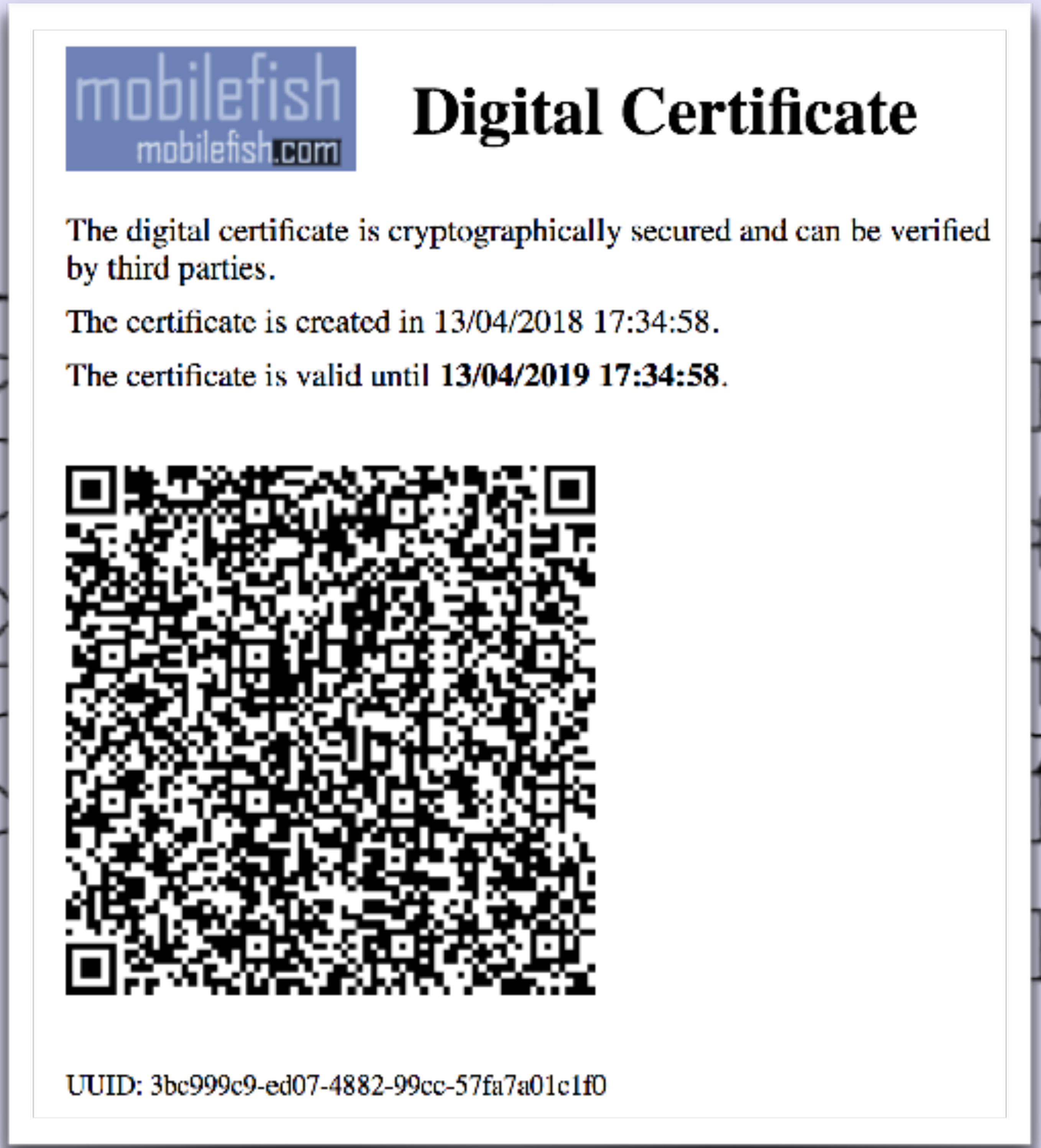
IOTA TUTORIAL 22

MAM Demo
Verifiable Claims



mobilefish
mobilefish.com

Digital Certificate

The digital certificate is cryptographically secured and can be verified by third parties.

The certificate is created in 13/04/2018 17:34:58.

The certificate is valid until **13/04/2019 17:34:58**.

