Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

re:claimID

https://reclaim.gnunet.org

10. 05. 21

Hansjürg Wenger

Motivation

- Identity and Access Management is an important component of today's Internet and Web
- Service providers authenticate users to authorize access with the help of Identity Providers (IdPs)
- Big tech companies open their IdPs for use by third parties
- Their motivation is not altruistic, they can gain more information about user's online behavior
- Moreover, the users lose control over their identity attributes shared, often the whole identity is exposed
- Federated IdPs (like eduroam or edu-ID) provide better control over shared attributes but are limited to higher education
- Self-Sovereign Identity (SSI) systems like **re**:claimID help the user to get back control over his online identities and their attributes!

Learning Objectives

Know:

- Function of identity management systems
- Centralized vs decentralized identity management
- (Personal) identity attributes
- Self-Sovereign Identity (SSI)

Learn:

- how reclaimID works

Terminology I

entity	existing person, machine, service, etc.
subject	entity specifying a person
attributes	structured information about an entity i.e. first name, surname, gender, address, cell phone number, e-mail address, social security number, etc.
identity digital identity	handle for a set of attributes related to an entity [1]

Terminology II

certified attributes	identity attributes certified by a thrusted third party e.g., date of birth (for age verification) certified by a government agency or university rectorate
user identity owner	entity whose personal information is requested by the service provider for authentication and authorization
service provider relying party	provides a service for users which requires authentication before authorization for use is granted

Terminology III

Identity Provider (IdP)	manages identity information for a user or on behalf of the user
authorization server	a server used by the service provider to authenticate and authorize the user usually provided by the IdP
credentials	are used by information systems to grant access to information or other resources e.g., username/password, biometric characteristics, security tokens, etc.

Entities ⇐ Identities ⇒ Attributes

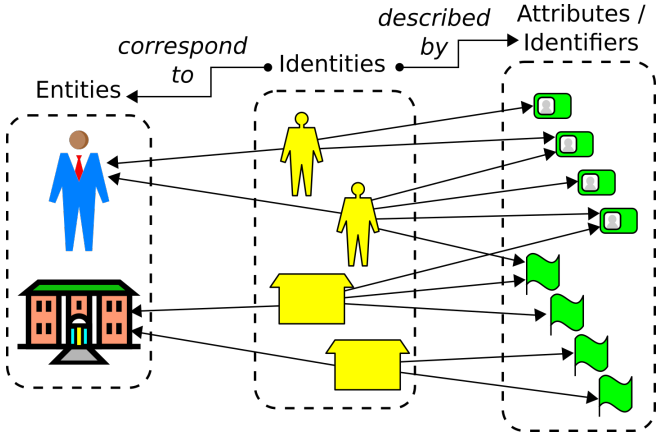

Figure 1: Identity Concept

Identity Management

- Some Internet services, Web sites, online stores, e-mail and social media providers need to know an **identity** of their users
- By proving their identity using security features (e.g. passwords and/or other factors) users are granted **access** to these services or resources
- The providers of these services must therefore collect credentials and access identity attributes
- The provider must protect any recorded identities from access by third parties in accordance with applicable data protection laws
- **Identity Management (IdM)** and **Identity and Access Management (IAM)** must therefore be implemented

Identity and Access Management (IAM)

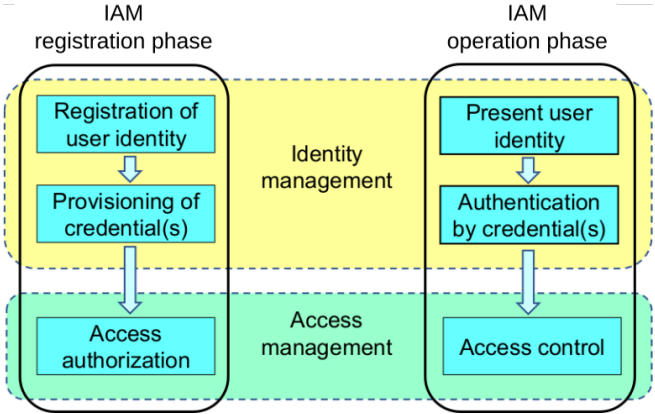

Figure 2: Identity and Access Management

IAM Registration Phase

The registration phase is used to register new users and to provision credentials for these users

- Identity management
  - Registration of user identity
    The registration process depends on the degree of verification of the attributes of an entity:
    - Some providers require strict verification of specified attributes
      (e.g., by physical presence and presentation of passport, ID card, or other certificates)
    - Others may offer self-service registration portals or Web sites
      (email address or cell phone numbers are often used to identify the user)

IAM Registration Phase (cont.)

