
Por dentro do Spring Security com OAuth2

Ronaldo Lanhellas

OAuth2 é um padrão para autorização, você não baixa, não instala, mas segue.

<https://oauth.net/2/>

A close-up photograph of actor Jim Carrey with a wide-eyed, shouting expression, showing his teeth. He is wearing a dark suit jacket over a white shirt. The background is dark and out of focus.

ALRIGHTY THEN

A documentação é grande mas vamos focar no que importa ...

- Conceitos Chave
- Fluxos de funcionamento do OAuth2
- OpenID
- Spring Security e seus projetos,
com *show me the code*

Precisamos conhecer algumas palavras chaves antes de começar...

- **Resource Server:** Quem sua API irá chamar, ou seja, a API que contém o recurso que você deseja acessar.
- **Client Application:** Quem está querendo acessar o recurso do Resource Server. Pode ser uma outra API, uma aplicação Web e etc.

Precisamos conhecer algumas palavras chaves antes de começar...

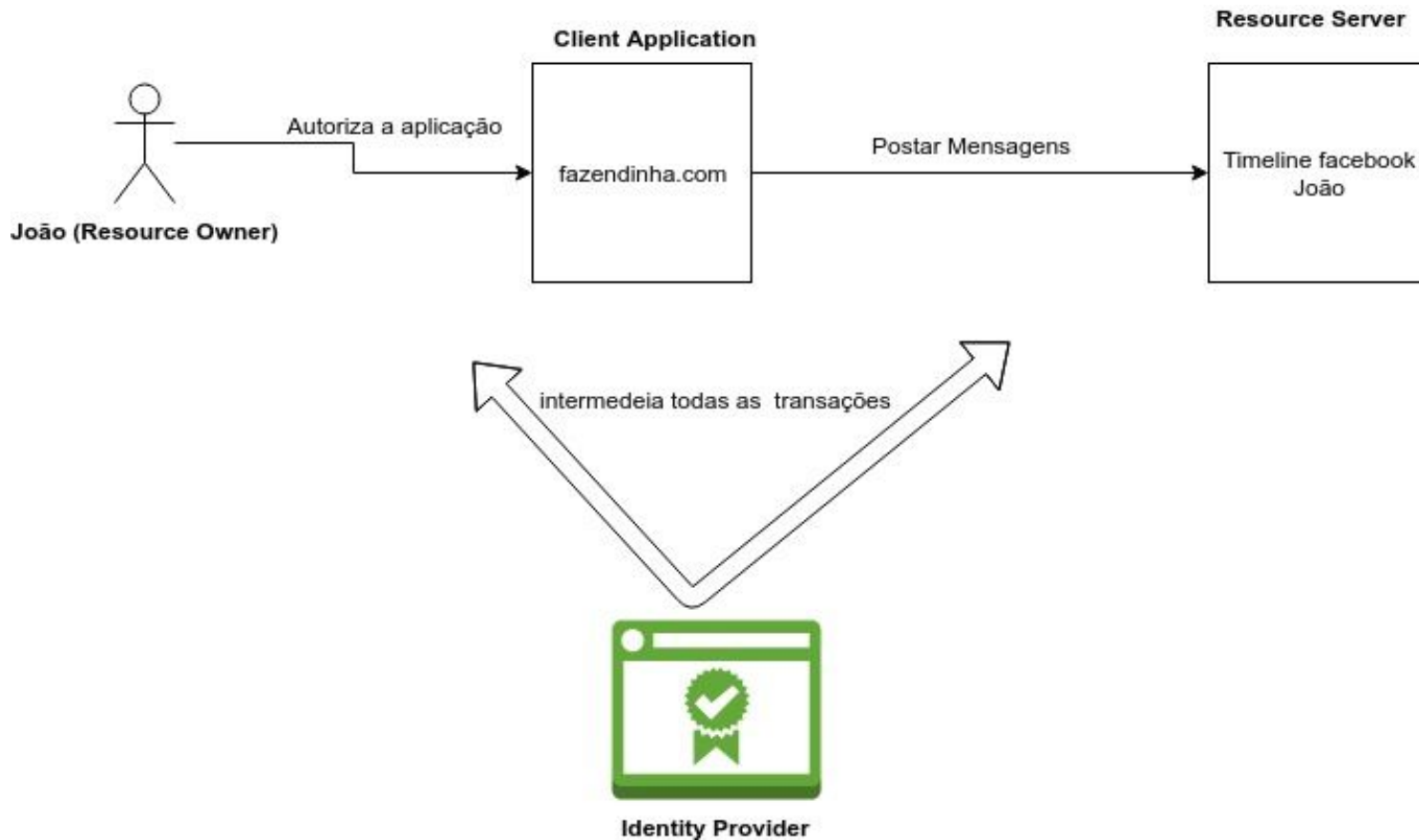
- **Authorization Server ou Identity Provider (IdP):** O responsável por gerenciar as permissões entre o **Client Application** e o **Resource Server**.

