Aalto University

# Electronic identity

**Tuomas Aura**

CS-C3130 Information security

Aalto University 2023

# Outline

1. Web single sign-on within organization
   - SAML and Shibboleth
2. Web single sign-on on the Internet
   - OAuth and OpenID Connect
3. Authentication to government services
   - Strong electronic identity

# Single sign-on (SSO)

- One authentication credential for many services. Why?
- For users:
  - Fewer accounts and passwords to remember
  - More convenient service access
  - Lower risk from old, forgotten accounts
- For intranet and extranet services within an organization:
  - Central user management within organization
- For public web sites:
  - Outsourcing credential provisioning and authentication
  - Tracking web users for advertising

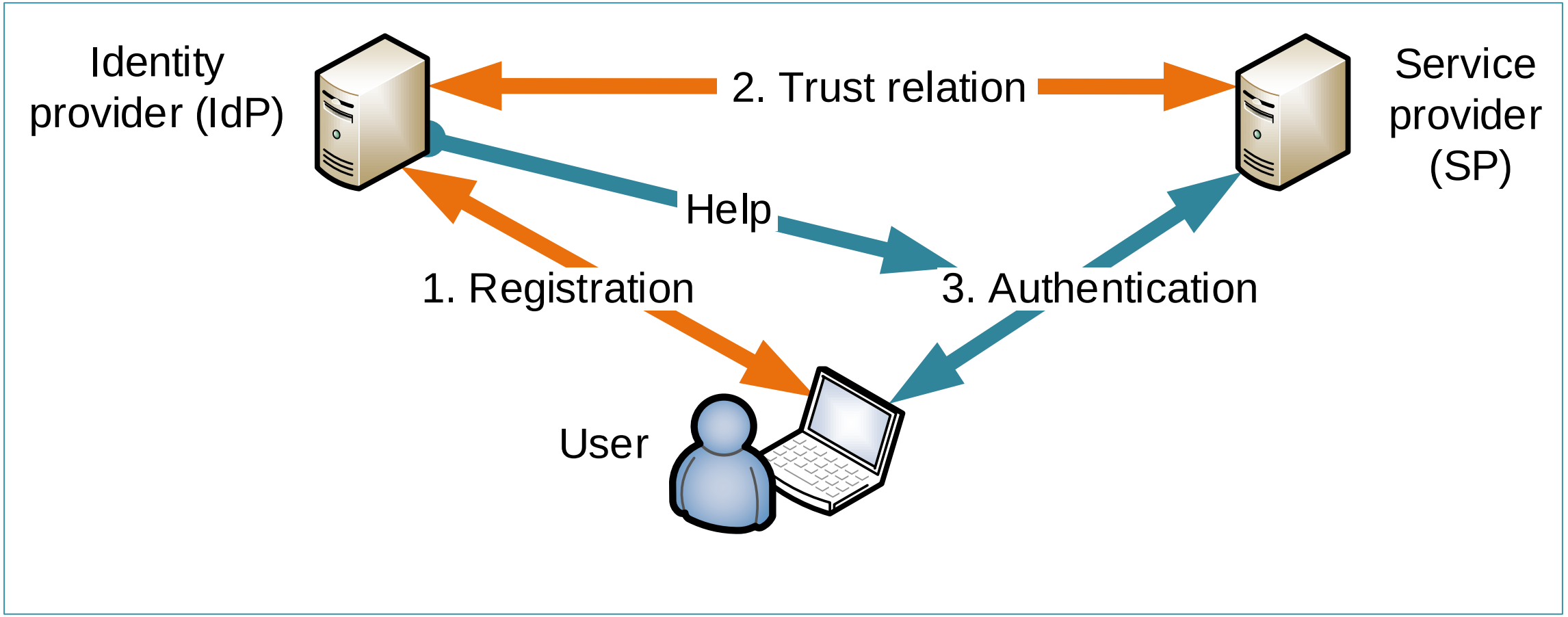
# WEB SINGLE SIGN-ON WITHIN ORGANIZATION

# SSO has a long history

Extra material

- Organizational and federated SSO:
  - Kerberos v5 (1993), GSSAPI, Windows AD (2000)
  - SAML 1.0, 1.1, 2.0 (2002-2005)
  - Shibboleth 1.3 (2005), 2.0 (2008) based on SAML

# General web SSO architecture




- Similar architecture, different terminology in each SSO standard

# How web SSO works

All messages must be protected with HTTPS

Web browser

Service provider (SP)

Identity provider (IdP)

IdP discovery: Which IdP to use?

HTTP Redirection or POST with client-side JavaScript (or user click)

1. Access request

2. HTTP GET or POST

3. User authentication: e.g. password

4. HTTP (GET or) POST, authentication result

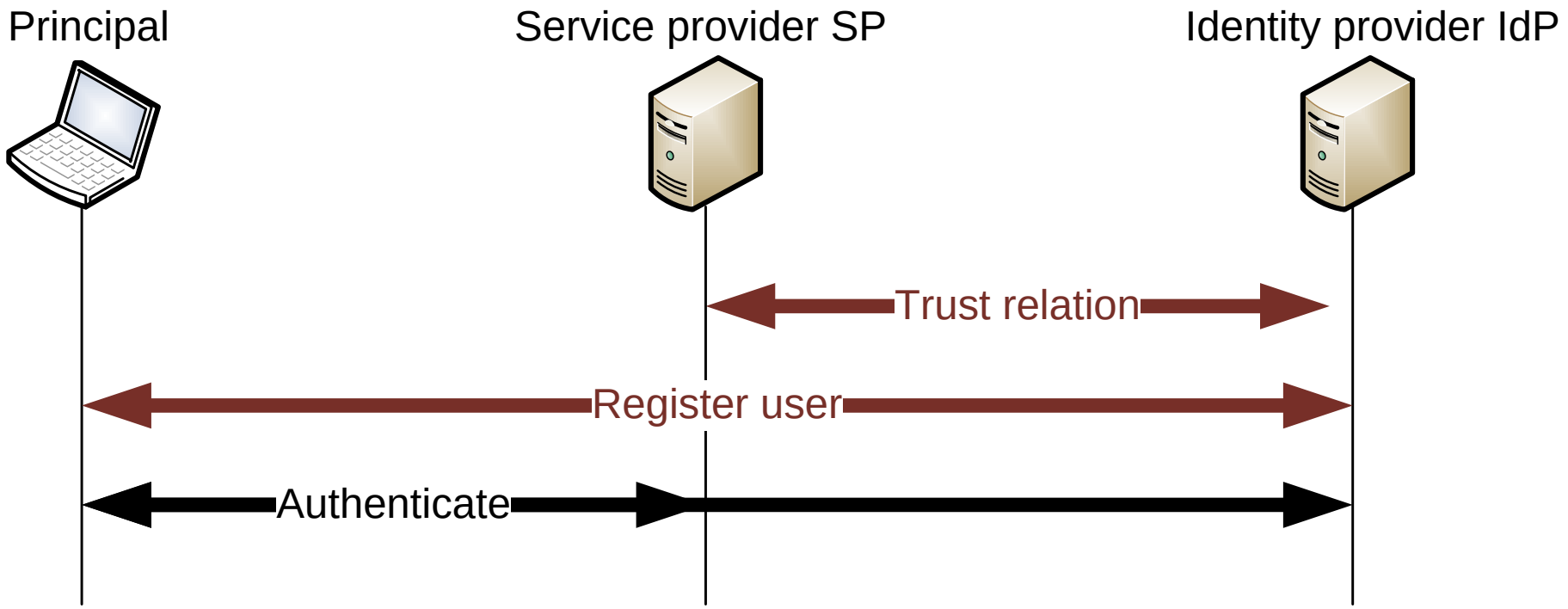
5. Service access

SP and IdP communicate via the web browser in the GET query or POST form; in some systems, also directly between servers

Cryptography needed to prevent user from tampering with the authentication results

# SAML AND SHIBBOLETH

# SAML 2.0 architecture




- Service provider (SP) and identity provider (IdP) establish a trust relation by exchanging metadata
- Principal (= user, subject) registers with the IdP

# Security assertion markup language, SAML 2.0

- Focus on XML-based assertions:

```
<saml2:Attribute FriendlyName="cn" Name="urn:oid:2.5.4.3"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
      <saml2:AttributeValue xmlns:xs=http://www.w3.org/2001/XMLSchema
      xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
      xsi:type="xs:string">Teemu Teekkari</saml2:AttributeValue>
</saml2:Attribute>
<saml2:Attribute FriendlyName="mail" Name="urn:oid:0.9.2342.19200300.100.1.3"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
      <saml2:AttributeValue xmlns:xs=http://www.w3.org/2001/XMLSchema
      xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
      xsi:type="xs:string">teemu.teekkari@aalto.fi</saml2:AttributeValue>
</saml2:Attribute>
```

- Federated identity management:
  - Cross-organizational authentication based on contractual relations

# SAML 2.0

- SAML is a complex family of specifications:
  - Assertions are statements by IdP about a principal, in XML
  - Protocols define message flows for requesting assertions
  - Bindings define message transport over HTTP, SOAP etc.
  - Profiles define useful combinations of the above
  - Metadata defines trust relations
  - So many possible combinations that SAML implementations are generally not interoperable
- For web SSO, SAML web browser SSO profile
  - Bindings: redirect, post, artifact

