Woleet: Digital Truth Provider

Vincent Barat, Gilles Cadignan
https://woleet.io
White Paper
June 30, 2017

Abstract

It is easy to trust data created in a well-known environment - for example within a company - risks appear when data travel in and out of this natural circle of trust. In order to solve this problem of losing trust, the common solution is to use trusted third parties, or central institutions who will ensure the integrity and authenticity of sensitive data. To achieve this goal, those entities must sometimes produce manual checks or implement centralised cryptography-based solutions. Those are expensive and add frictions and delays to critical business workflows.

Woleet offers a middleware platform enriched by a collection of software applications, leveraging the trustless technology offered by the Bitcoin protocol. Our solutions ensure data integrity and provenance by linking any kind of digital content to immutable Bitcoin transactions. This gives us the opportunity to reinvent business workflows, by offering a continuous level of authenticity over the whole Internet.

1 Introduction

Global volume of digital data exchanged between people or companies is surging to heights never seen before. This is not a tendency on the decline, as more and more applications and more and more participants are connected.

Solutions like digital timestamping, which is still better than manual checks, is expensive and applied only to a subset of very sensitive data like important contracts or financial documents. As a result, most of global data flows are completely excluded from this kind of security solutions. The worst part is that even when you're using those solutions and paying the price, you still entirely rely on central institutions which represent single points of failure.

The apparition of Bitcoin in 2008 introduced a brand-new kind of decentralisation, based on cryptography, game theory and distributed computing. The paper of Satoshi Nakamoto[1] showed us how to exchange value without the need of any central authority.

Value is something very hard to achieve, especially in a 100% virtual environment with no central authority. In order to reach value, you need to trust the underlying technology. Bitcoin, by privileging extreme security, neutrality and immutability gave us the opportunity to explore new ways to address digital trust. Woleet is a platform built on top of the Bitcoin ledger, linking digital content and signature to Bitcoin transactions.

2 System Architecture

2.1 Design principles

Woleet system architecture follows several design principles in order to offer a production-ready middleware that can be used by an existing or a new application to leverage the power of blockchain technology applied to cybersecurity.

2.1.1 Layered architecture

The Woleet architecture is based on our belief that the best way to push forward blockchain technology is to rely on layers. The most basic layer being the actual blockchain that should be intentionally limited for maximum security. We believe that more engineering should be made on top of this core of security (the root of trust) to achieve new features or capabilities. Woleet explore Layer 2 technology via protocols like Chainpoint decoupling the throughput of the original chain and allowing thousands of operations out of one Bitcoin transaction.

