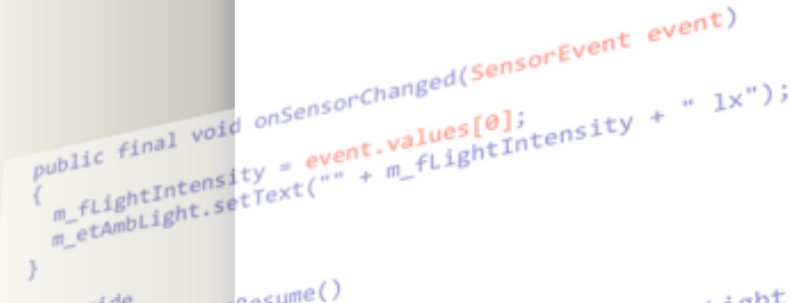
Authentication a Authorization

Jiří Šebek

b6b36nss



Definice

- Autentizace je, když entita prokáže identitu.
  - autentizace prokazuje, že jste tím, kým říkáte, že jste
  - podobá se identifikaci pomocí občanky například
- Autorizace je, když entita prokáže právo na přístup (access)
  - autorizace prokazuje, že máte právo poslat request
  - Příklad : jdete na koncert a máte lístek (nemusíte prokazovat totžnost :))

Motivace

- zabezpečení rozhraní API
- zajišťují, že přístup k nim mají jenom platní uživatelé
- mají přístup jenom k prostředkům, ke kterým mají oprávnění.
- Typy :
  - Basic autorization
  - API klíče
  - Oauth2
  - .. mnoho jiných

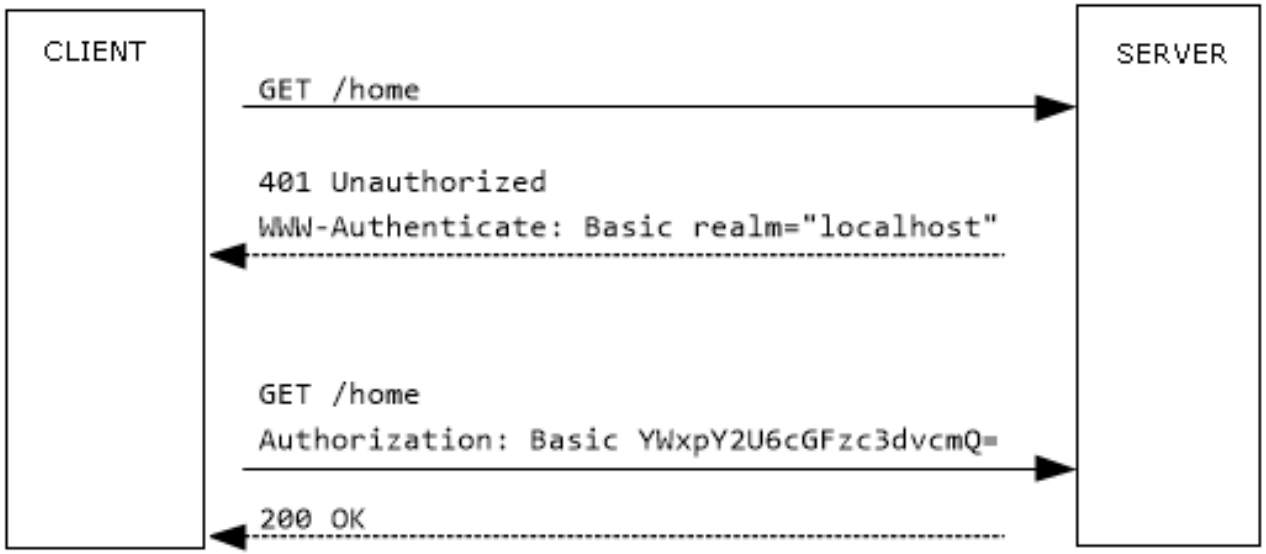
Basic autorization

- Jedná se o metodu, pomocí které se Rest klienti mohou autorizovat u REST Backendu
- Autorizují se pomocí username a hesla
- Využívá se **Encode64**

Princip

- Authorization: Basic <credentials>
  - Credentials – username:password

→ hash pomocí **Encode64**



Pomocné softwary

- Na generování hashe :
  - https://www.blitter.se/utils/basic-authentication-header-generator/

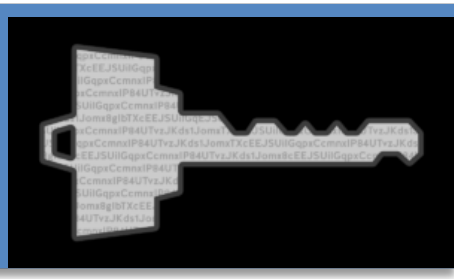
Kde použijeme

- Například na zabezpečení našich endpointu na BE
- Je výhodne v kódu vždy kontrolovat tuto hlavičku ručně ? Není lepší řešení ?
  - The Spring Security Configuration
  - **Inteceptors**
- Jendoduché řešení avšak není zabezpečená komunikace samotná, proto je často potřeba **SSL**, které zpomaluje proces