Precisamos conhecer algumas palavras chaves antes de começar...

- **Dono do Recurso (Resource Owner):** É quem de fato pode autorizar o acesso. Lembrando que não é ele quem chama, apenas autoriza. Vamos ver exemplos práticos, fique tranquilo.

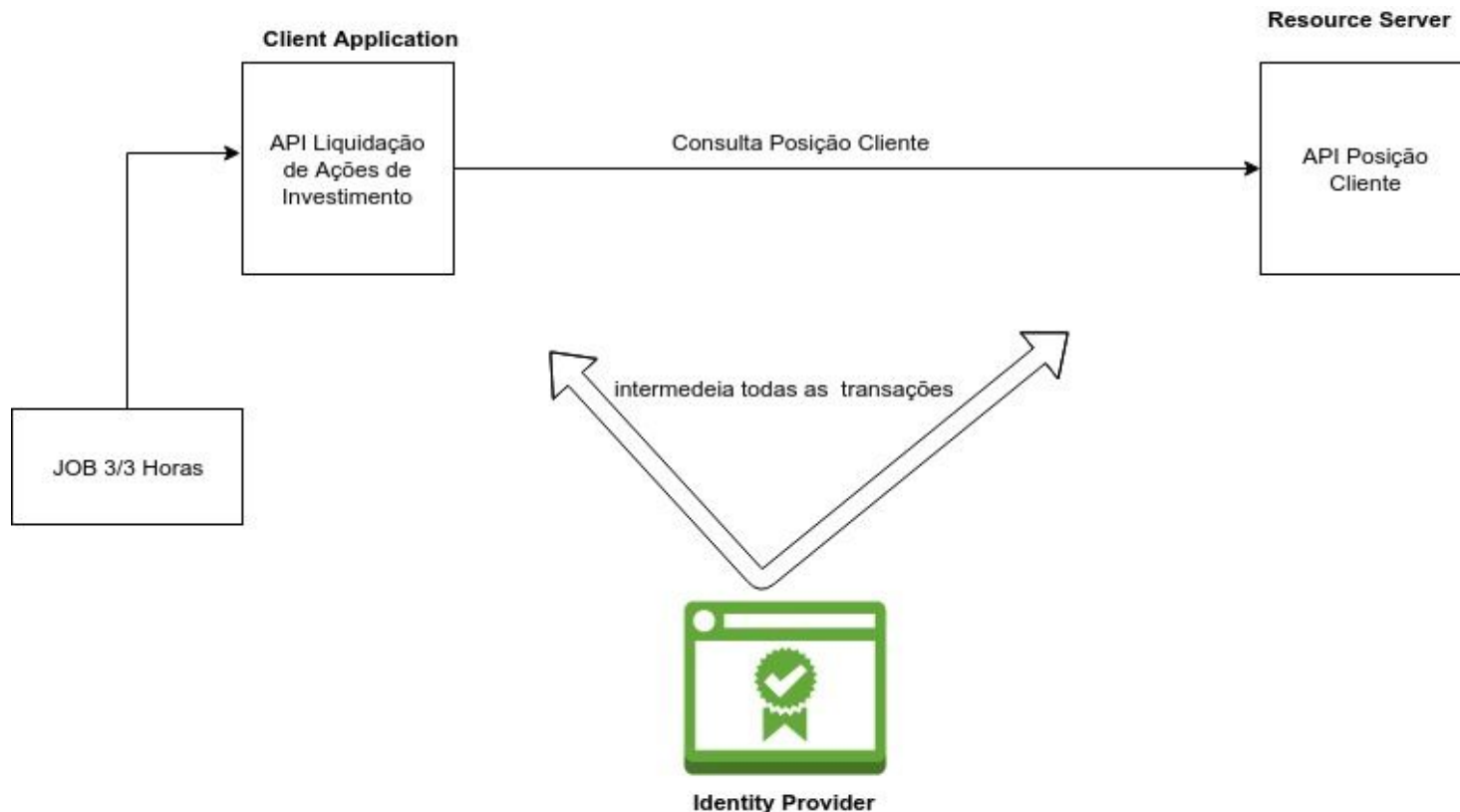
Como todos esses papeis se conversam ?

