OAuth 2.0.

O OAuth 2.0. é um protocolo de delegação, permitindo a um dono dum recurso autorizar uma aplicação para aceder ao recurso, sem que esta possa fazer passar-se pelo dono do recurso.

Entidades:

- Resource Owner (RO) - Tem acesso a uma API e pode delegar acesso a essa API. O resource owner é usualmente uma pessoa e é geralmente assumido que usa um browser.

- Protected Resource (PR) -

Componente sobre o qual o RO tem acesso. É geralmente assumido que o PR é uma web API.

↳ operações ler, escrever, ...

- Client - Peça de software que acede ao PR em lugar do RO.

- Authorization Server (AS) - Entidade confiável pelo PR que emite credenciais de segurança, chamadas OAuth Access tokens (AT), aos clientes.

Para adquirir um AT, o cliente envia o RO ao AS para que o RO autorize o cliente junto do AS.

3. O RO autentica-se no AS e autoriza o cliente para aceder a um conjunto de scopes (subconjunto de funcionalidades).

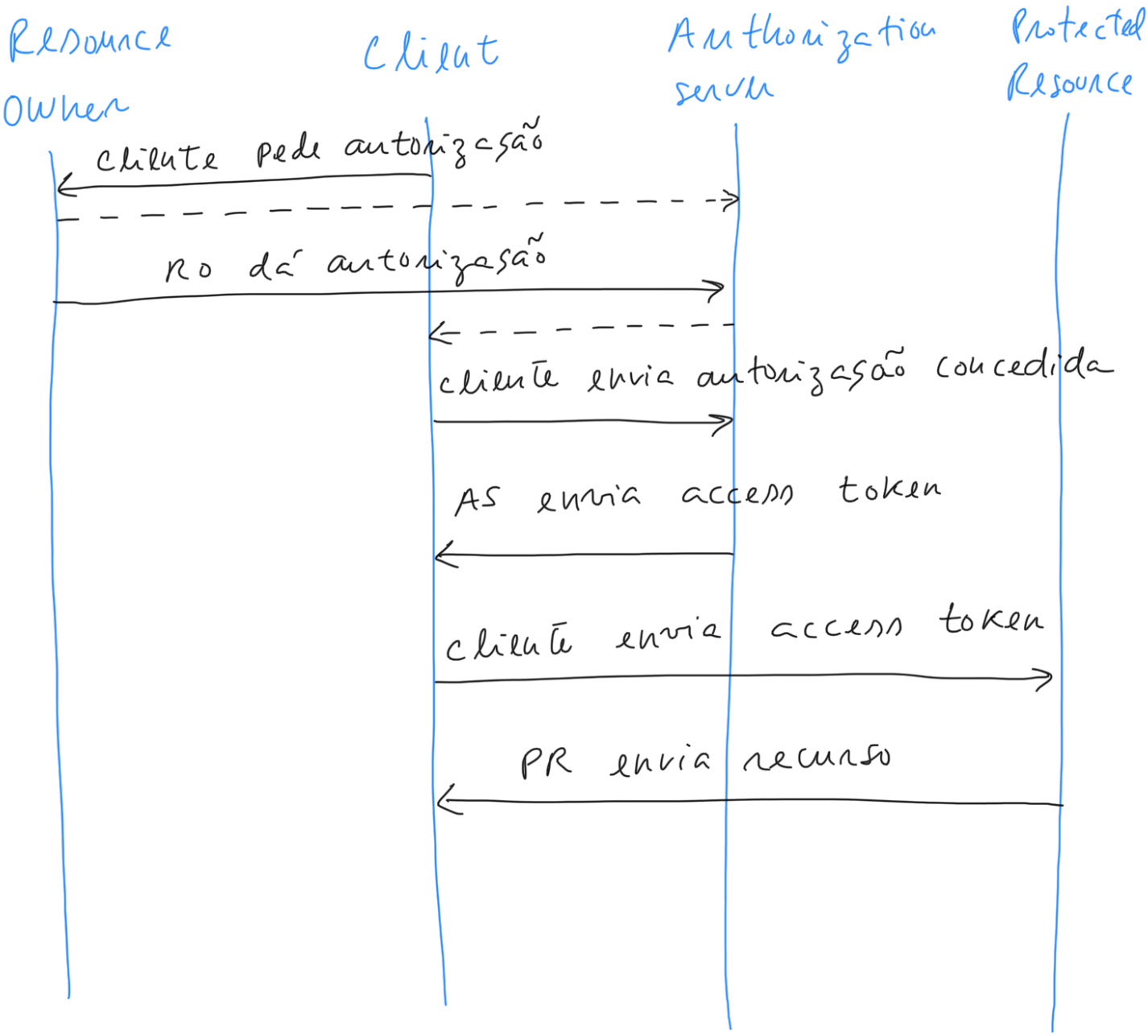
Assim que o AS autorize o cliente, o cliente pode solicitar um Access Token (AT) ao AS.