2.1.2 Frictionless API enabling multiple applications

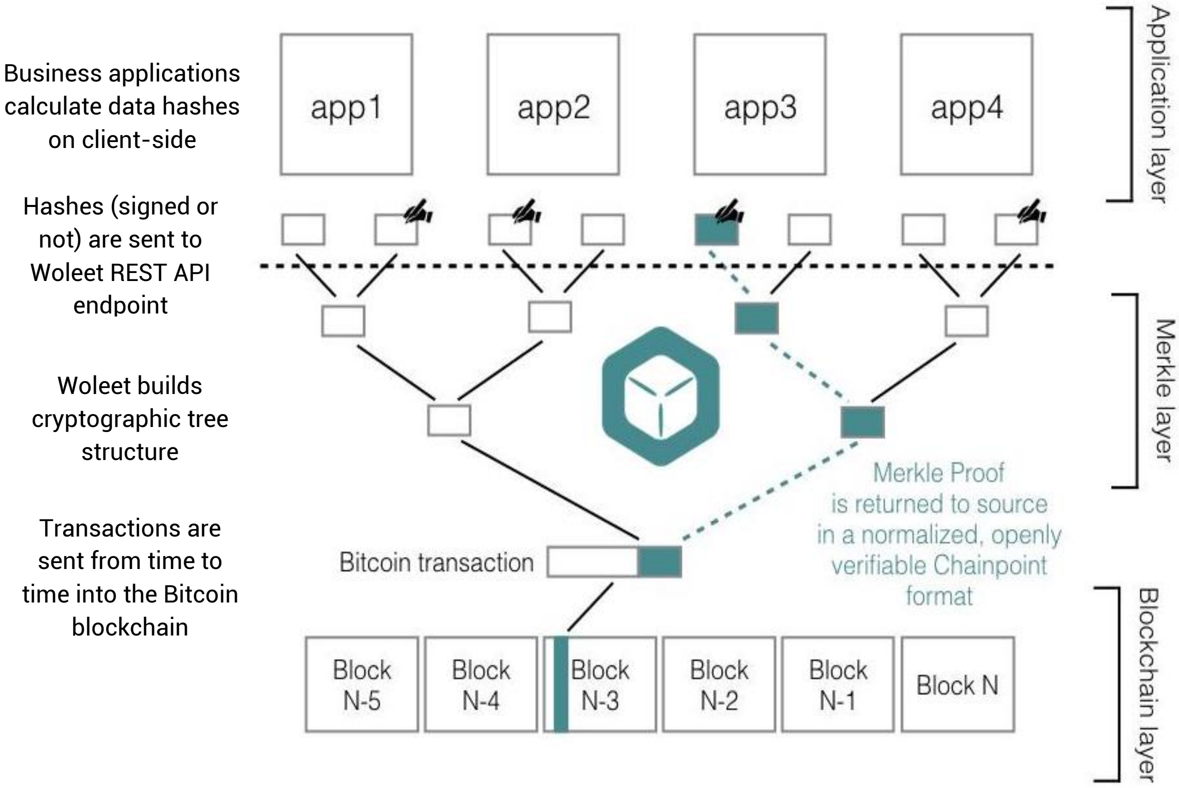
Another strong design principle of the Woleet platform is to minimize frictions for integration. We think that every existing or new system or application should be able to profit from our technology with minimum integration requirements. Even though blockchain marks a real change in the way we think about trust in digital exchanges, we think that IT architects can smoothly evolve to this new paradigm without breaking all the legacy.

2.1.3 Open formats and true disintermediation

Woleet avoids the complexity of the blockchain layer by producing interoperable proofs following open formats. That way, Woleet stays compatible with other systems following those formats and our proofs stays interoperable even if Woleet were to stop operating.

2.2 Global Architecture

The global architecture of Woleet is described as follows:


Business applications calculate data hashes on client-side
Hashes (signed or not) are sent to Woleet REST API endpoint
Woleet builds cryptographic tree structure
Transactions are sent from time to time into the Bitcoin blockchain
app1
app2
app3
app4
Application layer
Merkle layer
Merkle Proof is returned to source in a normalized, openly verifiable Chainpoint format
Bitcoin transaction
Block N-5
Block N-4
Block N-3
Block N-2
Block N-1
Block N
Blockchain layer

2.3 Layer 1: The Bitcoin blockchain

Bitcoin is the most mature blockchain, with more than 8 years of proof of work securing it and an efficient distributed autonomous consensus running it. In the beginning Bitcoin was intended to act as a new form of cash, a peer to peer currency with no central authority. Nowadays the core developers of Bitcoin favouring a list of very important properties making Bitcoin more a store of value than a real currency. The extreme resistance of Bitcoin in a hostile environment makes it perfect to be the ultimate settlement infrastructure. Let's explore those properties and compare them to other kind of blockchains.

1. **Permissionless**: Bitcoin is *permissionless*. That is to say that anybody running a piece of software that implements the consensus rules can plug into the network, send, receive or validate transactions. This is a very important feature as private blockchains and DLT (Distributed Ledger Technology) are not working this way. In Bitcoin or Ethereum the infrastructure is public, just like the Internet. In DLTs, users must support the infrastructure, maintain it, run nodes, miners all by themselves. New participants follow a much more painful process to get in and need permission. In the context of making global, open, international proofs that anybody can verify, a permissionless protocol is mandatory.