- Identity management (cont.)
  - Provisioning of credential(s)
    Credentials and security tokens used to access the service can be provided to the user using:
    - Outband mechanisms like physical hand-out, registered mail etc.
    - One-time web links (provided by email or cell phone message) to self set the access credentials (by the user)
- Access management
  - Access authorization
    Access to services and resources provided to a user must be configured acording to existing policies (of the service provider)

IAM Operation Phase

In the operation phase the registered users can access and use the provided service.

- Identity management
  - Present user identity
    The user presents his identity and the access credentials (on Web pages or login screens)
  - Authenticate by credential(s)
    According to the presented credentials the users is authenticated by the service provider
- Access management
  - Access control
    The access to certain services and resources is authorized based on the authentication previously performed

Identity Management / Identity and Access Management

Aproaches to implement an IdM/IAM:

- Use of an own local IdM/IAM by the service provider
- Use of an existing centralized Identity Provider (IdP)
- Use of a Federated Identity Management (FIdM)
- Use of a decentralized Identity Provider (IdP)
  under control of the user, i.e. a Self-Sovereign Identity

Each of these options has its advantages and disadvantages!

Local IdM / IAM

Implementing an own local IdM/IAM by the service provider:

Advantages:

+ Full control over the identities and attributes of customers/users

Disadvantages:

- Know-how in dealing with technical, security and legal aspects of IdM must be available
- The identity verification process must be performed by the service provider
- Users must configure their identity with fresh security credentials at each service provider and cannot rely on existing ones
- Reputation risk if identities of customers/users are stolen or misused

Centralized IdM / IAM

Using an existing centralized IdM/IAM:

Advantages:

+ Easy to implement using standards like OAuth or SAML
+ The identity verification process is delegated to the IdP
+ Users can rely on existing security credentials

Disadvantages:

- Service provider must rely on the availability of a third party service
- No control by the user over the attributes passed on by the IdP
- Usage statistics and profiles can be (mis)used by the IdP (e.g. tracking, advertisement, etc.)
- When user information is stolen or hacked, it impacts all services that use the IdP

Federated Identity Management (FIdM)

By using a Federated Identity Management (FIdM), identity and attributes are stored across multiple distinct identity management systems:

Advantages:

+ The process of identity verification is delegated to the IdPs of the participating parties
+ Users can rely on existing security credentials (SSO)
+ Users have control over attributes given to participating service providers

Disadvantages:

- Know how in dealing with technical, security and legal aspects of IdM must be available
- In addition know how to federate the IdPs must also be available
- When user information is stolen or hacked, it affects all services that rely on the FIdM

Decentralized IdM / IAM

Using a decentralized IdM/IAM:

Advantages:

+ The identity verification process is delegated to the user's IdP
+ Users have full control over used security credentials and attributes passed on
+ Identity and attribute verification process can still be delegated to trusted third parties

Disadvantages:

- (asynchronous) access to identity information must be implemented using a DHT or a block chain (persistence)
- The service provider must rely on the availability of these services
- Trust must be established "out-of-band"

Self-Sovereign Identity

Self-Sovereign Identity (SSI):

- decentralized approach to digital identities where every user acts as their own IdP
- gives individuals/users full control of their digital identities

Several solutions are under development:

- Blockchain-based: Sovrin, uPort, NameID, etc.
- W3Cs Decentalized IDentifier (DID)
  `https://www.w3.org/TR/did-core/`
- **re**:claimID
  `https://reclaim.gnunet.org/`

