



DEVOXX
FRANCE
2025



Guillaume Chervet

Principal IA Engineer
AXA France
[@guiChervet](https://twitter.com/guiChervet) 



Pourquoi dès aujourd'hui utiliser **OpenID Connect** côté client pour sécuriser vos fronts ?



DEVOXX
FRANCE 2025

Introduction



1. OIDC Client Side
2. OIDC Server Side
3. DPoP Demonstrating Proof-of-Possession

1. OIDC Client Side
2. OIDC Server Side
3. DPoP Demonstrating Proof-of-Possession

access_token

refresh_token

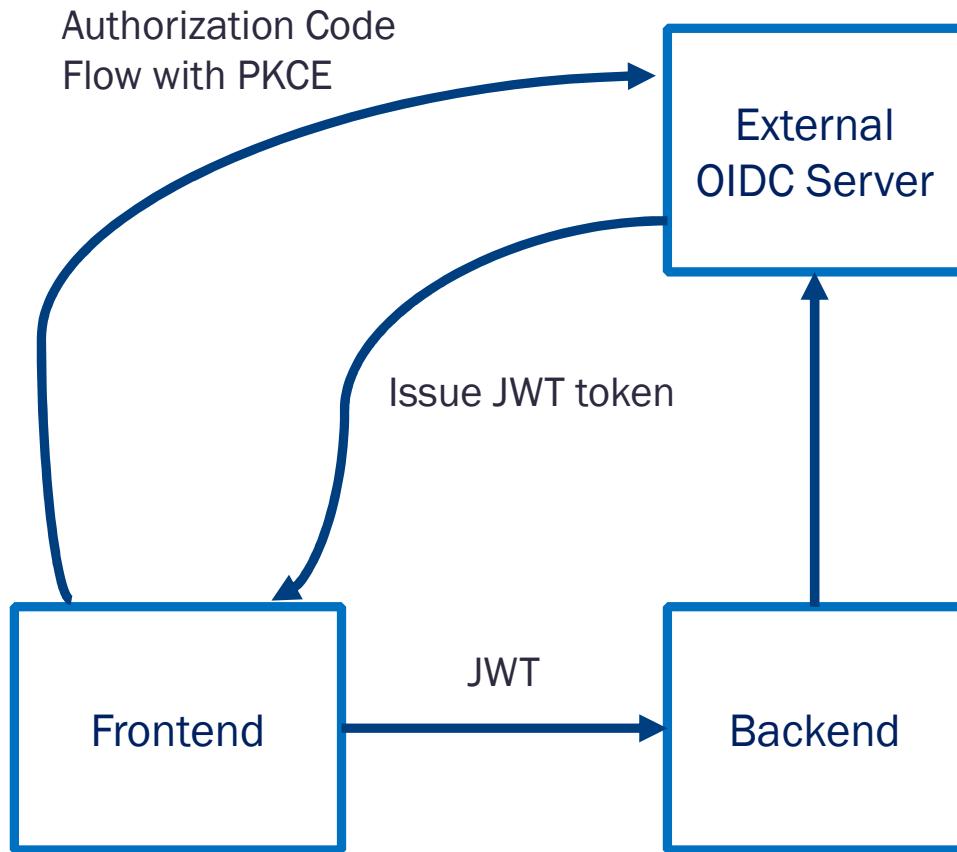
id_token

Authorization Code Flow with Proof Key for Code Exchange (PKCE)

The application is either on the client (Javascript, mobile application) or server side. Allows you to obtain renewal tokens.

Client Credentials Grant, server-to-server authorization

Used by clients to obtain an access token outside of a user's context.