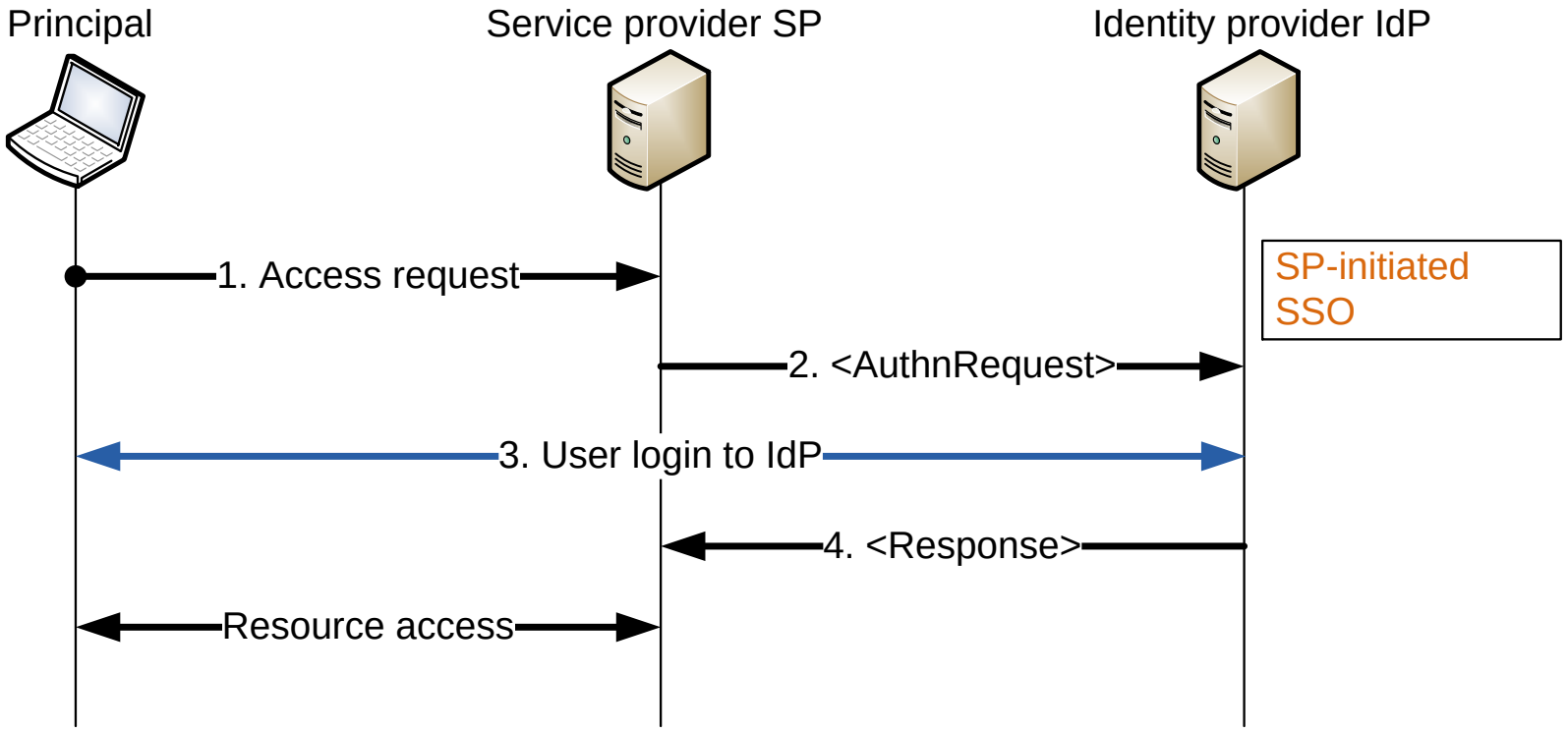
# SAML web browser SSO profile

Extra material

- IdP-initiated or SP-initiated SSO:
  - User first logs into the IdP, or first connects to SP
- Bindings to HTTP messages
  - Redirect: message from SP to IdP is sent in GET URL via user's browser, with help of HTTP redirection
  - POST: message between SP and IdP is sent in an HTTP form via user's browser
  - Artifact: reference number sent as redirect, and the actual message retrieved directly from the sender (artifact = random number)

# SAML web browser SSO profile

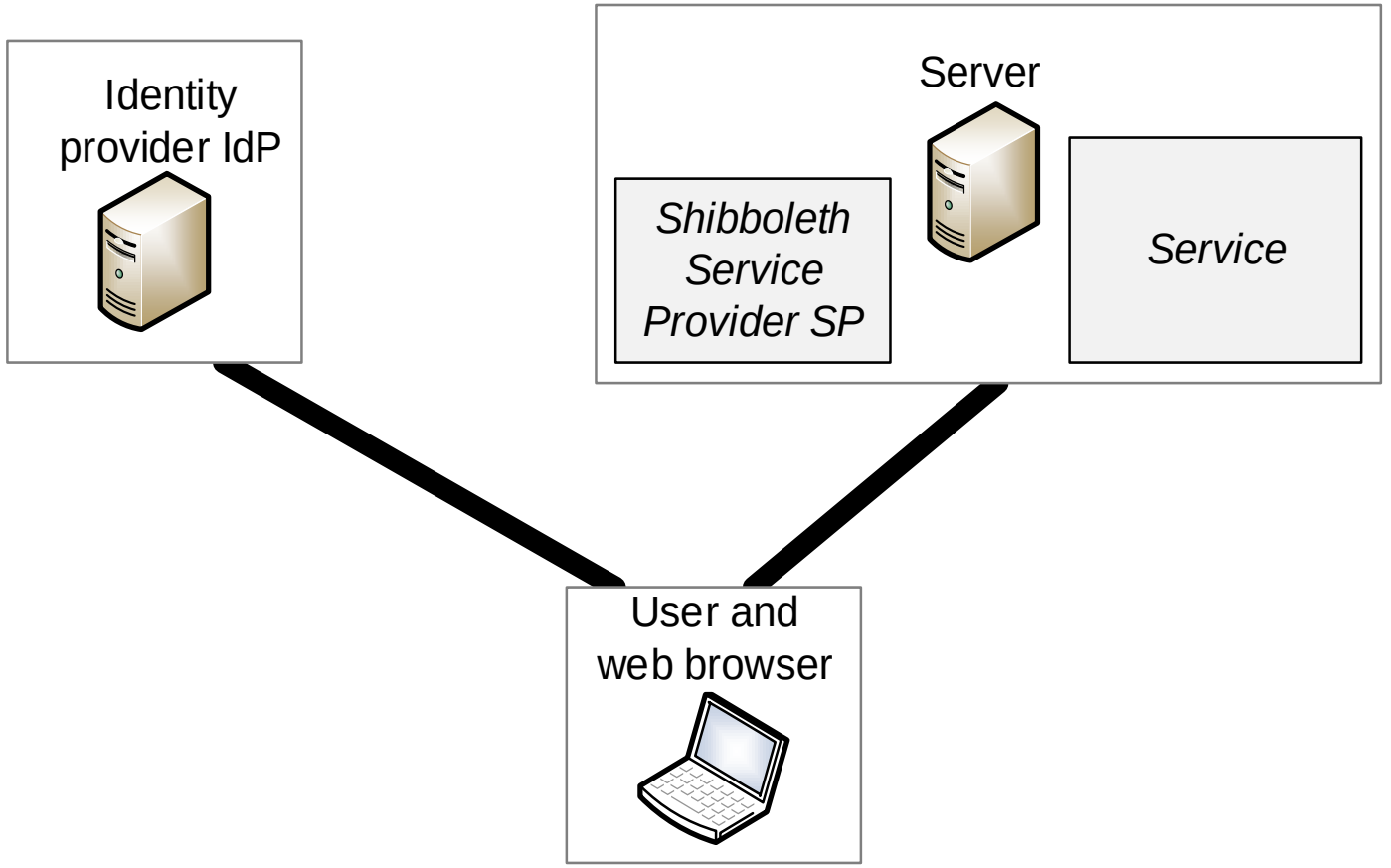
Extra material




- Protocol for SP-initiated SSO: AuthnRequest and Response
- How to send these messages over HTTP?
  → Choose bindings

# Shibboleth 2

- Open-source implementation of SAML 2.0
  - SAML web browser SSO profile
  - Used by research and educational institutions