This presentation focuses on the design and functionality of **re**:claimID.

re:claimID a nutshell

re:claimID = Decentralized directory service + Cryptographic access control

Design of **re**:claimID

Goals of **re**:claimID:

- users manage their attributes in a **namespace**
- users can selectively grant access to other parties
- the system ensures that attributes can be accessed asynchronously (i.e. whenever needed, even if the user is offline)
- trusted third parties can optionally guarantee authenticity
- integration into a standardised authorization and authentication protocol (OpenID Connect)
- access to attributes is authorized and enforced using Attribute-Based Encryption (ABE)

Directory Services and Name Systems → Namespaces

Directory Services and Name systems consist of **namespaces**
Namespaces:

- are owned by users or legal entities
- are managed by their owner
- contain name-value mappings (in form of **resource records**)

The owner of a namespace issues **attributes** in it.

Name systems therefore provide:

- storage-,
- resolution-, and
- delegation-

mechanisms for **self-issued** attributes.

Directory Services

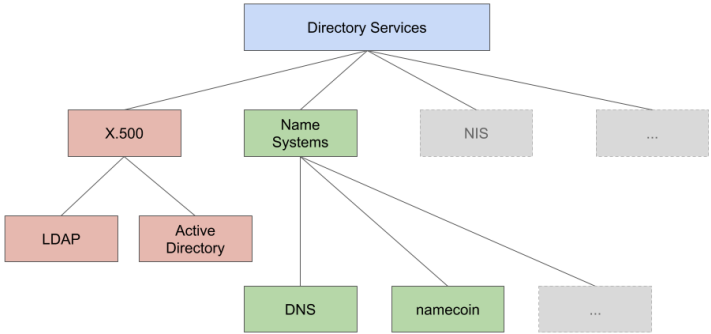

Figure 3: Directory Services

Name System Decision

re:claimID needs a name system with the following requirements:

- Decentralized
- Secure but with open name registration
- Supports encrypted and signed resource records
- Protects identity data from unwanted disclosure
- Allows users to enforce access control

The **GNU Name System (GNS)** provides:

- A decentralized name system
- A cryptographic access control layer
- User-defined namespaces

⇒ the **GNU Name System (GNS)** is used for **re**:claimID

Managing and Publishing Identity Information

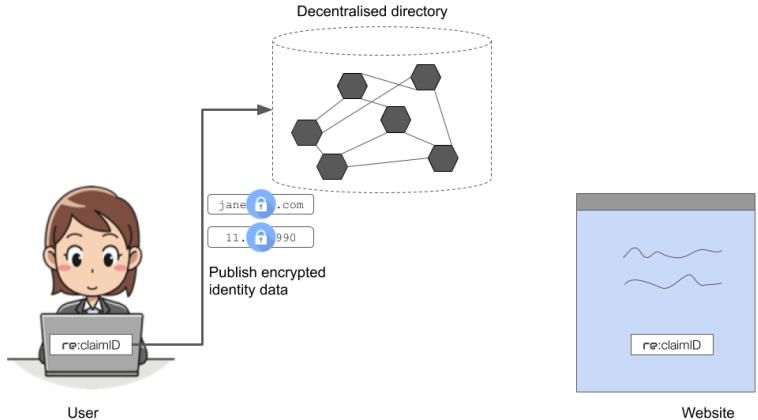

Figure 4: Publish Identity Information

The GNU Name System

- In GNS, a namespace is defined by a public/private Curve25519 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 under label l signed with private key $H(l, P)x$ are published in a distributed hash table (under key $q := H(H(l, P)P)$).
- Any peer is able to verify the signature as the corresponding derived public key $H(l, P)P$ is also published
- Records can only be resolved and decrypted if the public key P and the label l are known.

