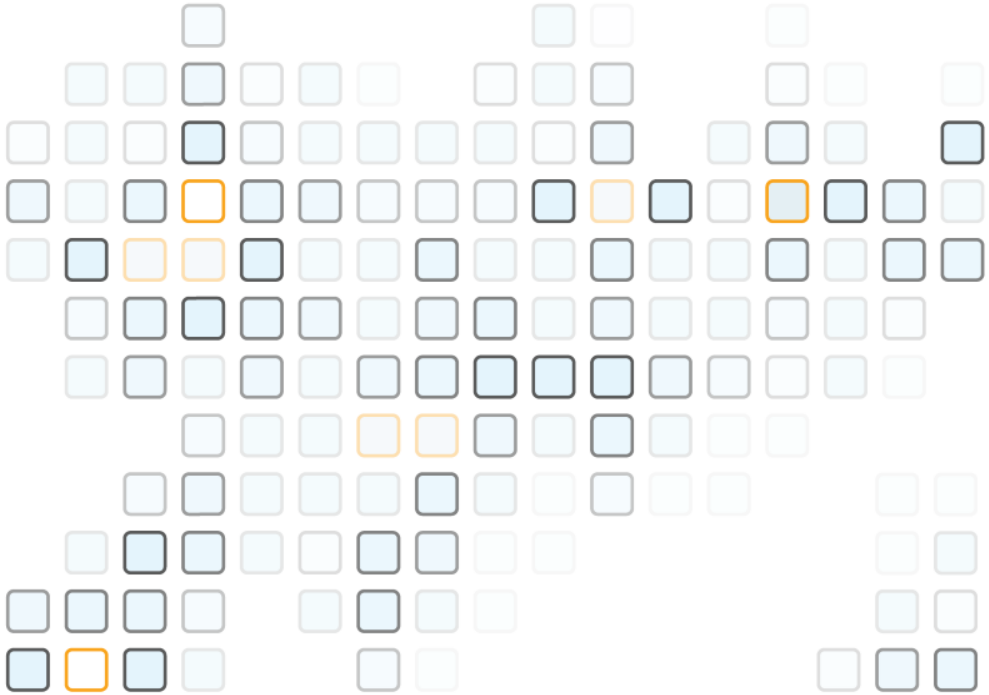


IS/STAG
is-stag.zcu.cz

OAUTH 2.0 v rámci IS/STAG

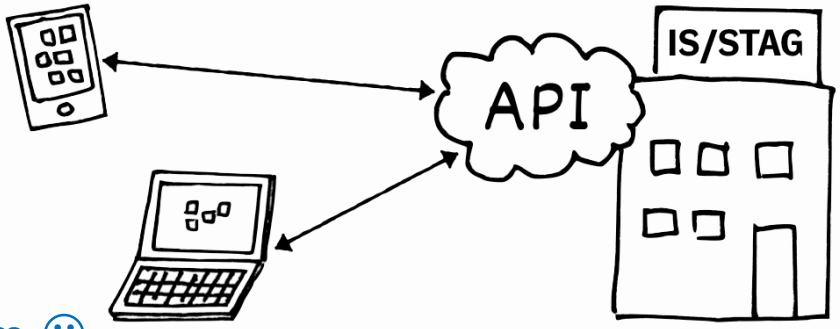
Lukáš Valenta
Ondřej Průcha

Úvod, obsah

- Součástí CRP 2022 je: Implementace OAuth 2.0 do vybraných částí IS/STAG
  - Kromě toho máme další vize, co s letitým modulem WS …
- Historie aplikačního rozhraní
  - Co nás trápí, jaké koule máme na noze…
- Plány
  - OAuth 2.0
  - Verzování API + info o době podpory API
  - Návrh struktury API „od píky“
- Dnes skutečně jen vize/plány…

Historie webových služeb nad IS/STAG

- Vytvořeny v roce 2007
- Tehdy vládl standard „WebServices“ (SOAP)
- REST v plenkách
  - Bez zažitých „best practices“
  - Bez „standardů“
  - Nějak jsme začali…
- + Nezkušený vývojář (já)
  - Snažící se vyhovět všem požadavkům ☺



- Vznikl modul, který se snaží pokrýt všechno možné
  - Způsoby přihlašování
  - Výstupy do různých formátů
  - Používáme modul i pro exporty z části portálové aplikace
  - **Za ty roky se na stávající API napojily stovky klientů, změny jsou těžké/nemožné**
- Dnes jsme chytřejší a zkušenější, zkusíme začít znovu

