Apostille

A blockchain notarization and timestamping service with transferable, updatable, branded, and conjointly owned notarizations

Authors: J. McDonald, Assistant Professor, Keimyung University
E-mail: jeff@nem.io
J. Oliverio, NanoWallet Developer
Email: oliverio.j@outlook.com

January 15, 2017
v.1.1

Abstract: Many timestamping services exist in one form or another in the blockchain sphere, but most of these are simple first generation systems for hashing a document to get its fingerprint and then timestamping that fingerprint in the blockchain with a transaction. These are one-off fixed transactions that have their reason to be culminating in being included in a block so that later they may be audited. These 1.0 timestamps are not updatable, cannot be transferred from one owner to another, and cannot have additional value and assets attached to the timestamp. Apostille, as the first 2.0 blockchain notarization service, solves these problems and offers a new set of features and business opportunities.

The Apostille system's key innovation is taking data which represents an underlying person, place, or thing and giving that data its own private key derived deterministically from its contents. This is used to create an account that represents the state of that object. These are dynamic accounts that can be used to prove authenticity, show ownership, and record updates, as well as be branded, transferable and conjointly owned.

Table of Contents

1. Introduction

The idea of timestamping documents to prove they existed and were in one's possession has been around since long before the blockchain, but with the invention of Bitcoin, many have seen the blockchain as a natural fit for this service taking advantage of the blockchain's usefulness as an open and auditable timestamped ledger.

Timestamping the hashed fingerprint of a document into the blockchain has come to be known as proof-of-existence (PoE). One of the first and most recognized services is proofofexistence.com, which is an open source website started in 2012. Users can drag and drop their document and get a hash representing the document's digital fingerprint that can then be stamped in the Bitcoin blockchain. Since PoE's start many other services have arisen, some with small improvements or cheaper fees have emerged, but these more or less rely on the same formula. These are one-time timestamping processes where a person uploads a document and stamps it in a blockchain. That is the limit of their utility.

Outside of blockchain technology, traditionally, legal notaries have verified and stamped paper documents to notarize their authenticity. These notaries gain their authority because they are registered and backed by the governments of the world. But since the invention of the blockchain, one can think of timestamping fingerprints of documents in the blockchain as a type of "blockchain notarization"; one not back by political authority, but instead by the blockchain's decentralized network and underlying technology.

The existing legal certification and notary systems backed by the governments and traditional organizations of the world offer many use cases for verifying an object's status. These include certificates of property ownership (e.g. car titles), certificates of achievement (e.g. diplomas, life event certificates), verification of witness (e.g., notary public), verification of a product quality (e.g. "Grade A" certification), etc.

These existing systems offer qualities to their certificates like being verifiable, witnessed, quality controlled by 2nd and 3rd parties, to be registered with official organizations, transferability, updatability, etc.

The Apostille system looks to add these properties which only previously existed outside of blockchains, to within a blockchain ecosystem. This is the logical next step forward in blockchain technology and is where the idea of transferable value within a blockchain system, e.g. cryptocurrency, and timestamping documents combine to bring the best features of both into one system.

Apostille seeks to do this by taking advantage of a naming service, multisignature accounts, messages, and blockchain assets. These can be found on a blockchain like Bitcoin but will need to incorporate different integrations and APIs of various projects, some of which are centralized; projects like Onename for the naming service, Bitpay for multisig, CounterParty and Colored Coins for assets, and the OP_RETURN field of a Bitcoin transaction for messaging. Alternatively, a better solution is for one to explore the option of Ethereum and write a smart contract attempting to replicate these features, but this too could prove to be challenging due the the complexity and risks in writing smart contracts. Instead, one 2.0 blockchain, NEM, offers all of these services by default with one universal API, so we will discuss using it as an example.

The name “Apostille” comes from the French word of the same spelling meaning *to certify, authenticate, or complete*. The French word derives its meaning from the Latin use of “post” meaning *after*, and the use of “illa” meaning *these*, and “verba” meaning *words*. At its root then we can say it means *after these words*.

The term was popularized during the Apostille Convention held under the Hague Convention in 1961 and signed by 112 countries making a system for international notarizations. The NEM blockchain is an international and decentralized technology and as so brings a new twist to the idea of an apostille. NEM Apostille transactions are not traditional legal notarizations endorsed and enforced by political entities and treaties, but instead are “blockchain notarizations” made on an internationally supported blockchain secured and enforced by computer code and cryptography.

2. NEM Features

The NEM blockchain is a 2.0 platform written from scratch. It was designed by professional enterprise-level developers to be a reboot of what an ideal blockchain should be. Because of this, how NEM works, and to fully understand how Apostille works, it is important to introduce NEM technology, which is in many ways fundamentally different from other blockchains. A more in-depth technical explanation of NEM technology can be found in the Technical Report.

2.1. Namespaces

Namespaces on the NEM system is a domain naming system, but one both like and unlike that of the internet. There are unique root-level domains and non-unique subdomains, typically used to classify fully qualified unique assets or naming systems. This allows one person with one unique root domain to create many different subdomains for their various projects or outside business accounts. It also helps to build and maintain a reputation system for services built on registered names. One such example is the blockchain supported NEM digital asset feature, named Mosaics, but others could be any third-party distributed naming systems an app builder would like to make.

This is useful in the Apostille system because it creates a system of authority and power and now a user can trust a blockchain notarization made from a legitimate and registered company. NEM's namespaces enable, for instance, one to own the namespace "foo company" and now no other person can claim that root domain. Blockchain notarizations published from that namespace can be trusted to have come from the real "foo company". This is useful, for example, with luxury good makers making certificates of authenticity on the blockchain for luxury items. Now, the certificate can be trusted because the manufacturer of the certificate can be clearly known from things like publishing their namespace on their website and packaging information. It is also useful for things like governments registering citizen identification on the blockchain. If those registrations are approved of by the official and unique namespace domain of that country, one can think of it as an officially endorsed blockchain ID.

2.2. Mosaics - Assets on the NEM Blockchain

Mosaics in NEM are essentially named assets inherent in the NEM blockchain, and not on a 2nd party layer. They can represent any kind of asset that a company, such as “foo company”, would like to issue. They have customizable names, descriptions, divisibility, quantities as either fixed or mutable, and transferability restrictions if necessary, and can have levies applied to them or be levies themselves on other mosaics.