⇒ Namespaces **cannot** be enumerated and queries/responses **cannot*** be observed

*Unless label l and identity key P are known

Identity Attributes in GNS

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

Label (l)	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

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

DHT key $\{$ $q := H(hP)$

Encryption $\begin{cases} k := HKDF(l_{attr}, P) \\ Record := Enc_k(Data) \end{cases}$

Signature $\begin{cases} d := h \cdot x \bmod n \\ Signature = Sig_d(Record) \end{cases}$

Authorizing Access

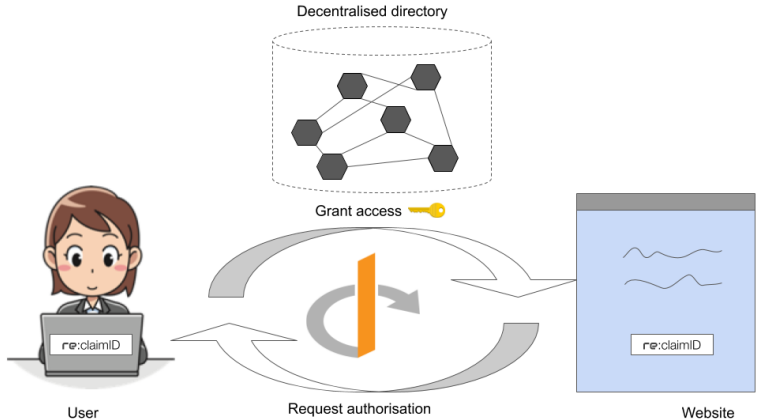

Figure 5: Authorizing Access

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”
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 **re**:claimID to revoke tickets.

Retrieve and Decrypt Attributes

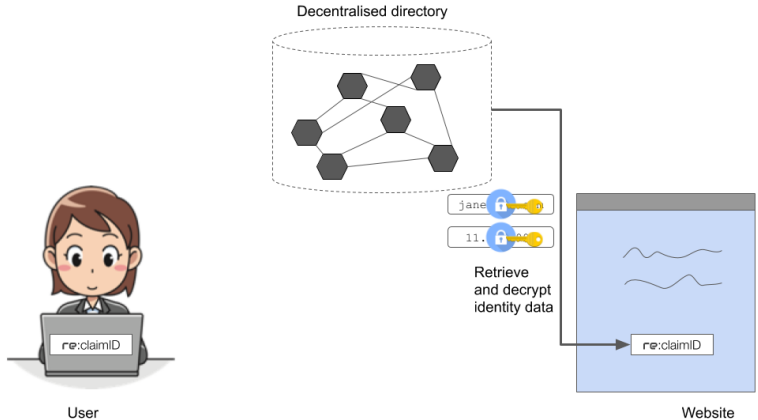

Figure 6: Retrieve Attributes

Retrieving Information

Given an identity with public key P, we can retrieve references using I_{ticket} and subsequently the associated identity attributes from GNS:

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

DHT key

$$q := H(hP)$$

Record decryption

$$\begin{cases} k := HKDF(I_{ticket}, P) \\ Data := Dec_k(Record) \end{cases}$$

Integration

- **re**:claimID implements the OpenID Connect protocol
- For Web sites, it is just like integrating any other IdP
- For users, the authorization flow looks just like with any other OpenID Connect IdP

Use Case "WooShop"

Figure 7: WooShop

"WooShop" – Goals

Goal:

- Build a webshop using the popular open source CMS **WordPress** with the eCommerce Plugin **WooCommerce**
- Use **re**:claimID as identity provider for shop users
- Use **GNU Taler** as payment service

"WooShop" – Prerequisites

Prerequisites:

- A running GNU Name System installation on the webserver and the user/browser system
- The **re**:claimID browser plugin
  (available for Chrome and Firefox)
- The GNU Taler Wallet browser plugin
  (available for Chrome and Firefox)
- A GNU Taler payment plugin for the WooCommerce web shop

Instructions about the installation can be found on the **re**:claimID Web site.

