re:claimID

Datenspuren 2019

Martin Schanzenbach

21.9.2019

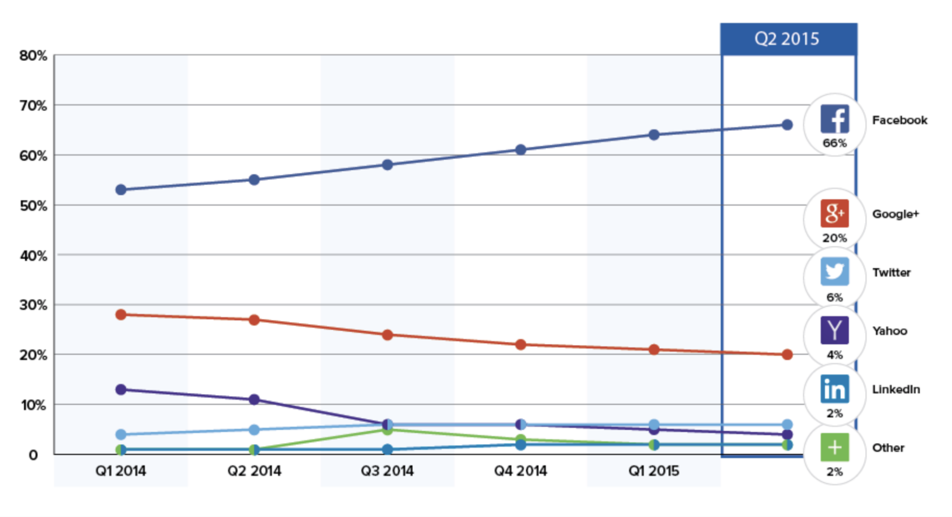
Fraunhofer AISEC

GNUnet

Motivation

Motivation

Identity Provider Market:



Motivation

Issues:

1. **Privacy** concerns:
   - Targeted advertisement, opinion shaping.
   - "Public safety": Mass surveillance and data collection.

Motivation

Issues:

1. **Privacy** concerns:
    - Targeted advertisement, opinion shaping.
    - “Public safety”: Mass surveillance and data collection.
2. **Liability** risks:
    - Data loss through leaks or hacks may result in existential legal implications (GDPR).

Motivation

Issues:

1. **Privacy** concerns:
    - Targeted advertisement, opinion shaping.
    - “Public safety”: Mass surveillance and data collection.
2. **Liability** risks:
    - Data loss through leaks or hacks may result in existential legal implications (GDPR).
3. **Oligopoly**:
    - “There can be only one (two)”.
    - IdP market tends to degenerate.
    - Federation not widely used.

Approach

Primary objective: We must enable users to exercise their right to digital self-determination:

Approach

Primary objective: We must enable users to exercise their right to digital self-determination:

1. Avoid third party services for identity management and data sharing.

Approach

Primary objective: We must enable users to exercise their right to digital self-determination:

1. Avoid third party services for identity management and data sharing.
2. Open, free and decentralized service which is not under the control of a single organization, consortium or business.

Approach

Primary objective: We must enable users to exercise their right to digital self-determination:

1. Avoid third party services for identity management and data sharing.
2. Open, free and decentralized service which is not under the control of a single organization, consortium or business.
3. Free software.

Approach

Primary objective: We must enable users to exercise their right to digital self-determination:

1. Avoid third party services for identity management and data sharing.
2. Open, free and decentralized service which is not under the control of a single organization, consortium or business.
3. Free software.

⇒ Empower users to **reclaim** control over their digital identities.

What does an IdP do?

1. Identity provisioning and access control
    - Allow management of identities and personal data.
    - Facilitate sharing of identity data with third parties.
    - Provide up-to-date information accessible even if user is offline.
    - Enforce authorization decisions of user.

What does an IdP do?

1. Identity provisioning and access control
    - Allow management of identities and personal data.
    - Facilitate sharing of identity data with third parties.
    - Provide up-to-date information accessible even if user is offline.
    - Enforce authorization decisions of user.

2. Identity information verification and attestation:
    - “this is Alice’s email address”: Email provider.
    - “this person is living in Germany”: Sovereign state.

