18.6.2016

Masterthesis Proposal

Decentralized Document Management

Supervisor: Prof. Dr. Stefan Traub
Student: Emmanuel Schwartz

Problem statement:

In the past few years it became common practice to distribute documents like invoices or other business papers electronically. Documents are sent via e-mail or offered for download on a supplier's web portal. This approach by no means prevents the necessity to deliver a hardcopy to the customer via traditional post service, but it is obvious that this approach saves cost only for the supplier and puts the burden to download and particularly archive the documents to the customer (private person).

Several problems can be identified:

- E-Mail is not reliable, even though providers are improving on this topic.
- E-Mail is normally not encrypted. The problem is not, that e-mail cannot be encrypted, but most private persons are not accustomed using cryptography.
- It's very annoying to log in to a company's web portal and manually download the invoices. Every portal is different and a lot of digital identities are necessary.
- There is no proof, that a customer received a document.
- A private customer does not have an idea of how to save and sort all these documents on their hard drive.
- A private customer does not do backups, even though everybody knows about the necessity of backup, a backup is only done after the loss of the first and important documents.
- A private pc is not save against malware attacks and theft.
- The documents cannot be accessed on any computer other than the customer's computer unless he stores the files on the cloud.

Upcoming technologies which might help to solve these problems:

Actually there are some interesting technologies emerging, which could help to solve these problems:

1. IPFS (https://ipfs.io)
   The idea of ipfs is mainly to access files not using a url and hardcoded filename but rather directly by its content represented by a documents hash value. Documents are not stored within a central server but rather by a distributed hash table. Currently it is used mainly for public unencrypted documents.
2. STORJ (https://storj.io)
   storj on the other hand uses local encryption for each file and stores it within a DHT which is made up of thousands of private or public computers. The storage for storj is not free, but anybody can rent its own harddisk.
3. Ethereum (https://www.ethereum.org)
   Ethereum is one of several implementations of a blockchain. Blockchain became famous after

the invent of cryptocurrency bitcoin. Ethereum takes this approach a little bit further and allows anybody to establish what's known as a smart-contract. This essentially is a small program which runs alongside the blockchain implementation and can implement any possible logic.

Initial idea

How can these technologies help us to solve the problems for our decentralized document management? Assume a vendor wants to send an invoice to its private customer. The following sequence of events might apply:

1. The invoice is stored on one of the DHT implementation like IPFS or STORJ. In the case of ipfs it must first be encrypted. The advantage is:
    a. No need for one single cloudprovider.
    b. The document is addressed by content and not by location.
2. The vendor adds a transaction to the blockchain. This transaction keeps a link to the sent document. The blockchain transaction provides the evidence, that the document has been sent and keeps a permanent not alterable link to the document. The document itself can also not be modified, otherwise the hash would become invalid.
3. If a customer wants to access his document, the only thing what he needs to know is his private key. This enables him to access the document on almost any places where a client-software is installed. This can also be a web application.

Questions needed to be solved

The master thesis should implement a proof of concept (poc) and give answers to the following questions:

- What is a possible architecture for our poc.
- What are the advantages and limitations?
- How can additional metadata be included?
- How can a document be signed within the proposed architecture?
- How about printing a document while preserving the signature and hash?
- How can the proposed architecture combined with traditional cloud systems.

Timeline

The following timeline seems to be suitable (roughly on month each):

1. Define and extend the problem statement. Look for currently available possible solutions (research topics)
2. Learn about the technologies. Install a demo network.
3. Define initial ideas for the solution. Try to implement special topics.
4. Describe the final architecture.
5. Implement the POC.
6. Write a paper and the master-thesis.