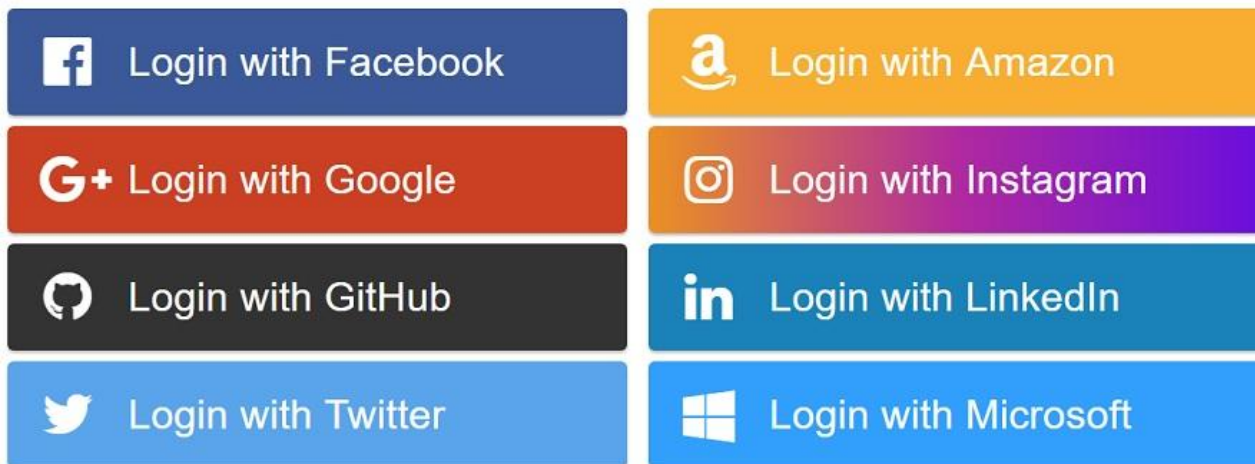


OAuth 2, OpenID Connect

Martin Kuba, ÚVT MU

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



OAuth 2 - zúčastněné strany

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

Co je OAuth 2

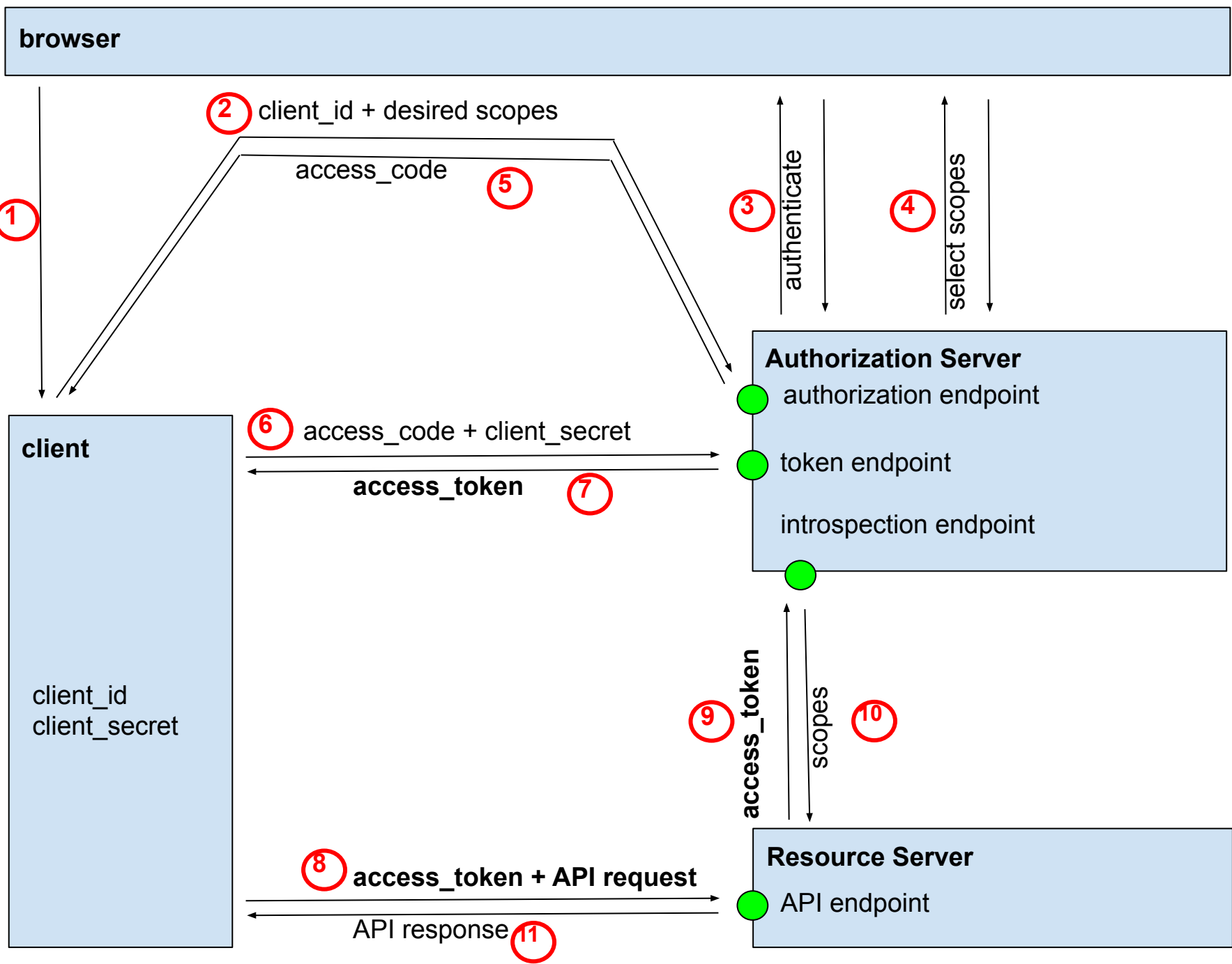
- otevřený standard specifikující protokol pro **autorizaci** přístupu k vyjmenovaným operacím nějakého API, ale lze jej využít i pro autentizaci v případě, že dané API má operace pro získání informací o uživateli
- umožňuje povolit pro konkrétního **poskytovatele služby** jen určité operace (ze všech operací) na API určitého poskytovatele API
- seznam implementací - <http://oauth.net/2/>