Caso 1



Como todos esses papéis se conversam ?

Caso 2



Só isso ? Acabou ? Não.

Precisamos entender os fluxos de autenticação.

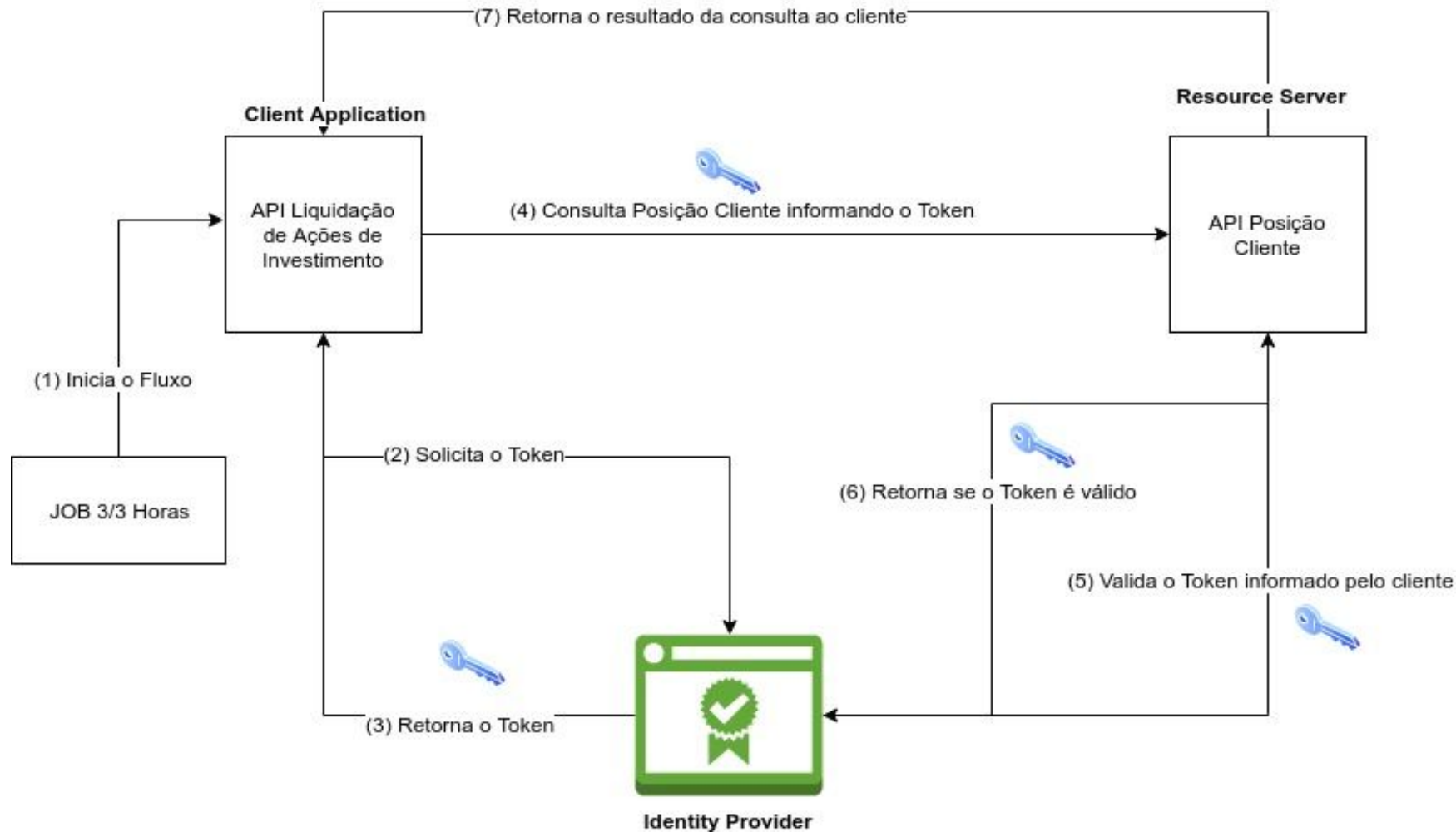
- São formas padronizadas de realizar a autenticação/autorização de uma aplicação.
- Existem pelo menos 4 e veremos todos aqui: **Client Credentials, Implicit, Resource Owner Password Credentials, Authorization Code.**

Client Credentials Flow

Autenticação de API para API

Client Credentials

- Se você tem 2 API's que precisam conversar entre si, esse é seu fluxo.
- Não temos intervenção humana nesse fluxo, tudo é automático, máquina conversando com máquina.
- É o mais simples de todos.