OIDC Client Side



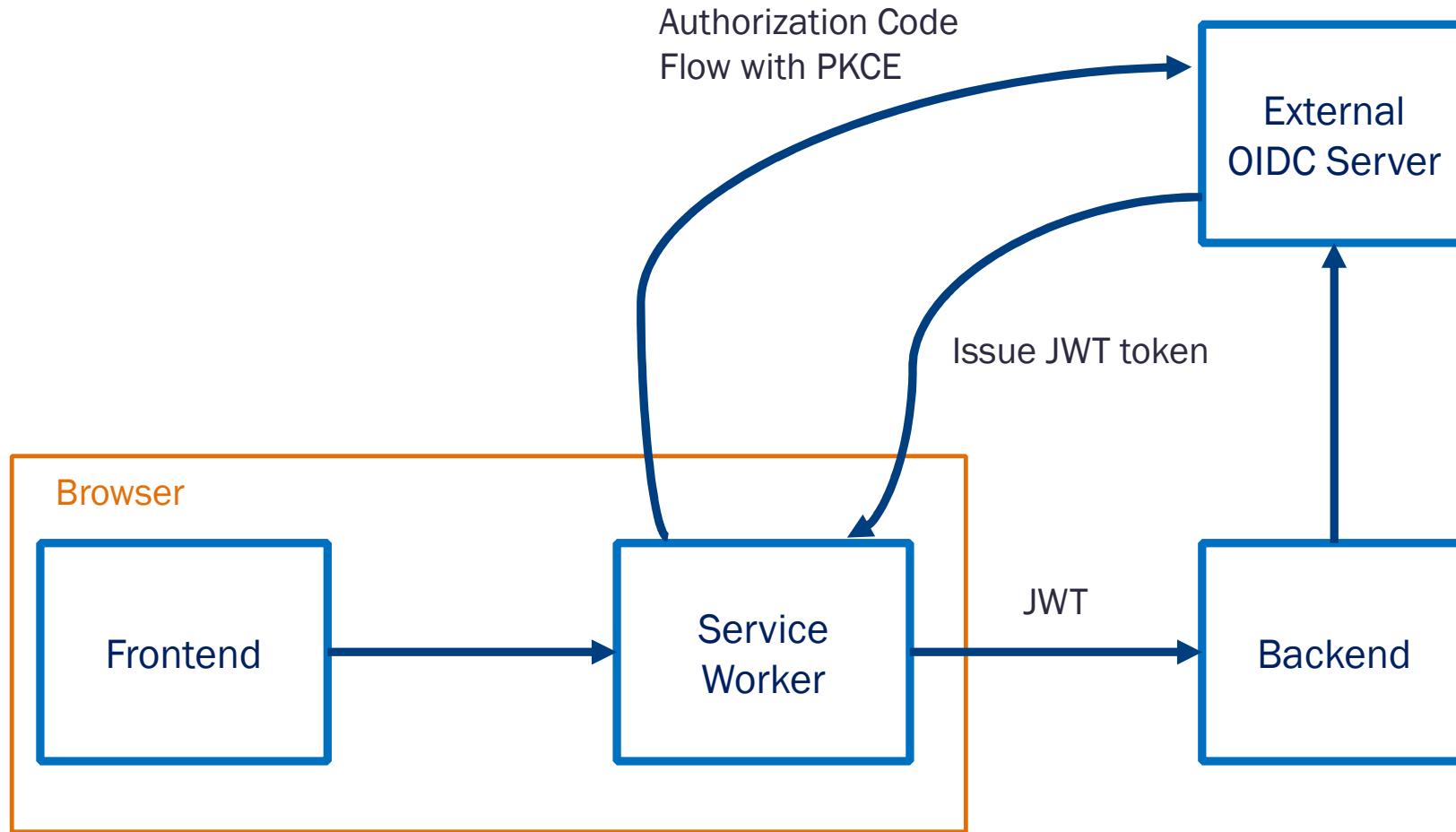
DEVOXX
FRANCE 2025

Guillaume Chervet – Avril 2025

Content-Security-Policies (CSP)

```
add_header Content-Security-Policy "default-src 'self'; script-src 'self' 'unsafe-inline'  
'unsafe-eval'; style-src 'self' 'unsafe-inline'; img-src 'self' data:; font-src 'self';  
connect-src 'self' https://demo.duendesoftware.com; media-src 'self'; object-src  
'none'; frame-src 'self' https://demo.duendesoftware.com; base-uri 'self'; form-action  
'self'; frame-ancestors 'self' https://demo.duendesoftware.com; block-all-mixed-  
content; upgrade-insecure-requests;";  
add_header X-Content-Type-Options nosniff;  
add_header X-Frame-Options SAMEORIGIN;  
add_header X-XSS-Protection "1; mode=block";  
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains;  
preload";  
add_header Referrer-Policy "no-referrer";  
add_header X-Permitted-Cross-Domain-Policies "none";
```