So not only can “foo company” make notarizations but can actually attach any kind of asset to the blockchain notarization they might wish. Governments might want to make a “taxes paid” or “registered citizen” mosaic asset. Private companies might want to make a “good for a redeemable amount of money” or “share of company” asset and pair it with a notarized contract.

This is useful in the Apostille system because now many different third parties can customize, brand, and attach mosaic assets that are digital representations of value or status to a blockchain notarization.

2.3. Messages

Messages in NEM come in three flavors: open, encrypted, and hex. They can be of any length up to 320 characters (272 encrypted), and messages can be strung together if needed.

This is useful in Apostille because after a blockchain notarization is made, messages on the blockchain can record updates and augmentations to the file and the person, place, or thing it represents. Additionally, they could be used as plain text descriptions, or links to additional information about the notarized content. These may need to be public, or private, or possibly written in hex as part of the backend of an application.

2.4. Multisig Contracts

Multisignature and multiuser accounts play an important and critical role in the Apostille system. Although other blockchain’s multisignature solutions resemble NEM’s multisignature accounts on the surface, there are subtle differences that set NEM apart. It is some of these differences that allow us to easily create the Apostille solution on NEM.

NEM’s multisig works by on-chain contracts. Unlike other blockchain solutions, the multisig account is not created by combining public keys from other accounts. Instead, a pre-existing and funded address is converted into a multisignature account and the cosignatories are assigned to it. The cosignatories can be assigned in any m-of-n combination where both the *m* and *n* can be any number of 1-32; this includes 1-of-1 multisignature contracts which are important in Apostille.

A 1-of-1 multisignature account is possible in NEM as the account being turned into a multisig account has its private key nullified; meaning, it no longer has any power to initiate transactions. Only a cosignatory’s private key can initiate transactions on the multisigged account’s behalf. It is therefore that accounts in NEM’s multisig implementation can be analogously thought of as parent/child accounts, where the parent accounts are the cosignatories and can make the child account make any transaction.

This is useful in the Apostille system because now a dedicated account representing a blockchain notarization can receive messages to update the blockchain notarization, and can receive assets/mosaics sent to the dedicated account locking in value or adding status to the blockchain notarization. But the other subtle yet critical utility about NEM’s multisignature solution is that, the dedicated blockchain notarization account be transferred from person to person. This means that the account is no longer “just an account” but instead is a certification account representing the state of the underlying content it is linked. We call these certification accounts “Apostille accounts”.

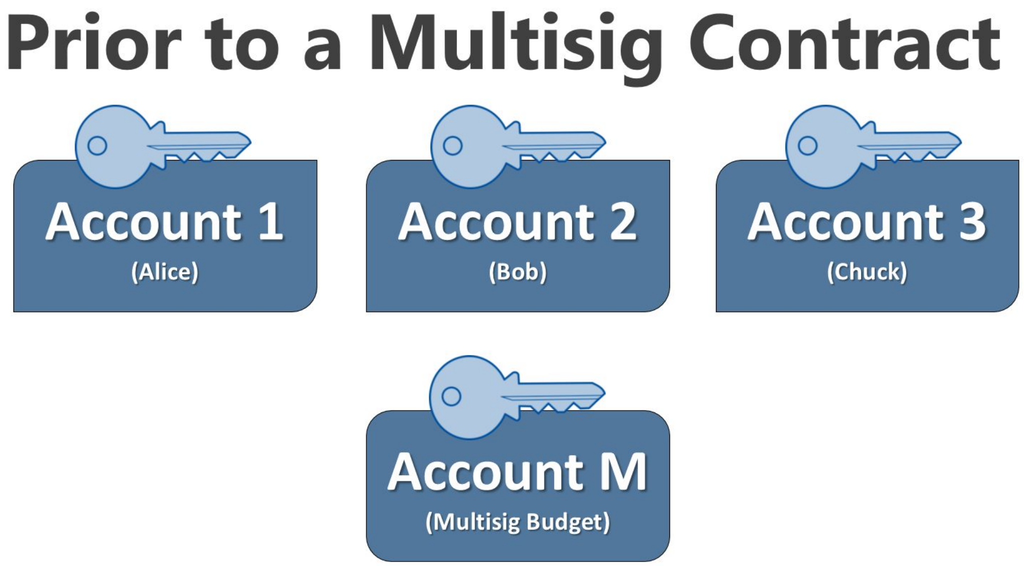

Figure 1. The accounts for Alice, Bob, Chuck and Account M are all each separate and each is controlled by its own private key.

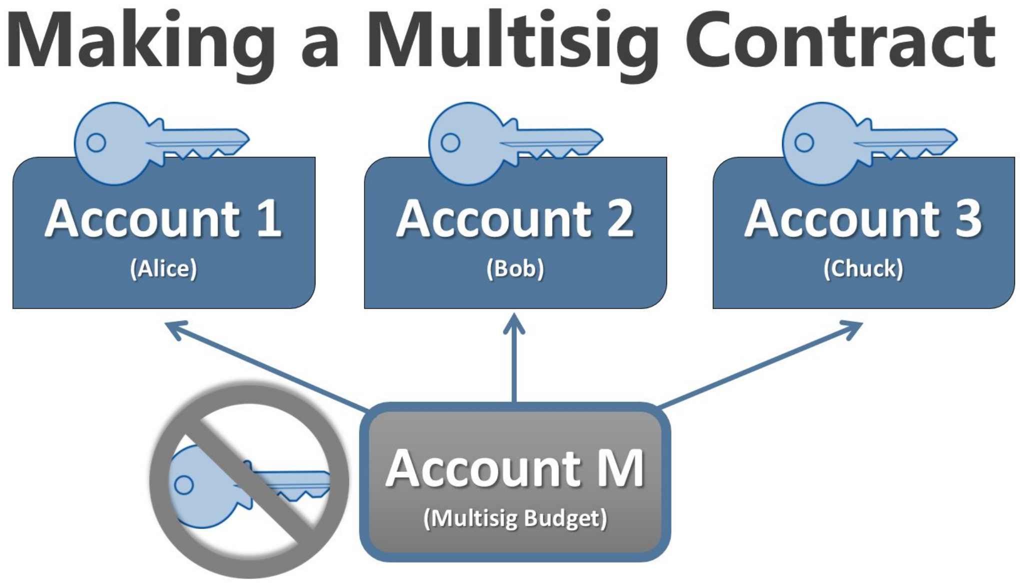

Figure 2. Making a multisig contract. The private key of the multisignature account M, is no longer important and not used. Meanwhile, Alice, Bob, and Chuck have custodial control over Account M.

