# OpenID Connect

Martin Kuba
makub@cesnet.cz

# Obsah

- co jsou OpenID Connect a OAuth 2
- principy OAuth 2
  - 4 zúčastněné strany
  - scope, access token
  - různé typy *authorization grant flow*
- rozšíření OpenID Connect nad OAuth2
- vyzkoušené implementace
  - OpenID Provider - *MitreID Connect*
  - server-side client - *Apache mod_auth_openidc*
  - JavaScript client - *oidc-client-js*
  - Resource Server  - *Apache mod_auth_openidc*

# OpenID Connect a OAuth2

- OpenID Connect (dále OIDC) je rozšíření autorizačního protokolu OAuth 2 o **autentizaci** a API pro získávání informací o uživateli
- OAuth 2 je **autorizační protokol**, jímž uživatel, vlastnící zdroje na resource serveru, zplnomocňuje cizí aplikaci, aby jeho jménem se zdroji zacházela
- z hlediska aplikací je OIDC obdoba SAML2, ale
  - není nutná výměna metadat mezi IdP a SP
  - uživatel si sám volí, která osobní data aplikaci zpřístupní
  - aplikace nejsou omezené na web (i mobilní, desktopové, command-line, SmartTV)

# OAuth 2

- definován v RFC 6749 z roku 2012
- používán firmami Google, Facebook, Microsoft, Twitter, LinkedIn, GitHub atd.
- je určen pro bezpečné **delegování přístupu**, ale byl od počátku používán i pro federované přihlášení

# OAuth2 - zúčastněné strany

- **resource owner** - uživatel
- **resource server** - server spravující uživatelova data, umožňuje určité operace nad nimi, právo k určitým operacím se nazývá **scope**
- **client** - aplikace, která chce přístup k operacím s uživatelovými daty (čtení, změny, mazání)
- **authorization server** - server, který autentizuje uživatele, ptá se jich které scopes chtějí povolit určitému clientovi, vydává **access token**

