OpenID Connect pro připojování služeb do AAI

Martin Kuba makub@cesnet.cz

AAI - Autentizační a Autorizační Infrastruktura

- e-INFRA AAI je
  - Perun IdM https://perun.cesnet.cz/
    - eviduje virtuální organizace, formuláře přihlášek, skupiny, jejich členy
    - eviduje zaregistrované služby
  - aplikace pro registraci služeb https://spreg.aai.cesnet.cz/
  - Proxy IdP https://login.cesnet.cz/
    - deleguje autentizaci na eduGAIN, Google, ORCID, GitHub, LinkedIn
    - přidává atributy podle údajů z Perun IdM
    - poskytuje službám SAML Identity Provider
    - poskytuje službám OIDC Authorization Server

Připojení služby přes SAML

- SAML (Security Assertion Markup Language) je XML formát/protokol pro předání informací o uživateli
- informace jsou ve formě atributů URN=hodnoty
- URN používají OID - ITU/ISO identifikátory objektů, např.
  - urn:oid:1.3.6.1.4.1.5923.1.1.1.9 - affiliation
  - urn:oid:2.5.4.20 - telephoneNumber
- služba musí
  - mít implementaci SAML Service Provider, např. Shibboleth SP, SSP
  - mít tajný klíč a X509 certifikát (stačí self-signed, PKI se nepoužívá)
  - vygenerovat svoje metadata - XML dokument popisující její
    - entityID - identifikátor ve tvaru URN, např. http://moje.cz/shibboleth
    - X509 certifikát
    - endpoints - URL na kterých komunikuje, záleží na implementaci
  - zaregistrovat svoje metadata u AAI
  - pravidelně si stahovat metadata Proxy IdP k sobě

OpenID Connect (OIDC)

- OIDC je nadstavba nad protokolem OAuth 2.0 (RFC 6749)
- definuje API pro získání informací o uživateli
  - tzv. userInfo endpoint, prosté URL volané HTTP metodou GET
  - formát dat je JSON objekt
  - položky v JSON jsou předdefinované, tzv. **claims**, např. sub, email, phone, given_name, nickname, gender, locale, …
  - skupiny claims mají samostatná práva přístupu, tzv. **scopes**
    - openid - jen nicneříkající identifikátor sub (subject)
    - email - email a email_verified
    - phone - phone a phone_verified
    - address - strukturovaná poštovní adresa v address
    - profile - všechny ostatní claims
- na rozdíl od SAML nemá definovaný mechanismus pro pojmenovávání dalších claims

Příklad claims z userInfo


```
{
  "sub": "3e65bd2aa4c818bd3579023939b546b69e1@einfra.cesnet.cz",
  "name": "Josef Novák",
  "preferred_username": "pepa",
  "given_name": "Josef",
  "family_name": "Novák",
  "nickname": "Pepan",
  "profile": "https://www.muni.cz/en/people/3988",
  "picture": "https://secure.gravatar.com/avatar/f320c89e39d15da1608c8fc31210b8ca",
  "website": "http://pepovo.wordpress.com/",
  "gender": "male",
  "zoneinfo": "Europe/Prague",
  "locale": "cs-CZ",
  "updated_at": "1508428216",
  "birthdate": "1975-01-01",
  "email": "pepa@gmail.com",
  "email_verified": true,
  "phone_number": "+420 603123456",
  "phone_number_verified": false,
  "address": {
    "street_address": "Severní 1",
    "locality": "Dolní Lhota",
    "postal_code": "111 00",
    "country": "Czech Republic"
  }
}
```