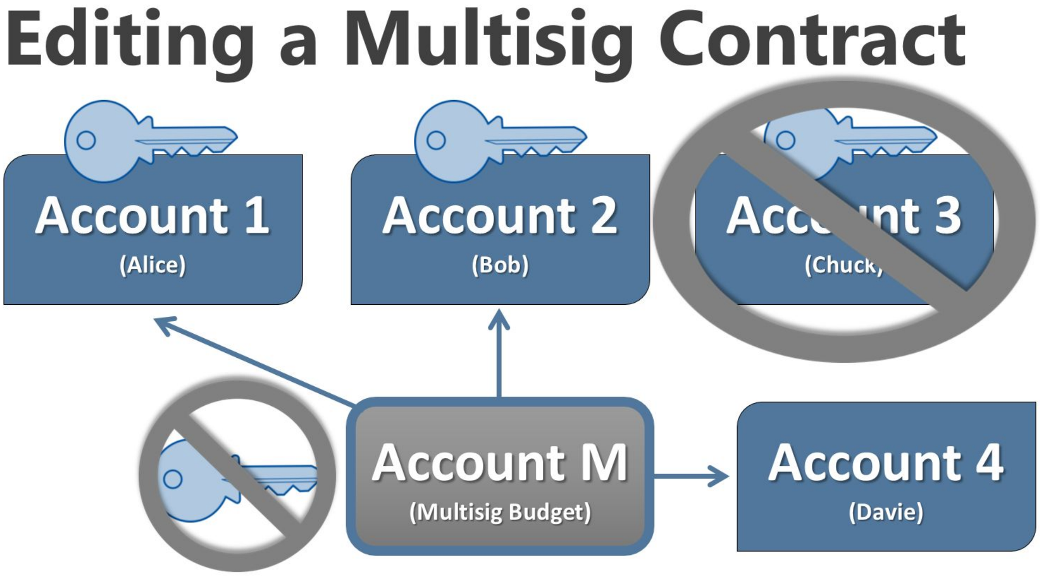

Figure 3. Editing a multisig contract. With only a few clicks, Chuck's account is removed and Davie's account is added.

3. The Apostille System

As one can gather from reading this paper thus far Apostille takes advantage of many different features to make a holistic blockchain notarization system, one in which notarizations are not static one-time timestamps, but instead can now be dynamic, moving, changing, and updatable values on the blockchain. Companies or applications using and customizing this service have an opportunity to make well-defined application framework conventions of how their Apostille accounts are made and interact with the authority given by namespaces, the value and status represented by digital assets sent to that account, and the information sent as memos.

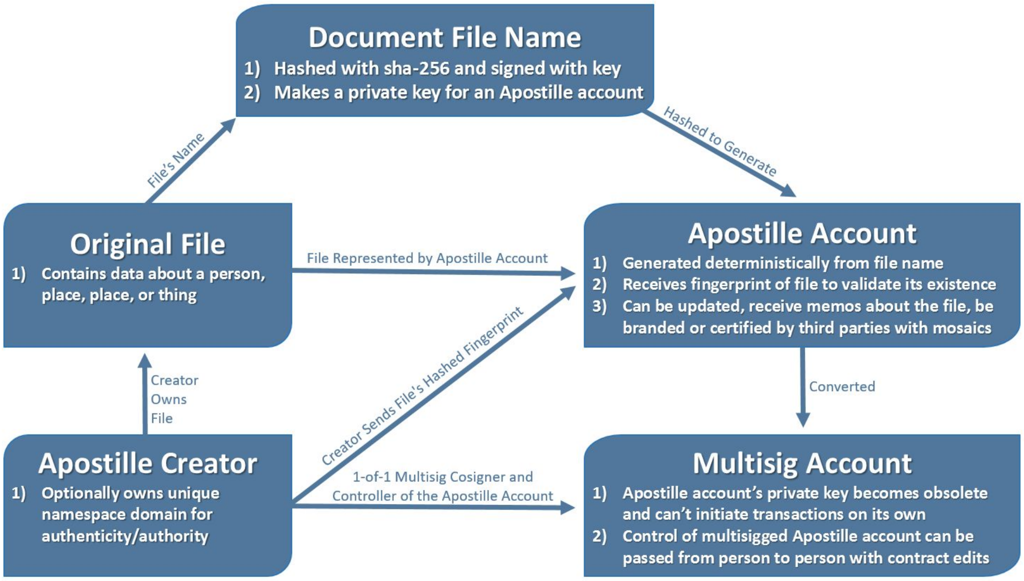
Document File Name
1) Hashed with sha-256 and signed with key
2) Makes a private key for an Apostille account
File's Name
Hashed to Generate
Original File
1) Contains data about a person, place, place, or thing
File Represented by Apostille Account
Apostille Account
1) Generated deterministically from file name
2) Receives fingerprint of file to validate its existence
3) Can be updated, receive memos about the file, be branded or certified by third parties with mosaics
Creator Owns File
Creator Sends File's Hashed Fingerprint
Converted
Apostille Creator
1) Optionally owns unique namespace domain for authenticity/authority
1-of-1 Multisig Cosigner and Controller of the Apostille Account
Multisig Account
1) Apostille account's private key becomes obsolete and can't initiate transactions on its own
2) Control of multisigged Apostille account can be passed from person to person with contract edits

Figure 4. Overview of the Apostille system framework and Apostille account creation.

If one is familiarized with the Colored Coin system in Bitcoin, then it becomes easier to understand the Apostille system. In Bitcoin, a colored coin is created by taking a Satoshi and tagging it with a value such as "car title" and then transferring that Satoshi around the blockchain making sure to track it and link it back to the time it was colored. In this situation, we can be considering to "color" the Satoshi with the designation of "car title". And while it is a visionary concept, it unfortunately, is built as a 2nd layer which is taxing for most Bitcoin organizations and wallets to implement. Furthermore such schemes usually require using the OP_RETURN, which is possible to use, but is something the Bitcoin Core developers have stated is a bad idea in version 0.9.0.

The NEM Apostille system proposes the same concept, but instead of applying it to a Satoshi, it applies it to an account created from a private key derived from the data being notarized. **These accounts can even be thought of as "colored accounts" which are far more suited and useful for tokenizing data value sets, and can signify anything a creator might want to represent, transfer, and update.** Bitcoins simply were not meant to be colored with information in the first place. Bitcoin was designed as "A Peer-to-Peer Electronic Cash System", emphasis in this case on a fungible "cash", not a peer-to-peer electronic *customized asset and message* exchanging system.