UUID: 3bc999c9-ed07-4882-99cc-57fa7a01c1f0

v1.0.0

INTRO

- In this video I will demonstrate a verifiable claims Proof-of-Concept using Masked Authenticated Messaging.

MUNICIPALITY OF HAARLEM IOTA POC

- In 2017, Xurux Solutions in collaboration with ICTU, were commissioned by the municipality of Haarlem (the Netherlands) to create a Proof-of-Concept in which the citizens of Haarlem logs into a website using an existing Identity Management System (called DigID) to retrieve a publicly verifiable claim. This verifiable claim is in fact a QR code.

- The QR code contains information such as the hash value of the citizens personal data (name, address and social security number), root and other relevant data.
  This hash value, called the attest hash, is stored on the IOTA Tangle (Masked Authenticated Messaging) using the previous mentioned root.

- Third parties, like housing corporations, can easily prove these verifiable claims. The QR code is scanned, to get the root and hash value. The attest hash value can now be retrieved from the Tangle and compared with the one stored in the QR code.

MUNICIPALITY OF HAARLEM IOTA POC

- More information about the municipality of Haarlem IOTA Proof-of-Concept can be found at:
https://github.com/Haarlem/digitale-waardepapieren

VERIFIABLE CLAIMS DEMO

- Based on the municipality of Haarlem IOTA Proof-of-Concept I have created the "IOTA MAM Demo: Verifiable Claims": https://www.mobilefish.com/services/cryptocurrency/mam_verifiable_claims.html

- This demonstration is created for educational purpose and is **NOT** the same as the Haarlem's PoC.

- It's main purpose is to demonstrate yet another Masked Authenticated Messaging use case.

WHAT IS A VERIFIABLE CLAIM

- A verifiable claim is a piece of information about an entity such as a name, government ID, home address or university degree.

- This verifiable claim is tamper-proof and whose authorship can be instantly cryptographically verified by the receiving party.

- Verifiable claims are also known as attestations.

VERIFIABLE CLAIMS DEMO EXPLAINED

- Bruce requires an attestation from Gotham City stating that he is a resident of this city and he is eligible for social housing.

- Gotham City issues a verifiable claim to Bruce, attesting that he is a resident of Gotham City and he meets all the conditions for social housing.

- The claim is hashed (also known as attesthash) and stored on the Tangle using the Masked Authenticated Messaging in restricted mode.

- Bruce shares this claim with the social housing cooperative because he wants to be eligible for a social rental home.

VERIFIABLE CLAIMS DEMO EXPLAINED

- The social housing cooperative needs to verify that Bruce's claim is signed by Gotham City.

- The social housing cooperative does this by first hashing Bruce's claim.
  Let call this the "calculated attesthash".

- Next the social housing cooperative extracts the "stored attesthash" from the Tangle.
  All relevant information, such as root and uuid are stored in Bruce's claim.

- If the "calculated attesthash" is the same as the "stored attesthash" than this is the proof that Gotham City has signed Bruce's claim.

VERIFIABLE CLAIMS DEMO EXPLAINED

- To make this all work Gotham City must provide the social housing cooperative with the side key because of the use of the Masked Authenticated Messaging restricted mode.

- The social housing cooperative does not need to have any connection to or interaction with Gotham City.

- Each time Bruce requests for a verifiable claim an Universally Unique IDentifier (UUID) is generated compliant with RFC-4122 Version 4. Such an UUID looks like "df346607-1d58-4130-b274-be6a084074ed" which is an 128-bit value formatted into blocks of hexadecimal digits separated by hyphens. One of the main reasons for using UUIDs is that no centralised authority is required to administer them.