What does an IdP do?

1. Identity provisioning and access control
   - Allow management of identities and personal data.
   - Facilitate sharing of identity data with third parties.
   - Provide up-to-date information accessible even if user is offline.
   - Enforce authorization decisions of user.

   ⇒ **re:claimID**
2. Identity information verification and attestation:
   - “this is Alice’s email address”: Email provider.
   - “this person is living in Germany”: Sovereign state.

   ⇒ **Not our department!***

*We will revisit this further on.

Introducing re:claimID

- re:claimID is a **self-sovereign** personal data sharing system.
- Other self-sovereign identity systems you may have head about:
    - Sovrin (Hyperledger)
    - uPort (Ethereum)
    - NameID (Namecoin)

- re:claimID is a **self-sovereign** personal data sharing system.
- Other self-sovereign identity systems you may have head about:
    - Sovrin (Hyperledger)⇐ **Permissioned blockchain**
    - uPort (Ethereum) .
    - NameID (Namecoin)

- re:claimID is a **self-sovereign** personal data sharing system.
- Other self-sovereign identity systems you may have head about:
    - Sovrin (Hyperledger)⇐ **Permissioned blockchain**
    - uPort (Ethereum)⇐ **Data shared off-chain: If user is offline data not accessible**.
    - NameID (Namecoin)

- re:claimID is a **self-sovereign** personal data sharing system.
- Other self-sovereign identity systems you may have head about:
  - Sovrin (Hyperledger)⇐ **Permissioned blockchain**
  - uPort (Ethereum)⇐ **Data shared off-chain: If user is offline data not accessible**.
  - NameID (Namecoin) ⇐ **Access control through central server (wat?)**

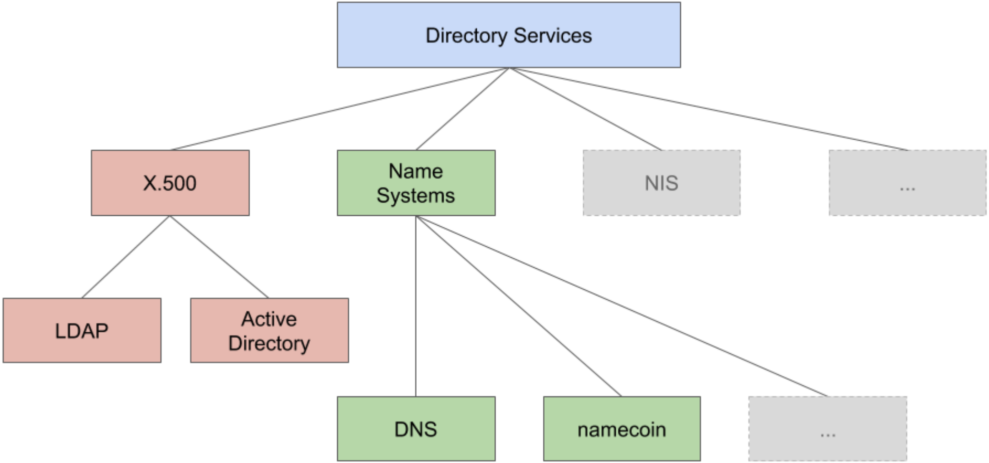
- re:claimID is a **self-sovereign** personal data sharing system.
- Other self-sovereign identity systems you may have head about:
    - Sovrin (Hyperledger)⇐ **Permissioned blockchain**
    - uPort (Ethereum)⇐ **Data shared off-chain: If user is offline data not accessible**.
    - NameID (Namecoin) ⇐ **Access control through central server (wat?)**

! re:claimID does **not** require a blockchain, is fully decentralized and allows asynchronuous data access.

In a nutshell

re:claimID = **Decentralized directory service** + **Cryptographic access control**

Directory services?