2. **Immutability and Censorship Resistance**: Bitcoin is Immutable, Since the third of January 2009, all transactions put in Bitcoin ledger have never been modified or deleted. This is a very important property in our context as we want to use Bitcoin as a witness of digital events, engraved in stone. Other popular public blockchain like Ethereum are not immutable as some transactions have been reversed. If one person or a little group of persons can act by themselves to modify the public ledger, we lose the censorship resistance and the immutability. This is the basis of a timestamping and anchoring system and again Bitcoin so far is respecting this principle.
3. **Security**: Bitcoin has been heavily secured from the very start. The Proof of Work algorithm protect the ledger by the hashing power of the network which must be undone to break into the system. With value attached to each token, Bitcoin has been a big honeypot, constantly attacked by numerous people. Bitcoin has been battle-tested every day since it came into existence. Until today the vulnerabilities of the system are rare and well known by developers who are always trying to find the most secure solution to fix them.
4. **Availability**: Bitcoin's availability rate is 99.99%. This rate is the result of the distribution of the network. You can always request information from the thousands of connected nodes and send transactions.

2.3.1 Managing transactions

Woleet infrastructure sends Bitcoin transactions from time to time to the Bitcoin network. The platform builds standard P2PKH transactions with OP_RETURN outputs only containing Merkle roots and no particular prefix. We use an in-house fee estimation algorithm, combined with full node estimation to reach the best possible fee regarding transactions priority.

2.3.5 Preferred Full Node Software

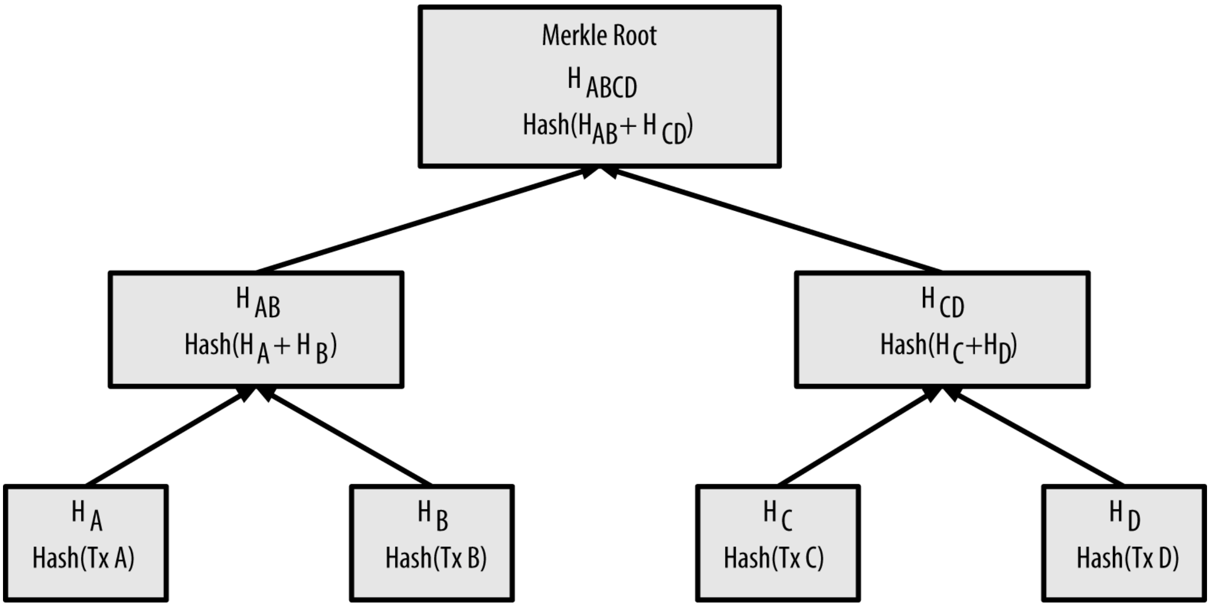
At the time of writing, Woleet is using Bitcoin Core distributions and trust this distribution to deliver solid, tested and peer-reviewed implementation of the Bitcoin protocol. We also estimate that the development process of Bitcoin Core, is open and not centralized in one company or one small group of developers.

2.4 Layer 2: Merkle layer

The Merkle layer of the Woleet platform contains the engine in charge of factorizing all the hashes to anchor them in the underlying blockchain. It acts as a hub capable of large economy of scale by building incrementally a binary tree structure - a Merkle tree - before the settlement of the root in a bitcoin transaction (in the OP_RETURN output)

2.4.1 Address scalability with Merkle trees

Merkle trees are common data structure used to verify integrity of large data files in operating systems. They also are used in Bitcoin for all the transactions included in a block. The schema below extracted from Mastering Bitcoin shows a classical Merkle tree structure.