Implicit Flow

Para SPA (Single Page Application)

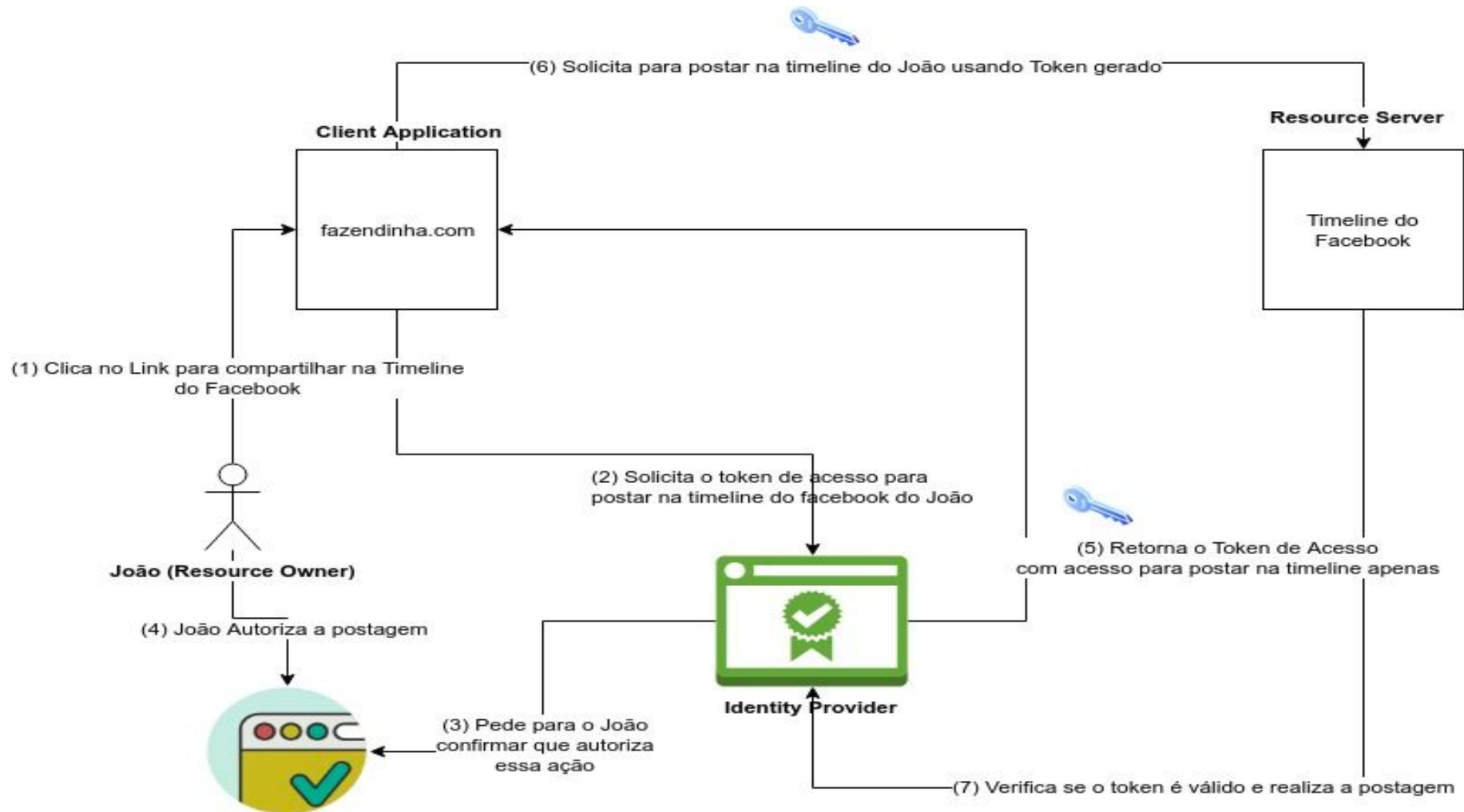
Implicit Flow

- Se você tem uma página renderizada no lado do cliente (Javascript, TypeScript, Angular e etc), então você precisa desse fluxo.
- Não confiamos na aplicação que está no lado do cliente, então precisamos ocultar o máximo de informações.