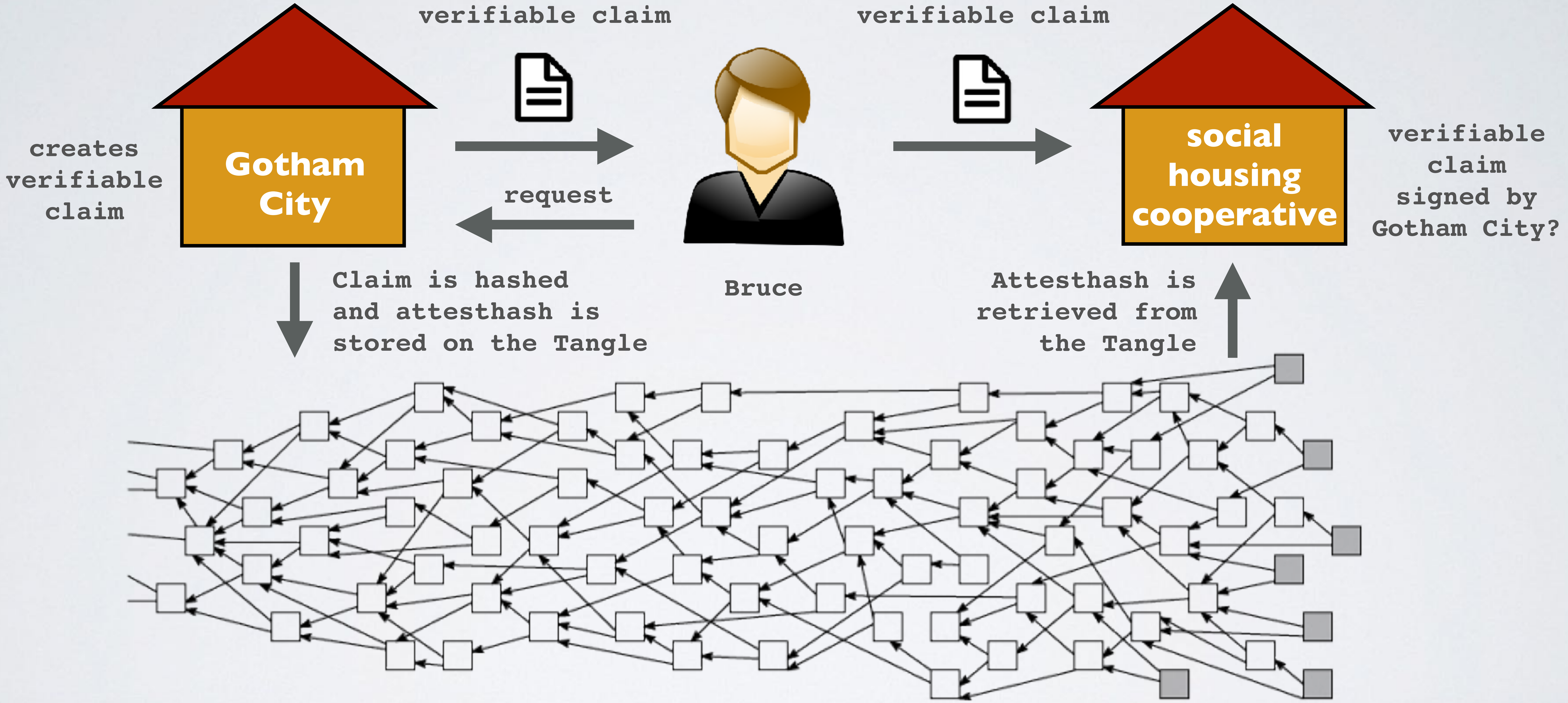
VERIFIABLE CLAIMS DEMO EXPLAINED

- The chance of generating the same UUID is quite small especially if the UUIDs are generated using sufficient entropy. More information see: https://en.wikipedia.org/wiki/Universally_unique_identifier

- The UUID is used as key for the HMACSHA384 keyed hash algorithm and should be stored in a database as a reference for later use.

- WARNING:
  In this demo I am using UUIDs because of its simple implementation. But it doesn't mean you should do it this way!

VERIFIABLE CLAIM PROCESS


verifiable claim
verifiable claim
creates verifiable claim
Gotham City
request
social housing cooperative
verifiable claim signed by Gotham City?
Claim is hashed and attesthash is stored on the Tangle
Bruce
Attesthash is retrieved from the Tangle

CREATE VERIFIABLE CLAIM

key (uuid = 3bc999c9-ed07-4882-99cc-57fa7a01c1f0)

Universally Unique IDentifier (UUID)

Claim

SSN: 1122-21122
firstName: Bruce
:

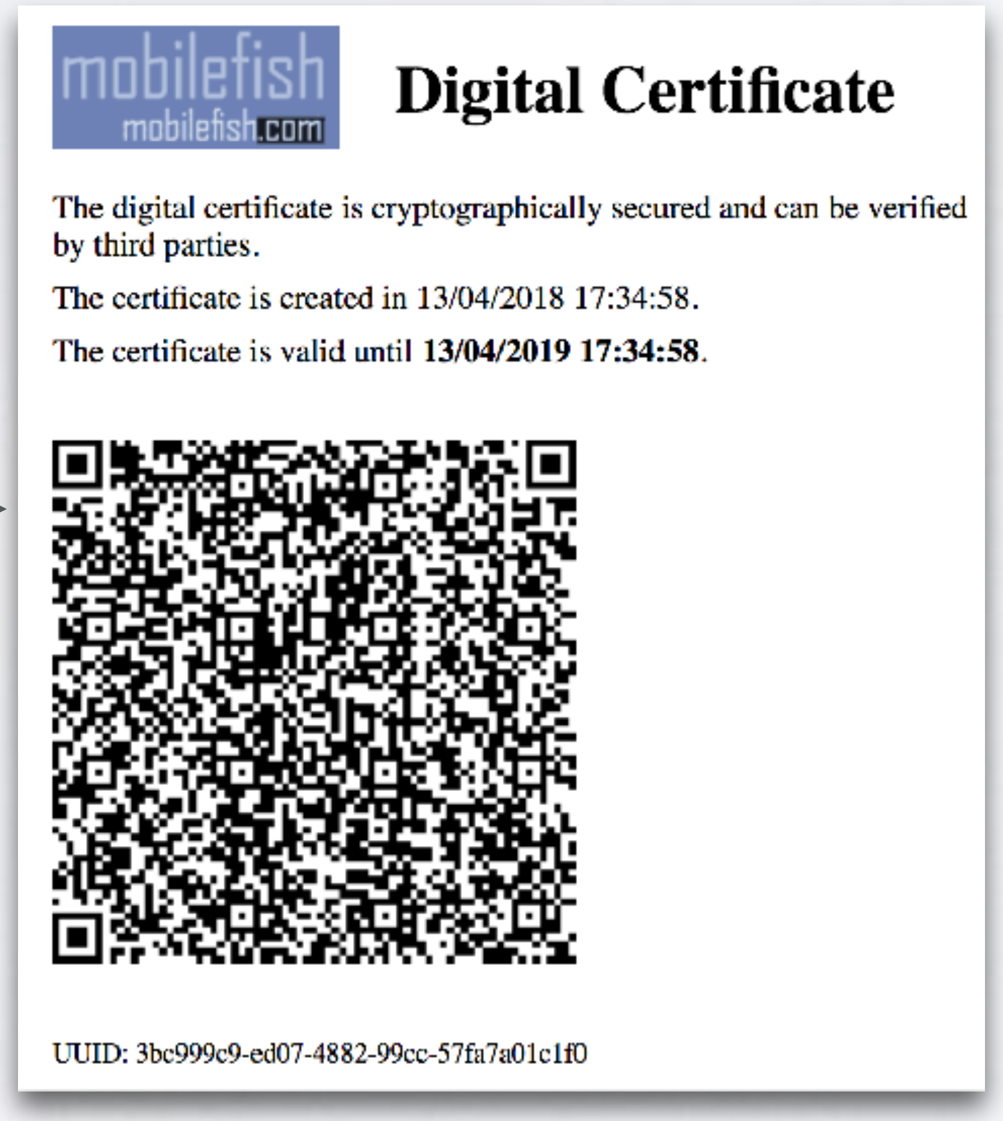
HMACSHA384

keyed hash algorithm

attesthash

Verifiable Claim

SSN: 1122-21122
firstName: Bruce
:
root: JUOCS…
uuid: 3bc999….