On the other hand, accounts on the NEM blockchain are designed by default to hold messages, hold mosaics (digital assets), and are designed to have control transferable from person to person via multisig contracts. They were built from the ground up to be much more than just an address. And because this all happens with APIs supported natively by the NEM blockchain servers, it is universal across the NEM blockchain ecosystem and all NEM wallets. No third party APIs are needed. Furthermore, the use of account states in NEM and the absence of Unspent Transaction Outputs (UTXO-inherent in Bitcoin) makes this possible. A UTXO system with a single native coin, allows for one common framework convention to take place, that of a single fungible coin and via the OP_RETURN, roundabout add ins. **A blockchain platform like NEM with acount states and customizable namespaces, mosaic assets, multisig contracts, and messaging options allows for a wide variety of application framework conventions to take place, Apostille being just one example.**

3.1. Preparing the Initiating Blockchain Notarization Account

While any account can initiate a blockchain notarization, it is preferably one that has a namespace and has published that namespace in their company information profile within their website, brochure, package information, etc. This is optional and is *not* a requirement but helps to bring legitimacy to a blockchain notarization especially when a consumer can deduce, "I trust this company, and only this company can use this namespace domain; therefore I trust the blockchain notarization made by this namespace domain to be an official blockchain notarization for this kind of product".

3.2. Types of Apostille Notarizations

The Apostille system allows a person to choose from two different kinds of notarizations depending on their business use case and privacy needs.

❖ *Public Sync:* Plain hashes are sent to a public synk address. This is useful for cases where a document that has been fingerprinted and stamped is meant to be shared freely.

- *Private, Transferable and Updatable:* Hashes are signed using the owner's private key and sent to a colored HD account (Apostille account) created from a file's dedicated private key. This is useful for cases where the contents of the notarized document should remain more private, or when a person would like to make a blockchain notarization that is updatable, transferable, conjointly owned, or hold extra value.

3.3. Making a File's Dedicated Private Key - Coloring HD Accounts - Creating an Apostille Account

A hierarchical deterministic (HD) account is generated from the file name, which is hashed using SHA-256 and then signed with user's private key.

The final signed hash is truncated to keep the first 64 characters that are now allocated to be the file's dedicated private key. Since the file's private key is created from a hashed filename, it will always be unique per each sender and can be thought of as the file's personal private key, not just as a random account's private key. When creating an HD account with a file's dedicated private key, we can consider that account to be "colored" and now be an Apostille account. For example, if one were to make a blockchain notarization for a car title, the color of the HD account in this case would be "car title".

Only the owner of the initiating account and file can retrieve the file's dedicated private key that makes the colored HD account, and it will always generate the same account for a given file name.

The formula is as follows:

```
HDprivateKey =
truncate(userKeyPair.sign(SHA256(fileName)));
```

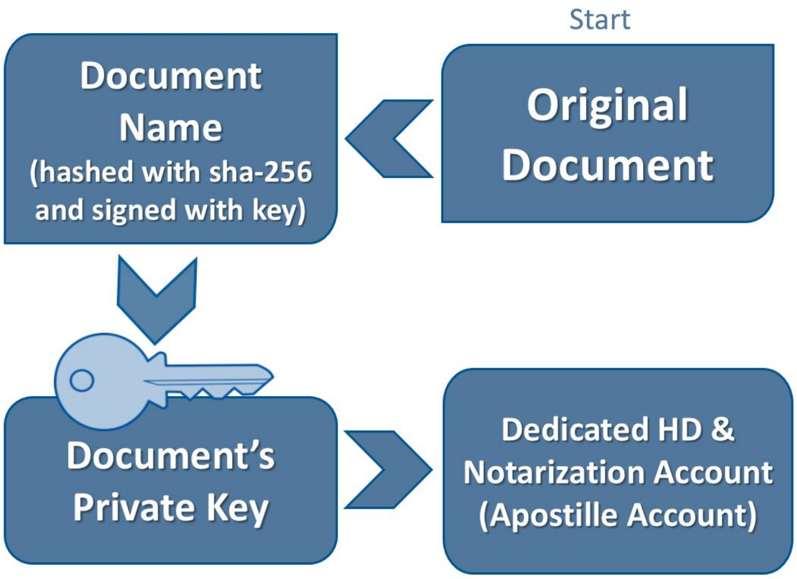
Start
Original Document
Document Name (hashed with sha-256 and signed with key)
Document's Private Key
Dedicated HD & Notarization Account (Apostille Account)

Figure 5. The process to create a colored HD account (Apostille account) from a document's name and account information which in turn make up its dedicated private key.

3.4. Apostille Hashes - Preparing the Timestamp Fingerprint - Making the Blockchain Notarization

An Apostille transaction message is the hash of the file data (document fingerprint) prepended with a custom Apostille hash header that begins with 4 byte magic bytes. The Apostille header helps us during an audit to determine the hashing algorithm used and if the hash was signed or not. Shown as follows:

```
ApostilleHash = Apostille header + fileContentHash;
```

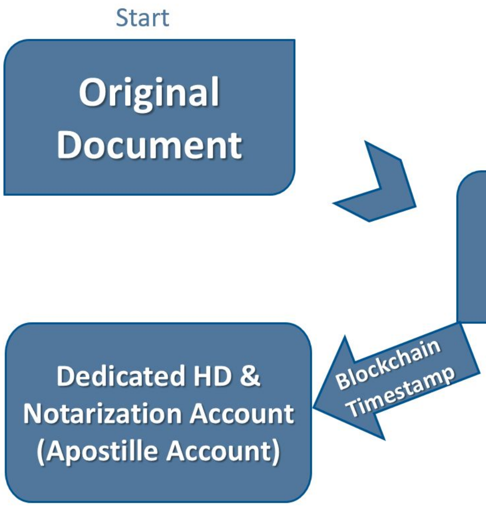
Start
Original Document
Document Fingerprint with Header
Blockchain Timestamp
Dedicated HD & Notarization Account (Apostille Account)

Figure 6. The Apostille process for creating an Apostille transaction message. The document's fingerprint plus Apostille header are sent in a message to a dedicated HD account.