O AT é apresentado ao PR pelo cliente para aceder à sua API, como delegado pelo RO.

As credenciais do RO nunca são expostas ao cliente.

Ex: RO - Dono de fotografias
Client - Serviço web de impressão de fotografias
PR - servidor onde estão as fotografias

Visão do processo OAuth

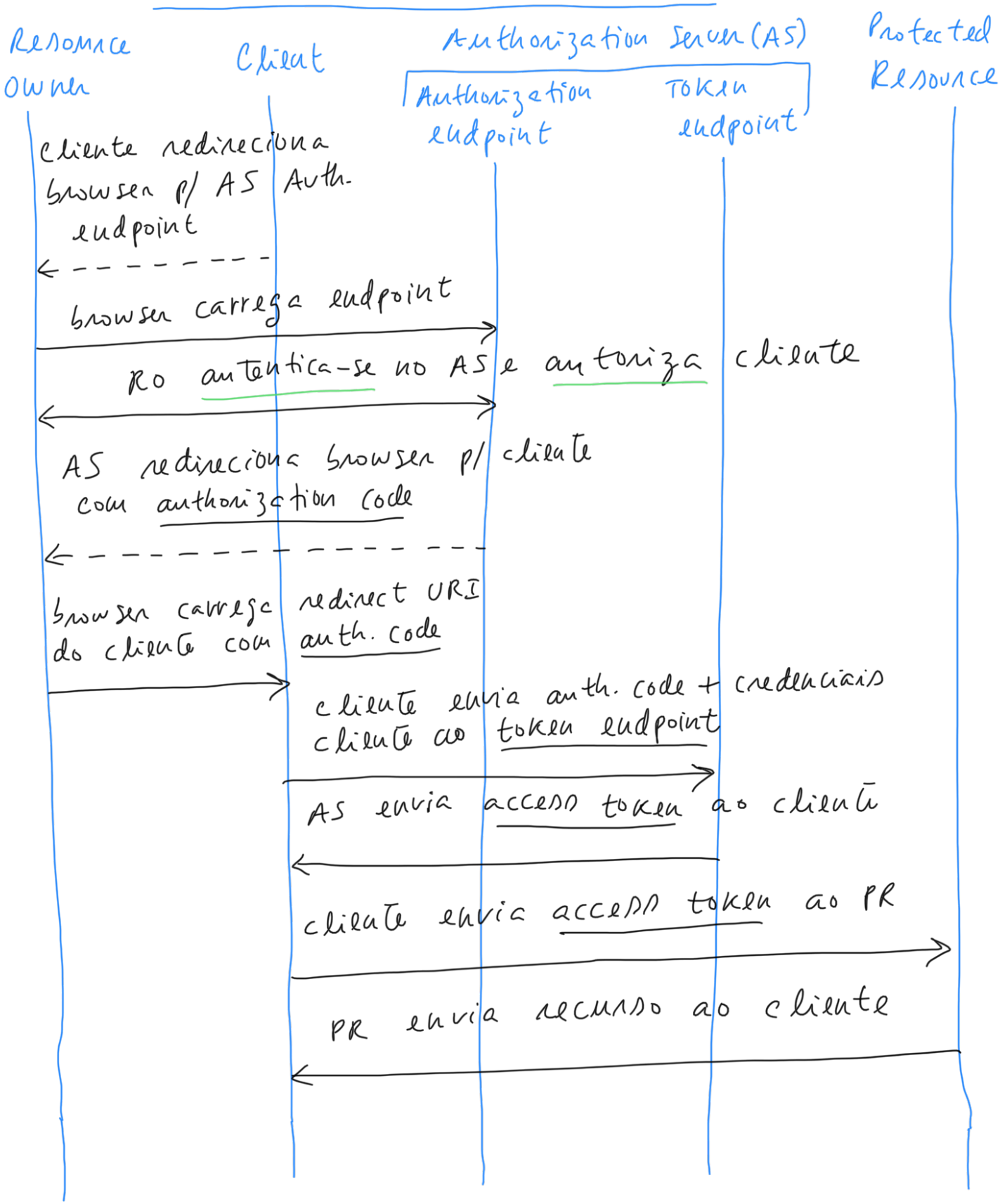


Nota: O AS tem que conhecer o cliente para que possam interagir. Assim, o cliente tem que se registar junto do AS.