In a nutshell

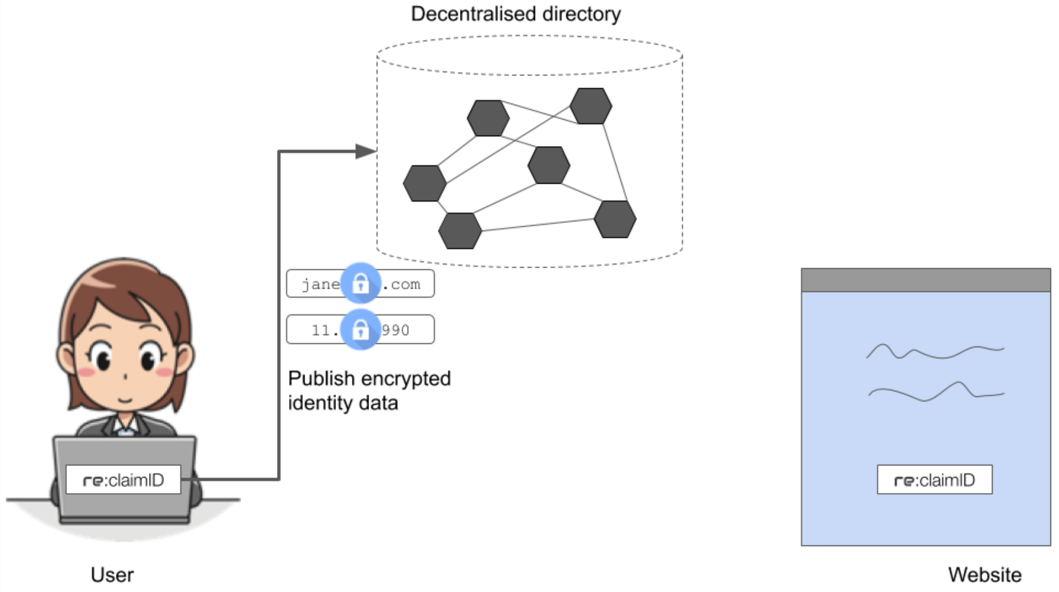
- Decentralized directory service
  - Secure **name system** with open name registration.
  - Idea "borrowed" from NameID.
  - Example: nslookup email.bob.org ⇒ "bob@example.com"
  - Our implementation uses the **GNU Name System (GNS)**

In a nutshell

- Decentralized directory service
  - Secure **name system** with open name registration.
  - Idea "borrowed" from NameID.
  - Example: nslookup email.bob.org ⇒ "bob@example.com"
  - Our implementation uses the **GNU Name System (GNS)**
- Cryptographic access control layer
  - Provided by GNS through encrypted and signed resource records.
  - Protects identity data from unwanted disclosure and allows users to enforce access control.

How does it work

Managing and publishing identity information



The GNU Name System

- In GNS, a namespace is defined by a public/private EC key pair:
  - x: Private key
  - P: Public key
  - G: Generator of the curve
  - n: Group order

The GNU Name System

- In GNS, a namespace is defined by a public/private EC key pair:
  - x: Private key
  - P: Public key
  - G: Generator of the curve
  - n: Group order
- Records are encrypted and signed using keys derived from (x, P).

The GNU Name System

- In GNS, a namespace is defined by a public/private EC key pair:
  - x: Private key
  - P: Public key
  - G: Generator of the curve
  - n: Group order
- Records are encrypted and signed using keys derived from (x, P).
- Encrypted records are published in a distributed hash table (under key q).

The GNU Name System

- In GNS, a namespace is defined by a public/private EC key pair:
    - x: Private key
    - P: Public key
    - G: Generator of the curve
    - n: Group order
- Records are encrypted and signed using keys derived from (x, P).
- Encrypted records are published in a distributed hash table (under key q).
- Any peer is able to verify the signature as the corresponding derived public key is also published.

The GNU Name System

- In GNS, a namespace is defined by a public/private EC key pair:
  - x: Private key
  - P: Public key
  - G: Generator of the curve
  - n: Group order
- Records are encrypted and signed using keys derived from (x, P).
- Encrypted records are published in a distributed hash table (under key q).
- Any peer is able to verify the signature as the corresponding derived public key is also published.
- Records can only be resolved and decrypted if the true identity and the label is known.

The GNU Name System

- In GNS, a namespace is defined by a public/private EC key pair:
    - x: Private key
    - P: Public key
    - G: Generator of the curve
    - n: Group order
- Records are encrypted and signed using keys derived from (x, P).
- Encrypted records are published in a distributed hash table (under key q).
- Any peer is able to verify the signature as the corresponding derived public key is also published.
- Records can only be resolved and decrypted if the true identity and the label is known.