Co nás konkrétně trápí

- Evidence klientů
  - Vždy jsme měli API „zcela otevřené“
  - Ve smyslu „kdokoliv jej může používat“
  - O našich klientech/aplikacích/webech obecně nic nevíme
    - Neví ale ani provozovatelé/administrátoři našeho systému u zákazníků
    - Máme jen mailing list, kam se mohou dobrovolně přihlašovat
  - Nemáme možnost klienty povolit/banovat/sledovat využití API…
- Způsob přihlašování uživatelů
  - Neumíme donutit vývojáře aplikací, aby to dělali „správně“
    - Historicky bohužel podporujeme metody, které jsou pro ně „snadné a lepší“, ale z hlediska bezpečnosti špatné (např. heslo uživatelů neputuje pouze nám, ale může jej znát i autor externí aplikace)
- Detailní řízení přístupu aplikací ke konkrétním webovým službám
  - „pusťte tuhle naší školní speciální aplikaci jen k této množině WS a dejte tomu tyhle speciální přihlašovací údaje“…
- … a další

OAuth 2.0

- Otevřený standard pro autorizaci
- Rozšířen, uznáván, používán, vznik 2012
- V IS/STAG máme již zkušenosti z pohledu klienta
  - např. Google API, GitHub
- Naučíme se taneční kroky v rytmu OAuthu 2.0 z pohledu serveru ☺

OAuth 2.0

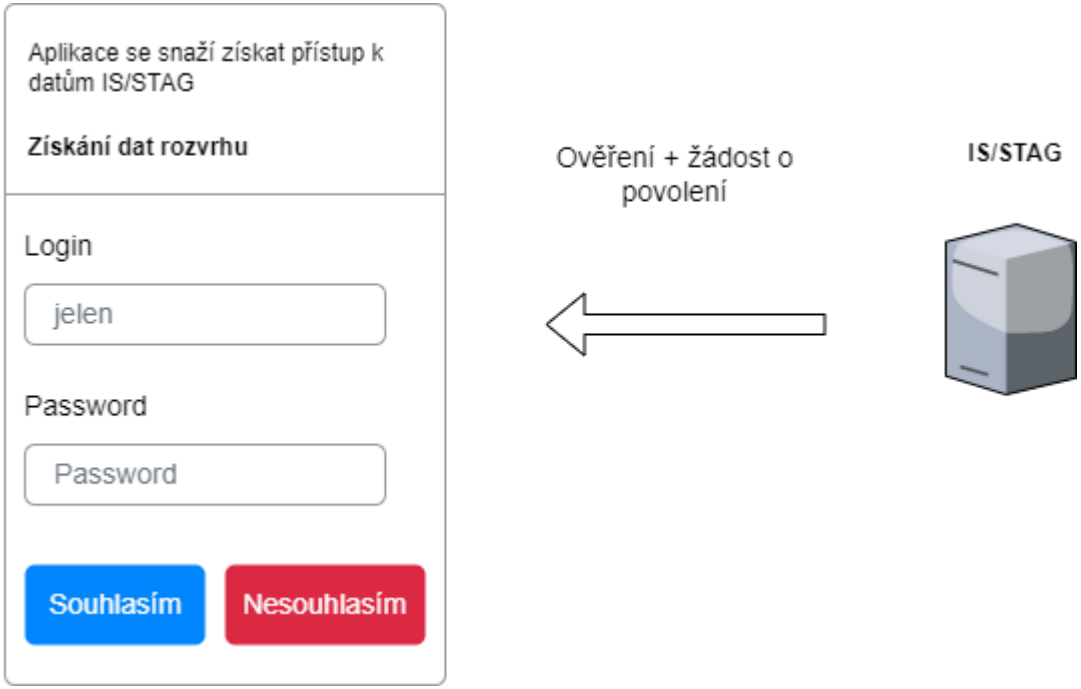
- Výhody
  - Aplikace třetích stran musí být registrovány, aby mohly požádat o data
    - Přístup lze dynamicky měnit dle potřeby
  - Možnost granularity práv uživatelů k datům
    - Lze definovat vlastní množiny práv a ty přidělovat
  - Uživatelé si mohou sami řídit přístup ke svým datům
  - Různé možnosti implementace dle úrovně zabezpečení
    - Počítá se i s variantou komunikace stroj-stroj