Co umožňuje OAuth

- není omezeno jen na web, lze i pro mobilní aplikace (Android, iOS), desktopové, SmartTV, embedded v set-top-boxech
- spolupráce dvou různých web serverů
- např. uživatel Google Disk může povolit jinému webu od firmy X, případně jejich mobilní aplikaci, čtení dokumentů a zápis jejich upravených verzí
- aplikace může přistupovat k API i bez uživatele

Jak to funguje

1. vývojář aplikace se zaregistruje u poskytovatele autorizačního serveru
 - Google API console <https://code.google.com/apis/console/>
 - Facebook developers <https://developers.facebook.com/apps/>
 - MUNI <https://spreg.aai.muni.cz/>
2. zaregistruje aplikaci, získá *client_id* a *client_secret*
3. při příchodu uživatele do aplikace přesměruje na OAuth server se žádostí o oprávnění k určitým operacím
4. aplikace získá od uživatele jednorázový kód
5. aplikace vymění kód a *client_secret* za token
6. aplikace volá API a prokazuje se tokenem



Registrace aplikace u Google

Create Client ID ✕

Client ID Settings

Application type

- Web application
Accessed by web browsers over a network.
- Service account
Calls Google APIs on behalf of your application instead of an end-user. [Learn more](#)
- Installed application
Runs on a desktop computer or handheld device (like Android or iPhone).

Your site or hostname [\(more options\)](#)

For example: `www.example.com` or `localhost`

Redirect URI

`https://www.example.com/oauth2callback`

[Learn more](#)

Zaregistrovaná aplikace u Google

[Vyhledávání](#) [Obrázky](#) [Mapy](#) [Play](#) [YouTube](#) [Zprávy](#) [Gmail](#) [Disk](#) [Další](#) ▼

[martinkuba@gmail.com](#) ▼ | [Nastavení](#) ▼ | [Nápověda](#) | [Odhlásit se](#)



MUNI Photometric Arch... ▼

Overview

Services

Team

API Access

API Access

To prevent abuse, Google places limits on API requests. Using a valid OAuth token or API key allows you to exceed anonymous limits by connecting requests back to your project.

Authorized API Access

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private. A single project may contain up to 20 client IDs. [Learn more](#)

Branding information

The following information is shown to users whenever you request access to their private data.