$\Rightarrow$ Namespaces **cannot** be enumerated and queries/responses **cannot*** be observed.

*Unless label and identity are known.

Identity attributes in GNS

Users may create a namespace (x, P) and use it as a digital identity containing personal information:

Label	Record Type	Value
l_{email}	`ATTR`	“email=alice@example.com”
l_{name}	`ATTR`	“name=Alice Doe”
l_{dob}	`ATTR`	“dob=1.3.1987”

where the labels are **random secret values** with high entropy.

Publishing information

Given a namespace (x, P), we can treat labels as shared secrets in order to selectively disclose information.

$$h := Hash(l_{attr}, P)$$

Publishing information

Given a namespace (x, P), we can treat labels as shared secrets in order to selectively disclose information.

$$h := Hash(l_{attr}, P)$$

$$\textbf{DHT key} \left\{ \qquad q := H(hP) \right.$$

Publishing information

Given a namespace (x, P), we can treat labels as shared secrets in order to selectively disclose information.

$$h := Hash(l_{attr}, P)$$

$$\textbf{DHT key} \left\{ \quad q := H(hP) \right.$$

$$\textbf{Encryption} \left\{ \begin{aligned} k &:= HKDF(l_{attr}, P) \\ Record &:= Enc_k(Data) \end{aligned} \right.$$

Publishing information

Given a namespace (x, P), we can treat labels as shared secrets in order to selectively disclose information.

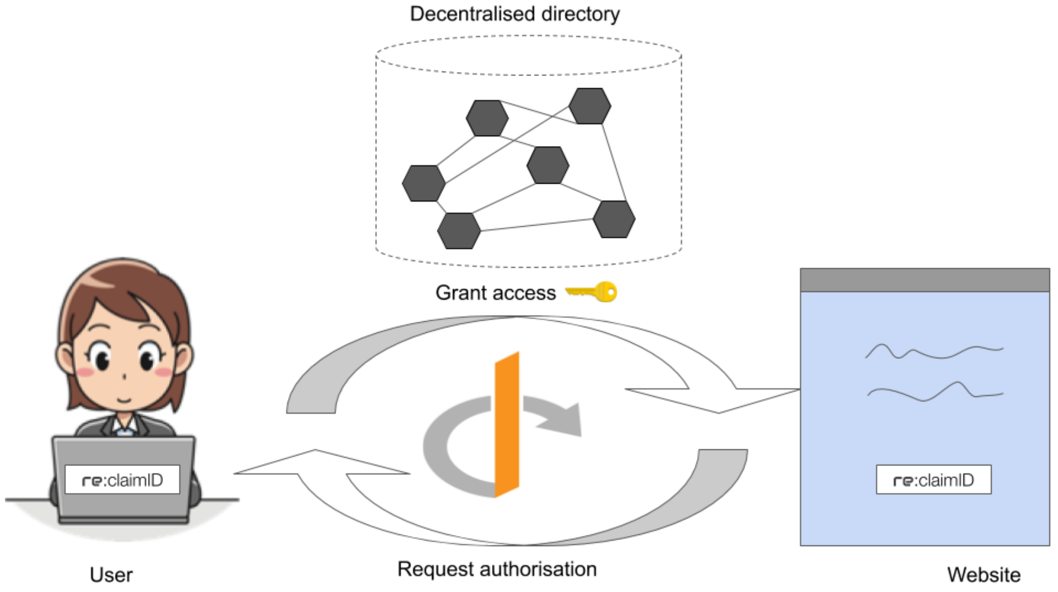
$$h := Hash(l_{attr}, P)$$

$$\textbf{DHT key} \left\{ \quad q := H(hP) \right.$$

$$\textbf{Encryption} \left\{ \begin{aligned} k &:= HKDF(l_{attr}, P) \\ Record &:= Enc_k(Data) \end{aligned} \right.$$

$$\textbf{Signature} \left\{ \begin{aligned} d &:= h \cdot x \ mod \ n \\ Signature &= Sig_d(Record) \end{aligned} \right.$$