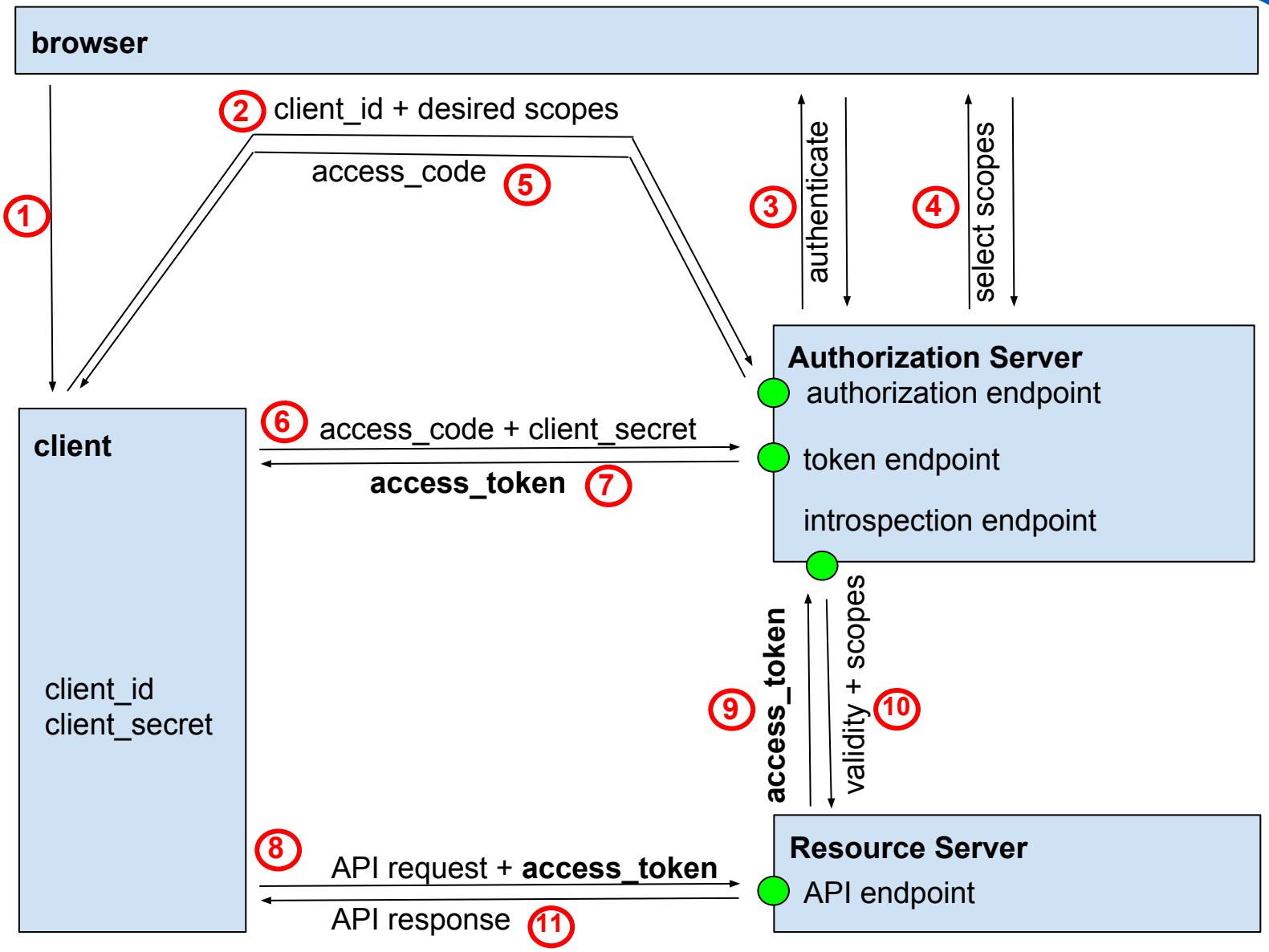
# OAuth 2 - příklad

- **resource owner** - já
- **resource server** - Google Calendar API na https://www.googleapis.com/calendar/v3
- **scopes**
  - čtení i zápis - `https://www.googleapis.com/auth/calendar`
  - jen čtení - `https://www.googleapis.com/auth/calendar.readonly`
- **client** - aplikace „Business Calendar“ pro Android od firmy Appgenix Software
- **authorization server** - https://accounts.google.com/

# OAuth 2 - registrace clienta

- před prvním použitím je nutné aplikaci (client) zaregistrovat u Authorization Server
- součástí registrace je
  - typ aplikace (web, user-agent-based, native)
  - seznam povolených URL s aplikací
  - seznam požadovaných scopes
- client získá **client_id** a **client_secret** pro autentizaci vůči Authorization Server

# OAuth 2 - schéma komunikace



# OAuth 2 access token

- access token (odznak přístupu) reprezentuje autorizaci udělenou uživatelem clientovi
- podle RFC 6749 je „opaque“ (neprůhledný)
- obvykle je ve formátu JWT (JSON Web Token) - digitálně podepsaný JSON
- Resource Server může buď rozparsovat token a ověřit podpis, nebo se na tzv. **introspection endpoint** autorizačního serveru zeptat na jeho platnost a význam, tj. seznam scopes
- uživatel může vydaný token zneplatnit

# OAuth 2 refresh token

- **access token** má krátkou dobu platnosti (například 60 minut)
- pokud client potřebuje delší přístup, může požádat o **refresh token** s dlouhou dobou platnosti
- refresh token může client vyměnit voláním token endpointu autorizačního serveru za nový access token

# Authorization Grant Flows

- OAuth 2 rozlišuje tři typy aplikací:
  - **web** - na serveru, může bezpečně uchovávat client_secret
  - **user-agent-based** - JavaScript, nemůže bezpečně uchovávat client_secret ani access token
  - **native** - mobilní nebo desktopová, nemůže chránit client_secret, ale access token může
- proto existují různé způsoby získání tokenu
  - authorization code grant - viz předchozí schéma
  - implicit code grant - AS vydá token clientovi přímo
  - resource owner password credentials grant
  - client credentials grant
  - device flow grant - pro SmartTV bez klávesnice