Product name: MUNI Photometric Archive
Google account: martinkuba@gmail.com
Home page URL: <https://www.cerit-sc.cz/login/>

[Edit branding information...](#)

Client ID for web applications

Client ID: 558708443072.apps.googleusercontent.com
Email address: 558708443072@developer.gserviceaccount.com
Client secret: 6_woe [redacted] -LxMvG
Redirect URIs: <https://www.cerit-sc.cz/login/google/auth>
JavaScript origins: <https://www.cerit-sc.cz>

[Edit settings...](#)

[Reset client secret...](#)

[Download JSON](#)

[Delete...](#)

[Create another client ID...](#)

Povolená API u Google

[Vyhledávání](#) [Obrázky](#) [Mapy](#) [Play](#) [YouTube](#) [Zprávy](#) [Gmail](#) [Disk](#) [Další](#) ▼

[martinkuba@gmail.com](#) ▼ | [Nastavení](#) ▼ | [Nápověda](#) | [Odhlásit se](#)

Google apis

MUNI Photometric Arch... ▼

All (56) Active (0) Inactive (56) Google Cloud Platform

Overview












Services

Team

API Access

All services

Select services for the project.

Service	Status	Notes
 Ad Exchange Buyer API ?	<input type="checkbox"/> OFF	Courtesy limit: 1,000 requests/day
 Ad Exchange Seller API ?	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
 AdSense Host API ?	Request access...	Courtesy limit: 100,000 requests/day
 AdSense Management API ?	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
 Analytics API ?	<input type="checkbox"/> OFF	Courtesy limit: 50,000 requests/day
 Audit API ?	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
 BigQuery API ?	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day • Pricing
 Blogger API v3 ?	Request access...	Courtesy limit: 10,000 requests/day
 Books API ?	<input type="checkbox"/> OFF	Courtesy limit: 1,000 requests/day
 Calendar API ?	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
 Custom Search API ?	<input type="checkbox"/> OFF	Courtesy limit: 100 requests/day • Pricing

Zaregistrovaná aplikace u Facebooku



Hledat aplikace



MUNI Photometri...

Aplikace ▶ MUNI Photometric Archive

Upravit aplikaci

+ Vytvořit novou aplikaci

Nastavení

Upravit nastavení

ID aplikace / API klíč

500753543283158

Tajný klíč aplikace

69ba4c6[redacted]74ba317

Název aplikace

muniastro

Režim pískoviště

Vypnuto

Platformy, na kterých aplikace běží

Přihlášení pomocí Facebooku

Vývojářská upozornění

Zobrazit vše

Nemáte žádná vývojářská upozornění.

Statistiky

Zobrazit vše

Uživatelé

0 Nových uživatelů denně

0 Aktivní uživatelé za den

Sdílení

0 Obsah sdílený za den

0,00 Ohlasy podle sdílení

Úlohy

Upravit úlohy

Úlohy

Správci:



Zaregistrované aplikace na MUNI

ADMINISTRATIVE

Manage Clients

Whitelisted Clients

Blacklisted Clients

System Scopes

PERSONAL

Manage Approved Sites

Manage Active Tokens

View Profile Information

DEVELOPER

Self-service client registration

Self-service protected resource registration


Home / Manage Clients

Refresh

+ New Client

Search...

« 1 2 3 4 5 »

Client	Information	
0 MUNI Perun RPC Resource Server  Registered 4 months ago	https://idm.ics.muni.cz/oauth/ address phone openid profile perun_admin email perun_api ➤ more information	Edit Whitelist Delete
0 Perun MUNI GUI Registered 4 months ago	https://perun.aai.muni.cz/silent-refresh.html https://perun.muni.cz/api-callback https://perun.muni.cz/silent-refresh.html https://perun.aai.muni.cz/api-callback perun_admin openid perun_api email profile offline_access ➤ more information	Edit Whitelist Delete