Authorizing access



Authorizing access

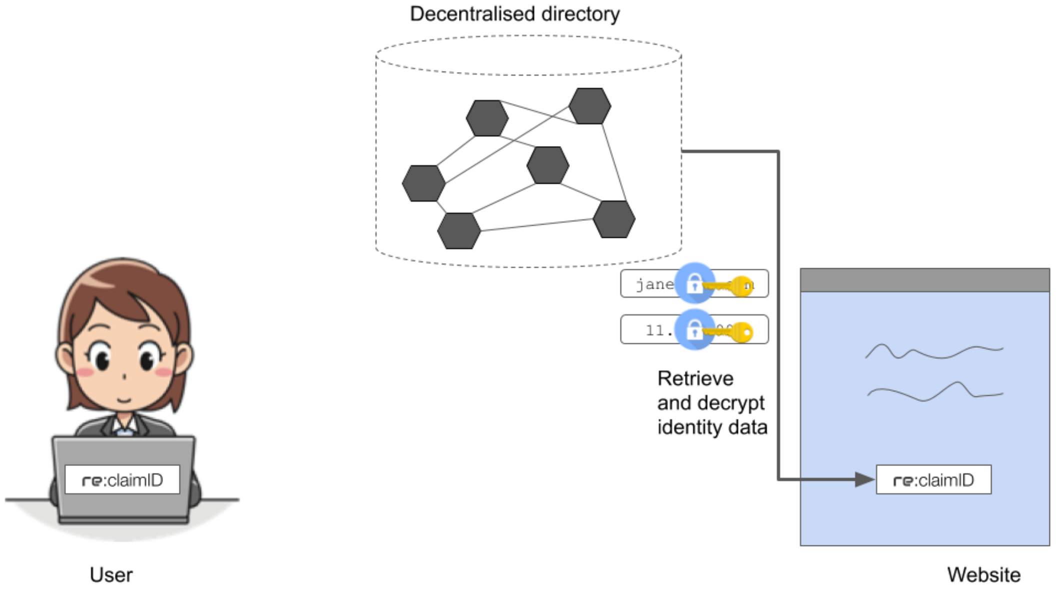
Label	Record Type	Value
l_{email}	`ATTR`	“email=alice@doe.com”
l_{name}	`ATTR`	“name=Alice Doe”
l_{dob}	`ATTR`	“dob=1.3.1987”

Authorizing access

Label	Record Type	Value
l_{email}	ATTR	“email=alice@doe.com”
l_{name}	ATTR	“name=Alice Doe”
l_{dob}	ATTR	“dob=1.3.1987”
l_{ticket}	ATTR_REF	l_{email}
	ATTR_REF	l_{dob}

- For each authorized party, the user publishes reference records under the secret label l_{ticket}
- l_{ticket} can be shared with a third party in order to authorize access to email and dob.
- Indirection enables us to revoke tickets.

Retrieve and decrypt attributes



Retrieving information

Given an identity with public key P, we can retrieve references using l_{ticket} and subsequently identity info from GNS.

$$h := Hash(l_{ticket}, P)$$

Retrieving information

Given an identity with public key P, we can retrieve references using l_{ticket} and subsequently identity info from GNS.

$$h := Hash(l_{ticket}, P)$$

$$\textbf{DHT key} \left\{ \quad q := H(hP) \right.$$

Retrieving information

Given an identity with public key P, we can retrieve references using l_{ticket} and subsequently identity info from GNS.

$$h := Hash(l_{ticket}, P)$$

$$\textbf{DHT key} \left\{ \quad q := H(hP) \right.$$

$$\textbf{Record decryption} \left\{ \begin{aligned} k &:= HKDF(l_{ticket}, P) \\ Data &:= Dec_k(Record) \end{aligned} \right.$$

Integration

- re:claimID implements the OpenID Connect protocol.
- For websites, it is just like integrating any other IdP (e.g. Google)
- For users, the authorization flow looks just like with anny other OpenID Connect IdP.

Demo

Who sais that, anyway?

Attestations

- Sometimes we need third party assurances to establish trust in identities.