Here, “H” is a cryptographic hash function (the default hash function used by Woleet is SHA256). This function takes any data and transforms it to a fixed-length digest which is unique for each piece of data. It is technically possible that different data generates the same hash result, but this probability is too low to be considered relevant (less than $1/2^{128}$).

Woleet uses Merkle trees to generate a single Merkle root from data hashes, instead of transaction hashes. A single binary tree can be large enough to contain thousands of data hashes as the leaves of the tree. That’s how we address scalability and get over the scaling limitations of the underlying blockchain.

2.4.2 Storing tree structure and metadata

Woleet platforms saves and backups the produced tree structures in a local database. Each Merkle tree made by the platform is kept in order to ease the generation of the anchoring receipts once Bitcoin transactions are confirmed by the network. It also helps to regenerate the receipt if it is lost by users. We strongly recommend to store anchoring receipts on the client side in order to be completely independent for proof verification.

2.5 Layer 3: Application Layer

Having a simple way to engrave proofs in public blockchain unlocks many possibilities on the application side. Woleet users are empowered to get real control over their data if they clearly identify where trustless technology can help them to overcome their trust issues.

2.5.1 Choosing data to secure

The first step is to identify which data is sensitive and needs verification by their clients, suppliers, auditors, or any other business partners. The granularity is also important as it is possible to secure high volumes without having additional costs.

2.5.2 Confidentiality preserved

If it’s possible to attach some metadata to anchoring requests, the actual content is not provided to the Woleet API. Thus, confidentiality is preserved as the platform works only with data hashes calculated only on the client-side.

2.5.3 Client-side software

Woleet provides a collection of client-side software tools capable of hash calculation and Bitcoin-based signature. As they are all open source, our tools can be adapted to specific needs of vertical use cases.

2.5.4 Easy external verification

Each time an anchor is confirmed by the blockchain, verification is possible by anyone having two pieces of information:

- the file/data to verify
- the anchoring receipt

Verification is then possible with any Chainpoint compatible tool.

3 Data anchoring

The Woleet API creates anchors following to the open source standard Chainpoint. Consequently, they can be verified using any tool compatible with this standard, without any interaction with Woleet, and so remain verifiable forever even if Woleet stops its operations.

3.1 Creating proofs of existence

To create a proof of existence for a specific data, one needs to create what we call an 'anchor'. An anchor is basically a proof of existence creation request. To do so, one only needs to compute the SHA256 hash of this data on the client-side and chose a name for the anchor. Since the platform doesn't need the actual files, there is no limitation to the size or the type of data, other than the time to compute the actual hash.

Newly created anchors are automatically collected by the platform and recorded in the Bitcoin blockchain: this can take from 10 min to a few hours, depending on the load the Bitcoin network and the level of priority of one’s user account. To check the state of anchors, one can pull them using the Woleet API, or associate a call back URL to an anchor that the platform will call whenever the anchor status changes.

Once an anchor is recorded in the Bitcoin blockchain, the user retrieves its associated anchoring receipt using the Woleet API. Anchoring receipts are following the ChainPoint standard (with some extensions for proofs of signature). The proof receipt is the only piece of data required to prove the existence/signature of a file at a given date (obviously the file itself is also required, since it is not included in the anchoring receipt). Thus, it is highly recommended to keep anchoring receipts (and files) in the client own data store, so that the client does not depend on the Woleet API to generate the proof receipt on-demand whenever they want to verify a file.

3.2 Verifying proofs of existence

Verifying a proof of existence using the Woleet API is straightforward: the API takes care of verifying that the proof receipt is valid and correctly anchored in a Bitcoin transaction, so the client just needs to check that the SHA256 hash of the file matches the proof receipt's hash property.
The verification process can also be made without the Woleet API with existing open source code. Woleet provides some examples on its GitHub repository and other compatible implementations exist made by other teams working on the Chainpoint standard.

3.3 About public and private anchors

An anchor can be public (which is the default) or private.