Implicit Flow

- Qualquer um pode usar o *inspect code* do navegador e ver seu código, suas credenciais e tudo mais.
- Pense na primeira **Lei de Murphy**: Se algo pode dar errado, dará.



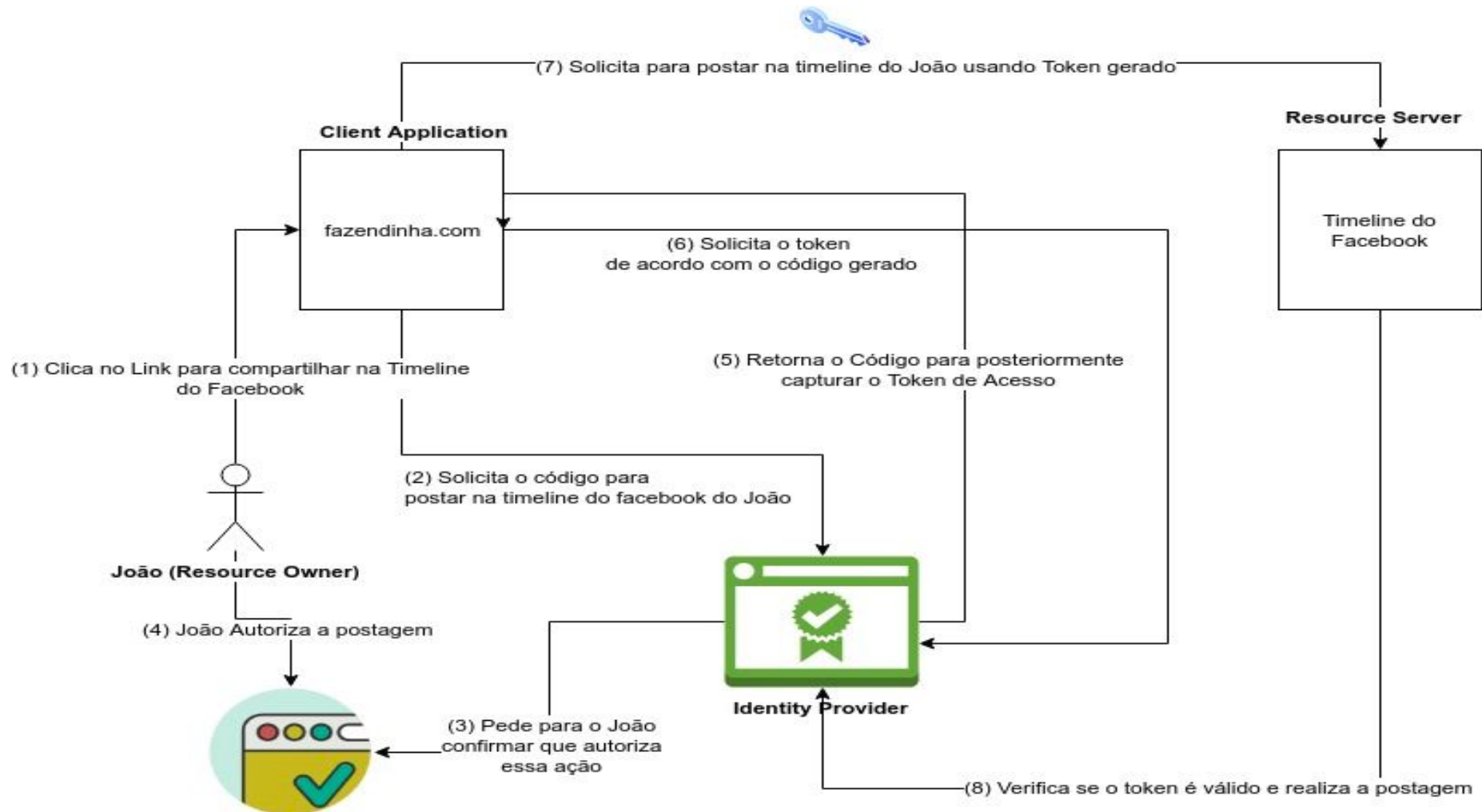


— Authorization Code Flow

Para aplicação Server-Side, mais confiáveis.

