

Technische Universität Berlin

Documentation

---

# **IOSL: Distributed Identity Management in the Blockchain**

---

*Authors:*

June 8, 2017

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.1.1	SONIC Project . . . . .	1
1.1.2	Blockchain . . . . .	1
1.1.3	Identity Management . . . . .	2
<b>2</b>	<b>Related Work</b>	<b>3</b>
2.1	Directory Service . . . . .	3
2.2	DHT and P2P Networks . . . . .	3
2.2.1	DHT characteristics . . . . .	3
2.2.2	Functioning Example . . . . .	3
2.2.3	The Network . . . . .	4
2.3	Blockchain Implementations . . . . .	4
2.4	Blockstack Use Case . . . . .	7
2.5	How to Store Social Records . . . . .	8
2.5.1	Solution 1 . . . . .	8
2.5.2	Solution 2 . . . . .	8
	Version 1 . . . . .	8
	Version 2 . . . . .	8
2.5.3	Solution 3 . . . . .	9
2.5.4	Solution 4 . . . . .	9
2.6	Security . . . . .	9
2.6.1	Elliptic Curves . . . . .	9
<b>3</b>	<b>Concept and Design</b>	<b>10</b>
<b>4</b>	<b>Implementation</b>	<b>11</b>
<b>5</b>	<b>Evaluation</b>	<b>12</b>
<b>6</b>	<b>Conclusion</b>	<b>13</b>
<b>A</b>	<b>Code</b>	<b>14</b>
	<b>Bibliography</b>	<b>15</b>

# 1 Introduction

## 1.1 Background

### 1.1.1 SONIC Project

Today's communication happens mostly through Online Social Network (OSN), but with their proliferation, some problems have arisen. The main issue is related to the fact that OSN are built in a closed and proprietary manner (think of Facebook, Twitter or LinkedIn) and don't allow a smooth communication among them. Therefore, the user is forced to create different accounts, and build different networks within for every platform. All these accounts and information correspond to the same user, but are heavily segregated from one another. This segregation is caused by the platform which aims to bind the user with a lock-in effect.

The SONIC project has been developed under suggestion of Telekom Innovation Laboratories to overcome this problem and facilitate a seamless connectivity between different OSN [1] allowing the migration of accounts between different platforms.

The project is supported by a distributed and domain-independent ID management architecture where a GSLS (Global Social Lookup System) is employed to map Global ID to URL of a social profile. This mapping is possible thanks to datasets called Social Records which are digitally signed.

### 1.1.2 Blockchain

Blockchain is a distributed ledger technology used to keep track of Bitcoin cryptocurrency transactions. Distributed ledgers create a data structure – like a chain – where records of every single Bitcoin transaction live. To prevent “double spend,” all Bitcoin transactions are validated and then permanently archived in the cryptographic ledger or chain. The validation is done via a peer-to-peer process that is hugely computer-intensive. It is supported by a global network of volunteers – known as “miners” – who are incentivized mainly by Bitcoin's mining reward.

In essence, Bitcoin uses cryptography to enable participants on the network to update the ledger in a secure way without the need for a central authority. The key to Blockchain was to agree on the order of entries in the ledger. Once this was in place, distributed control of Bitcoin was possible.

### 1.1.3 Identity Management

Add the references

Identity management refers to the process of employing emerging technologies to manage information about the identity of users and control access to company resources [1]. The goal of identity management is to improve productivity and security while lowering costs associated with managing users and their identities, attributes, and credentials.

Everyone is using internet to interact with different digital service platforms. It has many forms like accessing social sites, online shopping services, and interacting with your online banking account. Interactions with these service providers require that each user has digital identity so that user is authorized to access digital platform. There are certain reasons digital platform uses identities and storing all the information related to the user to grow their business and improve user experiences, and defend against certain attacks externally as well as to take care of the privacy of the user.

Identities were used in different ways like accessing personal computer we use only username and password. Digital platforms on the other hand like Facebook, yahoo use domain name with combination of username to differentiate different users because these platforms have many users so domain name can make username uniquely identifiable [2]. Anonymous credentials as another way of providing authentication assertions that don't reveal the user's identity to a service provider for example, U prove technology implemented by Microsoft.

The problem in these digital platforms is the absence of federated directories. Microsoft defines federation as "the technology and business arrangements necessary for the interconnecting of users, applications, and systems. This includes authentication, distributed processing and storage, data sharing, and more."

Federated directories interact and trust each other, thus allowing secure information sharing between applications. Companies are currently running isolated, independent directories that neither interact with nor trust each other. This is a result of applications having their own proprietary identity stores. Each proprietary directory requires its own method of user administration, user provisioning, and user access control. The problem with proprietary identity stores is that users require login for every application, which in turn burdens users with having to remember numerous username and password combinations.