Odeslání uživatele na OAuth server

```
public class FacebookServlet extends HttpServlet {

    private static final String client_id = "500753543283158";
    private static final String client_secret = "69ba4c[REDACTED]ba317";
    private static final String redirect_uri = "https://www.cerit-sc.cz/login/facebook/auth";
    private static final String LOGIN_URL = "https://www.facebook.com/dialog/oauth";
    private static final String TOKEN_URL = "https://graph.facebook.com/oauth/access_token";
    private static final String SCOPE = "email";

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        // Step 1
        if ("/login".equals(req.getPathInfo())) {
            //initiate facebook authorization
            String state = Integer.toString(random.nextInt(Integer.MAX_VALUE)); //random value protecting against XSRF
            req.getSession(true).setAttribute("state", state);
            //redirect to Facebook to ask for permission on email
            String loginRedirectURL = LOGIN_URL
                + "?client_id=" + urlEncode(client_id)
                + "&redirect_uri=" + urlEncode(redirect_uri)
                + "&state=" + urlEncode(state)
                + "&scope=" + urlEncode(SCOPE);
            resp.sendRedirect(loginRedirectURL);
        } else if ("/auth".equals(req.getPathInfo())) {
```

Uživatel se přihlásí k účtu ...

 Facebook

Přihlaste se pro používání vašeho účtu s aplikací MUNI Photometric Archive.

E-mail nebo
telefon:

Heslo:

Zůstat přihlášen(a)

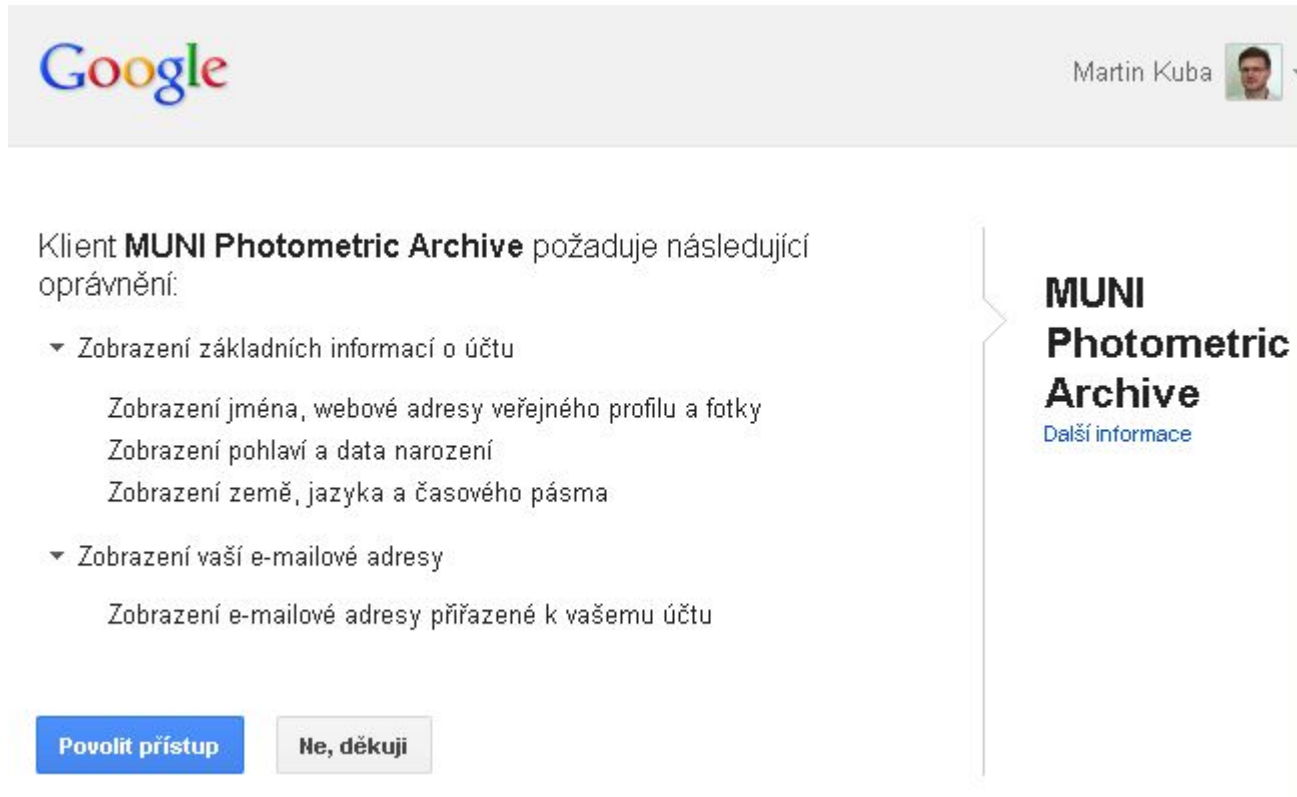
Přihlásit se nebo **Zaregistrujte se na Facebook**