P. ex., através duma "developer portal"

Após o registo, o AS atribui um client_id e um client_secret (no caso de se tratar dum confidential client).

Visão Detalhada do OAuth



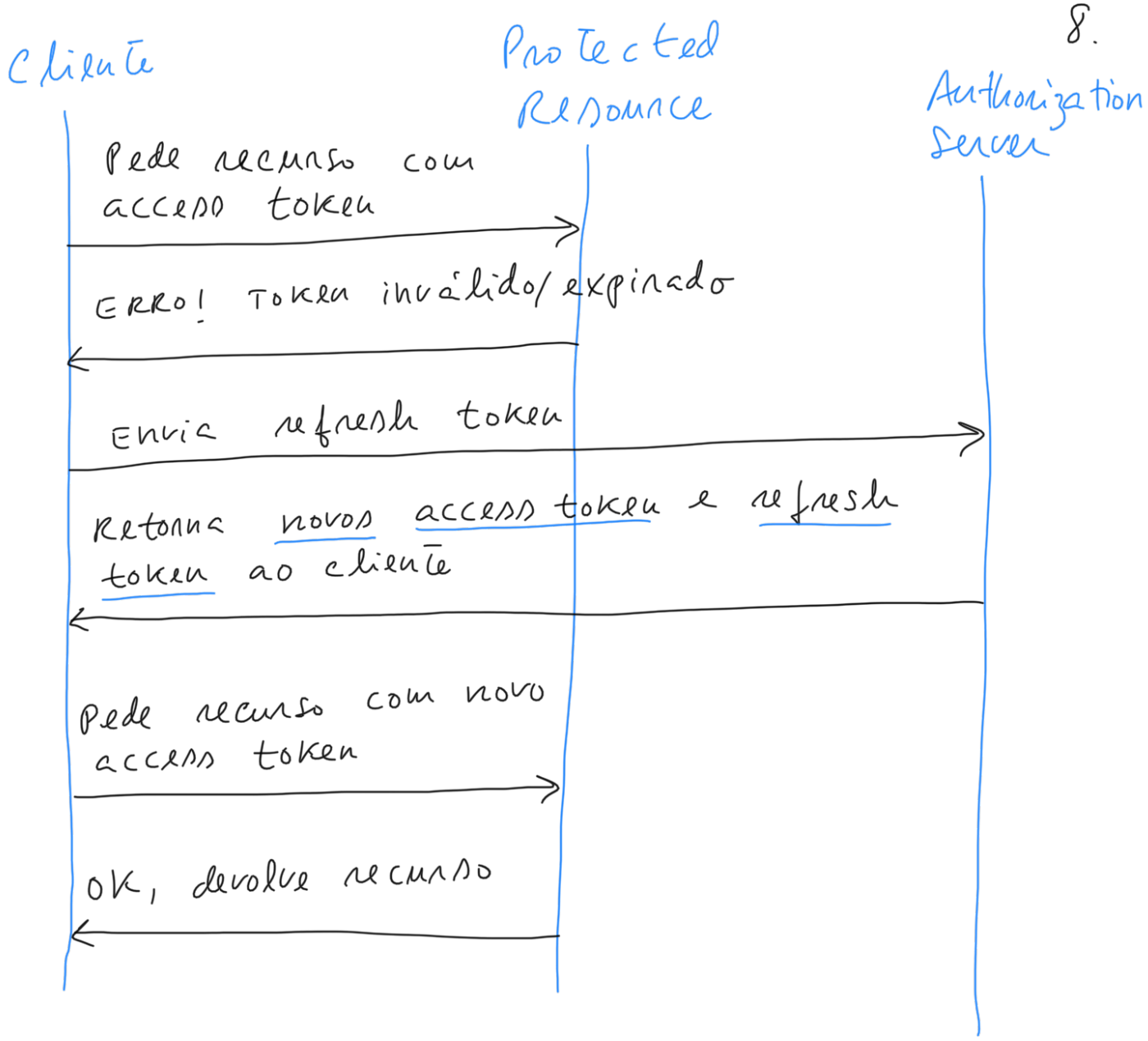
No OAuth são usados Bearer tokens (tokens de Portador)

↳ vantagem: cliente não precisa de descodificar / processar os Bearer tokens (são opacos) e, uma vez na sua posse, pode usá-los sem restrições (funciona como uma password).