The whole process then, of making a private Apostille blockchain notarization and creating the Apostille account, can be seen as the following:

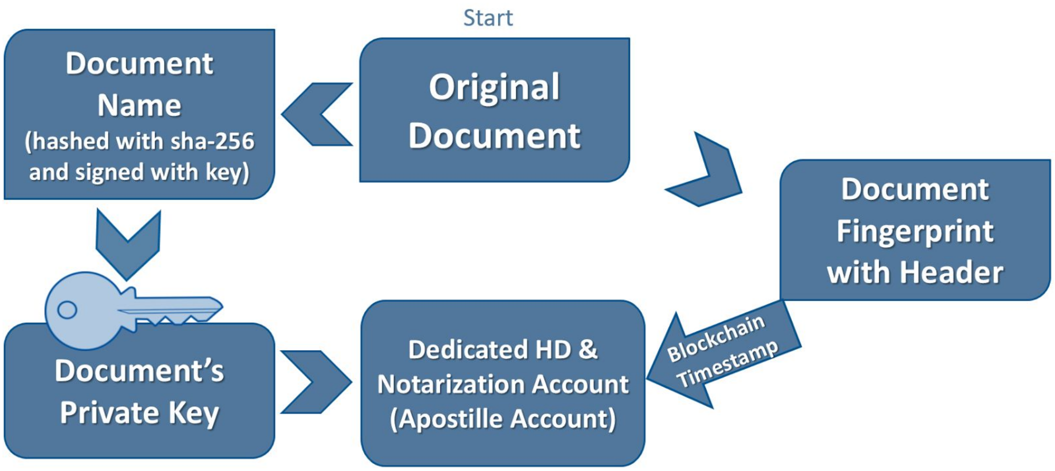
Start
Document Name (hashed with sha-256 and signed with key)
Original Document
Document Fingerprint with Header
Document's Private Key
Dedicated HD & Notarization Account (Apostille Account)
Blockchain Timestamp

Figure 7. The Apostille process for creating a document's private key, the dedicated HD account from that key, and the fingerprint of the original document that is sent to the dedicated HD account. Since it has been timestamped in the blockchain, it is now an Apostille blockchain notarization account.

3.5. Differences of Hashes

Apostille allows users to pick from two different options when making a blockchain notarization. These options are 1) *public sync,* or, 2) *private,* *transferable, and updatable*. Additionally, a user can further choose from a list of hashing algorithms. The user's choices will result in the various kinds of hashes shown below.

- Non-signed (public sync):

  ```
  0xFE 'N' 'T' 'Y' 0x01 + md5(data)
  0xFE 'N' 'T' 'Y' 0x02 + sha-1(data)
  0xFE 'N' 'T' 'Y' 0x03 + sha-256(data)
  0xFE 'N' 'T' 'Y' 0x08 + sha3-256(data)
  0xFE 'N' 'T' 'Y' 0x09 + sha3-512(data)
  ```

- Signed (private, transferable, and updatable):

  ```
  0xFE 'N' 'T' 'Y' 0x81 + sign(md5(data))
  0xFE 'N' 'T' 'Y' 0x82 + sign(sha-1(data))
  0xFE 'N' 'T' 'Y' 0x83 + sign(sha-256(data))
  0xFE 'N' 'T' 'Y' 0x88 + sign(sha3-256(data))
  0xFE 'N' 'T' 'Y' 0x89 + sign(sha3-512(data))
  ```

We start a hash with 0xFE for the NEM network to interpret the message as HEX

N, T and Y are for "Notary"; those letters are converted to hexadecimal represented as the magic bytes below:

- N: `0x4E`
- T: `0x54`
- Y: `0x59`

The common Apostille header for both types is FE4E5459; the two last hex characters of the header are defined by the user's choices (hashing method and signed or not).

For example:

- The header for a non-signed file hash using SHA-256 is `FE4E545903`.
- The header for a signed file hash using MD5 is `FE4E545981`.

Signed SHA-256 file hash example:
79952024fa6fd302abda4dfef63b1499b786c4269305bbf08d4058b417a528421d4f877c5a00d34e2addeba650b23812777448f2226515c801e4a424eefa6e04

Now with added Apostille header example:
fe4e54598379952024fa6fd302abda4dfef63b1499b786c4269305bbf08d4058b417a528421d4f877c5a00d34e2addeba650b23812777448f2226515c801e4a424eefa6e04

3.6. Auditing Blockchain Notarizations

To audit a file, we need to have the notarized document with a filename in a particular naming format. Documents that go through an Apostille blockchain notarization process will have their file name edited and pushed back to the user in a zipped folder that they may store it securely without editing it. The special naming format is as follows:

<Filename> - Apostille TX <transaction hash> -- Date YYYY-MM-DD.pdf

Original document file name example: *MyProject2016.pdf*

Hashed document file name example: *MyProject2016 - Apostille TX 0e94da29910ae64bb544e9de0e6a5c6440bd75e6bedafd81b5b4cf729ca25ef -- Date 2016-09-12.pdf*

When a user uploads the file to audit, the name is parsed and the blockchain notarization transaction is retrieved from the blockchain using the transaction hash.

We take the message in that transaction, cut the Apostille header and analyze it.

With the information from the header, we can know what algorithm the user selected for the Apostille notarization. So we can then hash the audited file with the right hashing method and look in the blockchain if it matches the transaction's message hash (without the header).

If the hash is non-signed as is the case with a *public* notarization, we simply verify that the uploaded file hash matches the one in the transaction message.

If the hash is signed as with the case with a *private, updatable, and transferable* notarization, we verify the signature using the signer's public key, the signed file hash (in the transaction message) and the hash of the uploaded file.

3.7. Transferring Control of a Blockchain Notarization's Colored HD Account

Since the notarized file's dedicated private key for its colored HD account (Apostille account) is only known by the maker of the blockchain notarization, the owner can freely turn that account into a 1-of-1 multisignature account, or any other combination of m-of-n they so choose.

A 1-of-1 multisignature contract is most convenient if a maker of the blockchain notarization wants sole ownership and wants to pass it to another singular 3rd party. A multisignature account of m-of-n, where *m and n* is greater than two, is useful for contracts that need to be conjointly owned, or shared, or when binding an agreement between multiple parties, or all of these.

NEM's multisig contracts can be made and edited in only a few clicks, and are supported by all wallets within the NEM system as they all use the same core API from NEM's blockchain servers. Likewise, to remove oneself and add another account as the parent/owner account of the colored HD account takes only a few clicks and can be done in a single transaction. No information need be exchanged with the new owner of the blockchain notarization account except knowing their account address. No smart contracts need be written, or special third party wallets used.