The need for blockchain based identity management is particularly noticeable in the internet age, we have faced identity management challenges since the dawn of the Internet. Prime among them: security, privacy, and usability. Blockchain technology may offer a way to circumvent these problems by delivering a secure solution without the need for a trusted, central authority[3]. It can be used for creating an identity on the blockchain, making it easier to manage for individuals, giving them greater control over their personal information and how they access it.

## 2 Related Work

### 2.1 Directory Service

Also known as name service, maps the names of network resources to their respective network addresses. Each resource on the network is considered an object by the directory server. Information about a particular resource is stored as a collection of attributes associated with that resource or object. One of the most representing example is the DNS, which offers internet domain names translation to ip addresses.

### 2.2 DHT and P2P Networks

A distributed hash table (DHT) is a class of a decentralised distributed system that provides a lookup service similar to a hash table: (key, value) pairs are stored in a DHT, and any participating node can efficiently retrieve the value associated with a given key

#### 2.2.1 DHT characteristics

*Autonomy and decentralization:* multiple nodes, no central coordinator;

*Fault tolerance:* reliable, no single point of failure (like Napster's central server) (Byzantine fault tolerance);

*Scalability:* the system scales. This means it's working efficiently regardless the workload;

*Load balancing* (optional);

*Data integrity* (optional).

#### 2.2.2 Functioning Example

Once these components are in place, a typical use of the DHT for storage and retrieval might proceed as follows. Suppose the keyspace is the set of 160-bit strings. To index a file with given filename and data in the DHT, the SHA-1 hash of filename is generated, producing a 160-bit key  $k$ , and a message  $put(k, data)$  is sent to any node participating in the DHT. The message is forwarded from node to node through the overlay network until it reaches the single node responsible for key  $k$  as specified by the keyspace partitioning. That node then stores the key and the data. Any other client can then retrieve the contents of the file by again hashing filename to produce  $k$  and asking any DHT node to find

the data associated with  $k$  with a message  $\text{get}(k)$ . The message will again be routed through the overlay to the node responsible for  $k$ , which will reply with the stored data.

### 2.2.3 The Network

Each node maintains a set of links to other nodes (its neighbors or routing table). Together, these links form the overlay network. A node picks its neighbors according to a certain structure, called the network's topology.

Aside from routing, there exist many algorithms that exploit the structure of the overlay network for sending a message to all nodes, or a subset of nodes, in a DHT. These algorithms are used by applications to do overlay multicast, range queries, or to collect statistic. (Flooding).

## Implementations Considered

Kademlia

TomP2P

## 2.3 Blockchain Implementations

### Blockchain introduction

Blockchain's distributed ledger system is used to keep track of Bitcoin transactions. Blockchain eliminates the need for central authorities and enables each user of the system to maintain their own copy of the ledger. It also keeps all copies of the ledger synchronized through a consensus algorithm. Bitcoin miners do the recording and validation of the transactions. The miners are necessary to prevent 'double spend.'

### Bitcoin

Blockchain is a distributed ledger technology used to keep track of Bitcoin cryptocurrency transactions. Distributed ledgers create a data structure – like a chain – where records of every single Bitcoin transaction live. To prevent “double spend,” all Bitcoin transactions are validated and then permanently archived in the cryptographic ledger or chain. The validation is done via a peer-to-peer process that is hugely computer-intensive. It is supported by a global network of volunteers – known as “miners” – who are incentivized mainly by Bitcoin's mining reward. In essence, Bitcoin uses cryptography to enable participants on the network to update the ledger in a secure way without the need for a central authority. The key to Blockchain was to agree on the order of entries in the ledger. Once this was in place, distributed control of Bitcoin was possible.

## **Namecoin**

Namecoin is an experimental open-source technology which improves decentralization, security, censorship resistance, privacy, and speed of certain components of the Internet infrastructure such as DNS and identities. Namecoin is a key/value pair registration and transfer system based on the Bitcoin technology. What does Namecoin do under the hood?

- Securely record and transfer arbitrary names (keys).

- Attach a value (data) to the names (up to 520 bytes).

- Transact the digital currency namecoins (NMC).

- Like bitcoins, Namecoin names are difficult to censor or seize.

- Lookups do not generate network traffic (improves privacy).