↳ existem outros tokens, ex: JWT (JSON Web Tokens), PoP (Proof of Possession), ...

Refresh tokens (RT) - Semelhante a um access token, mas um refresh token nunca é enviado ao Protected Resource.

O cliente envia um RT ao AS para solicitar novo access token sem envolver o Resource Owner (este pode até já não se encontrar online).


Cliente
Protected Resource
Authorization Server
Pede recurso com access token
ERRO! Token inválido/expirado
Envia refresh token
Retorna novos access token e refresh token ao cliente
Pede recurso com novo access token
OK, devolve recurso

Scopes - Mecanismo usado para limitar acesso do cliente aos recursos.

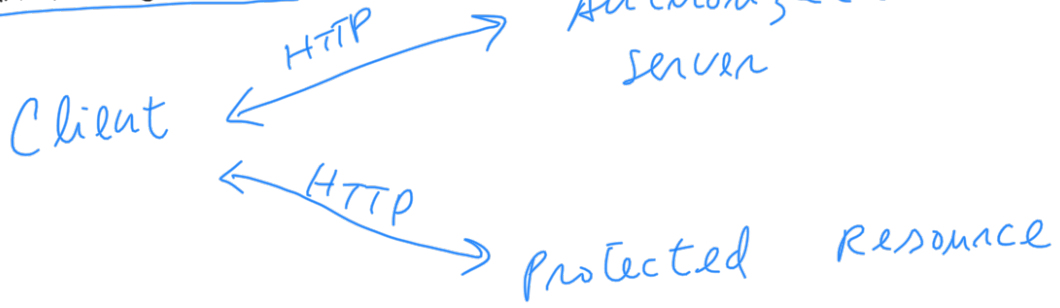
↳ definido pelo Protected Resource

↳ representado por conjunto de strings
ex: read-photo, read-metadata, ...

ou URI

Interação entre atores e componentes:
back channel e front channel

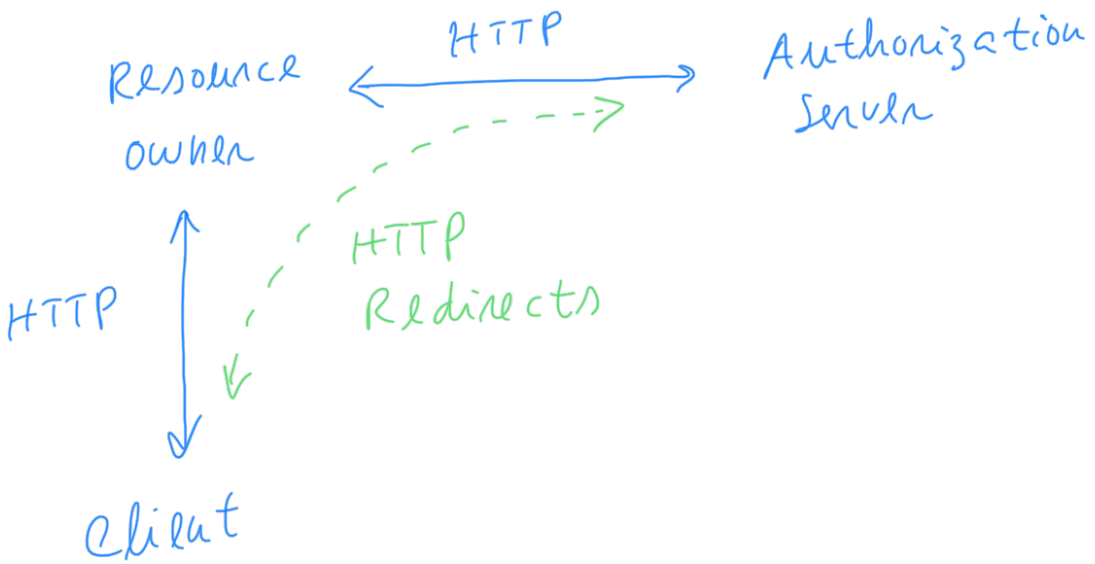
Back channel - Pedidos e respostas HTTP entre componentes OAuth (usando headers, query parameters, methods, body) que não envolvem o user agent (browser) nem o Resource Owner.