mobilefish mobilefish.com

Digital Certificate

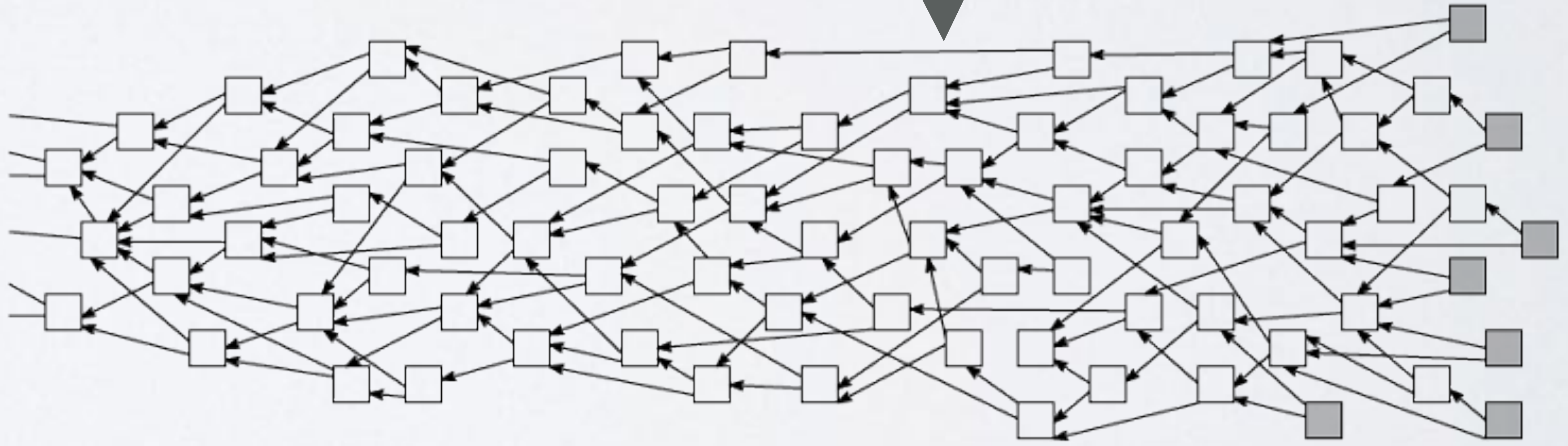
The digital certificate is cryptographically secured and can be verified by third parties.

The certificate is created in 13/04/2018 17:34:58.

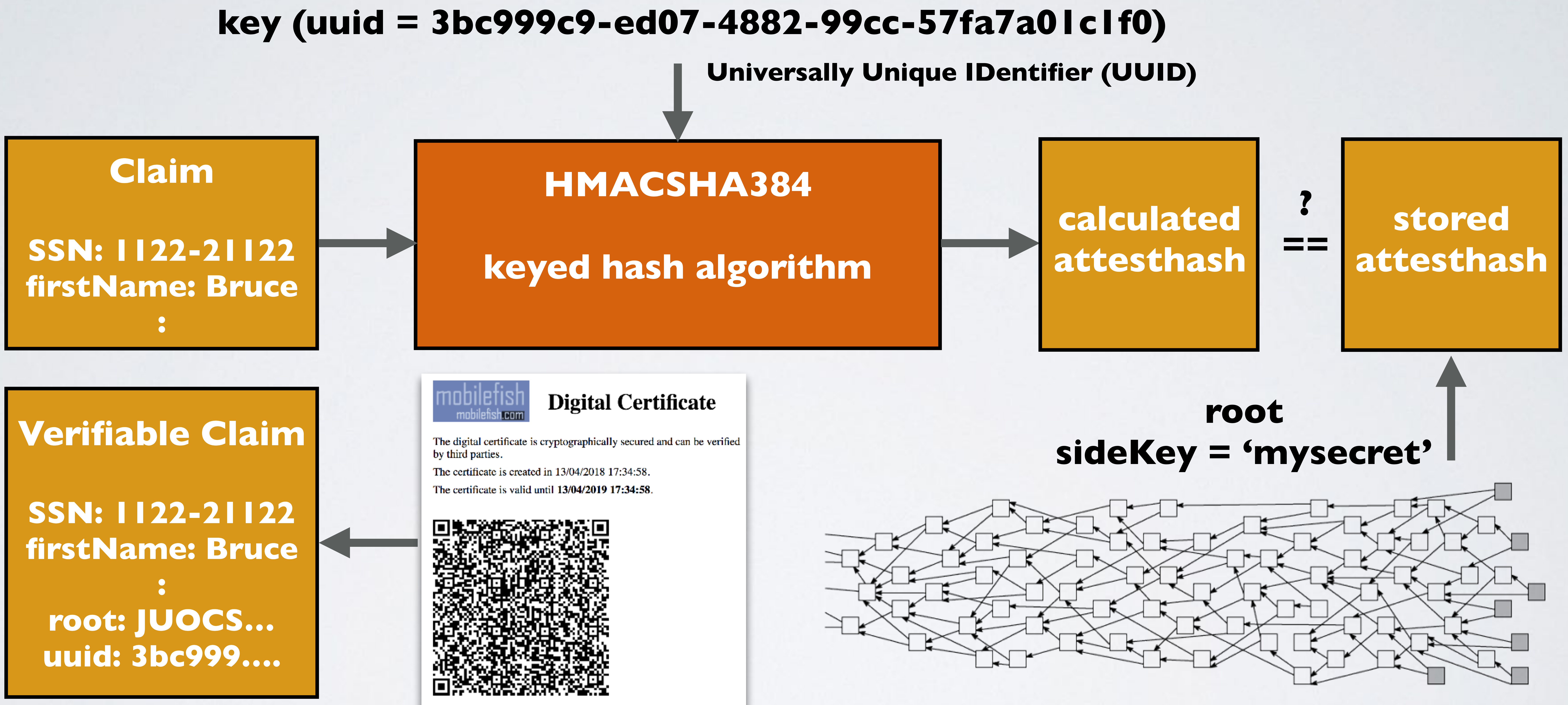
The certificate is valid until **13/04/2019 17:34:58**.