Step 1: Notarization Account Before a Multisig Contract

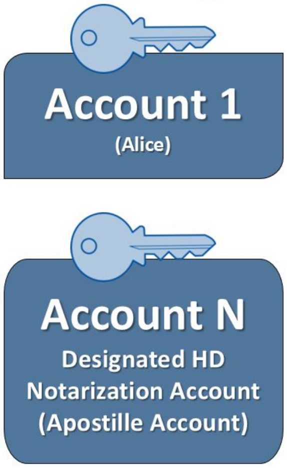

Figure 8: In this diagram, we can see that the notarization account has been made and stamped with a file's fingerprint hash. The file's dedicated private key was used to make Account N, but at this point, there is no multisig contract over it.

Step 2: Notarization Account After a Multisig Contract

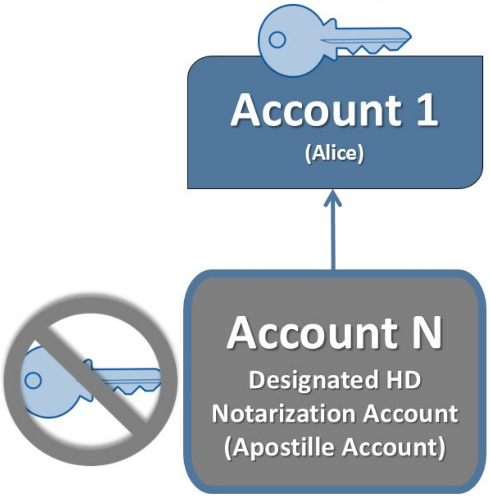

Figure 9: In this diagram, we can see that Account N has now been put under a 1-of-1 multisig contract. Alice now has full control of the notarization account, and the notarization account's private key is no longer allowed to initiate transactions.

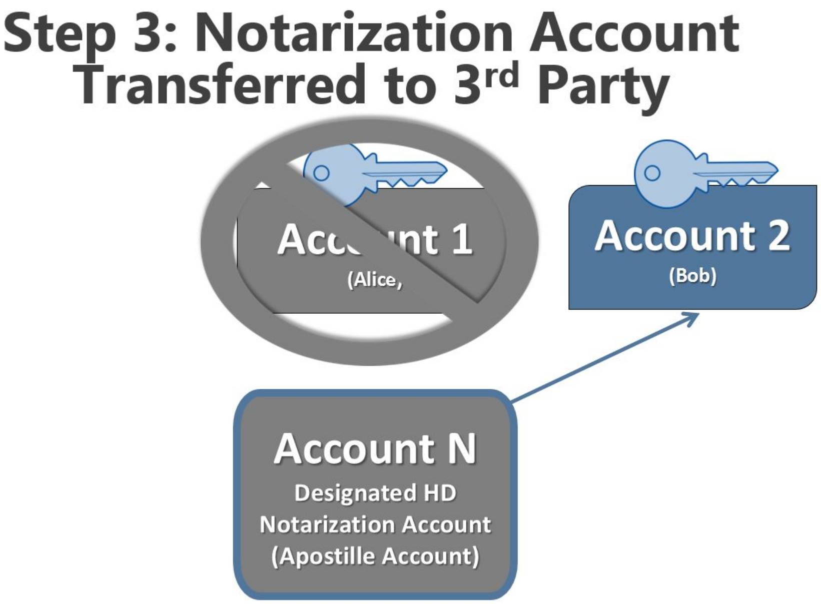
Step 3: Notarization Account Transferred to 3rd Party
Acc nt 1
(Alice,
Account 2
(Bob)
Account N
Designated HD Notarization Account (Apostille Account)

Figure 10: In this diagram, Alice has decided to transfer control of Account N to Bob. In a single transaction with a few clicks, she can remove herself and add Bob. Bob now has full control of Account N.

3.8. Updating a Blockchain Notarization

There are two ways a blockchain notarization can be updated. The first case is for an entirely new version of the document to be notarized by rehashing it and putting its new fingerprint on the blockchain. The second case is to amend the original notarization by sending messages/memos to the Apostille account. The latter case is also good for use cases where additional notes need to be recorded on an ongoing basis about the notarized product.

Sometimes documents or products change after they have been certified. If an item that is notarized is augmented, messages can be sent to the Apostille account reflecting any changes.

In the case of an Apostille account that has been transferred to third parties, who themselves have augmented the product, they can send messages from the blockchain Apostille account to itself representing changes made to the product.

In some cases, registered and certified third parties that are neither the creator nor owner of the object might augment it and want to record their changes. They can then send messages to the Apostille account from their account reflecting their work. In an ideal situation the third party's account would be registered with a namespace to help with reputation, but this is optional.

3.9 Making Multiple Party Contract Notarizations - Initiated, Signed, Updated, Controlled, and Transferred by Multiple Parties

One obvious use case of a blockchain notarization system is to have multiple parties make a multisig account together and then make contract together, jointly sign it and jointly upload its fingerprint onto the blockchain to that contract's Apostille account from their shared multisig account. This acts as proof that all parties agreed to the contract at the time it was signed.

The most elegant solution is for each party to own their own unique namespace, so that they may sign the contract with a registered account, but that is not a requirement.

Whether or not namespaces are used, the process is the same. All parties agreeing to a contract make a multisig contract over an account that will be used to initiate the Apostille transaction. That account will be under an n-of-n contract, and therefore any blockchain notarization coming from it must be signed by all parties. In NEM, this is done with push notifications via the blockchain network sent to light clients asking them if they want to sign.

The Apostille account that is created from this Apostille transaction will initially be just like any other regular account, but since the notarized file's dedicated private key will be known, the Apostille account can be put in multisig with the proper owners of it as cosigners.

These cosigners can then update, augment and renew the contract on chain by having all of them sign transactions over that Apostille account.

3.10 Private and Public Chains - Apostille for Mijin

Thus far this paper has mainly discussed using Apostille with the NEM public chain. But NEM has a sister blockchain technology, the private blockchain Mijin run by Tech Bureau. While Mijin and NEM are designed to meet different needs, they share much of the same underlying code and all of the same APIs. This makes any application running NEM Apostille capable of running on Mijin with very little effort in updating the code.

At the time of this paper, the current enterprise release of Mijin is running at 100’s tx/s, and the next generation update of the Mijin/NEM core codenamed “Catapult” is currently running at more than 4000’s tx/s on its private chain testnet. It is the first four-tiered blockchain technology utilizing microservices architecture, having one tier for the blockchain, one for API servers, one for MongoDB, and one for light clients, making it scale well beyond any other blockchain project. Catapult is the first and currently only blockchain to apply this common cloud scaling architecture to a blockchain system and marks a sharp shift from Bitcoin’s monolithic architecture.