OAuth 2.0

- Taneční kroky pro případ autorizace uživatelem
- 1. krok – aplikace žádá data

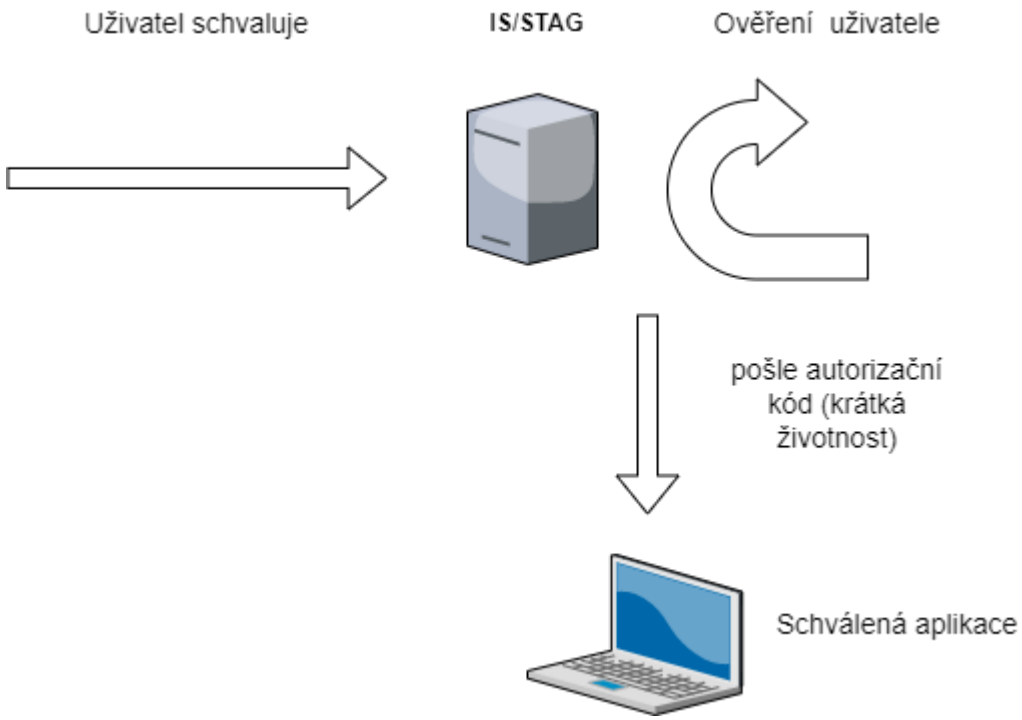
OAuth 2.0

- Taneční kroky pro případ autorizace uživatelem
- 2. krok – autorizace uživatele a povolení přístupu



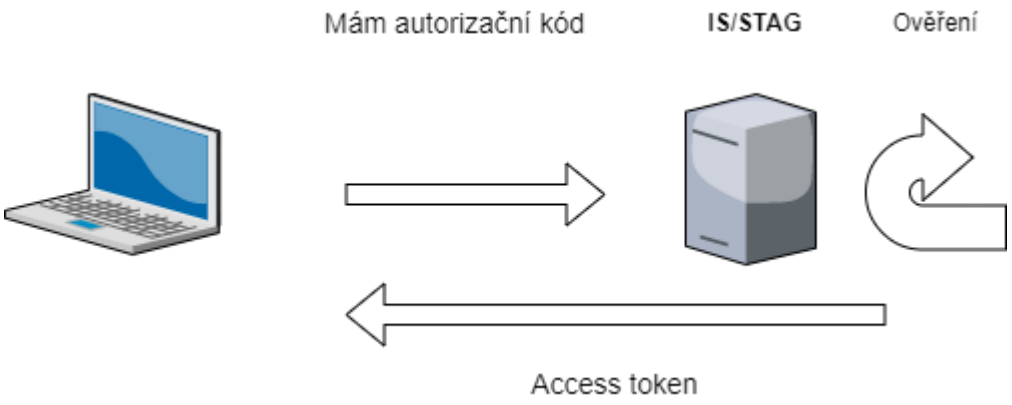
OAuth 2.0

- Taneční kroky pro případ autorizace uživatelem
- 3. krok – získání autorizačního tokenu



