, that are directly attributable to the design choices made for the underlying blockchain. This section explores some of these.

Expiry Date

Cryptography and data representation should not be mixed. Good security architectures (for example SASL, TLS, X.509, etc.) provide a means to negotiate cryptographic algorithms, as these are considered temporary, with an unknown expiry date. Data, especially immutable data, must survive changes of cryptographic algorithms over time. By fixing and persisting the cryptographic algorithm with the data, the entire blockchain is imbued with an unknown expiry date. It's like a gigantic "Year-2000" problem, except, you don't know when the problem will occur. Most probably, malicious users will know before system owners will.

Scalability

While a blockchain is distributed, it does not benefit from scalability of content typically attributed to distributed systems. Distribution is only used to solve the resilience and accessibility problems. The more computers that participate in a blockchain network, the more resilient to tampering or loss the data becomes, and the more parties can access the data. But the blockchain cannot store more information by adding new computer nodes. This is a big problem. To store more information, data storage capabilities need to be increased on all nodes participating in the blockchain. Nodes that cannot compete, will not be able to maintain the blockchain and be competed out of existence. Services connected to such a node will fail. In open Internet-based networks, control of content is not possible. Control of the lifetime of a node is therefore not in the hands of the operator of the node itself. The entire blockchain is limited by the node with the least amount of storage capacity.

Performance

As a blockchain is designed, its performance will degrade over time. As new computer hardware is made available, they can execute a proof-of-work quicker than older machines. This requires an adjustment to be made to the proof-of-work algorithm, to make sure it is still as difficult to generate new blocks, as earlier. Otherwise, in time, malicious entities will be able to hijack the blockchain, by injecting vast amounts of new blocks. This will confuse the consensus-algorithm into believing the maliciously injected blocks constitute a valid fork. As proof-of-work becomes more complex, older machines will no longer be able to compete generating new blocks. To stay operational over time, performing the same tasks as before, operators need to constantly upgrade hardware in their blockchain solutions. Performance therefore degrades over time, by design, as technology evolves. New inventions in the computer hardware field, will not benefit blockchain solutions, as they will drive an increase in complexity for the proof-of-work, rather than allow nodes to perform more constructive tasks. The entire blockchain is limited by the slowest node.

Homogeneity

Another consequence of the performance requirements required by proof-of-work is that blockchains only work in homogenous networks. Nodes need to be proportionally strong, or they will not be able to compete. Modern smart-city networks, embedded or IoT networks, require heterogeneous networks, with a wide range of types of nodes. Large, powerful nodes need to co-exist with weak embedded devices that must be able to operate over long periods of time without change. Blockchains are by design unsuited as a cooperative data-layer in such heterogeneous environments.

Longevity

The previous sections describe side-effects that all affect the longevity of a blockchain-based system. Traditionally, if a system works at a given time on a given platform, or hardware, it will continue to do so also over time, given that software patches are regularly administered. As long as the software (including the operating system) is kept up to date, the system will continue to run. With blockchain based systems, this is no longer the case. Not only are software-updates required, but hardware updates also. The only types of companies that can provide such updates in a timely fashion, are large data centers specialized for the purpose, such as cloud providers. They can constantly upgrade hardware. They can securely run a blockchain, at least until the underlying cryptographic algorithm is broken. Once a blockchain solution is implemented, it is very difficult to move it away from the cloud provider. The system is locked in. No wonder all major cloud providers invest heavily into promoting blockchain technologies.

Energy consumption

Due to the need for high performance computing to perform the proof-of-work, a lot of energy is wasted in unproductive work. The proof-of-work does not generate value, so consuming vast amounts of physical energy as a means to protect an otherwise unprotected blockchain seems like a great waste. If the goal is to create future-proof systems that are also sustainable, using blockchain is counter-productive.

Governance

In an open, Internet-based Smart City, a wide range of systems of different capabilities need to cooperate. Since they are operated on different domains by different actors, makes using blockchain very complicated. It raises huge issues with respect to governance. Even though a blockchain is designed to work in a distrusted environment, the need for maintaining nodes current so they can compete implies governance-related problems. Decisions made with respect to access, data storage, performance, etc., must be followed by everyone else. Having a federated organization is virtually impossible. Instead, centralized governance of the infrastructure is required, to assure a homogeneous infrastructure. This is not how open societies and smart

cities are intended to work. Cooperation across domains is very difficult, if the system is based on a blockchain.

Privacy

Apart from technical difficulties with a blockchain, perhaps the gravest problem with such systems, is their incompatibility with privacy principles. This makes blockchains unsuitable for storage of personal data.