Service Worker

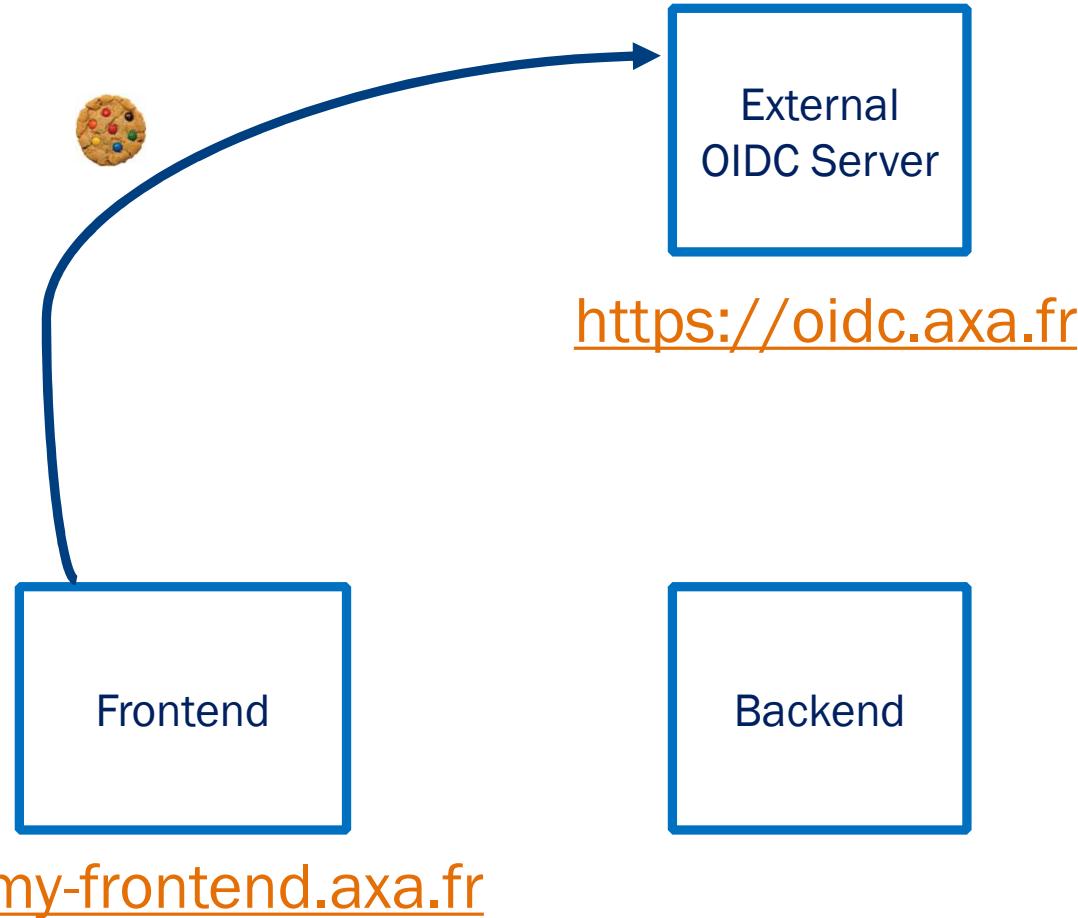


OIDC Client Side

Silent Signin

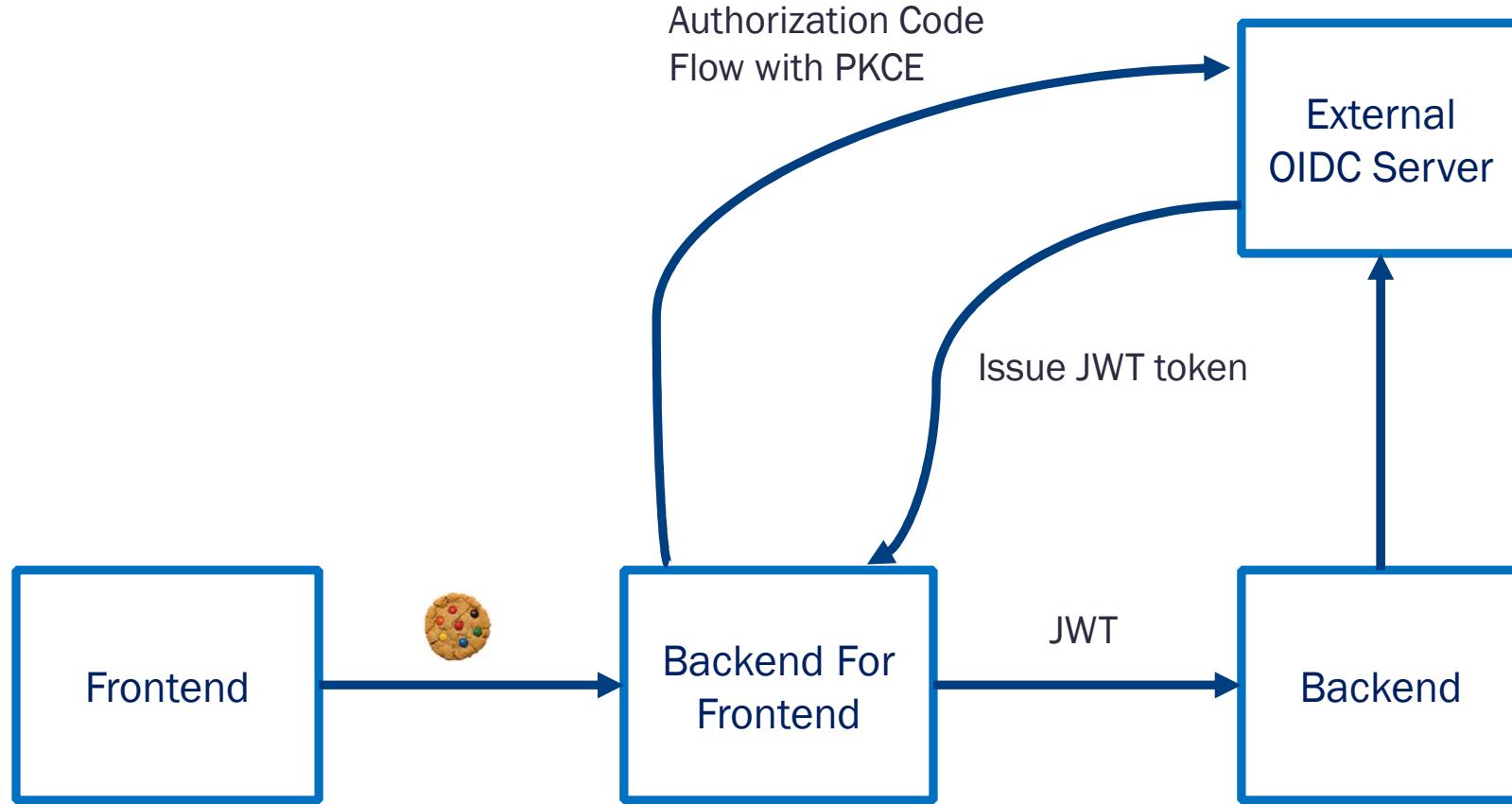
Third party Cookies

```
Set-Cookie:  
MyKey=MyValue;  
Domain=.axa.fr;  
Path=/; Secure;  
HttpOnly;  
SameSite=None
```