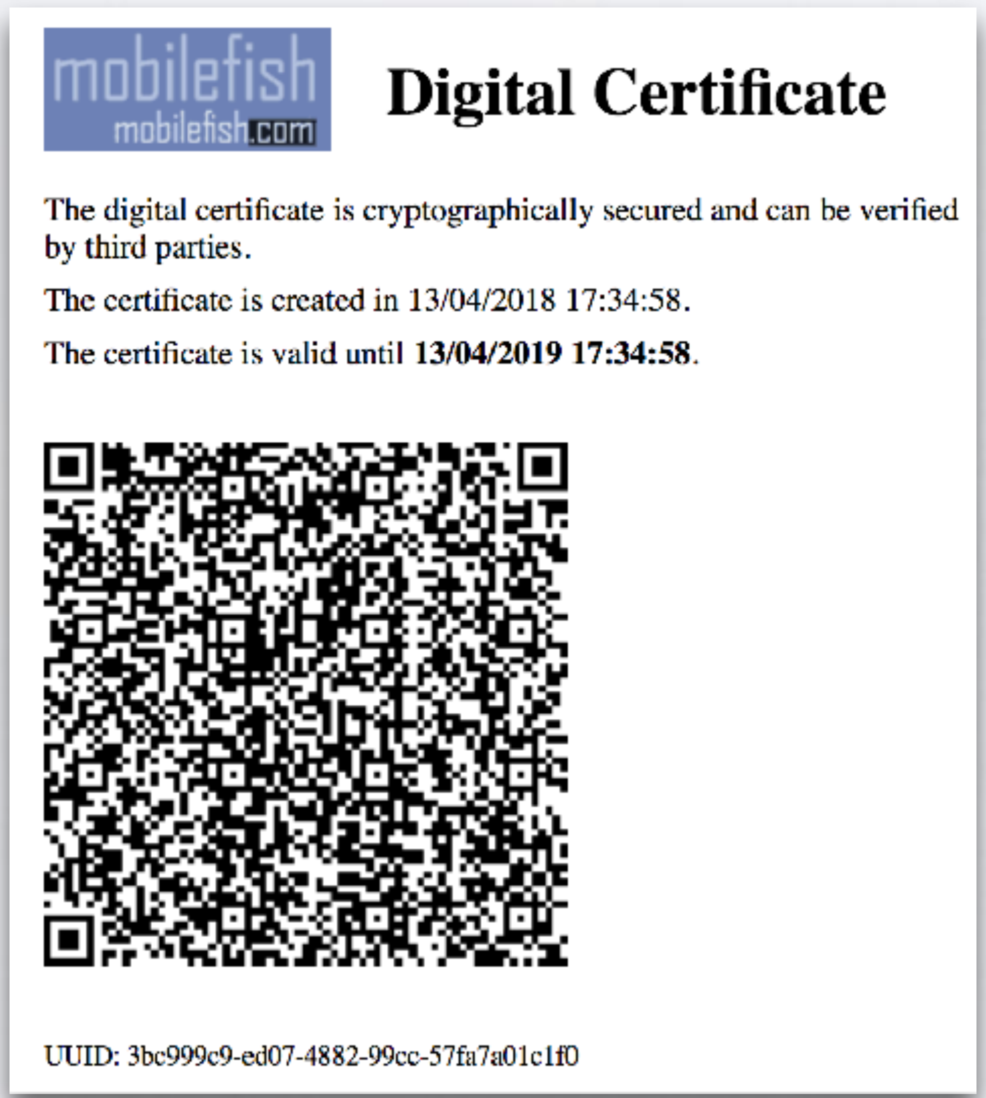
UUID: 3bc999c9-ed07-4882-99cc-57fa7a01c1f0

root
sideKey = 'mysecret'