[Zapomněli jste své heslo?](#)

... a schválí povolení k operacím




Obdobně u Google



The image shows a Google consent screen. At the top left is the Google logo. At the top right, the user's name "Martin Kuba" is displayed next to a small profile picture and a dropdown arrow. The main content area contains a heading "Klient **MUNI Photometric Archive** požaduje následující oprávnění:" followed by two expandable sections. The first section, "Zobrazení základních informací o účtu", lists: "Zobrazení jména, webové adresy veřejného profilu a fotky", "Zobrazení pohlaví a data narození", and "Zobrazení země, jazyka a časového pásma". The second section, "Zobrazení vaší e-mailové adresy", lists: "Zobrazení e-mailové adresy přiřazené k vašemu účtu". At the bottom left are two buttons: "Povolit přístup" (highlighted in blue) and "Ne, děkuji". On the right side, there is a vertical sidebar with the text "MUNI Photometric Archive" and a link "Další informace".

Google

Martin Kuba 

Klient **MUNI Photometric Archive** požaduje následující oprávnění:

- ▼ Zobrazení základních informací o účtu
 - Zobrazení jména, webové adresy veřejného profilu a fotky
 - Zobrazení pohlaví a data narození
 - Zobrazení země, jazyka a časového pásma
- ▼ Zobrazení vaší e-mailové adresy
 - Zobrazení e-mailové adresy přiřazené k vašemu účtu

Povolit přístup **Ne, děkuji**

MUNI Photometric Archive
[Další informace](#)

Aplikace vymění kód od uživatele a vlastní `client_secret` za token

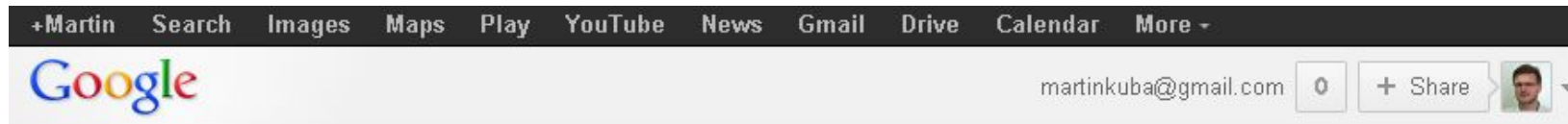
```
} else if ("/auth".equals(req.getPathInfo())) {  
    //process google authorization  
    //check state for XSRF attack  
    String state = req.getParameter("state");  
    String state1 = (String) req.getSession(true).getAttribute("state");  
    if (state == null || !state.equals(state1)) {  
        resp.sendError(HttpServletResponse.SC_FORBIDDEN, "state does not match, probably a XSRF attack");  
        return;  
    }  
    //get code  
    String code = req.getParameter("code");  
    if (code == null) {  
        resp.sendError(HttpServletResponse.SC_FORBIDDEN, "code not present");  
        return;  
    }  
    //exchange code for token  
    RestTemplate restTemplate = new RestTemplate();  
    MultiValueMap<String, String> map = new LinkedMultiValueMap<>();  
    map.add("client_id", client_id);  
    map.add("client_secret", client_secret);  
    map.add("redirect_uri", redirect_uri);  
    map.add("code", code);  
    map.add("grant_type", "authorization_code");  
    JsonNode jsonNode = restTemplate.postForObject(TOKEN_URL, map, JsonNode.class);  
    String accessToken = jsonNode.path("access_token").asText();  
    String expires = jsonNode.path("expires_in").asText();  
    Log.debug("accessToken={ } expires={ }", accessToken, expires);  
}
```

Aplikace volá API

- aplikace volá určitá URL
- prokazuje se tokenem
- odpověď je obvykle JSON

```
//use token for getting user data  
JsonNode userData = restTemplate.getForObject(USER_INFO_URL+"?access_token={access_token}", JsonNode.class, accessToken);  
String userId = userData.path("id").asText();  
String userEmail = userData.path("email").asText();  
String userName = userData.path("name").asText();
```

Uživatel může odebrat oprávnění



Authorized Access to your Google Account

Connected Sites, Apps, and Services

You have granted the following services access to your Google Account:

Android Login Service — Full Account Access [Revoke Access](#)

sfg.google.com — Google Calendar, Google Calendar [Revoke Access](#)

OAuth2 Login Demo — Profile Information [Revoke Access](#)