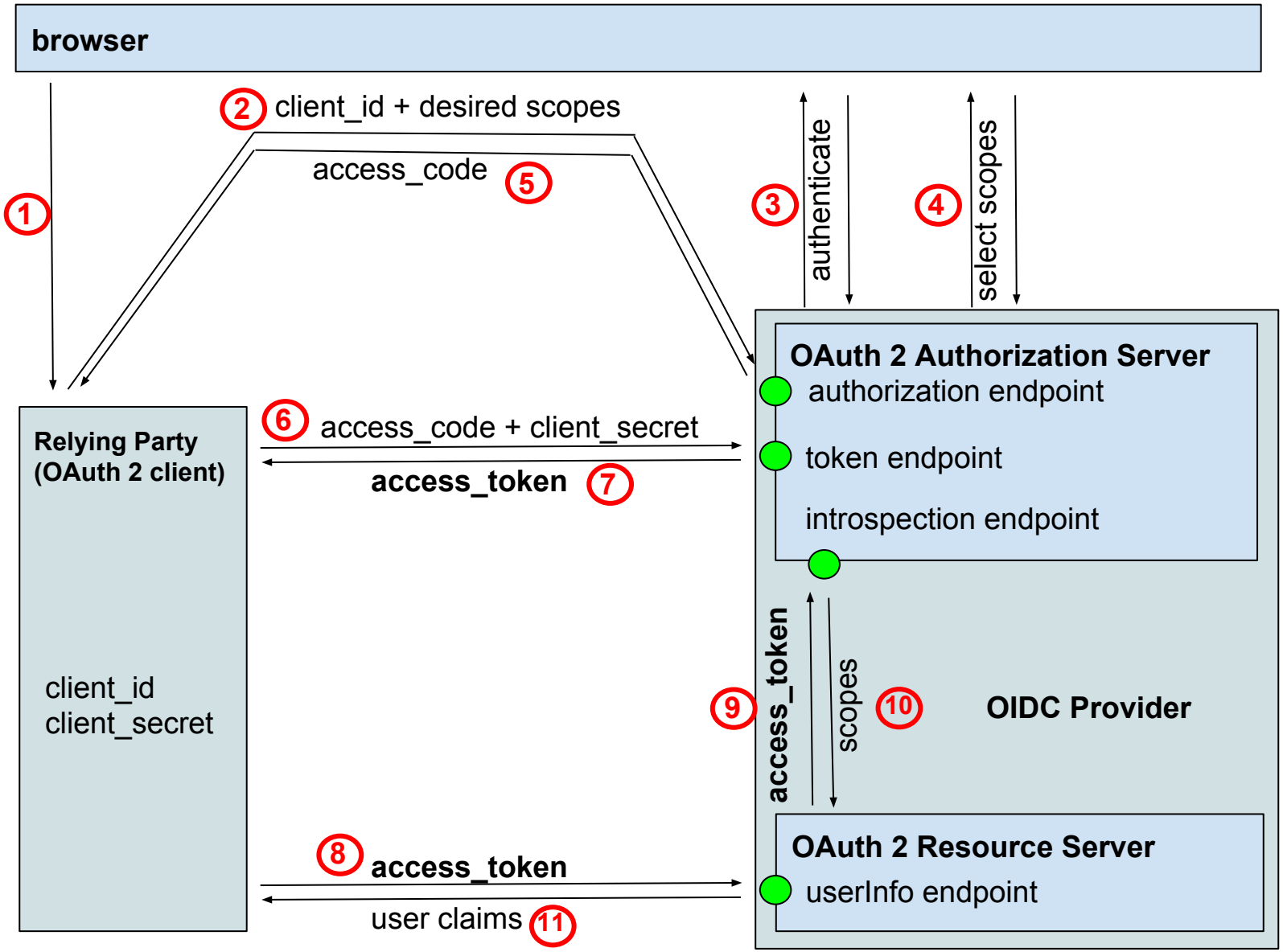
# OAuth 2 - shrnutí

- OAuth 2 umožňuje aplikaci požádat uživatele o oprávnění k operacím s jeho daty
- uživatel po přihlášení na autorizačním serveru schválí buď všechna požadovaná, nebo jen některá oprávnění
- aplikace získá **časově omezený** access token představující povolená oprávnění jednat za uživatele voláním Resource Serveru