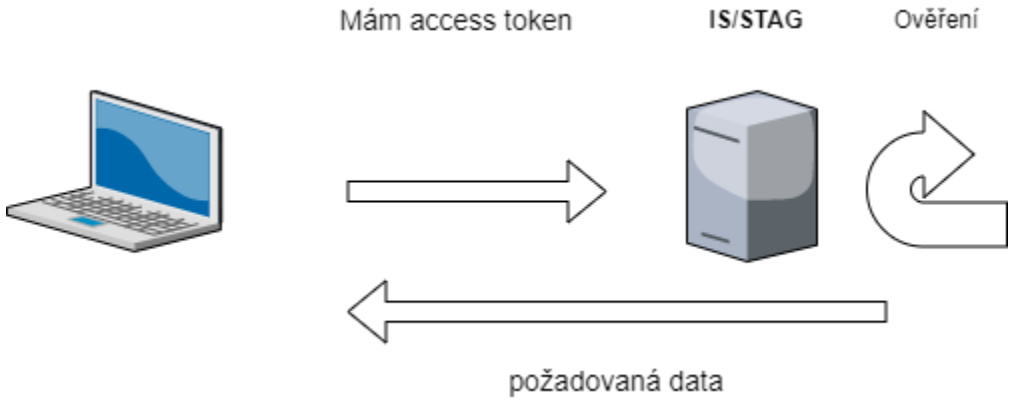
OAuth 2.0

- Taneční kroky pro případ autorizace uživatelem
- 4. krok – výměna autorizačního tokenu za přístupový (access)