Google Developers — Google+ You [Revoke Access](#)

DevRates — Profile Information [Revoke Access](#)

ColorNote — [Revoke Access](#)

EasyPolls.net — Profile Information [Revoke Access](#)

MUNI Photometric Archive — Profile Information [Revoke Access](#)

MyTracks — Drive API [Revoke Access](#)

Nastavení aplikací

Vaše jméno, profilová fotka, úvodní fotka, pohlaví, síť, uživatelské jméno a vaše ID číslo jsou vždy dostupné všem uživatelům Facebooku, a to včetně aplikací (důvody). Aplikace navíc mohou přistupovat k vašemu seznamu přátel a ostatním údajům, které jste na svém profilu nastavili jako veřejné.

Aplikace, které používáte	Chcete na Facebooku i jinde používat aplikace, plug-iny, hry a weby?	Zapnuto	Upravit
 CiteULikeAuth		Přátelé	Upravit ✕
 Cities I've Visited		Přátelé	Upravit ✕
 Vimeo		Přátelé	Upravit ✕
 Heureka.cz		Přátelé	Upravit ✕
 Uplay		Přátelé	Upravit ✕
 Geocaching.com		Přátelé	Upravit ✕

 **MUNI Photometric Archive** Poslední přihlášení: 21 únor [Zavřít](#)

Viditelnost aplikace:  **Přátelé** ▼

Tato aplikace vyžaduje:

- Vaše základní informace [?]
- Vaši e-mailovou adresu (makub@ics.muni.cz)

Poslední přístup k údajům: Základní informace [Zobrazit podrobnosti](#) · [Další informace](#) Dnes

Kdy si přejete být upozorněni? **Když vám aplikace pošle upozornění** ▼

[Odebrat aplikaci](#) · [Nahlásit aplikaci](#)

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

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 klientovi přímo
 - resource owner password credentials grant
 - client credentials grant
 - device flow grant - pro SmartTV bez klávesnice

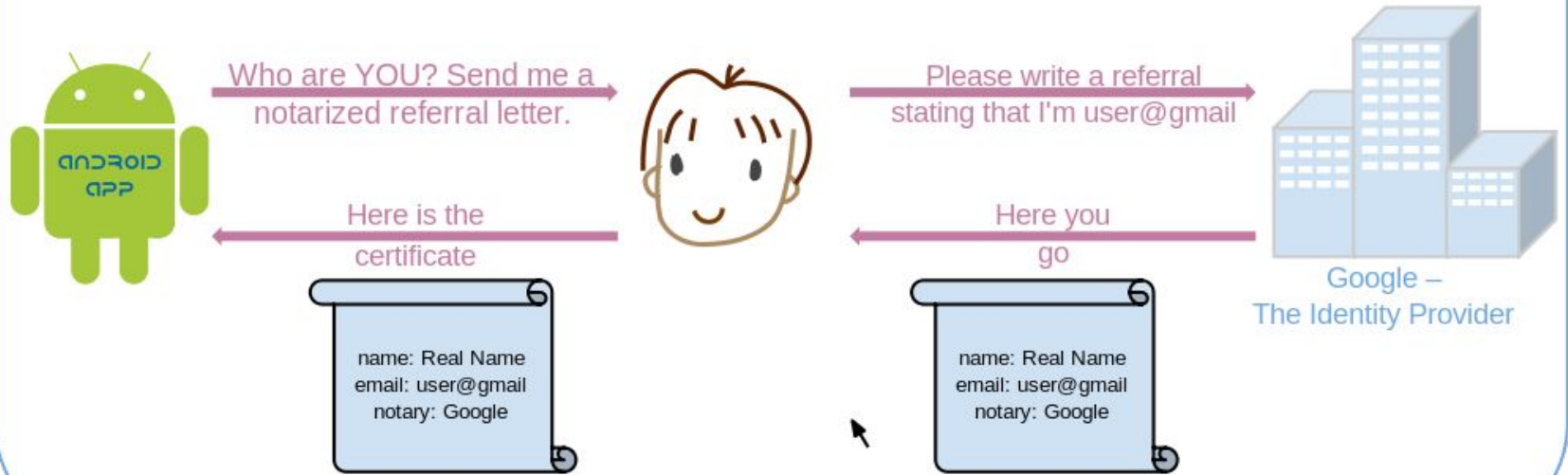
OAuth shrnutí

- uživatel i aplikace jsou zaregistrovány na autorizačním serveru
- oba mají své heslo (client_secret u app)
- poskytovatel API v dokumentaci uvádí možná oprávnění (scopes)
- aplikace žádá uživatele o konkrétní oprávnění (povolení k určité množině operací)
- pokud uživatel schválí, aplikace získá token
- uživatel může kdykoliv token revokovat