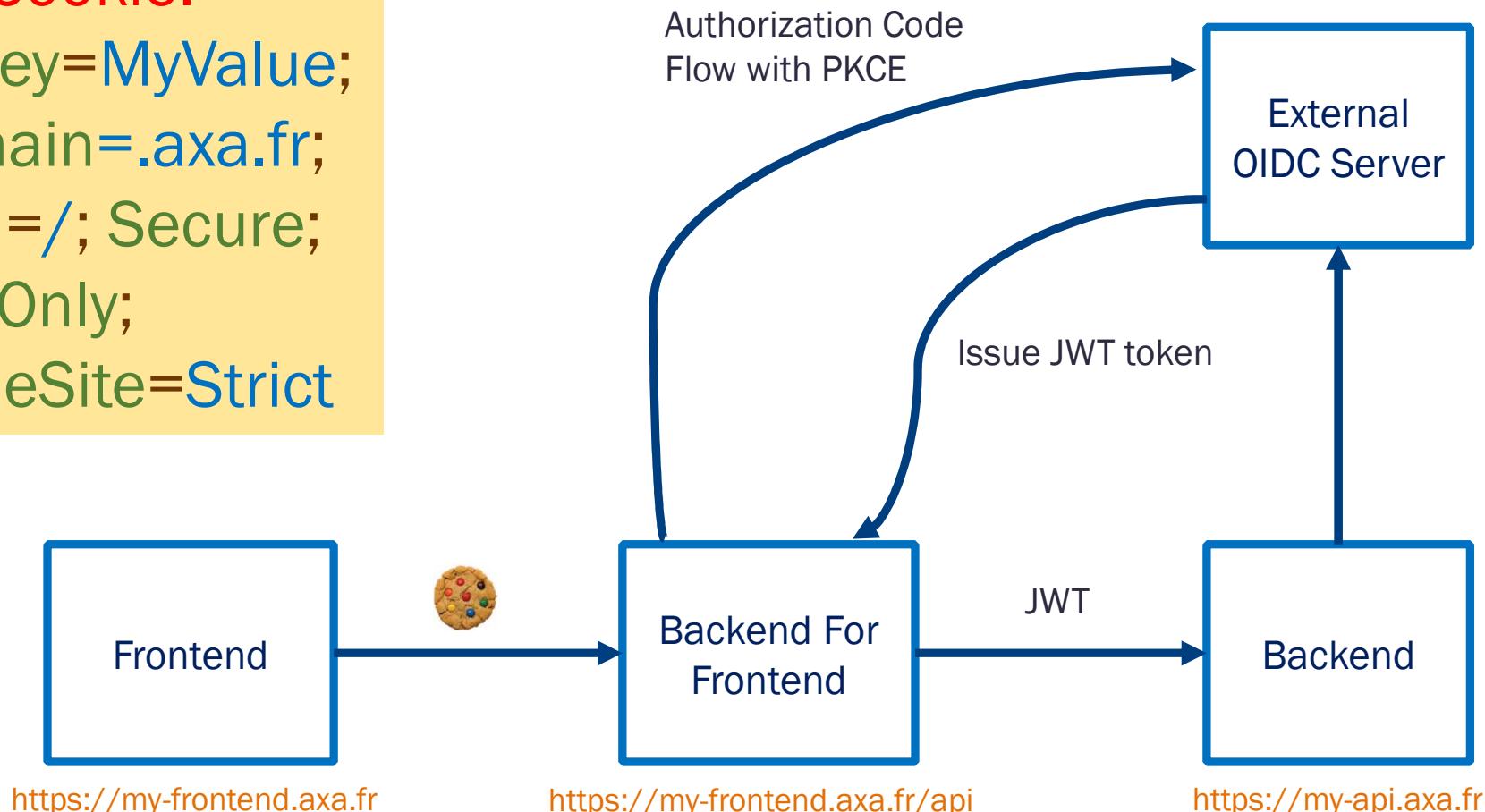
OIDC Client Side

1. OIDC Client Side
2. OIDC Server Side
3. DPOP Demonstrating Proof-of-Possession