# OpenID Connect

- OAuth 2 zajišťuje přihlášení, ale nedefinuje, jak získat údaje o uživateli, každá služba poskytovala jiné API
- OpenID Connect definuje
  - **userInfo endpoint** - API pro získání údajů o uživateli
  - **scopes** - openid, profile, email, address, phone
  - **claims** - sub, name, family_name, given_name, middle_name, nickname, preferred_username, profile, picture, website, gender, birthdate, zoneinfo, locale, updated_at, email, email_verified, address, phone_number, phone_number_verified
  - mapování scopes na claims
  - **id_token** který může (ale nemusí) obsahovat claims
  - metadata v JSON na `/.well-known/openid-configuration`

# OpenID Connect - schéma



# 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": true,
 "address": {
   "street_address": "Severní 1",
   "locality": "Dolní Lhota",
   "postal_code": "111 00",
   "country": "Czech Republic"
 }
}
```

# Příklad obsahu access tokenu


```
{
 "kid": "rsa1",
 "alg": "RS256"
}
{
 "sub": "3e65bd2aa4c818bd3579023939b546b69e1@einfra.cesnet.cz",
 "azp": "7652ad4c-4ee6-4ad1-b571-3576574f383e",
 "iss": "https://login.cesnet.cz/oidc/",
 "exp": 1508431816,
 "iat": 1508428216,
 "jti": "5b4f8eb8-3688-4588-8f31-ecb78ba48a76"
}
```

# Příklad odpovědi z Introspection


```
{
  "active":true,
  "scope":"address phone openid profile email",
  "expires_at":"2017-10-19T18:50:16+0200",
  "exp":1508431816,
  "sub":"makub",
  "user_id":"makub",
  "client_id":"7652ad4c-4ee6-4ad1-b571-3576574f383e",
  "token_type":"Bearer"
}
```


- specifikace na https://tools.ietf.org/html/rfc7662#section-2.2

# Příklad metadat OIDC serveru

Zabezpečeno | https://login.cesnet.cz/oidc/.well-known/openid-configuration