## **Hyperledger / Hyperledger Fabric**

Hyperledger is an open source collaborative effort created to advance cross-industry blockchain technologies. It is a global collaboration, hosted by The Linux Foundation, including leaders in finance, banking, Internet of Things, supply chains, manufacturing and Technology. Hyperledger Fabric is a business blockchain framework hosted on Hyperledger intended as a foundation for developing blockchain applications or solutions with a modular architecture. Hyperledger Fabric allows components such as consensus and membership services to be plug-and-play. Hyperledger Fabric establishes trust, transparency, and accountability based on the following principles:

- Permissioned network – Provides collectively defined membership and access rights within your business network

- Confidential transactions – Gives businesses the flexibility and security to make transactions visible to select parties with the correct encryption keys

- No cryptocurrency – Does not require mining and expensive computations to assure transactions

- Programmable – Leverage the embedded logic in smart contracts to automate business processes across your network

## **Ehtereum**

Ethereum is a decentralized platform that runs smart contracts: applications that run exactly as programmed without any possibility of downtime, censorship, fraud or third party interference. These apps run on a custom built blockchain, an enormously powerful shared global infrastructure that can move value around and represent the ownership of property. This enables developers to create markets, store registries of debts or promises, move funds in accordance with instructions given long in the past (like a will or a futures contract) and many other things that have not been invented yet, all without a middle man or counterparty

risk. On traditional server architectures, every application has to set up its own servers that run their own code in isolated silos, making sharing of data hard. If a single app is compromised or goes offline, many users and other apps are affected. On a blockchain, anyone can set up a node that replicates the necessary data for all nodes to reach an agreement and be compensated by users and app developers. This allows user data to remain private and apps to be decentralized like the Internet was supposed to work.

## **Storj**

Storj is a peer-to-peer cloud storage network implementing client-side encryption would allow users to transfer and share data without reliance on a third party storage provider. The removal of central controls would mitigate most traditional data failures and outages, as well as significantly increase security, privacy, and data control. Peer-to-peer networks are generally unfeasible for production storage systems, as data availability is a function of popularity, rather than utility.

## **Swarm**

Swarm is a distributed storage platform and content distribution service, a native base layer service of the ethereum web 3 stack. The primary objective of Swarm is to provide a decentralized and redundant store of Ethereum's public record, in particular to store and distribute dapp code and data as well as block chain data. From the end user's perspective, Swarm is not that different from WWW, except that uploads are not to a specific server. The objective is to peer-to-peer storage and serving solution that is DDOS-resistant, zero-downtime, fault-tolerant and censorship-resistant as well as self-sustaining due to a built-in incentive system which uses peer to peer accounting and allows trading resources for payment. Swarm is designed to deeply integrate with the devp2p multiprotocol network layer of Ethereum as well as with the Ethereum blockchain for domain name resolution, service payments and content availability insurance.

## **IPFS**

The InterPlanetary File System (IPFS) is a peer-to-peer distributed file system that seeks to connect all computing devices with the same system of files. In some ways, IPFS is similar to the Web, but IPFS could be seen as a single BitTorrent swarm, exchanging objects within one Git repository. In other words, IPFS provides a high throughput content-addressed block storage model, with content-addressed hyper links. This forms a generalized Merkle DAG, a data structure upon which one can build versioned file systems, blockchains, and even a Permanent Web. IPFS combines a distributed hashtable, an incentivized block exchange, and a self-certifying namespace. IPFS has no single point of failure, and nodes do not need to trust each other.



## Filecoin

Filecoin is a distributed electronic currency similar to Bitcoin. Unlike Bitcoin's computation-only proof-of-work, Filecoin's proof-of-work function includes a proof-of-retrievability component, which requires nodes to prove they store a particular file. The Filecoin network forms an entirely distributed file storage system, whose nodes are incentivized to store as much of the entire network's data as they can. The currency is awarded for storing files, and is transferred in transactions, as in Bitcoin. Files are added to the network by spending currency. This produces strong monetary incentives for individuals to join and work for the network. In the course of ordinary operation of the Filecoin network, nodes contribute useful work in the form of storage and distribution of valuable data.

## 2.4 Blockstack Use Case

Blockstack is a particular implementation of a decentralized DNS system based on blockchain. It combines DNS functionality with public key infrastructure and is primarily meant to be used by new blockchain applications.

According to the company: "under the hood, Blockstack provides a decentralized domain name system (DNS), decentralized public key distribution system, and registry for apps and user identities" [add reference].

The real breakthrough is the architecture the system is built on. It can be described as a three-layer design with the blockchain as the first and lower tier, the storage system as the upper and the peer network as middle layer.

The architecture is shown in ??.