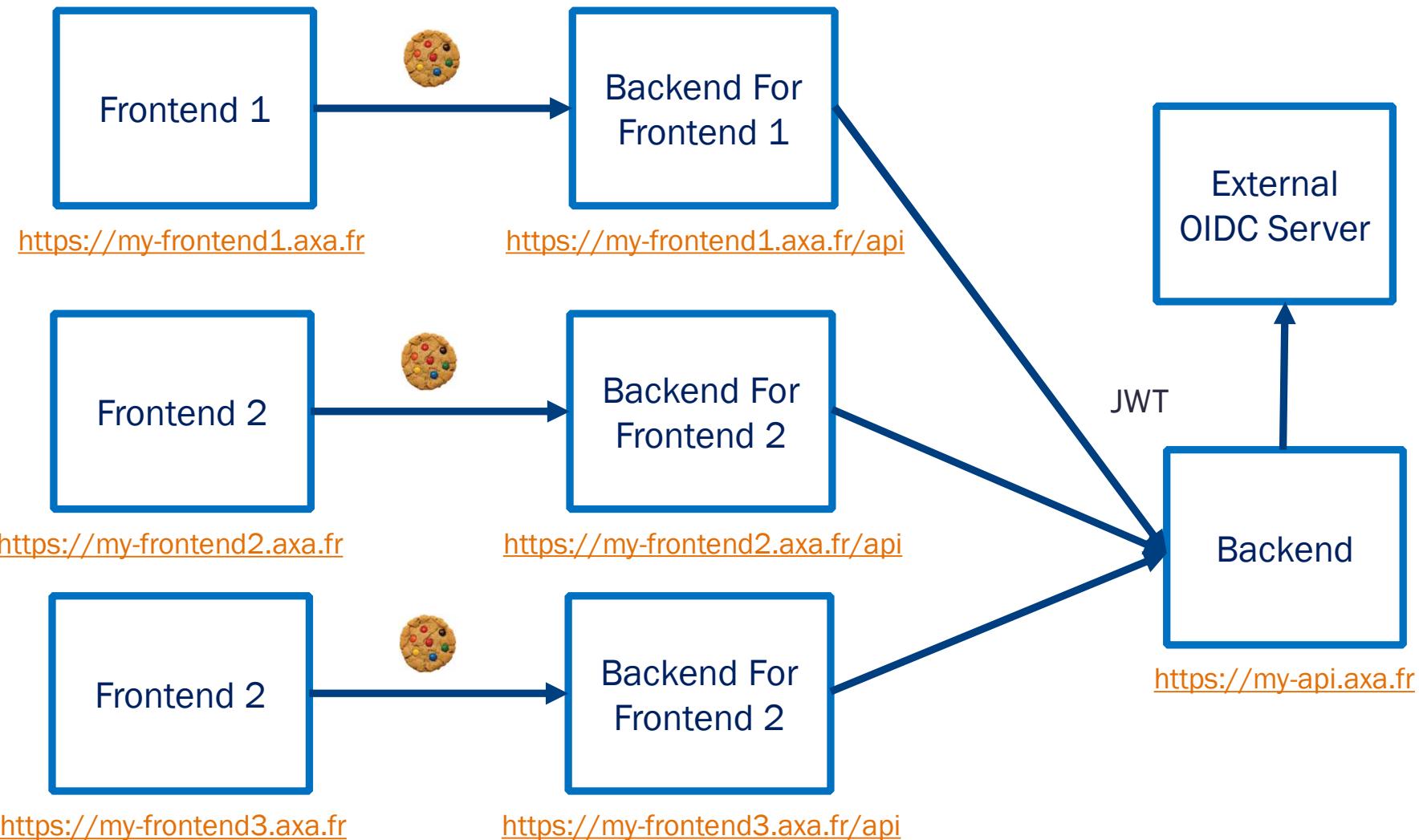


OIDC Server Side

Set-Cookie:
MyKey=MyValue;
Domain=.axa.fr;
Path=/; Secure;
HttpOnly;
SameSite=Strict