```
{
    request_parameter_supported: true,
    claims_parameter_supported: false,
    introspection_endpoint: "https://login.cesnet.cz/oidc/introspect",
  - scopes_supported: [
        "openid",
        "profile",
        "email",
        "address",
        "phone",
        "offline_access",
        "groupNames"
    ],
    issuer: "https://login.cesnet.cz/oidc/",
  + userinfo_encryption_enc_values_supported: […],
  + id_token_encryption_enc_values_supported: […],
    authorization_endpoint: "https://login.cesnet.cz/oidc/authorize",
    service_documentation: "https://login.cesnet.cz/oidc/about",
  + request_object_encryption_enc_values_supported: […],
    device_authorization_endpoint: "https://login.cesnet.cz/oidc/devicecode",
  + userinfo_signing_alg_values_supported: […],
```

# OIDC terminologie

- místo „client“ používá výraz „Relying Party“
- Authorization Server + Resource Server s userInfo endpointem se nazývá „OpenID Provider (OP)“
- RP odpovídá funkcí SAML2 SP
- OP odpovídá funkcí SAML2 IdP

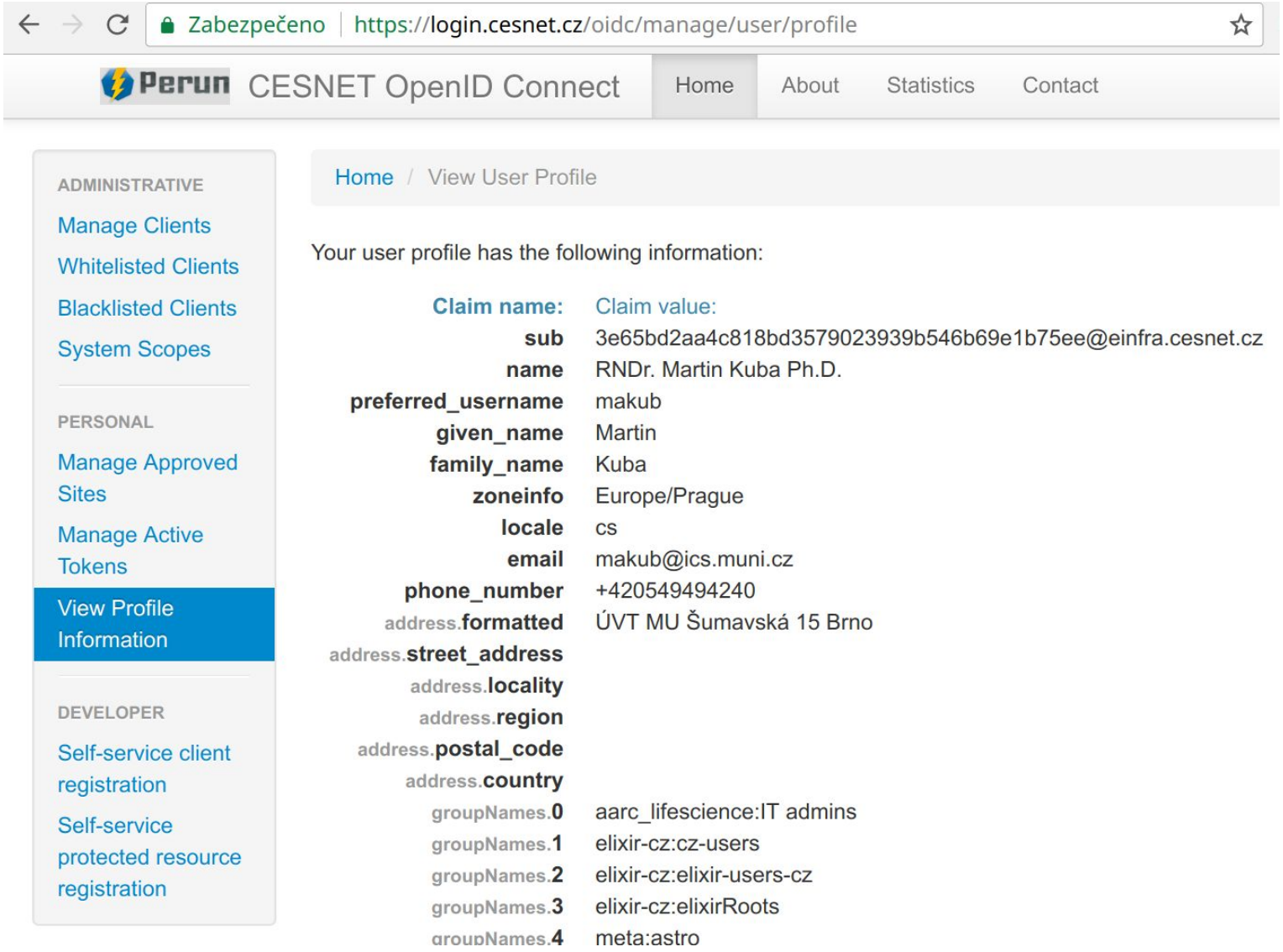
# Vyzkoušené implementace

- OpenID Provider - MITREid Connect
  - webová aplikace v Javě, založená na Spring Security
  - https://github.com/mitreid-connect/OpenID-Connect-Java-Spring-Server
- JavaScriptový client - oidc-client-js
  - JavaScriptová knihovna s certifikací kompatibility
  - https://github.com/IdentityModel/oidc-client-js/
- autentizační modul do Apache - mod_auth_openidc