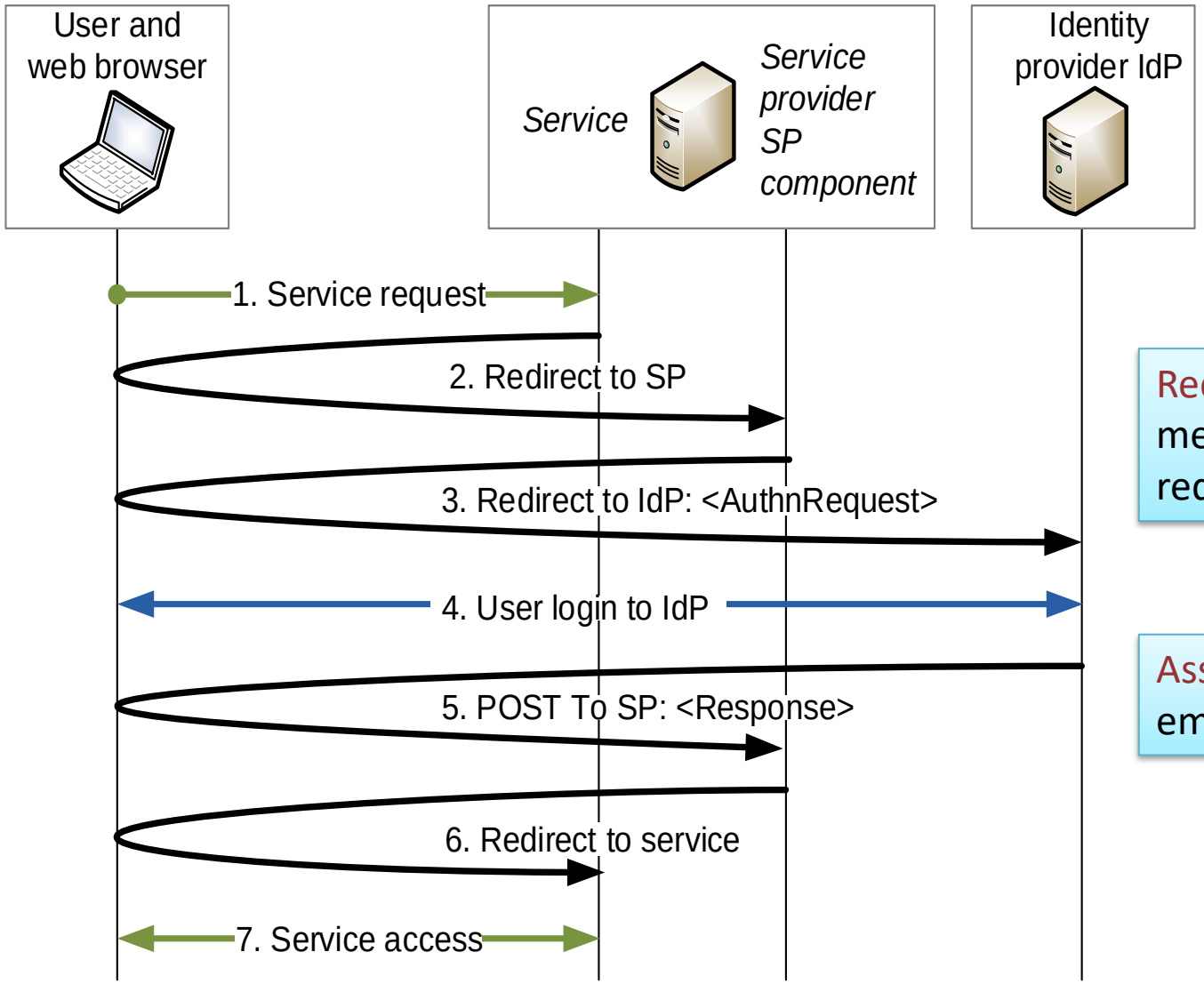


# Shibboleth federation

- Federation is a group of IdPs and SPs that
  - share metadata (including public keys) in one signed file
  - agree on an attribute schema for assertions
  - agree on a CA for TLS
  - have a service agreement that sets out rules for the federation
- Examples: HAKA federation, eduGAIN
- When authenticating, where are you from (WAYF) page for IdP selection

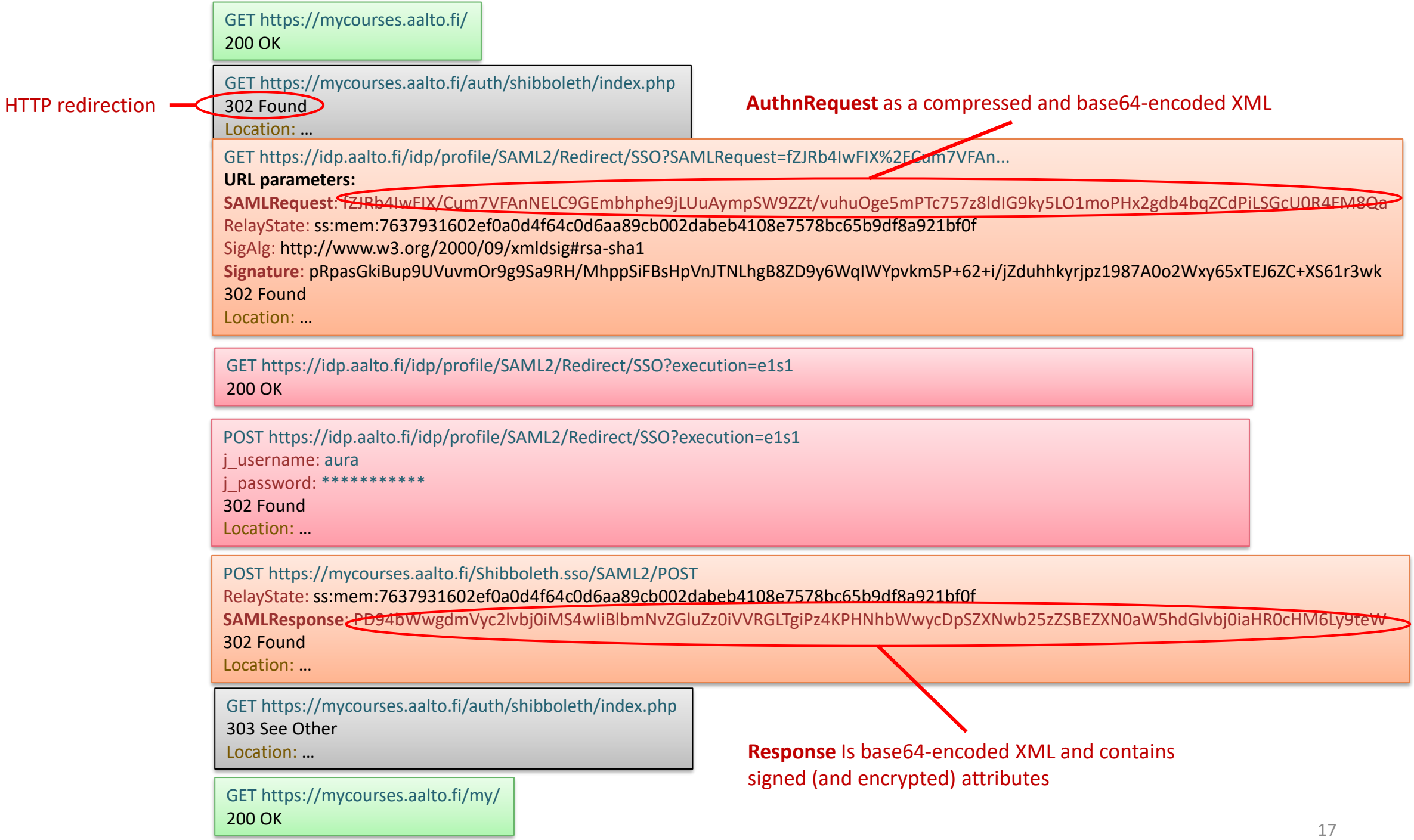
University members can try attribute test service: https://rr.funet.fi/haka/