Authorization Flow

- Se você tem uma aplicação renderizada Server-Side (JSF, JSP e etc) então esse é seu fluxo.
- Suas credenciais ficam “escondidas” no servidor então podemos ter um pouco mais de confiança aqui.
- Só tem 1 passo a mais que o Implicit Flow.



— Password Flow

Para aplicações que não podem
ser Authorization Code Flow
mas ainda são muito confiáveis.

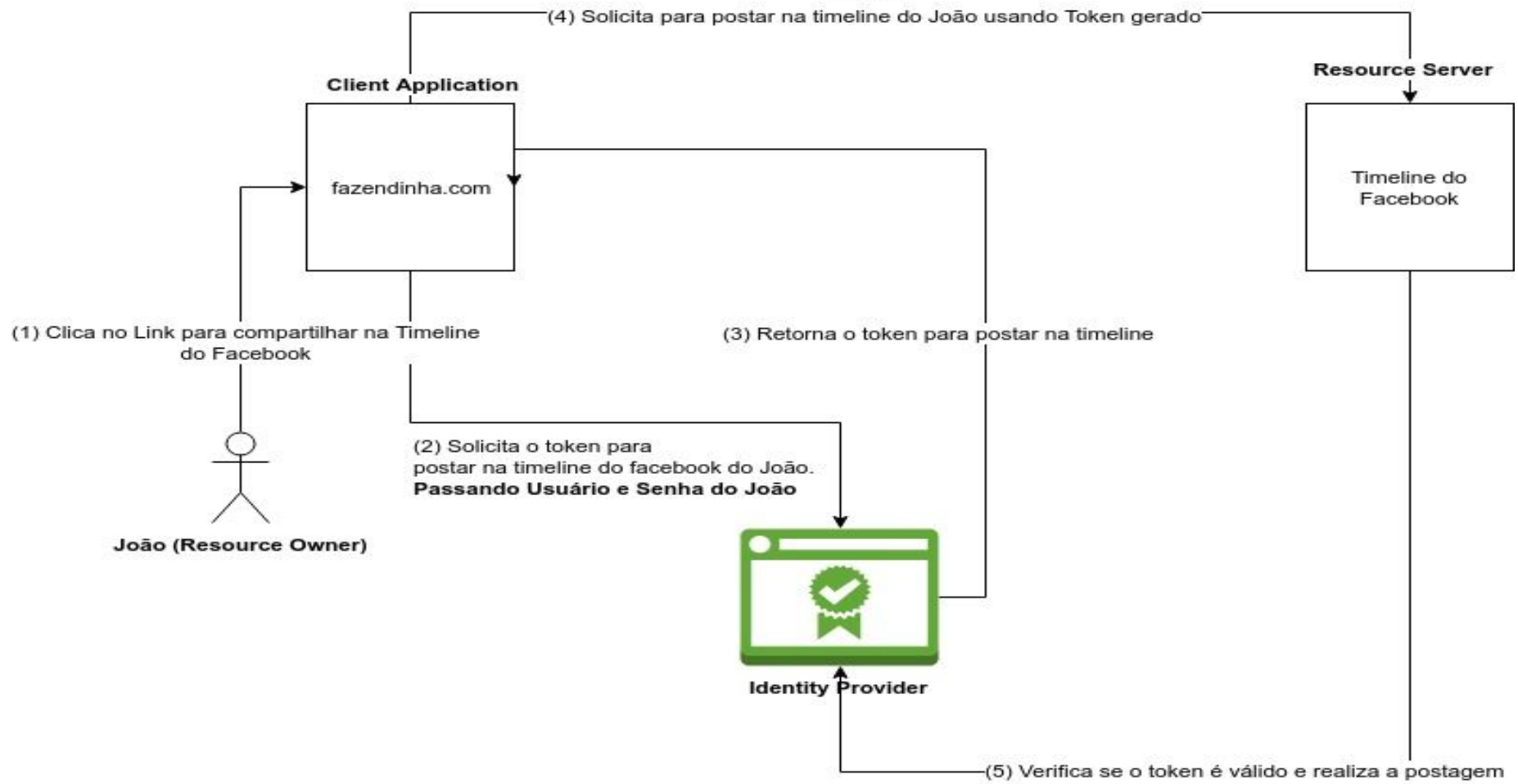


Password Flow

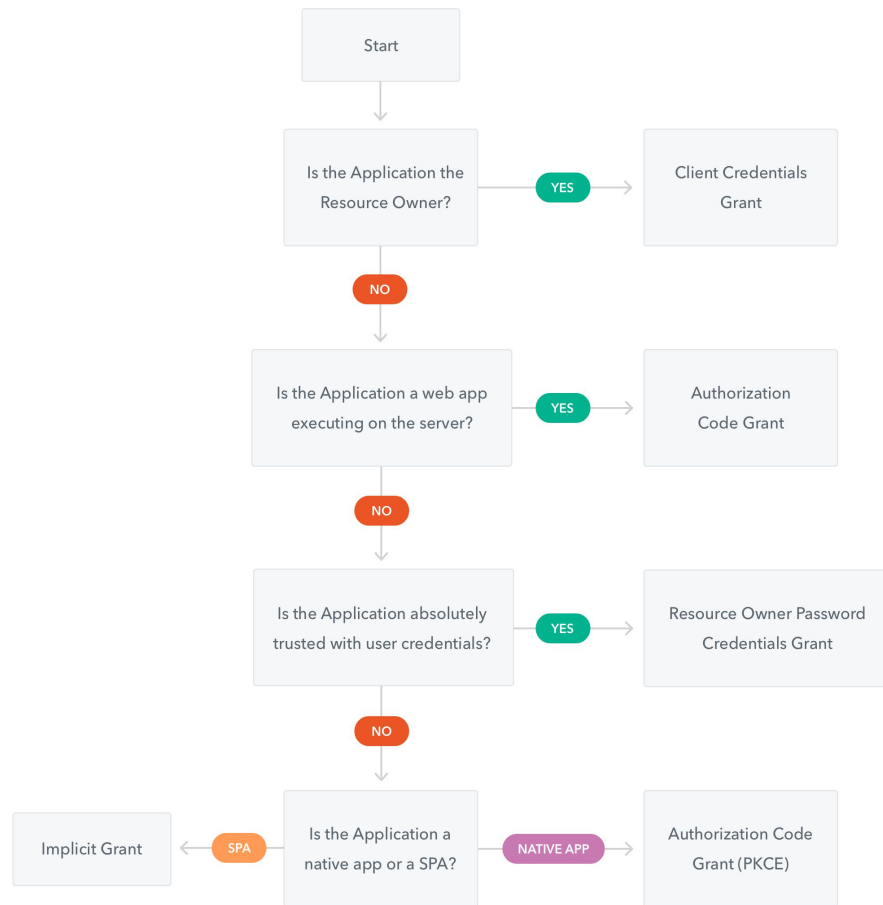
- Se por algum motivo você está renderizando páginas no lado do servidor (Server-Side) e não pode usar Authorization Code Flow, então só lhe resta o Password Flow.
- Nesse fluxo não existe autorização prévia do usuário, você já está passando o usuário e senha dele então partimos do princípio que tudo já está autorizado.

Password Flow

- Você tem que realmente confiar na aplicação que você está usando esse fluxo.
- Repense ... Você realmente precisa desse fluxo ? Normalmente quem usa Password Flow quer na verdade só um Client Credentials.



Se ainda assim você não consegue decidir qual usar...



E o OpenID ? Onde entra nisso tudo ?

- É uma “camada” a mais para o OAuth2, adicionando mais recursos.
- O OAuth cuida da autorização e o OpenID da autenticação.
- O OpenID gera o famoso **id_token** , que diferente do **access_token** (gerado pelo OAuth), cuida das informações de autenticação do usuário.

E o OpenID ? Onde entra nisso tudo ?

- OAuth2 pode usar o OpenID para autenticar um usuário que irá autorizar algo, mas lembre que seu objetivo final é autorizar, não autenticar.

E o OpenID ? Onde entra nisso tudo ?

- 1) Você acessa o site fazendinha.com e clica em Logar com facebook.
- 2) Você é redirecionado para tela de login do facebook e após logar volta para o site da fazendinha.com, já logado pois o fazendinha.com recebeu informações importantes sobre você através do **id_token** e criou sua sessão de login.

Até aqui você usou o **OpenID** ...

E o OpenID ? Onde entra nisso tudo ?