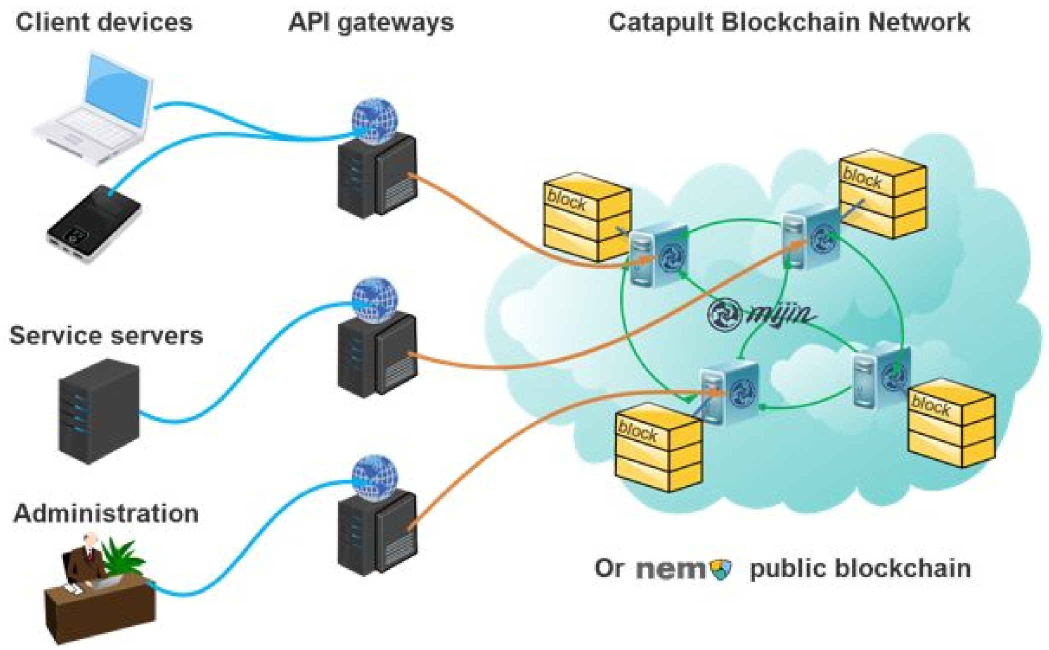

Figure 11. A representation of the Catapult blockchain.

Mijin private chains used for Apostille notarizations can also be anchored to the NEM public chain. This is a process by which a hash of the Mijin chain or its recent block header is itself notarized on the NEM public chain. This gives users and auditors of the Mijin chain and the blockchain notarizations within it nearly the same level of assurance of immutability as having used the public chain directly.

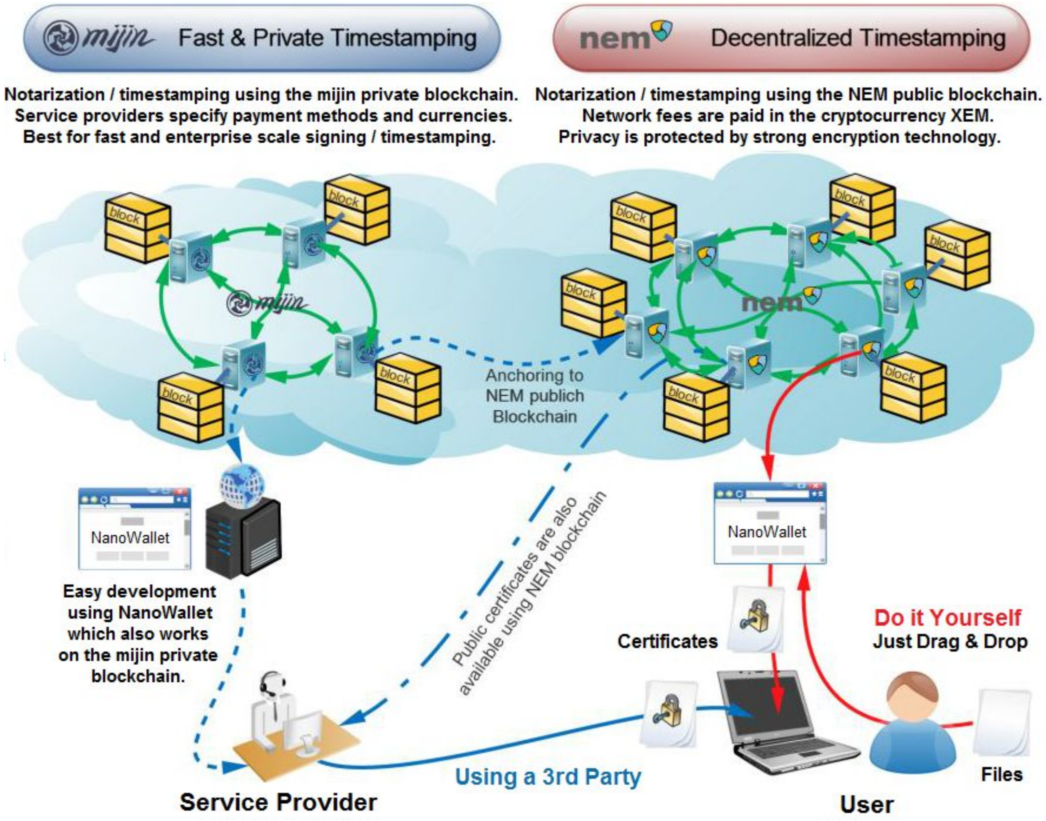
mijin Fast & Private Timestamping
nem Decentralized Timestamping
Notarization / timestamping using the mijin private blockchain.
Service providers specify payment methods and currencies.
Best for fast and enterprise scale signing / timestamping.
Notarization / timestamping using the NEM public blockchain.
Network fees are paid in the cryptocurrency XEM.
Privacy is protected by strong encryption technology.
block
mijin
nem
Anchoring to NEM publich Blockchain
Public certificates are also available using NEM blockchain
NanoWallet
Easy development using NanoWallet which also works on the mijin private blockchain.
Certificates
Do it Yourself
Just Drag & Drop
Files
Using a 3rd Party
Service Provider
User

Figure 12: The Mijin and NEM blockchain networks using the NEM Apostille notarization system.