# Shibboleth protocol

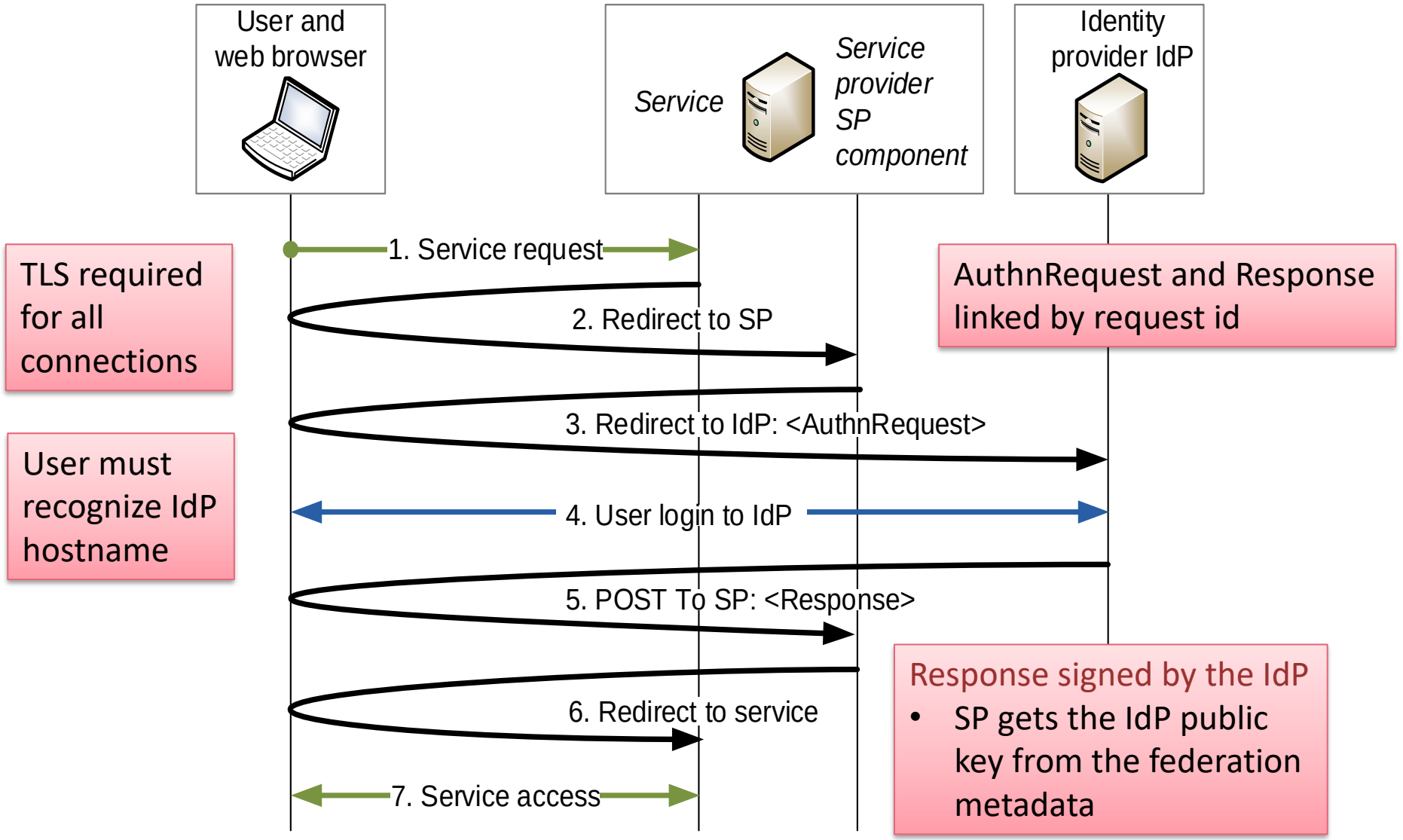



Redirect-POST binding: messages sent with HTTP redirect via user's browser

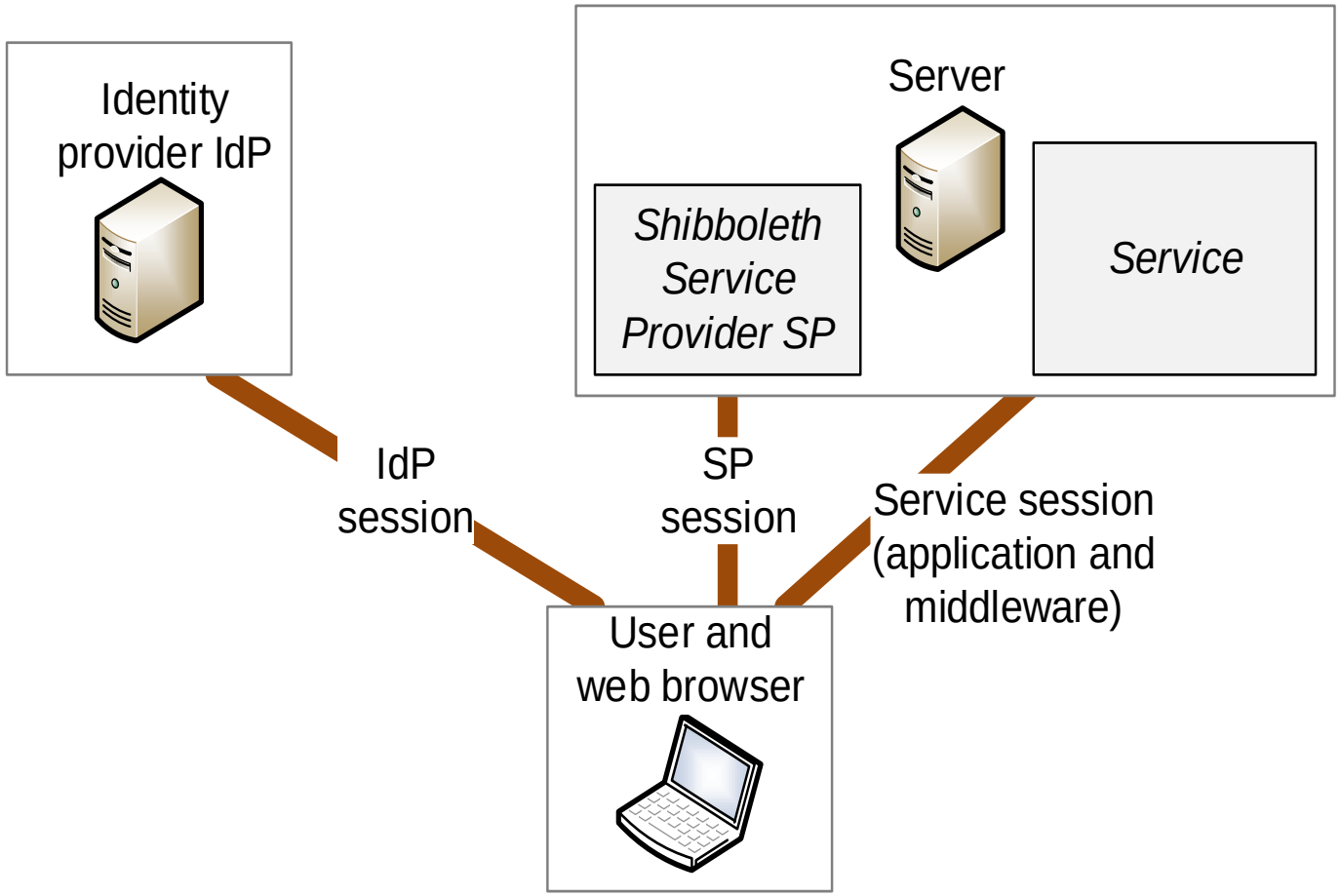
Assertions are XML embedded in the Response


GET https://mycourses.aalto.fi/
200 OK
HTTP redirection
GET https://mycourses.aalto.fi/auth/shibboleth/index.php
302 Found
Location: ...
AuthnRequest as a compressed and base64-encoded XML
GET https://idp.aalto.fi/idp/profile/SAML2/Redirect/SSO?SAMLRequest=fZJRb4IwFIX%2FCum7VFAn...
URL parameters:
SAMLRequest: fZJRb4IwFIX/Cum7VFAnNELC9GEmbhphe9jLUuAympSW9ZZt/vuhuOge5mPTc757z8ldIG9ky5LO1moPHx2gdb4bqZCdPiLSGcU0R4FM8Qa
RelayState: ss:mem:7637931602ef0a0d4f64c0d6aa89cb002dabeb4108e7578bc65b9df8a921bf0f
SigAlg: http://www.w3.org/2000/09/xmldsig#rsa-sha1
Signature: pRpasGkiBup9UVuvmOr9g9Sa9RH/MhppSiFBsHpVnJTNLhgB8ZD9y6WqIWYpvkm5P+62+i/jZduhhkyrjpz1987A0o2Wxy65xTEJ6ZC+XS61r3wk
302 Found
Location: ...
GET https://idp.aalto.fi/idp/profile/SAML2/Redirect/SSO?execution=e1s1
200 OK
POST https://idp.aalto.fi/idp/profile/SAML2/Redirect/SSO?execution=e1s1
j_username: aura
j_password: ***********
302 Found
Location: ...
POST https://mycourses.aalto.fi/Shibboleth.sso/SAML2/POST
RelayState: ss:mem:7637931602ef0a0d4f64c0d6aa89cb002dabeb4108e7578bc65b9df8a921bf0f
SAMLResponse: PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPHNhbWwycDpSZXNwb25zZSBEZXN0aW5hdGlvbj0iaHR0cHM6Ly9teW
302 Found
Location: ...
Response Is base64-encoded XML and contains signed (and encrypted) attributes
GET https://mycourses.aalto.fi/auth/shibboleth/index.php
303 See Other
Location: ...
GET https://mycourses.aalto.fi/my/
200 OK

# Shibboleth security



# Sessions in Shibboleth




- Many different sessions → logout is confusing (which sessions end?)
- Logout is a problem in all SSO systems: hard to implement, and not obvious how it should work

# Sessions in Shibboleth

- Shibboleth implements two kinds of sessions:
  - IdP session between browser and the IdP implemented with IdP cookies
    → user only needs to type in password once
  - SP session between browser and the SP component with SP cookies
- Additional application sessions:
  - Applications and web-application frameworks implement sessions using cookies or with fields in URLs or web forms
- No single logout
  - Logging out of SP does not usually log the user out of the IdP
    → can log back to SP without password
  - Logging out of IdP does not log the user out of SPs
  - Logging out of one SP does not log the user out of other SPs
  - Application sessions complicate the situation further

  → Shibboleth logout behavior is difficult to understand

# WEB SINGLE SIGN-ON ON THE INTERNET

# History of web SSO

Extra material

- Long struggle for control over web SSO:
  - Microsoft Passport (~1999), LiveId, CardSpace, Microsoft Account
  - Liberty Alliance SAML-based alternative to Passport (2001-)
  - OpenID (2005), 2.0 (2007), focus on openness
  - OAuth 1.0 (2010), 2.0 (2012), authorization only
- Finally, wide adoption:
  - Facebook, Google+, Twitter, Microsoft finally accepted by SPs and users
  - OpenID Connect (2014) protocol based on OAuth 2.0 but for identity authentication, support for web-browser and mobile-app clients