CHECK VERIFIABLE CLAIM



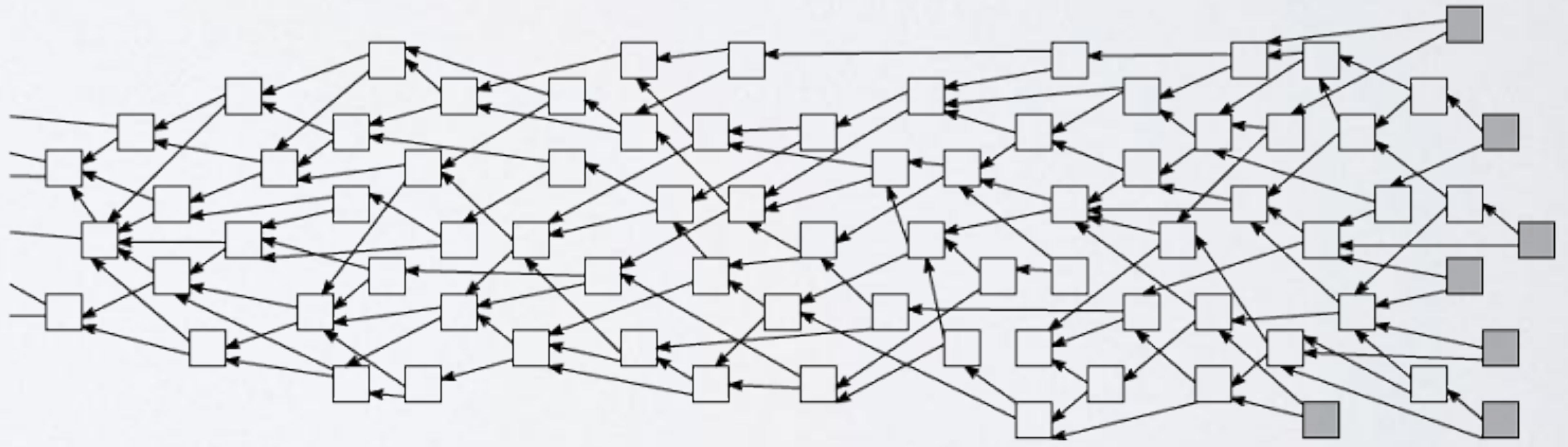

Digital Certificate

The digital certificate is cryptographically secured and can be verified by third parties.

The certificate is created in 13/04/2018 17:34:58.

The certificate is valid until **13/04/2019 17:34:58**.

UUID: 3bc999c9-ed07-4882-99cc-57fa7a01c1f0



WHAT IS HMACSHA384

- HMACSHA384 is a type of keyed hash algorithm that is constructed from the SHA-384 hash function and used as a Hash-based Message Authentication Code (HMAC). The output hash is 384 bits in length.

- An HMAC can be used to determine whether a message sent over an insecure channel has been tampered with, provided that the sender and receiver share a secret key. The sender computes the hash value for the original data and sends both the original data and hash value as a single message. The receiver recalculates the hash value on the received message and checks that the computed HMAC matches the transmitted HMAC.

WHAT IS HMACSHA384

- Any change to the data or the hash value will result in a mismatch, because knowledge of the secret key is required to change the message and reproduce the correct hash value. Therefore, if the original and computed hash values match, the message is authenticated.

- Please note:
  In our demo the key for the HMACSHA384, which is the uuid, is not secret because MAM restricted mode is being used.

FUTURE IMPROVEMENT TO THIS POC

- An improvement to this PoC would be, after the social housing corporative has scanned and processed the QR code, the verifiable claim should be revoked.
- This "revoked" information should be stored on the Tangle.
- If the social housing corporative scanned the same QR code again it will automatically detect that the verifiable claim is invalid.