Attestations

- Sometimes we need third party assurances to establish trust in identities.
- Currently, IdPs such as Facebook/Google implicitly provide this assurance (i.e. vouch for the truthfulness and correctness).

Attestations

- Sometimes we need third party assurances to establish trust in identities.
- Currently, IdPs such as Facebook/Google implicitly provide this assurance (i.e. vouch for the truthfulness and correctness).
- Claim: Those parties are not actually the authorities over (most of) your personal data! Examples:
    - Real name (State/Self-asserted/Other organization)
    - Phone number (Provider)
    - Address (State/Self-asserted)
    - Citizenship (State)
    - Age (State)
    - Email address (Mail provider)

Attestations

- What users actually need is a **collection of credentials**.

Attestations

- What users actually need is a **collection of credentials**.
- Those credentials are issued by a **variety of different entities**, including the user.

Attestations

- What users actually need is a **collection of credentials**.
- Those credentials are issued by a **variety of different entities**, including the user.
- Credentials are ideally **preserving the privacy** of the individual, e.g. using zero-knowledge proofs.

Attestations

- What users actually need is a **collection of credentials**.
- Those credentials are issued by a **variety of different entities**, including the user.
- Credentials are ideally **preserving the privacy** of the individual, e.g. using zero-knowledge proofs.
- Those ideas are already finding their way into standards:
  - W3C: “Verifiable Credentials”
  - OpenID Connect: “Aggregated Claims”

Attestations

- What users actually need is a **collection of credentials**.
- Those credentials are issued by a **variety of different entities**, including the user.
- Credentials are ideally **preserving the privacy** of the individual, e.g. using zero-knowledge proofs.
- Those ideas are already finding their way into standards:
  - W3C: "Verifiable Credentials"
  - OpenID Connect: "Aggregated Claims" $\Leftarrow$ **working on it**.

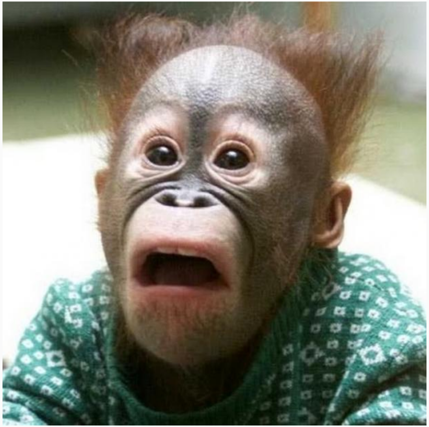
Using re:claimID

Installing re:claimID

1. Install the webextension:
   `https://addons.mozilla.org/firefox/addon/reclaimid/`

Installing re:claimID

1. Install the webextension:
   `https://addons.mozilla.org/firefox/addon/reclaimid/`
2. Install **GNUnet** $>=$ 0.11.6



Installing re:claimID

Get help installing GNUnet and/or re:claimID at our workshop today!

- Right after this.
- Time: 12:15 PM – 15:00 PM
- Location: Seminarraum

Summary

Status

- Get it at `https://reclaim-identity.io`.
- Demo websites exist:
  - `https://demo.reclaim-identity.io`
  - `https://eusec.clouditor.io`
- Roadmap:
  - User-friendly packaging (of GNUnet)
  - Ship GNUnet inside browser plugin (yes, that might even work).
  - “1.0” by end of 2019

Questions?

https://reclaim-identity.io

https://gnunet.org

schanzen@aisec.fraunhofer.de
6665 201E A925 7CC6 8FDE 77E8 8433 5131 EA3D ABF0
– or –
schanzen@gnunet.org
3D11 063C 10F9 8D14 BD24 D147 0B09 98EF 86F5 9B6A

References

1. Matthias Wachs, Martin Schanzenbach and Christian Grothoff. *A Censorship-Resistant, Privacy-Enhancing and Fully Decentralized Name System*. **13th Intern ational Conference on Cryptology and Network Security**, 2014.
2. Martin Schanzenbach, Georg Bramm, Julian Schütte. *reclaimID: Secure, Self-Sovereign Identities Using Name Systems and Attribute-Based Encryption*. **17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (TrustCom)**, 2018