# OAUTH 2.0 AUTHORIZATION

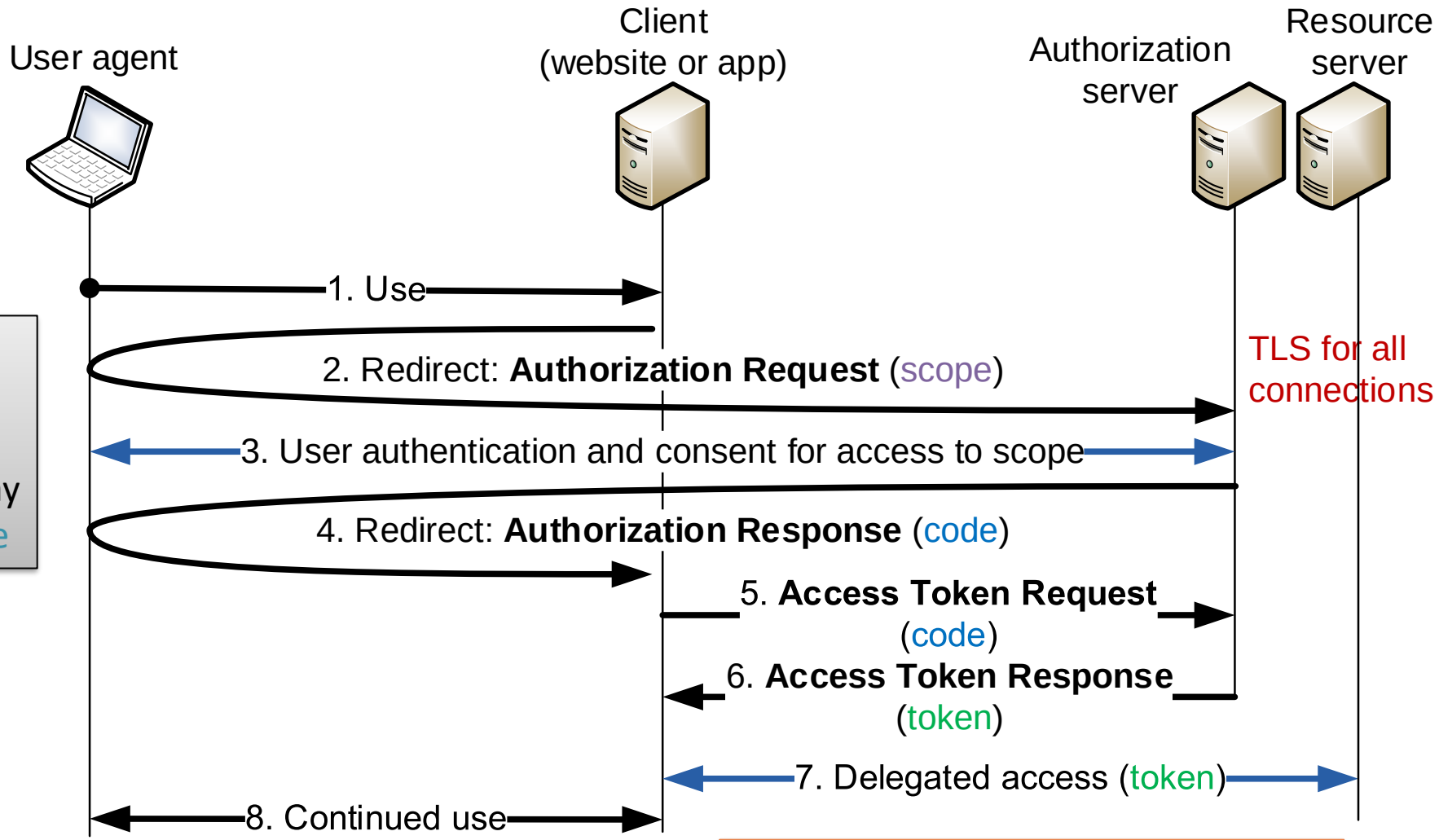
OAuth 2.0 is not for authentication or SSO. We cover it before OpenID Connect because OpenID Connect builds on the OAuth 2.0 standard

# OAuth 2.0

- OAuth was designed for authorization (i.e., delegation)
  - User authorizes a web app, mobile app, or another online service service to access their data in an online service
- Examples:
  - Authorize a website or app to update Facebook for you
  - Authorize continuous integration tool to monitor a GitHub repository
- Standardized by IETF (RFC 6749, RFC 6750)
  - App developers can use a well-designed, secure protocol
  - No interoperability: many incompatible ways to implement

# OAuth 2.0 authorization

(RFC 6749 section 4.1)




Resource owner (user) authorizes client to access the resource; may restrict the scope

TLS for all connections

Bearer token: access to the resource by just presenting the token

# Why OAuth 2.0

- Authorization is better than sharing your user account and credentials with third-party services
  - Password sharing avoided
  - Access can be revoked without changing user password
  - Scope of the delegated access rights can be limited
- Can OAuth it be used for user authentication?
  - Maybe, but early attempts had serious security flaws

# Using OAuth for authentication?

Extra material

- The message flow in OAuth looks like OpenID or SAML → services were tempted to use OAuth also for authentication
  - User would prove its identity by delegating access to a (dummy) resource associated with the user account
- In principle, this was a bad idea:
  - OAuth access token enables client to access a resource on the service provider
  - The client in OAuth does not know or care who gives it the token, as long as the token works for accessing SP
  - The protocol does not prevent the client from sharing the token with others, e.g., subcontractors or server instances

  → Malicious client can impersonate user to other clients
- Authentication based on OAuth has now been standardized as OpenID Connect, which prevents the token forwarding attack
  - http://openid.net/developers/specs/

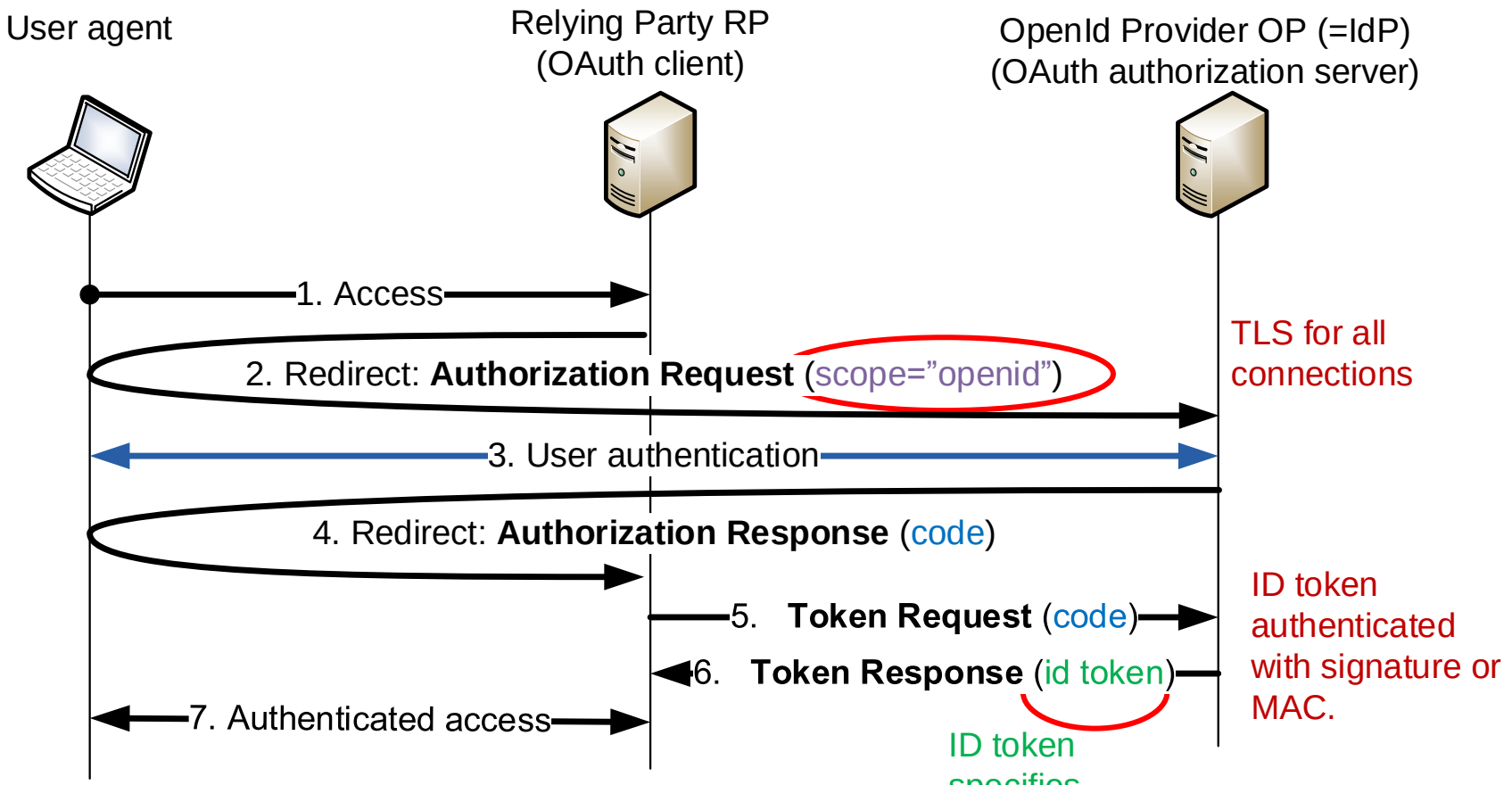
# OPENID CONNECT

# OpenID Connect

- Authentication built on OAuth 2.0, JavaScript, REST APIs, JSON data formats
  - OAuth access token is replaced with id token, which is not a bearer token; the token is bound to the specific relying party (OAuth client)
- Implementations are usually not interoperable
  - Many options: MAC with pre-shared key between OP and RP vs. JSON Web Signature on messages
  - OP provides both server-side RP code and client-side JavaScript
    → no need for interoperability