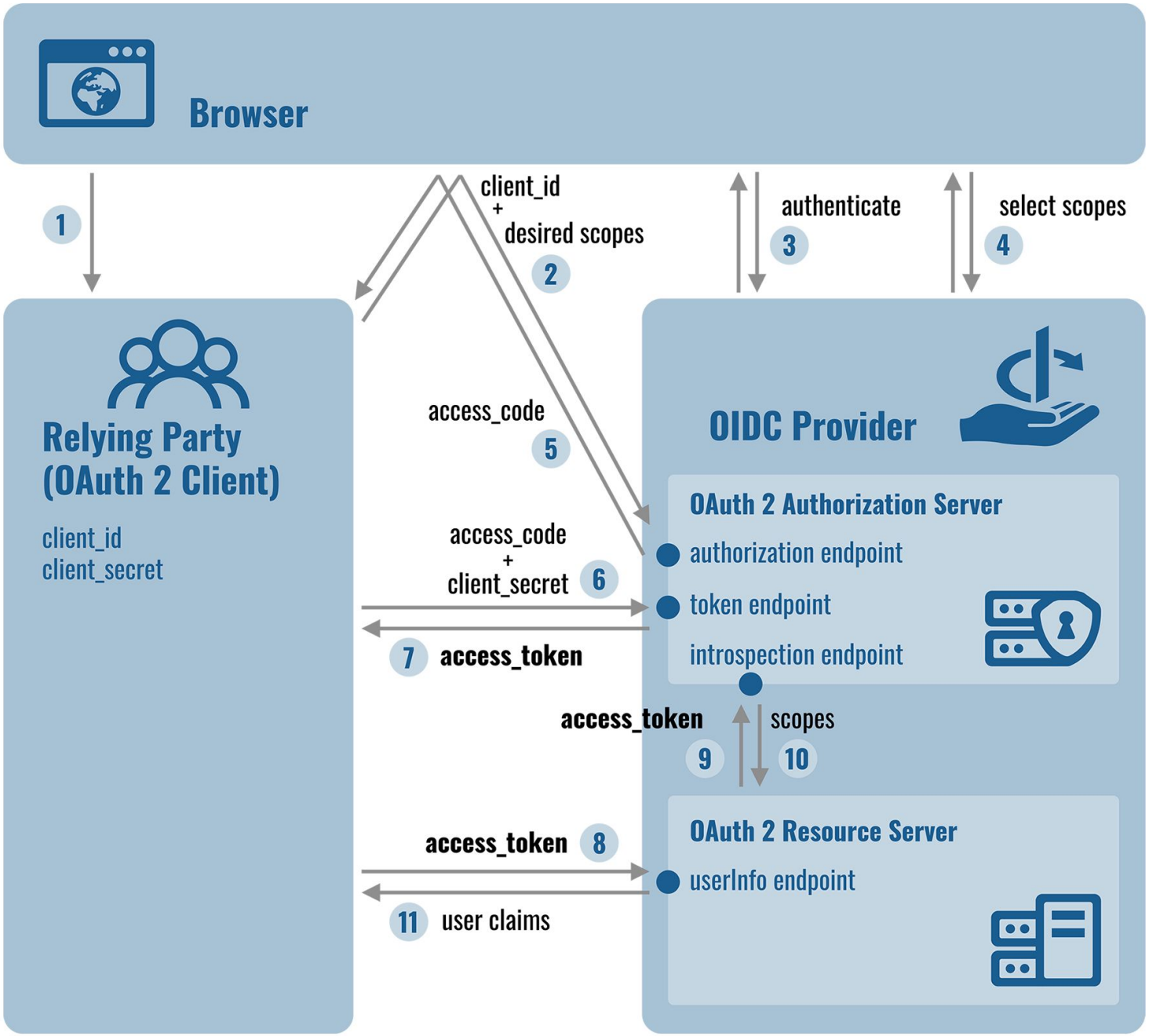
Připojení služby přes OIDC

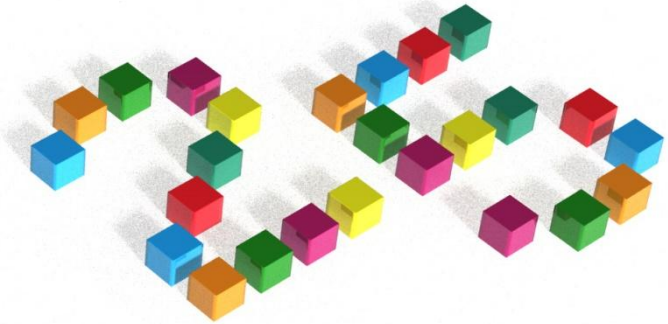
- oproti SAML je mnohem jednodušší
  - nepotřebuje svůj tajný klíč, certifikát, ani svoje metadata
  - SAML používá XML a XML-Signature, OIDC používá JSON a JWT (JSON Web Tokens)
- služba musí
  - míst implementaci OIDC (vlastní, nebo např. mod_auth_openidc modul pro web server Apache)
  - zaregistrovat se u Autorizačního Serveru (https://spreg.aai.cesnet.cz/) a tím získat dva řetězce - client_id a client_secret
  - nastavit ve své implementaci OIDC:
    - URL na metadata Autorizačního Serveru https://login.cesnet.cz/oidc/.well-known/openid-configuration
    - svoje client_id a client_secret
    - požadované scopes (jaké data o uživateli chce)

Konfigurace pro Apache s mod_auth_openidc

```
OIDCProviderMetadataURL https://login.cesnet.cz/oidc/.well-known/openid-configuration
OIDCProviderMetadataRefreshInterval 3600
OIDCClientID your_client_id_replace_with_yours
OIDCClientSecret your_client_secret_replace_with_yours
OIDCScope "openid profile email"
OIDCRedirectURI /oauth2callback
OIDCCryptoPassphrase randompassword

   <Location /oauth2callback>
      #non-existent location for returning from OIDC server
      AuthType openid-connect
      Require valid-user
   </Location>
   <Location /cgi-bin/>
      #actually protected URLs
      AuthType openid-connect
      Require valid-user
   </Location>
```



Browser
1
client_id
+
desired scopes
2
authenticate
3
select scopes
4
Relying Party
(OAuth 2 Client)
client_id
client_secret
access_code
5
OIDC Provider
OAuth 2 Authorization Server
authorization endpoint
token endpoint
introspection endpoint
access_code
+
client_secret
6
7
access_token
access_token
scopes
9
10
OAuth 2 Resource Server
access_token
8
userInfo endpoint
11
user claims



Děkuji za pozornost!