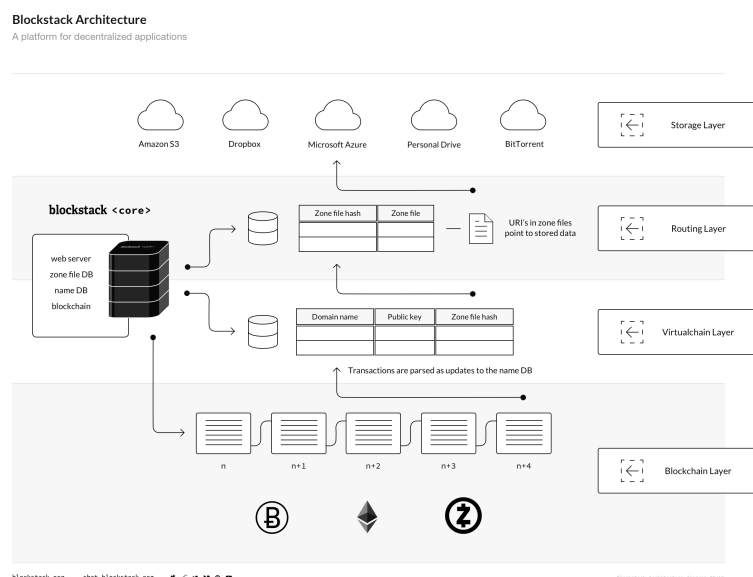


Figure 2.1: Basic Blockstack architecture

## 2.5 How to Store Social Records

### 2.5.1 Solution 1

The most intuitive solution would be to switch from the DHT table to the Blockchain, moving each social record through a transaction. Unfortunately after few considerations, this approach does not seem to be optimal:

- Store the entire social record would occupy a high amount of memory in the distributed node. In particular the limit for the data structure is 44kb, but is best practice to keep the records light.
- Check the real identity of the uploader would be impossible, one option could be to allow only a specific wallet address to write into it.
- Look up would not be efficient.

### 2.5.2 Solution 2

#### Version 1

This solution is inspired by Blockstack architecture and requires to create a DHT to map the GlobalID to the address of the transaction contained inside the blockchain.

The transaction holds in the data field the hash of the Social Record. In this manner, the blockchain has the role of validation infrastructure rather than storage system. The actual data is stored in the distributed hash table. This solution could be vulnerable in case a malicious node tries to restore an old version of the social record by injecting the wrong address in the DHT table, causing a corrupted record.

<< Not clear >>

Another vulnerability could be in the consistency of the data, some node could be inconsistent if the DHT doesn't get updated. Therefore, the version control should be handled by the blockchain. Most of the vulnerabilities depend on the fact that a transaction stored in the blockchain has a different address every time we update the information contained within the Social Record.

#### Version 2

The third solution is based on the second one but requires each user to hold an Ethereum Wallet. In such design each transaction contains both the Wallet public key and the hash of the Social Record within the data field. Every time the Social Record gets updated, its hash changes and the new transaction will have a different address.

The idea is to use the wallet public key as id, and map the GlobalID with the Wallet ID which will remain always the same. Searching inside the blockchain looking for the ID, the ledger will return the latest version. Compared to the second solution, the former provides version check through the blockchain rather than DHT.

### 2.5.3 Solution 3

Blockstack offers an already implemented solution using a layered architecture similar to the one proposed in solution 2. It allows also to choose among different blockchain implementations and storage systems. Nevertheless, the underlying idea is the same as solution 2: use the blockchain as a validation system and use a DHT to map ids (keys/GlobalIDs) with the corresponding values (Social Records).

According to Blockstack, the applications built on this system are server-less and decentralized. Developers start by building a single-page application in Javascript. Then, instead of plugging the frontend into a centralized API, they plug into an API run by the user. Developers install a library called 'blockstack.js' and do not have to worry about running servers, maintaining databases, or building out user management systems [add reference].

### 2.5.4 Solution 4

Trying to simplify the complexity of the process, it could be possible to have in the contract a *key-value* array where the key is the GlobalId and the value is the hash already presented in the solution 2.

Given the fact that the GSLS is an open source not centrally managed server, a safety threat may arise. In order to ask to the server to conclude a particular transaction, a password is required (otherwise a malicious user could update the social record on behalf of someone else), but of course in case of a malicious node the password could be sniffed.

In order to solve this problem, the following approaches are suggested:

.

## 2.6 Security

### 2.6.1 Elliptic Curves

ECC is the next generation of public key cryptography, and based on currently understood mathematics, it provides a significantly more secure foundation than first-generation public key cryptography systems like RSA.

With ECC, you can use smaller keys to get the same levels of security. Small keys are important, especially in a world where more and more cryptography is done on less powerful devices like mobile phones. While multiplying two prime numbers together is easier than factoring the product into its component parts, when the prime numbers start to get very long, even just the multiplication step can take some time on a low powered device. While you could likely continue to keep RSA secure by increasing the key length, that comes with a cost of slower cryptographic performance on the client. ECC appears to offer a better tradeoff: high security with short, fast keys.

## **3 Concept and Design**

## 4 Implementation

## 5 Evaluation

## 6 Conclusion

# A Code



# Bibliography

- [1] Sebastian Göndör and Hussam Hebbou. “SONIC: Towards seamless interaction in heterogeneous distributed OSN ecosystems”. In: *Wireless and Mobile Computing, Networking and Communications (WiMob), 2014 IEEE 10th International Conference on*. IEEE. 2014, pp. 407–412.