  - dvě role - Relying Party nebo OAuth2 Resource Server
  - https://github.com/zmartzone/mod_auth_openidc

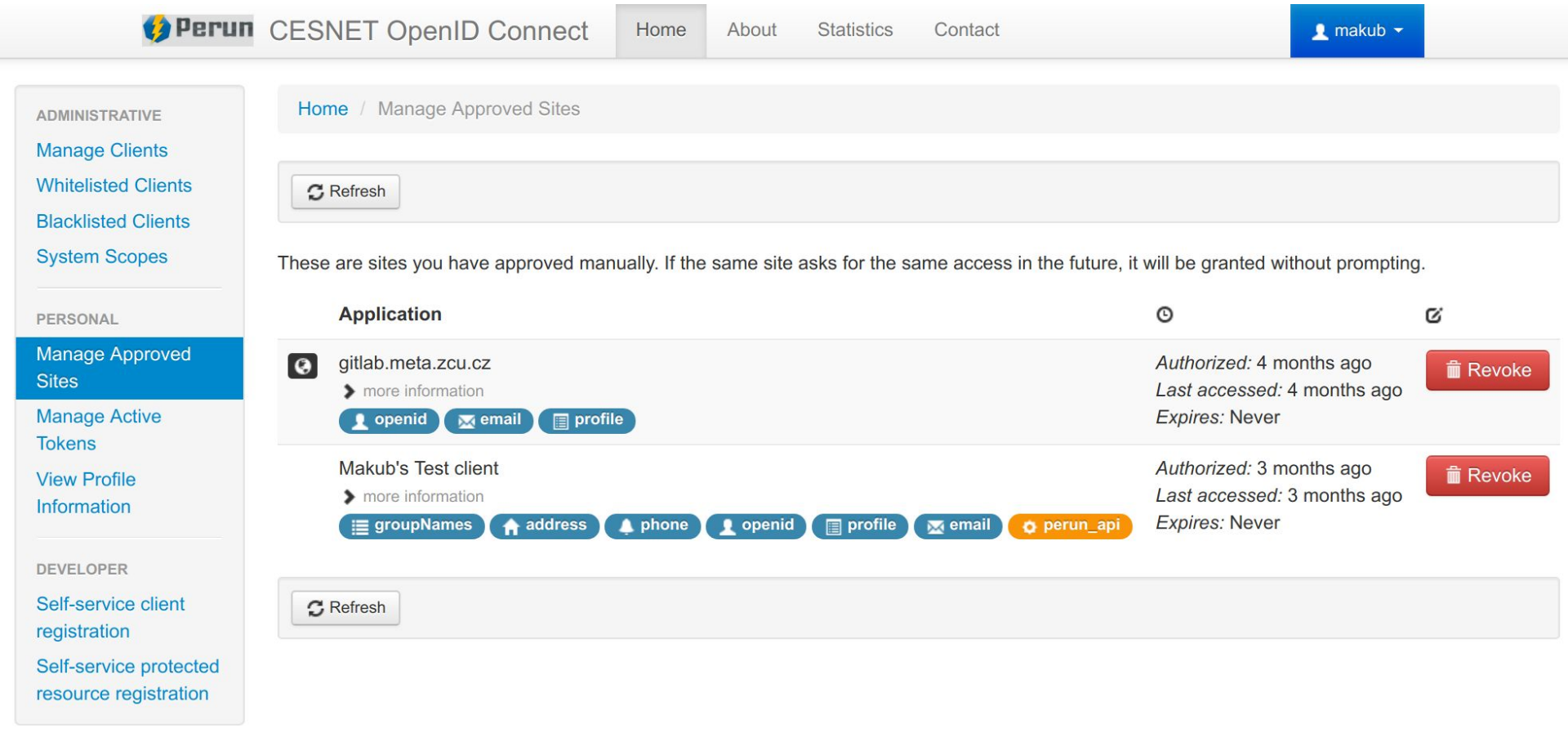
# MITREid Connect

- Maven webapp overlay s vlastními modifikacemi
- používá Hibernate pro práci s databází, podporuje PostgreSQL, MySQL, Oracle, HSQL
- umíme
  - napojit na libovolný zdroj dat o uživatelích (Perun)
  - přebírat autentizaci uživatele z Apache
  - přidávat vlastní scopes a claims
  - modifikovat odpověď z introspection endpointu
  - modifikovat vydávaný access token

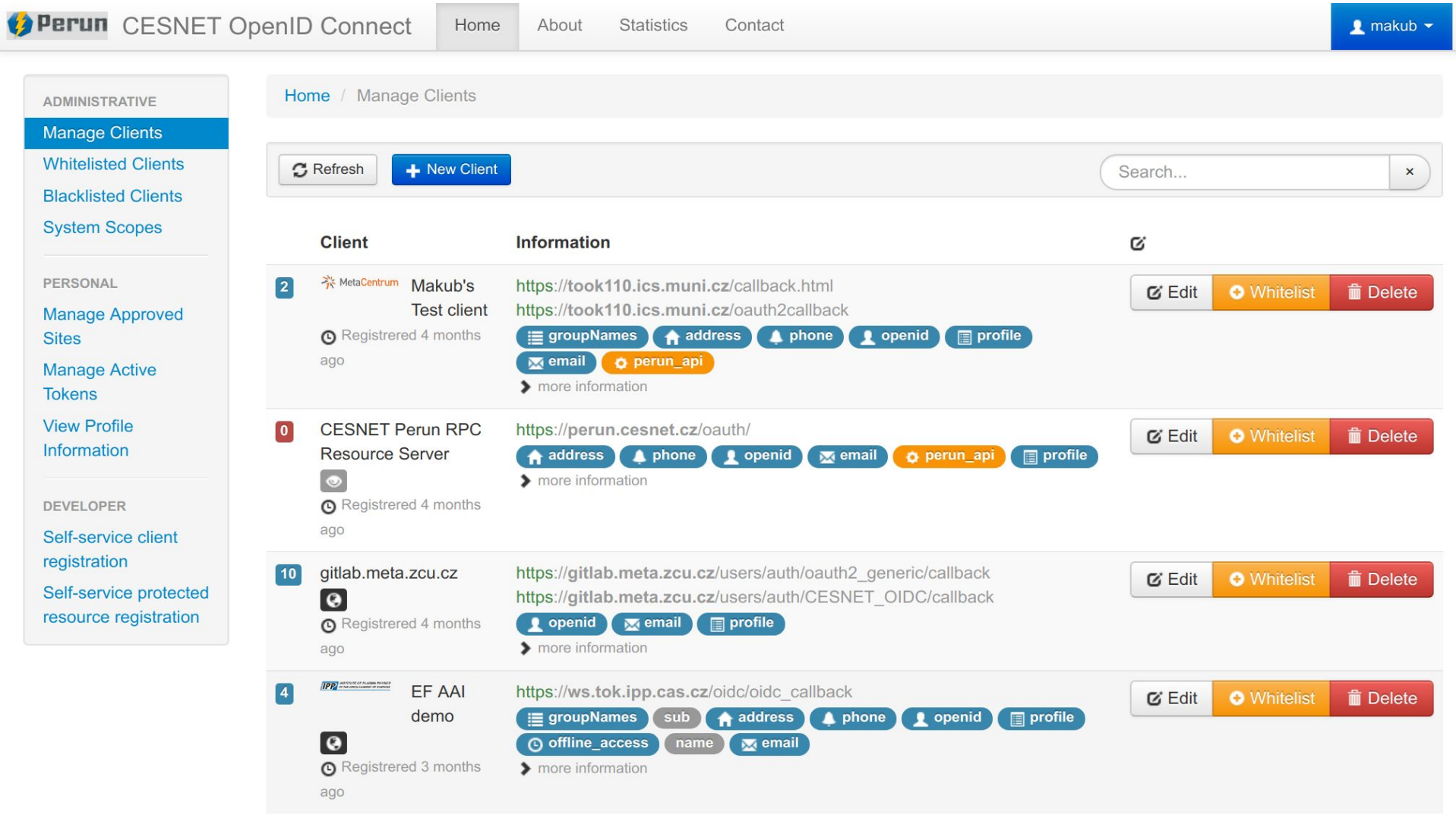
# MITREid - profil uživatele



# MITREid - schválené aplikace



# MITREid - správa aplikací



# OpenID Connect

Děkuji za pozornost

cesnet

makub@cesnet.cz

cerit scientific cloud