# OpenID Connect

Authorization code flow – one of many protocol variants




Important difference to OAuth: not a bearer token!
binding the token to the client prevents token forwarding

# What does “open” mean?

Extra material

- Original OpenID gave freedom to the user:
  - Anyone could become an identity provider
  - User could choose any identity provider
  - Services supposed to accept any identity and IdP chosen by the user
  - Worked on any web browser without proprietary software (originally even no JavaScript or TLS)
- OpenID Connect is not open in any sense
  - No interoperability between competing implementations
  - Web sites decide which IdPs they support

# Common features of Internet SSO !

- Authentication
  - RP learns user identity (username or some other id)
- Attribute attestation
  - IdP tells RP further information about the user
- Authentication-server discovery
  - WAYF page, or automatic way to resolve the user identity to IdP address
  - For automatic discovery, user identifiers must be globally unique, e.g., name@domain or URL
- Client registration
  - Many IdPs  allow creation of new accounts online
  - Difficult to combine with proper identify proofing
- Authorization
  - User delegates access rights to a client app or to another service
  - Best implemented as a separate step from identity authentication: e.g., OpenID Connect for authentication + OAuth 2.0 for delegating access rights

# STRONG ELECTRONIC IDENTITY

# For comparison


- Passport and identity card are traditional methods for authenticating a person
  - Photo and signature
- Biometric passport (ICAO9303)
  - Chip contains all the same information as printed passport (name, birth date etc.)
  - Biometric data: photo and fingerprint
  - Electronic signature for authenticity
- Access to passport chip for authorized readers only
  - Active authentication with signature optional and designed for authentication only

→ Cannot be used online

# Strong authentication

- Goal: online authentication equivalent to checking national identity card or passport
- Is it needed?
  - Services are moving to the web, and strong authentication is required by law for access to government services and personal information
  - Saves travel and customer service cost in the initial id check for new users, e.g., buying an insurance or enrolling to university
  - Increasing trust in online commerce (?)

# Strong authentication regulation

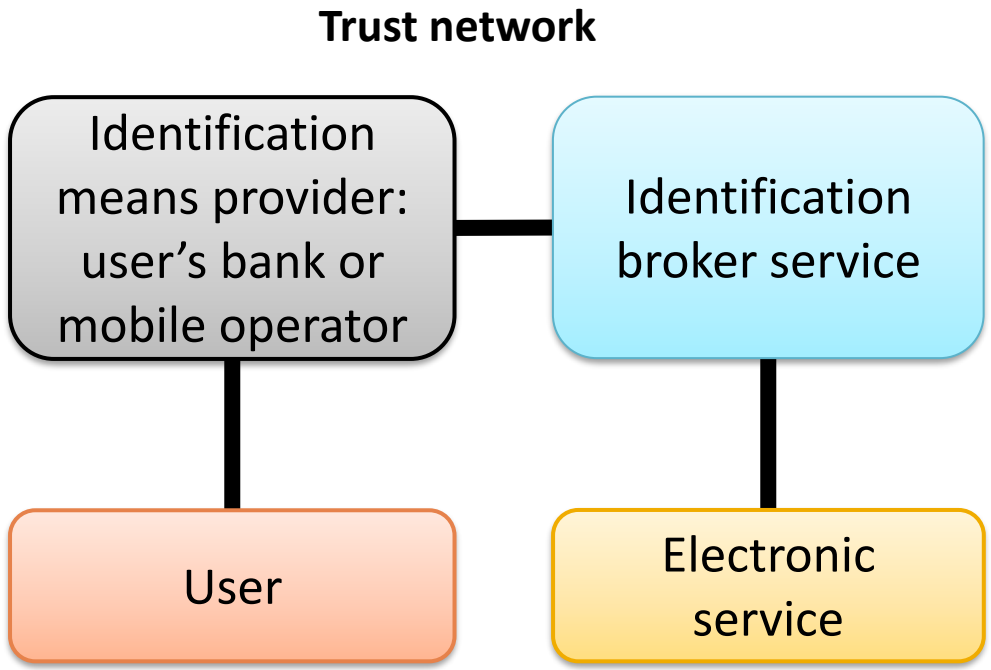
- European eIDAS regulation
  - Implemented in Finnish law: *Laki vahvasta sähköisestä tunnistamisesta ja sähköisistä luottamuspalveluista*

  https://www.finlex.fi/fi/laki/ajantasa/2009/20090617 (617/2009, updated 2016)
- Two-method authentication, any two of:
  - something you know (password or PIN code)
  - something you have (physical token)
  - something you are (biometrics)
- Identity proofing in person or electronically

# Finnish Trust Network (FTN)

- Regulated electronic authentication network:
  1. Identification means providers (banks, mobile operators, population register)
  2. Identification broker services
  3. Electronic services

  – Identification means providers must work with all approved brokers (at regulated price)
  – Electronic service chooses one broker
  – Supervised by Finnish Transport and Communications Agency (Traficom)

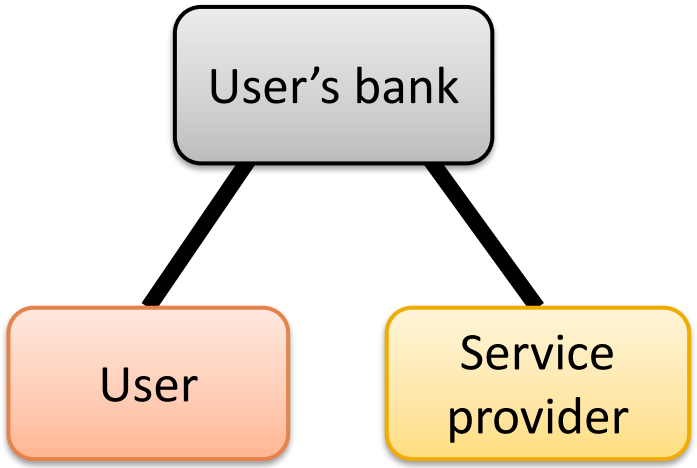



- Protocol between broker and identity means provider is OpenID Connect or SAML2
  – Identification means provider is the OpenID provider (IdP); broker is the relying party

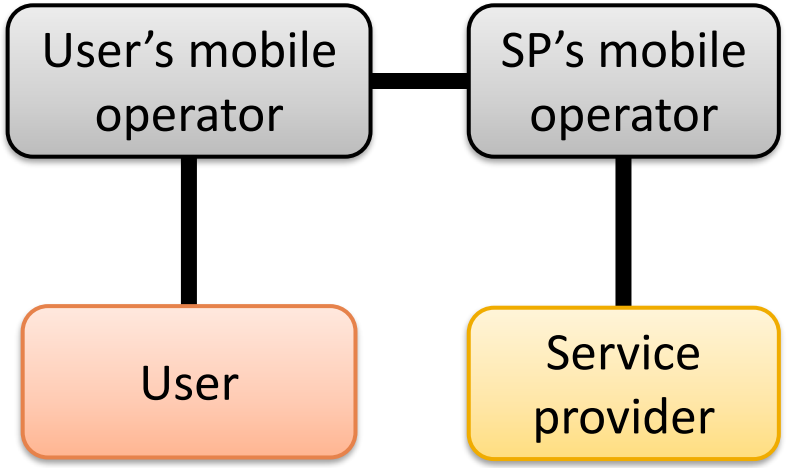
- Protocol between electronic service and broker is typically also OpenID Connect
  – Electronic service forwards the browser to broker, broker forwards to IdP

# Older models

**TUPAS**




**Mobile certificate**




## Three-corner model

- Each service had to set up a shared key with each bank; only the largest banks were supported in all online services
- TUPAS has been mostly discontinued but may still be used for bank login

## Four-corner model

- "Roaming" agreements between operators

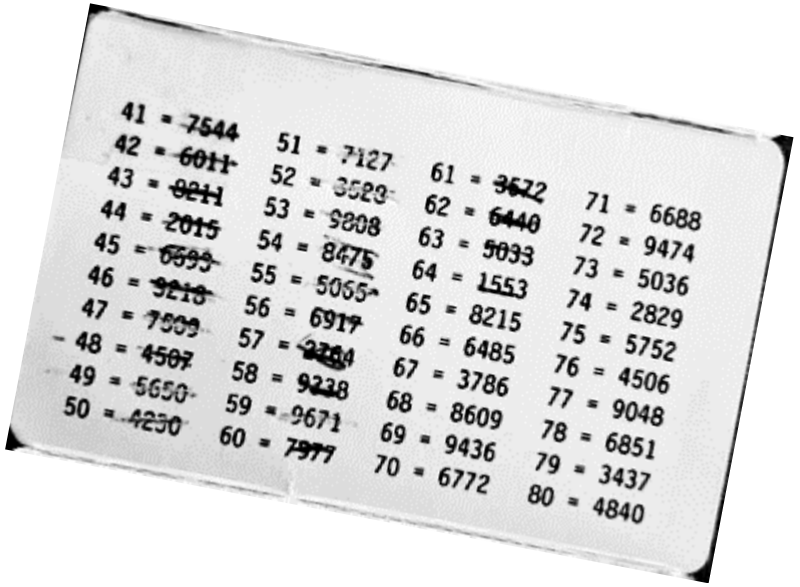
# Finnish electronic identity card


- Finnish identity cards (HST-kortti) have a chip with two RSA key pairs and certificates:
  - (1) authentication and encryption key
  - (2) Signature key
  - https://dvv.fi/fineid-maaritykset
- Digital and Population Data Services Agency (VRK) acts as the CA
- Not popular – why?
  - Similar cards widely used by Finnish government employees and healthcare workers