By design, blocks cannot be changed. This also means, they cannot be corrected. A blockchain can be amended with new information. But the incorrect information cannot be removed and will always be available. Furthermore, information cannot be deleted in a blockchain, unless the entire tail ending at the block being deleted, is to be deleted also. Since personal data can only be stored for a given time, which is related to the purposes of processing, personal data must be possible to delete.

To reconcile privacy requirements with blockchain deployments, different methods are required to process the personal data "off-chain". This might include obfuscating the data, encrypting the data, storing the keys off-chain, or simply store hashes or links in the blockchain, keeping all the personal data off-chain. But this defeats the original purpose: Maintaining resilient and immutable storage of the data, just for the sake of using blockchain technologies.

Access

A blockchain is designed to operate in a distrusted environment. For this reason, there are no access controls, such as authentication and authorization, to gain access to blocks in the chain by default. Working with confidential or private information using blockchain technology is therefore complicated. Most solutions restrict access to the nodes, thus creating "private blockchains". This defeats many of the original purposes of working with a blockchain in the first place. Furthermore, such controls are necessarily handled outside of the blockchain itself, in a proprietary manner, making interoperation difficult. To have a blockchain-based system with access control interoperate across domains is not possible, unless control is given one of the parties, or a third party (which is contrary to the principles of interoperation).

5 A Ledger for open, smart societies

From the previous sections it should be clear that a blockchain is not suitable for open networks, or any form of network that requires cooperation between entities from different domains, aim to be globally scalable, or process personal information. Instead, a ledger for these use cases must:

- Provide resilient persistence of data for variable amounts of time (but not necessarily indefinitely).

- Distribution must provide scalability of content, as well as accessibility and resilience.
- Allow for corrections and deletions of personal data.
- Provide auditable and traceable records of changes made to data.
- Protect the privacy of data subjects related to stored personal data.
- Retain functionality over time.
- Allow for flexible use of cryptographic algorithms.
- Support heterogenous networks, where weak nodes can participate alongside powerful nodes.
- Use a federated infrastructure that allows for interoperation across domains.
- Access control by default.

6 The Neuro-Ledger™

The Neuro-Ledger™ is a Distributed Ledger that complies with the above-mentioned requirements. It does this by changing the original premise on which blockchains where defined: Instead of assuming a network of distrusted nodes, we assume a network of strongly identified nodes, on which trust relationships are built. This Trust-based approach have several important consequences, as outlined in the following sections.

Communication

Communication is only possible along trust-lines. Trust is established through consent. The communication protocol used, is XMPP, with extensions from IEEE IoT Harmonization P1451.99. XMPP and P1451.99 are federated by nature, which means cooperation across domain boundaries is built in by default.

All entities are strongly identified, both using a network identity, and a cryptographic legal identity. To connect to the network, each node connects to a broker, called a Neuron, on a domain. Brokers interoperate across domains, making it possible for nodes to communicate with each other, given they have consented to do so. Each node is authenticated using SASL and state-of-the-art authentication mechanisms, which is negotiated during authentication. The network identity of each sender is always presented to each receiver. Furthermore, each participant has a cryptographically secured legal identity, that is attested to by the broker operator, or *Trust Provider*, as defined in IEEE P1451.99. The cryptographic algorithms used to protect the legal identity is also negotiable between the entity and Trust Provider and can change over time.

Nodes participating in a Neuro-Ledger™ are all connected to such a federated network, and exchange blocks over this network, in accordance with the built-in consent-based authorization mechanism provided. Instead of basing our ledger on distrust, XMPP and IEEE P1451.99 makes it possible to base it on Trust-based computing instead.

No proof-of-work necessary

Injecting a block into a trust-based network is much more difficult, than to perform a proof-of-work computation in a distrusted network. Proof-of-work is designed to be relatively simple to perform, but expensive enough to avoid frivolously performing. You don't need to impersonate or spoof the identity of another node, in order to inject a block, just compute the proof-of-work.

But if you want to inject blocks into an XMPP/1451.99 network, you can only do so, if you impersonate a node with which other nodes have trusted connections to. This is astronomically much more complex, than to perform a proof-of-work operation. It basically amounts to breaking the credentials of a node, or the cryptographic algorithm used to authenticate the node with the network.

If, through lateral attacks, social engineering, or other methods, a trusted link is established, or network credentials compromised, leading to malicious blocks being injected, these can still be easily identified and ignored. First, blocks are signed with the legal identity of each participant and attested to by each trust provider with a second cryptographic signature, making them doubly difficult to falsify. This provides two additional layers of very strong cryptographic protection. Furthermore, if malicious blocks are injected, they can be clearly identified using the two signatures, and deleted without endangering the entire ledger (see below).

Some of the benefits of not requiring proof-of-work, is that the ledger can stay operational over time, there is no need for competition between nodes; it supports heterogeneous networks of old or weak machines, together with new and powerful machines. There is no need of special hardware to perform proof-of-work in a competitive way.