- 3) fazendinha.com é um joguinho e você arrasa na pontuação e quer compartilhar na sua timeline do facebook.
- 4) Então você clica no botão “compartilhar pontuação no facebook” .
- 5) Todo processo de Authorization Code Flow é feito, inclusive aparece um *popup* pra você aceitar que o fazendinha.com pode postar na sua timeline.

E o OpenID ? Onde entra nisso tudo ?

6) Mas você não precisa logar denovo, você já fez isso com o OpenID, lembra ?

7) Por fim sua pontuação é compartilhada no facebook.

Agora você usou **OAuth2...**

Chega de teoria, vamos
falar de Spring Security ...



Um milhão de maneiras de fazer ...

- Existem 4 projetos diferentes do Spring que tentam resolver segurança com OAuth.
- Como saber qual escolher ?
- Cada tutorial na internet faz de uma forma diferente.



Matriz de Comparação


Spring Security (5.1+)	Spring Security OAuth (2.2+)	Spring Cloud Security (1.2+)	Spring Boot OAuth2 (1.5+)
---------------------------	---------------------------------	---------------------------------	------------------------------

<https://github.com/spring-projects/spring-security/wiki/OAuth-2.0-Features-Matrix>


O que a Pivotal pretende com isso ? OAuth será atualizado em todos os projetos ?

NÃO






O plano é ter
todas as
funcionalidades
do OAuth no
Spring Security :)



Spring Boot 2.0
não tem suporte
nativo ao Spring
Security OAuth :(



Spring Security
OAuth não
receberá mais
grandes
atualizações :(



Mas há uma luz no fim
do túnel, ainda
podemos usar Spring
Boot 2.0 com Spring
Security OAuth...

Spring Security OAuth Boot 2 AutoConfig

Uma dependência para resolver nossa vida !!!!

`spring-security-oauth2-autoconfigure`

Showwww
meeeeeee the
coooooooooode



Identity Server , a.k.a IdP

- Vamos por partes ... Baby Steps ...
- Quais dependências devemos adicionar no pom.xml ?

Identity Server , a.k.a IdP

- E a anotação @EnableAuthorizationServer ? Porque precisamos dela ?

Identity Server , a.k.a IdP

- Acabou ? Não, nem começamos.
- Precisamos definir quais usuários podem autenticar para autorizar a geração de tokens (Implicit Flow, Password Flow e Authorization Code Flow).

Identity Server , a.k.a IdP

- Podemos definir esses usuários sem usar o Spring Security OAuth2

- Vamos ver a classe

WebSecurityConfig



Identity Server , a.k.a IdP

- Estamos quase lá ...
- Já temos os usuários, precisamos das aplicações que o nosso IdP irá controlar.

Identity Server , a.k.a IdP

- Vamos olhar a classe ***AuthConfig*** com calma.
- É nela que definimos todas as nossas aplicações.

Identity Server , a.k.a IdP

- Vamos olhar a classe ***AuthConfig*** com calma.
- É nela que definimos todas as nossas aplicações.

Resource Server , a.k.a API que você vai chamar :D

- **Quais dependências precisamos no pom.xml ?**
- **Exatamente as mesmas que usamos no IdP, então vamos poupar tempo e pular essa parte.**

Resource Server , a.k.a API que você vai chamar :D

- Vamos começar configurando e entendendo o arquivo **application.yml**

Resource Server , a.k.a API que você vai chamar :D

- E as anotações `@EnableResourceServer` e `@EnableGlobalMethodSecurity` ?

Resource Server , a.k.a API que você vai chamar :D

- Agora precisamos definir nossos *Endpoints* que serão chamados pelos nossos clientes.
- Mas não é só definir, precisamos **PROTEGER**. Vamos para a classe **AccountController**.

Isso é só a ponta do IceBerg ...

- Spring Security OAuth e o Spring Security são projetos complexos com milhares de funcionalidades



Isso é só a ponta do IceBerg ...

Mas não se deixe abalar, segue os links pra você começar a jornada:

- *<https://projects.spring.io/spring-security-oauth/docs/oauth2.html>*
- *<https://docs.spring.io/spring-security-oauth2-boot/docs/current-SNAPSHOT/reference/htmlsingle/>*

E claro ... *<https://stackoverflow.com/>*

E se ainda assim precisar de um ombro amigo ...

ronaldo.lanhellas@itau-unibanco.com.br

Conte comigo !!