Example of how poor deployability can cause failure of otherwise good technology

# Bank authentication

- Banks used to have one-time code lists
- Now a mobile app where you approve the transaction
  - More secure because it cannot be copied and user without the PIN code



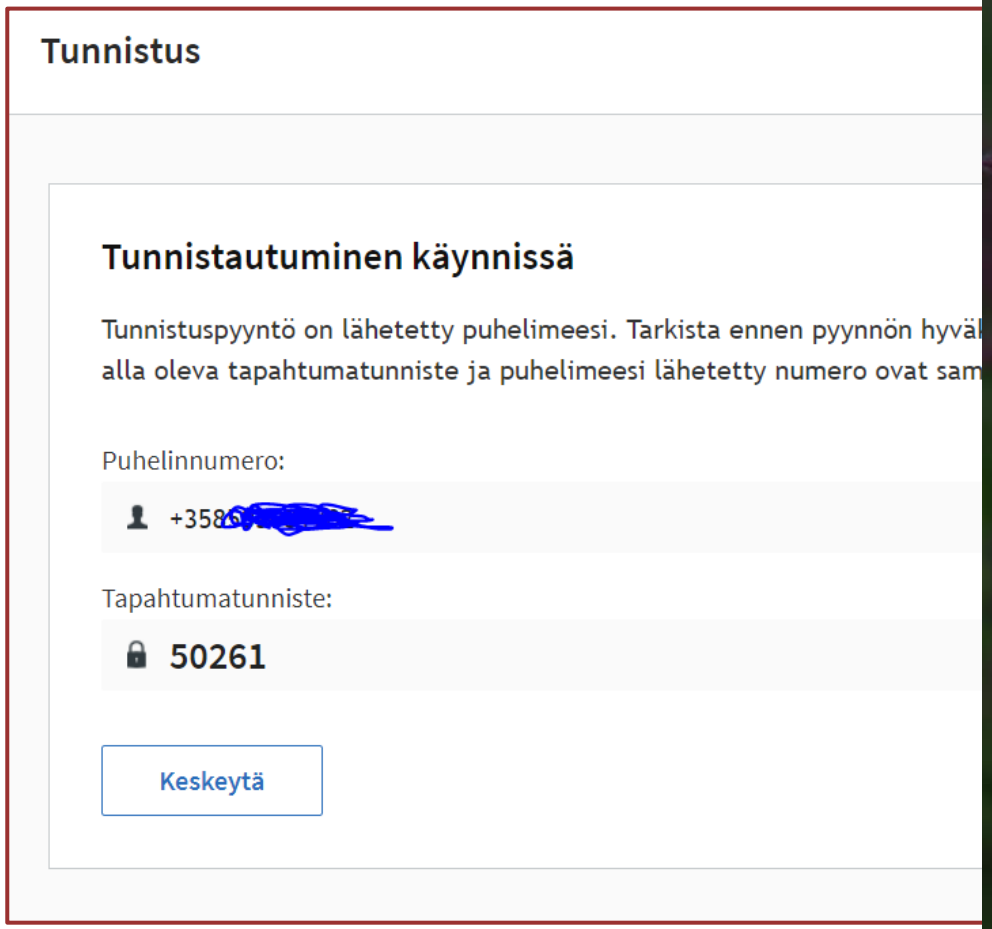

  - Less secure if you use the phone also for banking and other service access. The phone becomes a single point of failure:
    If the phone OS is hacked, the hacker can capture the PIN and tamper with the service access from the app or browser.

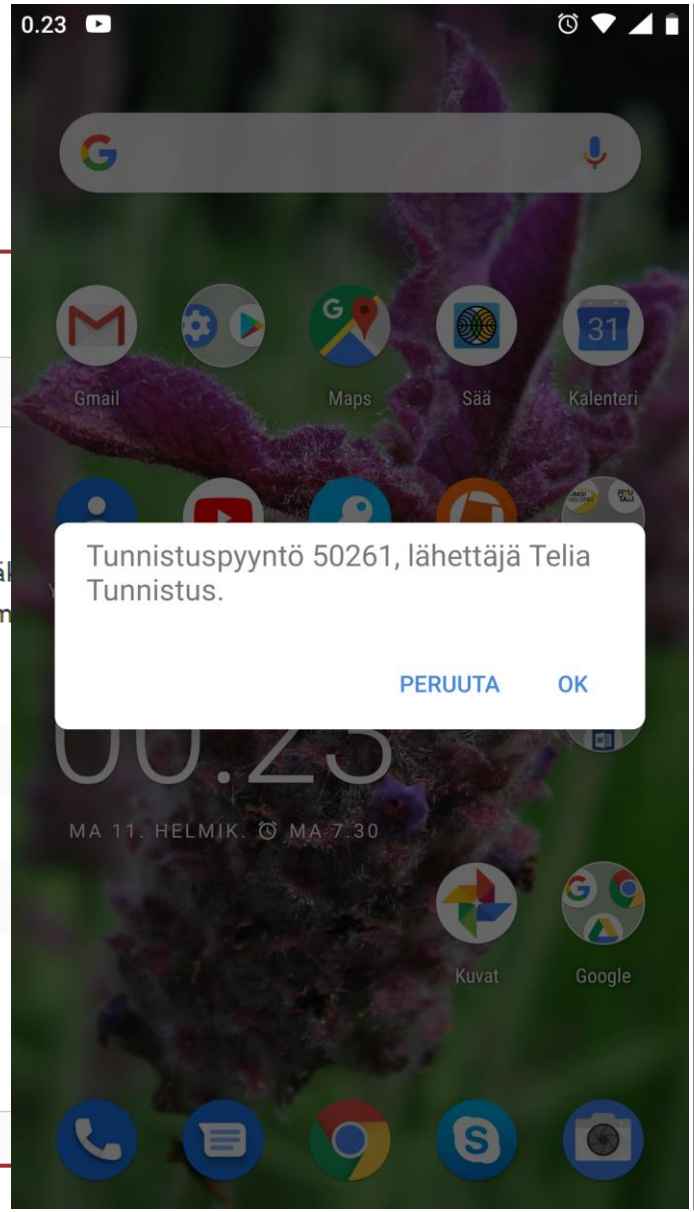
# Mobile signature service (MSS)

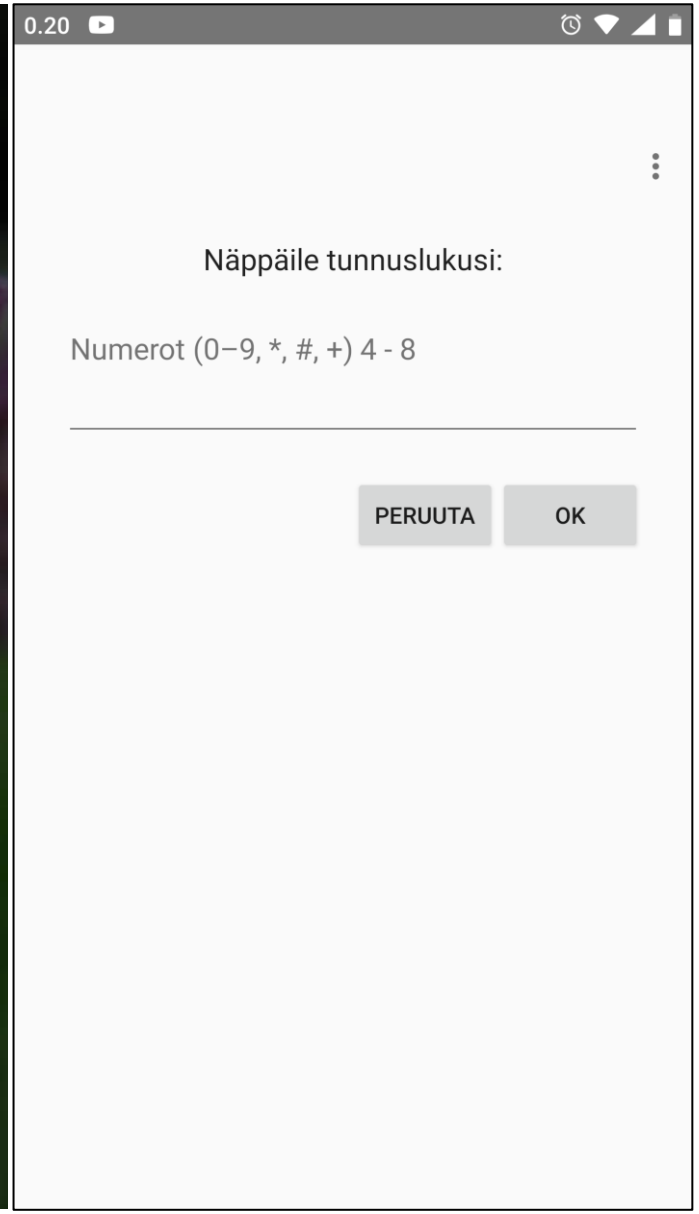
- Signature key on the mobile-phone SIM
  - Used for strong authentication, potentially also document signing
  - Protocols depend on a mobile signature service provider (MSSP)
  - ETSI standard, https://mobiilivarmenne.fi/