Interceptor

- V javě se jedná o třídy ve kterých se dá definovat chování které se vykoná před nebo po zavolání některého requestu
- Má metody :
  - PreHandle – před zavoláním kontroleru
  - PostHandle – před posláním response
  - AfterCompletion – po poslání response

API klíče

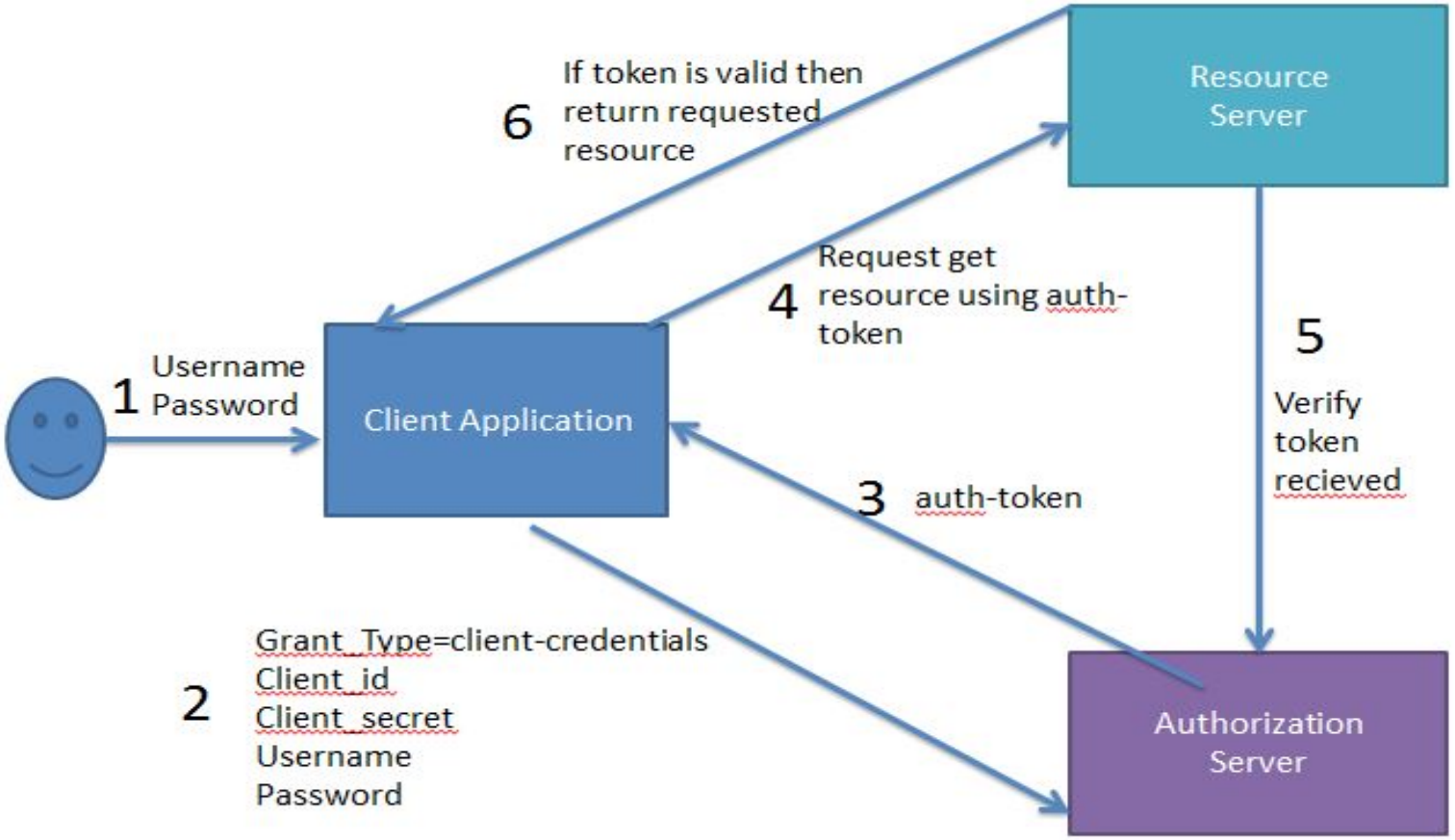


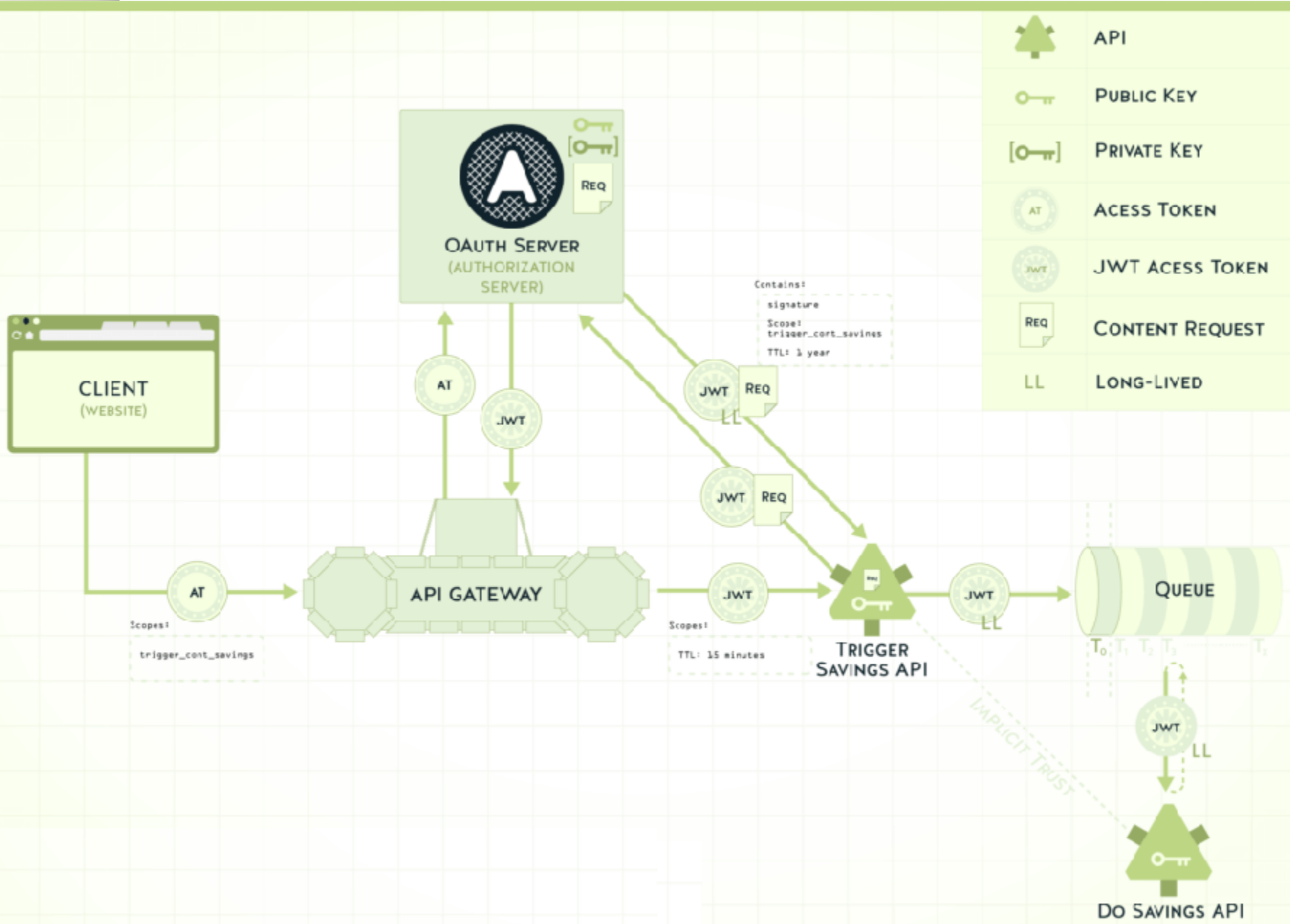
- V tomto přístupu je každému generovanému uživateli přiřazena unikátní hodnota (**API klíč**), což znamená, že uživatele známe
- Když se uživatel pokusí znovu vstoupit do systému, použije se jeho **API klíč** (metoda jak generovat klíč je velmi důležitá), aby dokázal, že je stejný uživatel jako předtím
- Rychlejší než basic auth
  - POZOR ! - **není** způsob autorizace

OAuth2

- Nejlepší možnost ze tří možností
- Autorizace i autentizace

OAuth2


Resource Server
6
If token is valid then return requested resource
4
Request get resource using auth-token
5
Verify token recieved
1
Username
Password
Client Application
3
auth-token
2
Grant_Type=client-credentials
Client_id
Client_secret
Username
Password
Authorization Server


API
Public Key
Private Key
Acess Token
JWT Acess Token
Content Request
LL
Long-Lived
OAuth Server
(Authorization Server)
Req
Contains:
signature
Scope: trigger_cont_savings
TTL: 1 year
Client
(website)
AT
JWT
JWT
Req
LL
JWT
Req
AT
Scopes:
trigger_cont_savings
API Gateway
JWT
Scopes:
TTL: 15 minutes
Trigger
Savings API
JWT
LL
Queue
Implicit Trust
JWT
LL
DO Savings API