4. Uses Cases

The following are use cases made to show the utility and range of Apostille accounts and notarizations.

4.1.1. Car Title

A car that has had its certificate of ownership notarized on the blockchain can have messages sent to its colored HD account reflecting what repairs and maintenance had been done and at what mileage. If those messages came from trusted car maintenance namespace accounts, one could believe that indeed the work was done properly. Mosaics created by trusted insurance or government organizations can be sent to the Apostille account, showing it had paid insurance or the car had been certified to meet certain standards.

4.1.2. Government Registration

A government could send a mosaic or message to the colored blockchain notarization HD account showing the car has been registered and taxes paid. In such cases, mosaics on the NEM system can be made to be "non-transferable" meaning that the controller of the colored HD account cannot send an official government mosaic representing the car's status to any other third parties. The only option would be to return to the original sender, which is the government in this case. Governments can also leave links that are either for data that is on or off the blockchain in the Apostille account that lead to more information, history, or registration of the car.

4.1.3. Digital Media Licenses

A digital media blockchain license can have messages sent to it representing how many times a product has been streamed. It can also have messages sent to it detailing the terms of the license and limits put upon it. A namespace and digital asset can originate from the license account (Apostille account for that license) and these assets can be sent to others representing rights in the license.

4.1.4. Luxury Consumer Goods and Anti-Counterfeiting

Companies with luxury goods can make a namespace account that only they can control. They can publish this name on their company profile. They can then make a blockchain notarization for each and every item using things like serial numbers, high definition scans, chemical makeup, and so on, to uniquely identify and register each item. Since each item is unique, and each is fingerprinted, and each blockchain notarization comes from a registered and recognized source, any competitor offering counterfeit products without an accompanying blockchain notarization will be easily identified.

It could occasionally be updated with additional information regarding the condition of the product as it is maintained or repaired by registered and licensed professionals.

A luxury good item could have a message sent about a recall to the Apostille account.

And in all these cases, any third party wanting to purchase the used item (along with the colored Apostille account), can trace back the items authenticity and history.

4.1.5. Two-Party Legal Contracts

As mentioned in section 3.9, the Apostille system is an ideal candidate for making on-chain notarizations of contracts signed by multiple parties which can then later be updated or transferred if desired.

4.2 Custom Use Cases

While this paper describes and exact application framework convention for which the Apostille service has been built and released in NEM's NanoWallet, the system is highly adaptable and additional functions can be added in the future. Furthermore, the NanoWallet is open-sourced, written in Javascript, and connects to NEM with easy to use APIs so other parties looking to build specific business apps can make a new application framework convention for their app that interprets the rules for multisignature contracts, namespaces, mosaic assets, and messages with differently defined rules to meet their organizational needs.

5. Conclusion

This paper has proposed a blockchain powered account certification and notarization solution made possible on the NEM block chain platform. We believe the Apostille service is a unique and a notable improvement in blockchain technology. Bitcoin's UTXO system has proven very valuable and resilient over time for spending money, but has not been so useful for building greater blockchain applications. NEM has switched to an account state system and architecture with built-in features supported by the blockchain core protocol, thus enabling a new variety of application framework conventions to be set in motion and a new set of blockchain applications to be built.

Under the Apostille service, accounts in NEM can now be thought of as states representing objects. An Apostille account uses a file's dedicated private key to make a specially marked (colored) hierarchical deterministic (HD) account. When combined with a system of mutlisignature contracts, namespaces, mosiac assets, and messages, a well-defined and dynamic system can be built that follows strict rules. An application framework convention with these properties is powerful because 1) multisignature contracts allow these Apostille accounts to be transferred or conjointly owned, 2) using namespaces denotes authority, authenticity, and authorship, 3) mosaic assets can be sent either to or from the Apostille account representing value or status, and lastly, 4) messages can be sent to or from the Apostille account representing additional information and updates.

The Apostille service is an upgrade of the earlier blockchain notarization services where a notarization is only a one-time digitally fingerprinted record of a document on a blockchain. By combining the features of messages, namespaces, mosaics, and multisig and NEM's account system all being utilized with NEM's APIs, a dynamic blockchain notarization system where blockchain notarizations can be branded, registered, transferred, and updated is created. We believe that the Apostille service can unlock a new host of applications suitable for commercial applications not yet recognized on the blockchain, including apps for representing authenticity, identity, object state and status, property ownership, confidential records and more.

Acknowledgements

The Apostille project was financially supported by the NEM Community Fund. We would like to thank all the NEM community members that took time to show support for this project. The authors would also like to personally thank Lon Wong and Takao Asayama for their guidance and pearls of wisdom during the process of making Apostille. We would like to give credit to the NEM developers and NEM Core team. They have worked diligently for many years to make NEM technology the great platform it is today. And lastly we are immensely grateful for the greater NEM community for its backing and contributions.

Bibliography

Apostille. (2014). *Collins English Dictionary – Complete and Unabridged, 12th Edition*. Retrieved October 2016, from http://www.thefreedictionary.com/Apostille

Apostille Convention. (2016, October 24). In *Wikipedia, The Free Encyclopedia*. Retrieved October 2016, from https://en.wikipedia.org/wiki/Apostille_Convention

Araoz, M. (2012). *Proof of Existence: About*. Retrieved October 2016, from https://proofofexistence.com/about

Bitcoin Core Developers. (2014, March). *Bitcoin Core version 0.9.0 released*. Retrieved January 2017, from Bitcoin.org: https://bitcoin.org/en/release/v0.9.0#opreturn-and-data-in-the-block-chain

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Retrieved 28 April 2014. https://bitcoin.org/bitcoin.pdf

NEM Core Team. (2015). *NEM Technical Reference*. Retrieved October 2016, from nem.io: https://www.nem.io/NEM_techRef.pdf

NEM Team. (2016). *Mosaics and Namespaces*. Retrieved Ocotober 2016, from nem.io: https://blog.nem.io/mosaics-and-namespaces-2/

NetSPI. (2013, July). Magic Bytes - Identifying Common File Formats at a Glance. Retrieved from NetSPI: https://blog.netspi.com/magic-bytes-identifying-common-file-formats-at-a-glance/

Rosenfeld, M. (2012). *Overview of Colored Coins*. Retrieved October 2016, from Bitcoil: https://bitcoil.co.il/BitcoinX.pdf

Swan, M. (2014). *Blockchain: Blueprint for a New Economy*. O'Reilly Media .