Mobiilivarmenne

# MSS user experience

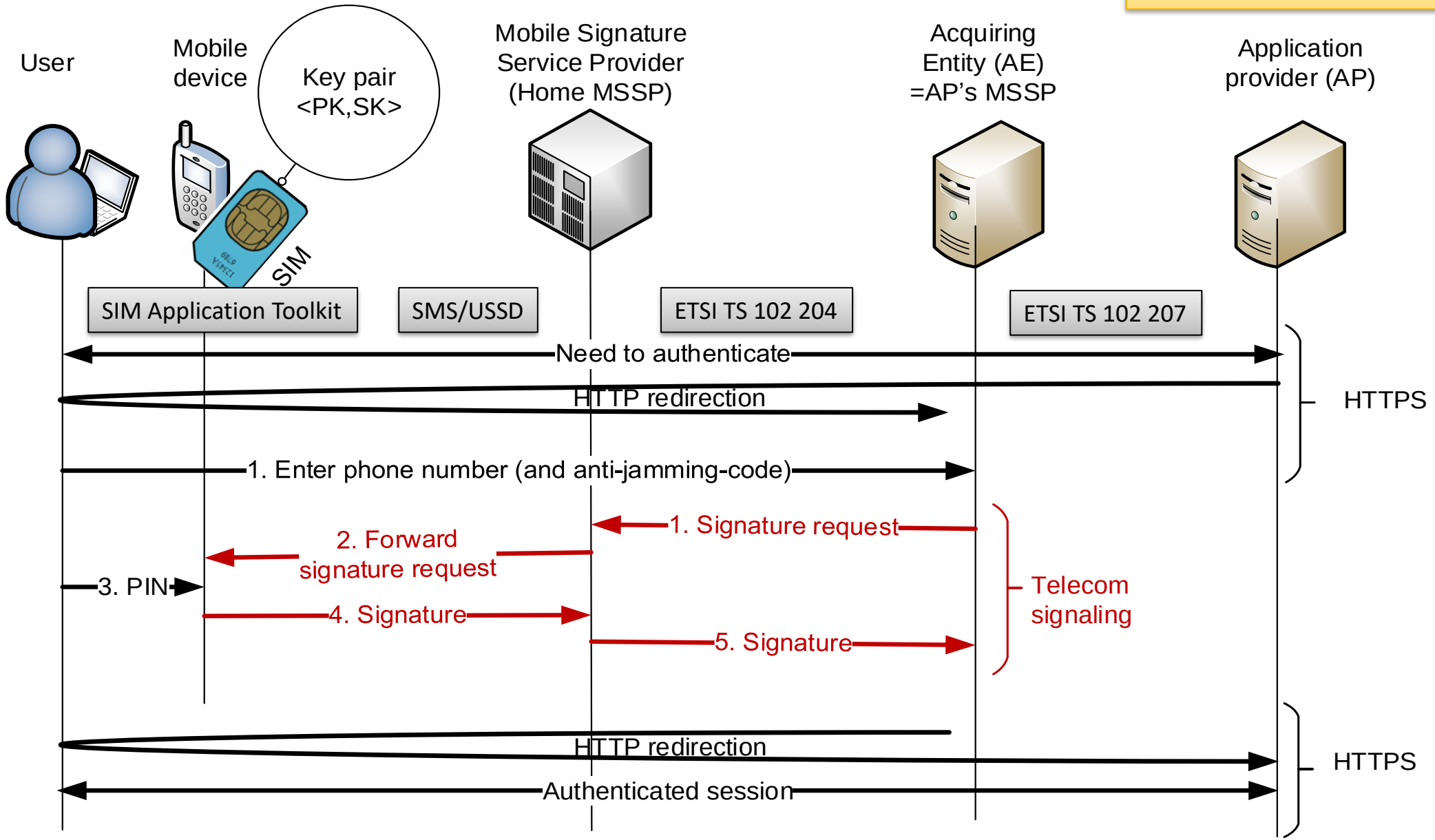


# MSS authentication

MSS standard message flows with four-corner model



# Mobile signature service (MSS)

- Advantages: everyone has a SIM card, and operators have 24/7 service for revocation
- "Signature keys are in your own pocket"
  - However, signatures can only be requested via the operator's mobile signature service provider (MSSP)
- Deployment and adoption has been slow
  - Requires identity proofing, i.e., verification of the subject identity before issuing the certificate
  - Operators want a fee for every transaction

# Identity proofing

- Identity proofing, i.e., verifying the subject identity before issuing the certificate
  - Electronic identify card: in person at police office
  - Banks: in person at the bank, government issued identity document required
  - Mobile signature: in person at operator shop, or online with bank credentials

# AUTHENTICATION TO GOVERNMENT SERVICES

# KaPA

- Integrating all the strong authentication methods to all services is a lot of work
- *Suomi.fi -palveluväylä* (Suomi.fi data exchange layer)
  - Enterprise service bus for government services
  - Design based on Estonian X-road
  - Includes common electronic identification for public online services
  - Ministry of Finance, implementation by Population register (VRK)

# European eIDAS regulation

- Goal: Cross-border access to government online services in EU
  - https://esignature.ec.europa.eu/efda/home/
- Technology-agnostic regulation
- Electronic identification (eID)
  - Cross-border acceptance of national id schemes
  - Assurance levels: low, substantial and high
- Electronic trust services (eTS)
  - Electronic signature, electronic seal, time stamp, electronic delivery service, website authentication
- EU Trusted List: list of trust service providers that issue (or subcontract issuers of) qualified digital certificates

# Implementing eIDAS in Finland

- Traficom registers identification and trust services in Finland
- Digital and Population Data Services Agency, Digi- ja väestötietovirasto* is the only qualified trust service provider on the trusted list
  - Qualified certificates, signatures, timestamps, registered delivery services, web site authentication
- Other strong electronic identification services (banks and MSS) are not qualified

Extra material

# CORPORATE IAM

# Corporate IAM

Extra material

- Federated identity and authentication is not sufficient:
  - Need to configure access permissions for users in the services
  - Need to monitor and audit access control state in the system
  - Need to revoke access rights
- Identity and access management (IAM) systems
  - Define roles and groups for the organization
  - Enable centralized role assignment, revocation and monitoring
- Example:
  - student enrolls to university, then becomes employee, then graduates, finally leaves employment
- Typical implementation: central IAM server, and an IAM agent at each supported service

  → more expensive to develop and deploy than federated authentication; better overall picture and effective revocation

# SUMMARY

# List of key concepts

- Single sing-on, pseudo SSO, proxy-based pseudo SSO, federated SSO, identity and access management IAM
- SAML 2.0, service provider SP, identity provider IdP, federation metadata, assertion
- Shibboleth 2, server discovery and WAYF, identity vs. attribute, use consent, login session, local vs. global logout
- OAuth 2.0, limited authorization, delegation, resource, scope, authorization token, bearer token
- OpenID Connect, identity token, JSON Web Signature
- Strong authentication, Finnish Trust Network, HST-kortti, Mobile signature service MSS, MSSP, three-corner model, four-corner model, national service bus
- eIDAS, trusted list, qualified digital certificate, identity proofing, assurance level

# References


- OpenID Connect, http://openid.net/developers/specs/
- SAML 2.0 Technical Overview, http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf
- OAuth specification http://tools.ietf.org/html/rfc6749
- MSS standard
  http://www.etsi.org/deliver/etsi_ts/102200_102299/102204/01.01.04_60/ts_102204v010104p.pdf,
  http://www.etsi.org/deliver/etsi_ts/102200_102299/102207/01.01.03_60/ts_102207v010103p.pdf
- Mikael Linden, Identiteetin- ja pääsynhallinta (in Finnish)
  http://www.cs.tut.fi/kurssit/TLT-3600/iam-sem2011.pdf

# Exercises

- Learn to use developer the web browser's developer mode (Ctrl-Shift-I in Chrome, F12 in Edge, ⌥ ⌘I in Safari). Capture the AuthnRequest and Response messages when logging into university services and external services and compare the messages. (If the messages are not what you expected, maybe a slightly different SAML 2.0 binding is used.)
- Find examples of OAuth 2 and OpenID Connect in online services. Inspect the traces. There are useful online tools for decoding message parts. Do login buttons like Facebook and Google login use any recognizable protocol?
- Look at the Haka federation metadata for Shibboleth 2 (XML format). How does this create trust between an IdP and SP? What ways are there to limit the trust between organizations?
- Why exactly is TLS needed at each stage in Shibboleth/OAuth/OpenID Connect, or is it?
- Compare the logout (and re-login) behavior of services like MyCourses, Sisu and Exam. Which sessions get deleted, when and how?
- Despite similarities in the web implementation, OpenID Connect, older OpenID, SAML, OAuth, and bank authentication (FTN or old Tupas) have different goals and make different assumptions about the relations between entities. What differences are there?
- Find out about the eIDAS requirements for different assurance levels of identification services.