"WooShop" Architecture

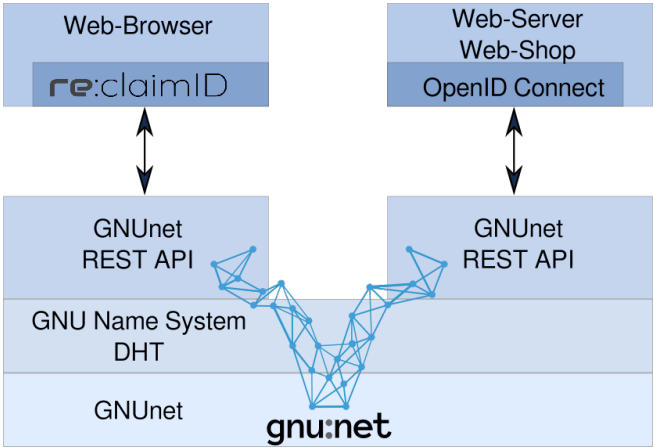

Figure 8: WooShop Architecture

"WooShop" Demo Video

A demo video of the Taler and WooShop workflow can be found here:

https://gnunet.org/ schanzen/2021-01-18-reclaimID-Taler-Shopping.webm

Technologies I

- nameID
  `https://nameid.org/`
- OAuth 2.0 RFC6749[2]
- OpenID Connect 1.0
  `https://openid.net/connect/`
- **re**:claimID
  `https://reclaim.gnunet.org`
- GNUnet
  `https://gnunet.org`

Technologies II

- GNU Name System
  `https://gnunet.org/gns.html`
- GNU Taler
  `https://taler.net`
- Decentalized IDentifier
  `https://www.w3.org/TR/did-core/`
- WordPress
  `https:/wordpress.com`
- WooCommerce
  `https://woocommerce.com/`

Figures I

- Figure 1 "Identity Concept" [7/46]
  Audun Jøsang, CC BY 3.0 (modified)
  <https://creativecommons.org/licenses/by/3.0>,
  via Wikimedia Commons
- Figure 2 "Identity and Access Management" [7/46]
  Josang, CC BY-SA 4.0 (modified)
  <https://creativecommons.org/licenses/by-sa/4.0>,
  via Wikimedia Commons
- Figure 3 "Directory Services" [22/46]
  © M. Schanzenbach, Fraunhofer AISEC
- Figure 4 "Publish Identity Information" [24/46]
  © M. Schanzenbach, Fraunhofer AISEC

Figures II

- Figure 5 "Authorizing Access" [28/46]
  © M. Schanzenbach, Fraunhofer AISEC
- Figure 6 "Retrieve Attributes" [30/46]
  ls
- Figure 7 "WooShop" [33/46]
  © 2021, WGH1, BFH
- Figure 8 "WooShop Architecture" [36/46]
  © 2021, WGH1, BFH

Acronyms I

ABE	Attribute-Based Encryption.
CMS	Content Management System.
DHT	Distributed Hash Table.
DID	Decentalized IDentifier.
FIdM	Federated Identity Management.
GNS	GNU Name System.
IAM	Identity and Access Management.
IdM	Identity Management.
IdP	Identity Provider.

Acronyms II

OAuth	OAuth 2.0 Authorization Framework [2].
OIDC	OpenID Connect.
SAML	Security Assertion Markup Language.
SSI	Self-Sovereign Identity.
SSO	Single Sign-On.
W3C	World Wide Web Consortium.

References I

- Security techniques — entity authentication assurance framework. Standard ISO/IEC 29115:2013, International Organization for Standardization, Geneva, CH, 2013.
- D. Hardt (Ed.). The OAuth 2.0 Authorization Framework. RFC 6749 (Proposed Standard), October 2012. Updated by RFC 8252.
- Martin Schanzenbach. **re**:claimID: A gnunet application for self-sovereign, decentralised identity management and personal data sharing. url: https://media.ccc.de/v/ds19-10383-re_claimid, 9 2019.

References II

Martin Schanzenbach, Georg Bramm, and Julian Schütte.
re:claimID: Secure, self-sovereign identities using name systems and attribute-based encryption.
CoRR, abs/1805.06253, 2018.

Acknowledgements

- Partly based on
  - **re***:claimID: Secure, Self-Sovereign Identities using Name Systems and Attribute-Based Encryption* [4]
  - **re***:claimID: a gnunet application for self-sovereign, decentralised identity management and personal data sharing* [3]

- Special thanks to **NGI TRUST** for funding the project!