Srovnání s OpenID 1.0/2.0

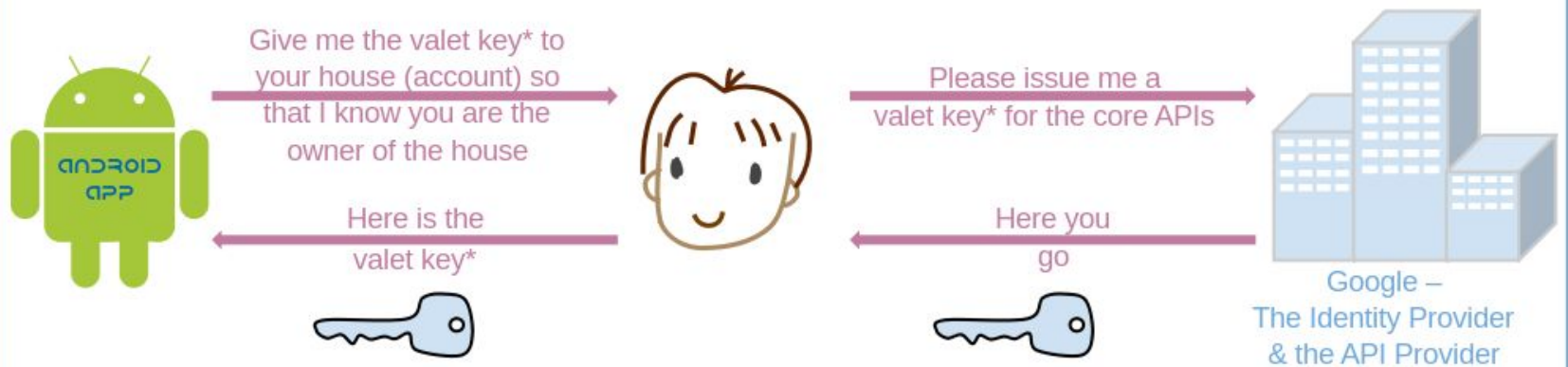
- otevřený standard
 - OpenID - pro autentizaci
 - OAuth - pro autorizaci
- aplikace se registrovat
 - OpenID - nemusí
 - OAuth - musí
- poskytovatel totožnosti
 - OpenID - kdokoliv, ale jak mu věřit ?
 - OAuth - konkrétní, API-specific

OpenID Authentication



vs.

Pseudo-Authentication using OAuth



*valet key = limited scope
OAuth Token

adapted from a drawing by @_nat_en

Srovnání se SAML

- SAML

- autentizace, jako OpenID 1.0/2.0
- potřebuje Discovery Service/WAYF
- uživatel nemá kontrolu nad vydávanými údaji
- IdP a SP se musí vzájemně dohodnout
- SP nemůže požádat uživatele o více informací
- uživatel může schválit všechny nebo nic

- OAuth

- autorizace
- autentizace jako nulová autorizace
- uživatel má kontrolu nad vydávanými oprávněními
- může je i zpětně revokovat

OpenID Connect

- OAuth 2 zajišťuje přihlášení, ale nedefinuje, jak získat údaje o uživateli, každá služba poskytuje 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`

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"  
  }  
}
```

Jednotné přihlášení na MUNI

- systém pro autentizaci podporující protokoly SAML, OpenID Connect a OAuth 2
- uživatelem je každý, kdo má učo a (primární) heslo
- registrace aplikací na <https://spreg.aai.muni.cz/>
- autorizační server na <https://oidc.muni.cz/oidc/>
- tři stroje s plovoucí IP adresou kvůli High Availability
- aplikace: <https://inet.muni.cz/>, <https://portal.muni.cz/>, <http://o365.muni.cz/>, ...
- návod: <https://it.muni.cz/sluzby/jednotne-prihlaseni-na-muni/navody/jak-pripojit-sluzbu-k-jednotnemu-prihlaseni-muni>

Konec

Děkuji za pozornost