Global Scalability through Federation

Instead of forcing all nodes to store all blocks and identifying each block by the hash digest of its content, the Neuro-Ledger™ is federated. This means the identity of a block is a hash digest on a client: `64ec88...@client@example.com`. This means the block is accessible to anyone with consent-based authorized access privileges to the client. Each client also signs the block with its legal identity maintained by the Trust Provider hosting the client account (`example.com` in this example), which can be easily validated by anyone, as the Trust Providers publishes the public keys for legal identities in their domain. The hash digest part still assures contents of the block is immutable; any manipulation of the contents of a block will be easily detected. Using federated identities also has the following implications:

- Distribution provides resilience, accessibility, but also limitless scalability of content storage.
- The network can be operated by different domains, which cooperate in a standardized manner. There is no need for third parties to operate the infrastructure.

- Blocks do not need to be stored on all nodes. Instead, they can be stored on multiple trusted nodes, for resilience, and then on all other, by reference or necessity, until used, and on demand.

No chains

Employing a consensus algorithm to establish a chain in a blockchain is a consequence of designing a blockchain for distrusted networks. There is no need for links to the previous block, establishing a chain of blocks in a trust-based environment. Why should one node be affected by unrelated operations taking place on another node? There is no need to compete in a trusted network. Therefore, there is no need for a consensus algorithm, and no need for chains at all. Instead, the Neuro-Ledger™ is just a big distributed set of blocks, which may, or may not be related, based on their contents. Blocks can still be ordered in time, up to clock differences between nodes, as they are time stamped. Not using chains has the following consequences:

- Blocks can be deleted, without affecting information in other blocks.
- A propagation mechanism in the protocol makes sure changes made on the source is propagated across the network.
- Information in a Neuro-Ledger™ can be removed when it is no longer needed, complying with important privacy requirements. Blocks have a Time-to-Live, or TTL, or best-before date.

Updatability

Blocks in a Neuro-Ledger™ can be updated, fulfilling another important privacy-requisite. But it can only be effectively updated by the source that created (and owns) the original block. When a block is updated, a new block is generated, and the contents of the original block is entirely removed and replaced with a link to the new block. A receiver of the updated block immediately detects that the hash digest no longer matches, but can follow the link to the new block, validate its credentials, hash digest, signatures and source, making sure it is the same as the original. Only when all checks have passed, is the update accepted. This method removes the incorrect information from the ledger, making sure only correct information remains. The change is also traceable and can be audited. A changed block can be changed many times, each time, creating a new link to the next, updated link.

Variable block size

Many blockchains have fixed-length blocks. The Neuro-Ledger™ uses variable-sized blocks. This makes it easier for clients to decide when to create blocks. It also avoids forcing nodes to store a lot of empty space, just because nodes were not able to fill blocks in time.

Distributed Object Database

The Neuro-Ledger™ comes with a distributed object database. It integrates with the local object database, with .NET Standard libraries for developers, defined in the IoT Gateway repository[1]. This database is populated locally on the node, as any object database. The libraries use class definitions to persist objects, load objects and search for objects in the object database seamlessly. The Neuro-Ledger™ serializes changes made to the local object database into blocks and distributes them on the network. It also receives blocks from other nodes, deserializing changes and applying them to the local object database. This makes development of distributed services utilizing distributed object databases based on ledger technologies easy.

Privacy Attributes

Privacy attributes in class definitions are used to identify personal data in objects automatically. This makes it easy to automatically create de-personalized information (anonymized, obfuscated, aggregated, etc.) from personal information automatically. This makes it possible to remove personal information from blocks automatically, when distributing them across certain domain boundaries. This makes it possible for a node to distribute one set of blocks on a network where personal information is permitted, and another set of blocks containing anonymized, obfuscated or aggregated information on another network.

7 Summary

Using blockchain-based ledgers is dangerous for many reasons. Very few use cases can safely be implemented using blockchains in the long term. Implementing systems processing personal or sensitive data, or cyber-physical systems, utilizing blockchains, should be avoided, and can even be of national security concern. But ledgers solve an important short-term problem, making their use desirable by system architects and engineers. By changing the axioms on which blockchains where built, a new type of distributed ledger can be built that do not suffer from the same vulnerabilities as a traditional blockchain. It is for this purpose, the Neuro-Ledger™ was designed.

8 References

[1] European Parliamentary Research Service (EPRS), Panel for the Future of Science and Technology, “Blockchain and the General Data Protection Regulation – Can distributed ledgers be squared with European data protection law?”, PE634.445 – July 2019.

[1] The IoT Gateway is not included in the Neuro-Ledger™.