A private anchor is only discoverable by its owner (see the /anchors endpoint). Thus, the owner needs to provide the anchoring receipt along with the data to anyone wanting to verify the proof.

A public anchor is discoverable by anyone knowing the hash of the data (including people with no Woleet account, see the /anchorids endpoint). This allows anyone to retrieve the proof receipt using only the data hash as input, and then to verify it using the Woleet API or any other mean:

- use the /anchorids endpoint to retrieve the anchor identifier by its hash
- use the receipt/{anchorid} endpoint to retrieve the proof receipt (which includes the anchor's metadata).
- use the receipt/verify endpoint (or any other Chainpoint compatible tool) to verify the proof receipt and get the data or signature timestamp.

4 Proof of signature

The Woleet API creates proofs of signature that are an extension of the same standard proposed by Woleet (we are actively involved in the standardization process). Thus, the existence and timestamp of a signature is verifiable using the same tools used to verify proofs of existence.

When it comes to verifying the validity of the signature and the identity of the signatory, some additional processing is performed: since this processing can be fully performed client side with no additional data, proofs of signature remain verifiable forever even if Woleet stops its operations.

4.1 Creating proofs of signature

To create a proof of signature for a file, one also needs to create an anchor, and so to compute the SHA256 hash of the file and chose a name for the anchor, but some additional data is required: the clients' public key (the one associated with the private key used to sign the SHA256 hash of the file) and the clients' signature itself. Optionally, one can provide a

URL allowing to verify your identity by ensuring they own the public key and the TLS certificate of the URL.

4.2 Verifying proofs of signature

Verifying a proof of signature using the Woleet API is also straightforward: the API takes care of verifying that the proof receipt is valid and correctly anchored in a Bitcoin transaction, then verifies the signature, and optionally verifies that the signatory owns the public key and the TLS certificate, so one just needs to check that the SHA256 hash of the file matches the proof receipt's *signedHash* property.

For proof of signature, an additional verification process is performed:

- check that the SHA256 hash of the signature property matches its target_hash property
- check that the signature property is a valid signature of the *signedHash* property for the public key stored in the pubKey property
- additionally, if an *identityURL* property is available:
    - call *identityURL* to make the callee sign some random data using the public key pubKey
    - check that the returned signature is valid
    - get the TLS certificates of the URL (it must be an HTTPS URL) to get insight about the signee identity

5 Use Cases

The Woleet platform has many real-life use cases that can be applied to almost every business when it comes to addressing digital trust. As the time of writing this report, Woleet provides two technical primitives that can be extrapolated to higher level business uses cases:

- **Proof of existence**: Data anchoring using the Woleet platform offers a powerful and affordable way to prove the certain date of existence of any type of data. Having a piece of data anchored in a block can serve as a protection of intellectual property or can be shared as proof of transparency on sensitive data such as financial documents or contracts. The main advantage of using Woleet technology is that the produced proof can be shared with anyone in the world having an Internet access and an open source (and free) software capable of verifying the blockchain receipt.
- **Proof of digital signature**: with its proof of signature technology, Woleet introduce a new technique to deal with digital signature. By using Bitcoin keys for identification, the process gains fluidity and becomes way cheaper than the actual digital signature solutions.

On top of these two primitives, multiple topics can be addressed:

- Intellectual property protection
- Traceability
- Implementation of transparency policies
- KYC

- Identity processes
- Contractualization
- Integrity check

6 Conclusion

Woleet current value proposition is to enter into the new trustless paradigm offered by the Bitcoin protocol. By favouring mature technologies and open standards, Woleet offers a long-term vision, by choosing viable and resilient technical components. One can confidently build solutions on top of our technical stack without worrying about compatibility and sustainability of the whole system infrastructure. Despite the fact trustless cybersecurity is a new technology, Woleet guarantees to its users that it provides the most mature technology available in the blockchain space.

References

1. Bitcoin: A Peer-to-Peer Electronic Cash System: https://bitcoin.org/bitcoin.pdf
2. Rebooting Web of Trust initiative
3. W3C Working group
4. chainpoint.org
5. Opentimestamps
6. Bitcoin Core GitHub repository