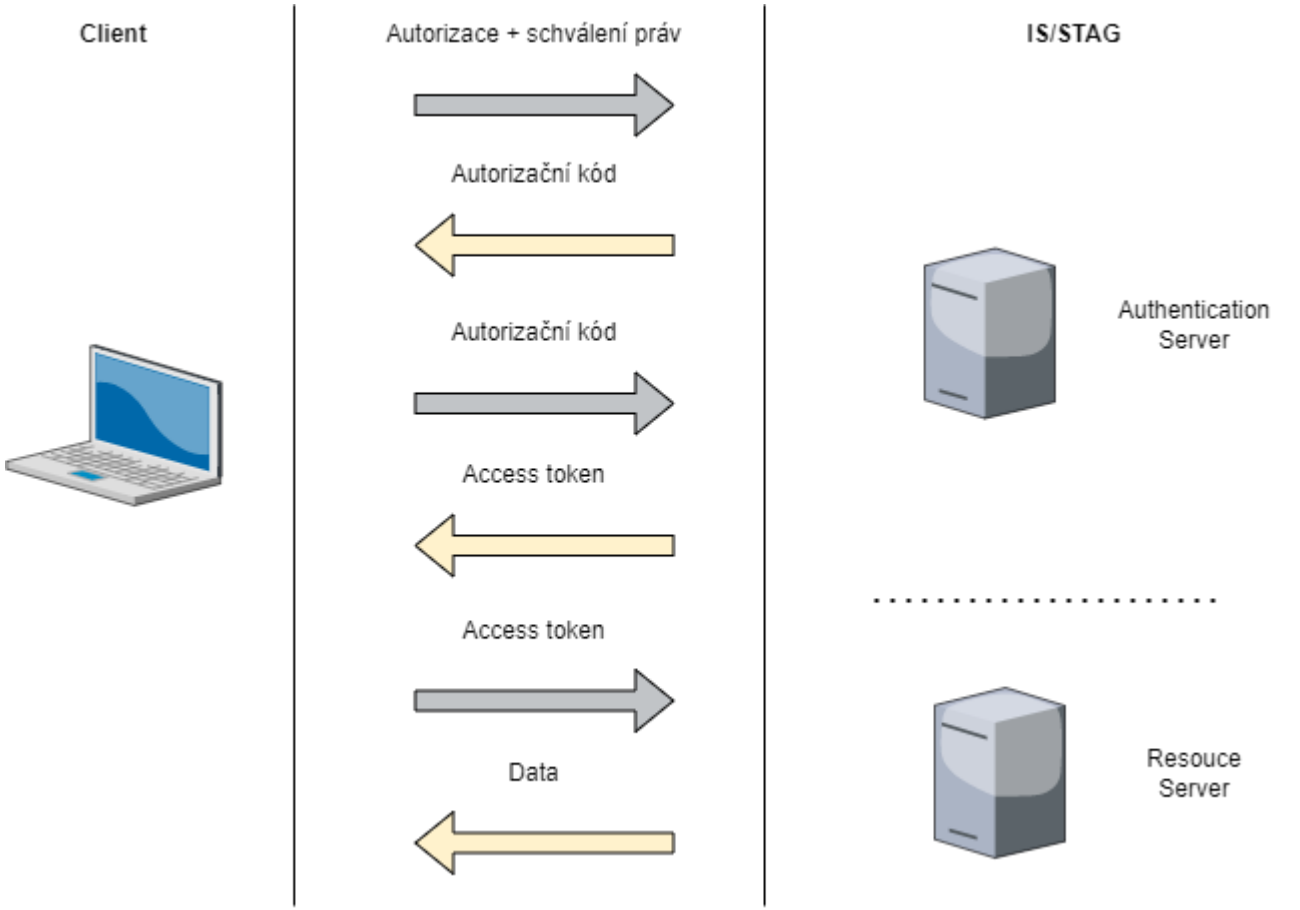
OAuth 2.0

- Taneční kroky pro případ autorizace uživatelem
- 5. krok – vlastní získání dat



OAuth 2.0

- Souhrn kroků



Verzování + plánované rušení API

- Aktuálně garantujeme téměř všechna rozhraní tak, jak jsme je kdy vytvořili
  - Některá tedy garantujeme skoro 15 let ☺
  - Možno dělat pouze zpětně kompatibilní změny (např. přidání nepovinných parametrů)
    - Tedy to, čím nerozbijeme současné klienty
  - Čím dál častěji dochází k chybám při snaze držet tak stará rozhraní
- Dříve jsme dopředu bohužel nepřemýšleli o tom, co a jak dlouho garantovat
  - Klienti nemají takové informace
  - Zatím všichni berou, že to rozhraní je „věčné“

Verzování + plánované rušení API

- Vize: u webových služeb budou **jasně** uvedeny informace o tom, do kdy jsou v dané podobě garantované
  - Nelze garantovat u všech WS neměnnost donekonečna
  - Především budou označeny
    - Naprosto negarantované služby – interní záležitosti (zveřejněné např. „z dobré vůle“, věci co se hodí „na hraní“)
    - Extra dlouhodobě držené služby – např. zakoupené někým na zakázku, kde víme, že službu využívá externí systém a podobně. Garantované tu dobu na jakou zněla původní domluva, pak přejdou do třetí kategorie:
    - Ostatní/běžné WS budou garantovány vždy na X měsíců dopředu (tj. pokud se rozhodneme je změnit/zrušit, oznámíme to těchto X měsíců předem). X předpokládáme něco v rozmezí 6 – 18
  - Taková informace bude v popisech WS uvedena
  - Pokoušeli jsme se cosi takového zavést již dříve, ale skončili jsme u popisu u nás na produktovém webu
- V okamžiku, kdy budeme potřebovat API opravdu změnit, kdy už nepůjde (či nebude koncepční) držet zpětnou kompatibilitu:
  - Vydáme novou verzi daného API: /api/predmety/**v2**/…
  - Původní budeme garantovat ještě tu dobu, jakou jsme dopředu informovali
  - Návrh URL pro nové API bude vždy obsahovat identifikátor verze API (v1, v2)

Vize struktury API „od píky“

- 15 let starý návrh se od současných podob REST API dost liší
  - My jsme vycházeli ještě spíše z „procedurálního“ návrhu, dávno se již API navrhují jinak („dle REST“)
- Vybereme si nějaké best-practices či metodiku a podle toho API redesignujeme
  - Zapracujeme i verzování
  - Pokusíme se maximálně zjednodušit
  - Zařadíme jednotlivé WS i do „OAuth scopes“ pro lepší řízení přístupů

/predmety/getPredmetyByFakulta?fakulta=FAV	/predmety/v1/fakulta/FAV/
/predmety/getPredmetyByPlan?stpIIdno=123	/predmety/v1/plan/123/
/predmety/getPredmetInfo?pracoviste=KIV&zkratka=PPA1	/predmety/v1/info/KIV/PPA1
/znamky/uploadZnamek *(+ POST data s XML/JSON)*	/znamky/v1/zapis *(+ POST data…)*

Závěr

- V rámci letošního CRP jsme si předsevzali:
  - „Prokopnout“ a oživit OAuth 2.0
  - Technicky začlenit do naší aplikace
  - Rozmyslet vše související
- Zároveň bychom chtěli provést vše, o čem jsem mluvil
  - To ale bude trvat déle než do konce 2022…
  - A chceme si velmi rozmyslet všechny důsledky, než to vydáme (závazek na dalších 15 let? ☺)

- Otázky, diskuse?

Děkuji za pozornost