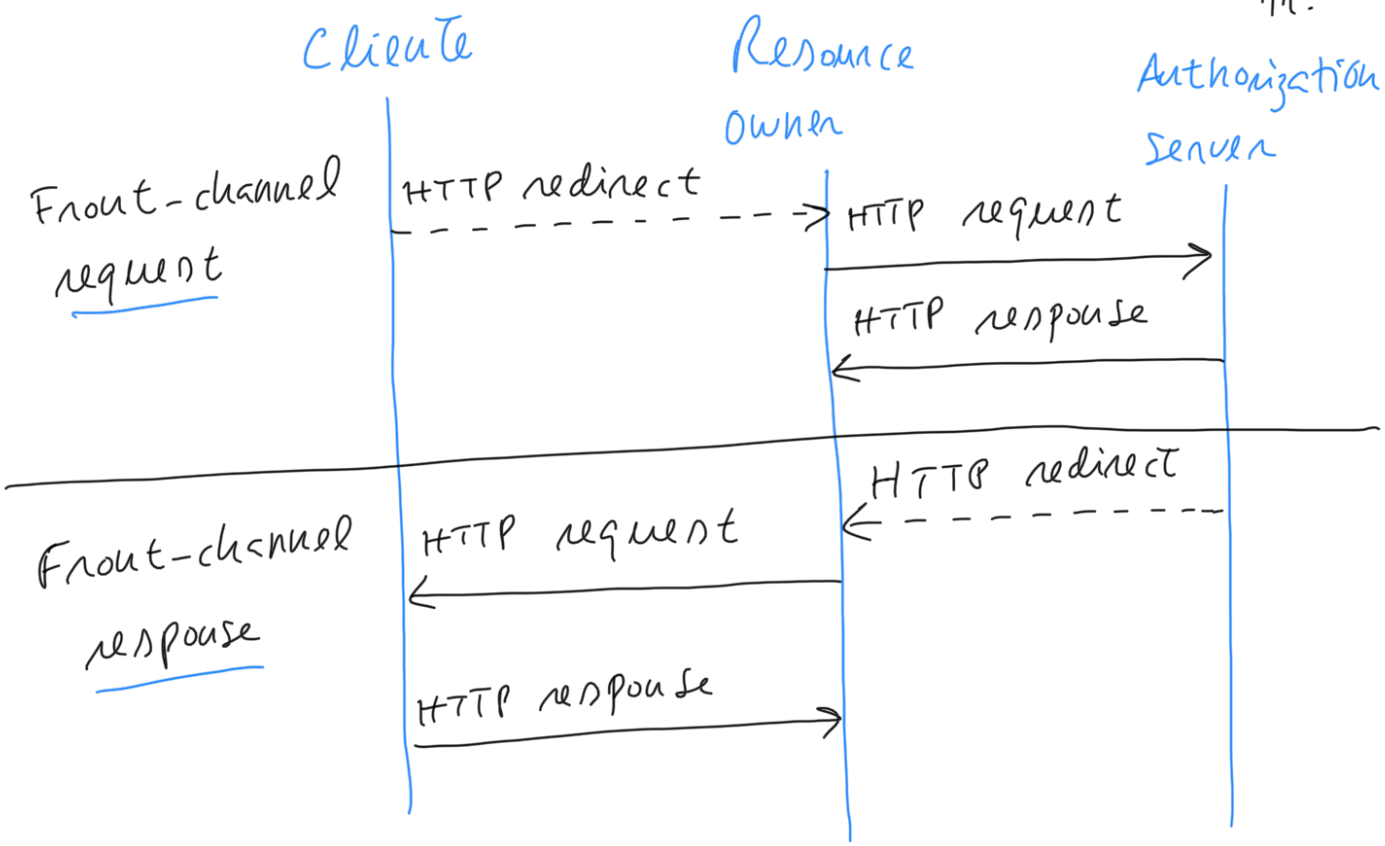


Front-channel - Comunicação

front-channel é um método de usar pedidos HTTP para comunicar indiretamente entre dois sistemas por intermédio duma browser.

ex: Resource Owner autentica-se no AS sem mostrar as credenciais ao cliente.





DEMO: Authorization Code Grant Type

↳ existem outros fluxos, como:

- implicit grant type
- client credentials grant type

Notas:

- Formas de enviar um Bearer token:
  - HTTP Authorization header (preferencial)
  - form-encoded request body parameter
    ↳ usa POST + forms
  - URL-encoded query parameter
    ↳ access token pode ser capturado pois vai no URL request