OIDC Server Side



OIDC Server Side

1. OIDC Client Side
2. OIDC Server Side
3. DPoP Demonstrating Proof-of-Possession

Conclusion

Basic practices

	Basic practices
OIDC Client Side	<ul style="list-style-type: none">• Activate Content-Security-Policies (CSP) to protect from Cross-site scripting (XSS) attacks• Set up callback route management before any javascript fetch to protect from Cross-site scripting (XSS) attacks
OIDC Server Side	<ul style="list-style-type: none">• Cookie SameSite=Strict; to protect against Cross-site request forgery CSRF• Cookie HttpOnly; to protect to protect from Cross-site scripting (XSS) attacks• Cookie Secure to guaranty HTTPS• Cookie Content : Use a Strong and Up to Date Cipher algorithm



Merci !

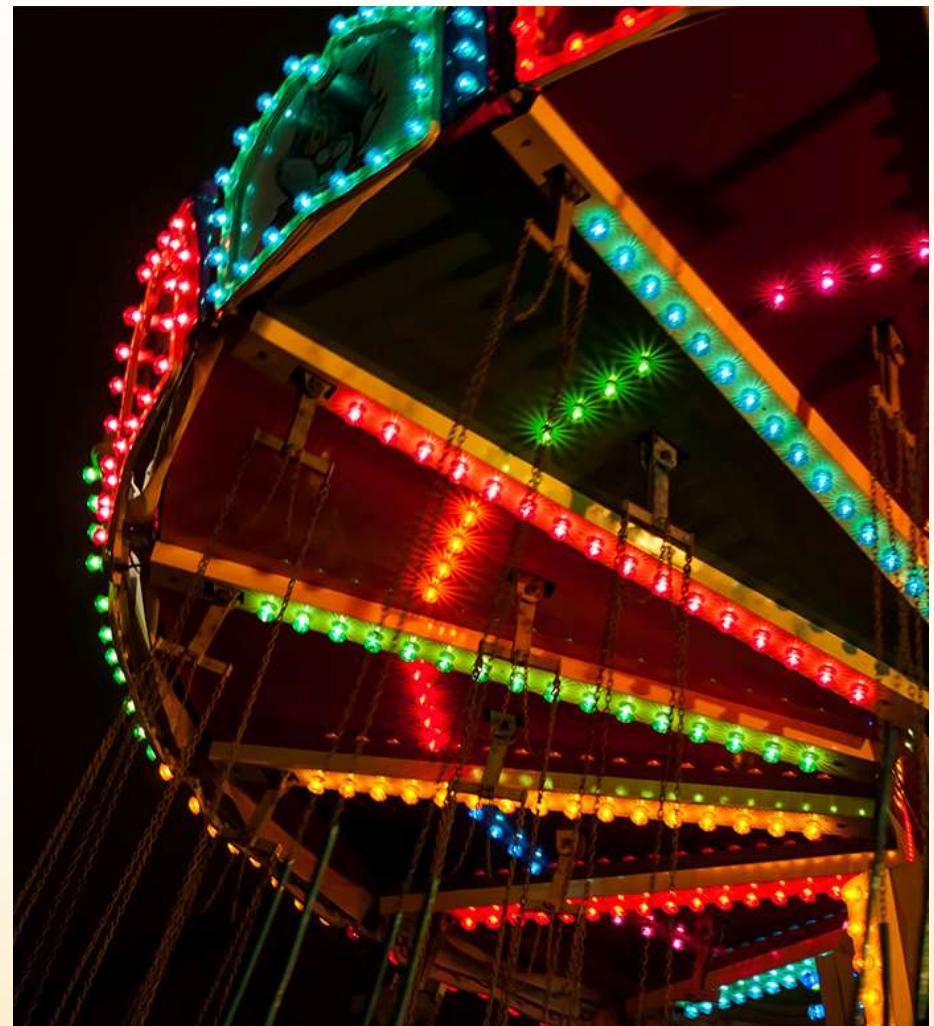
DEVOXX
FRANCE

Café

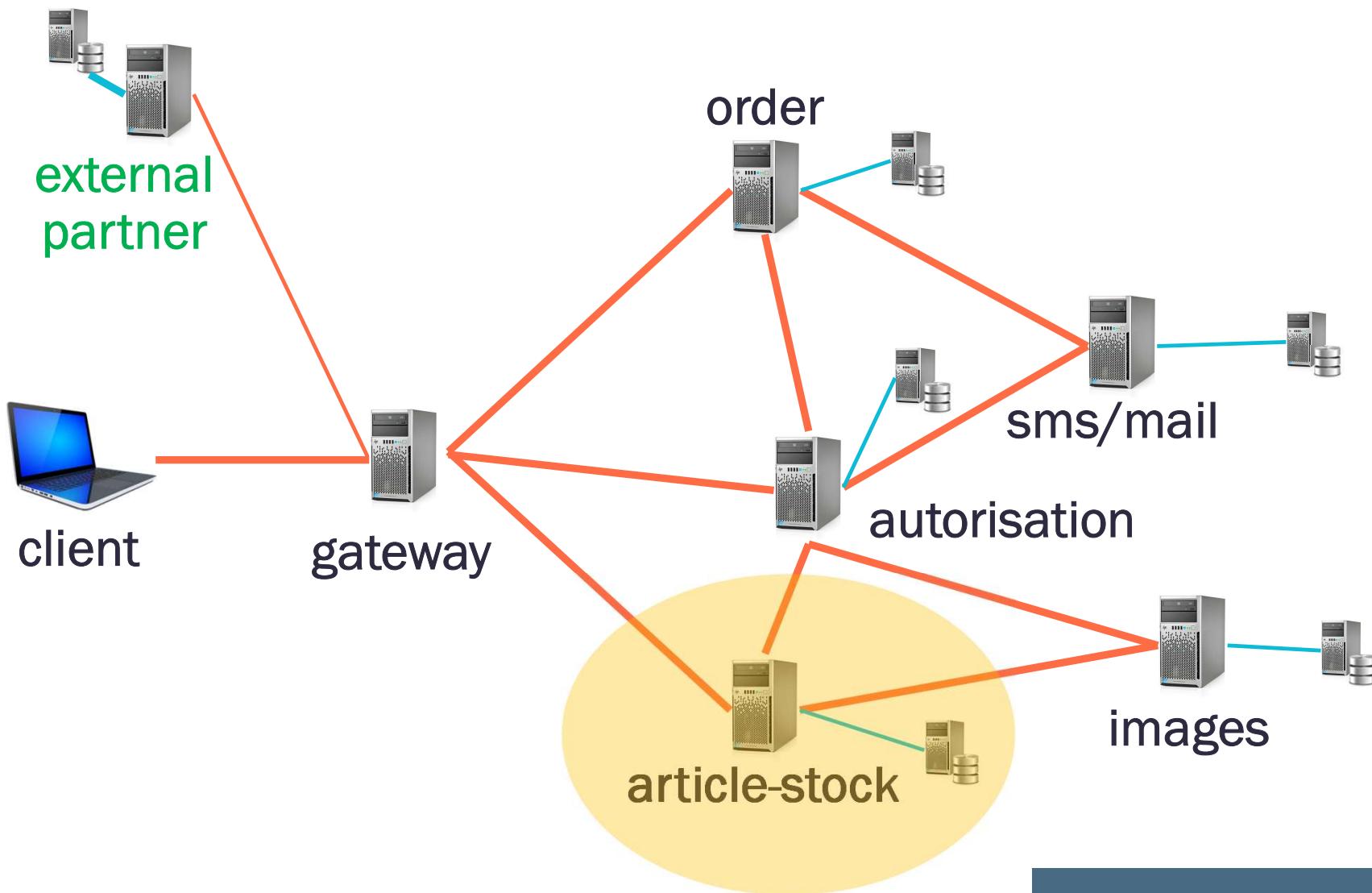
Merci 😊



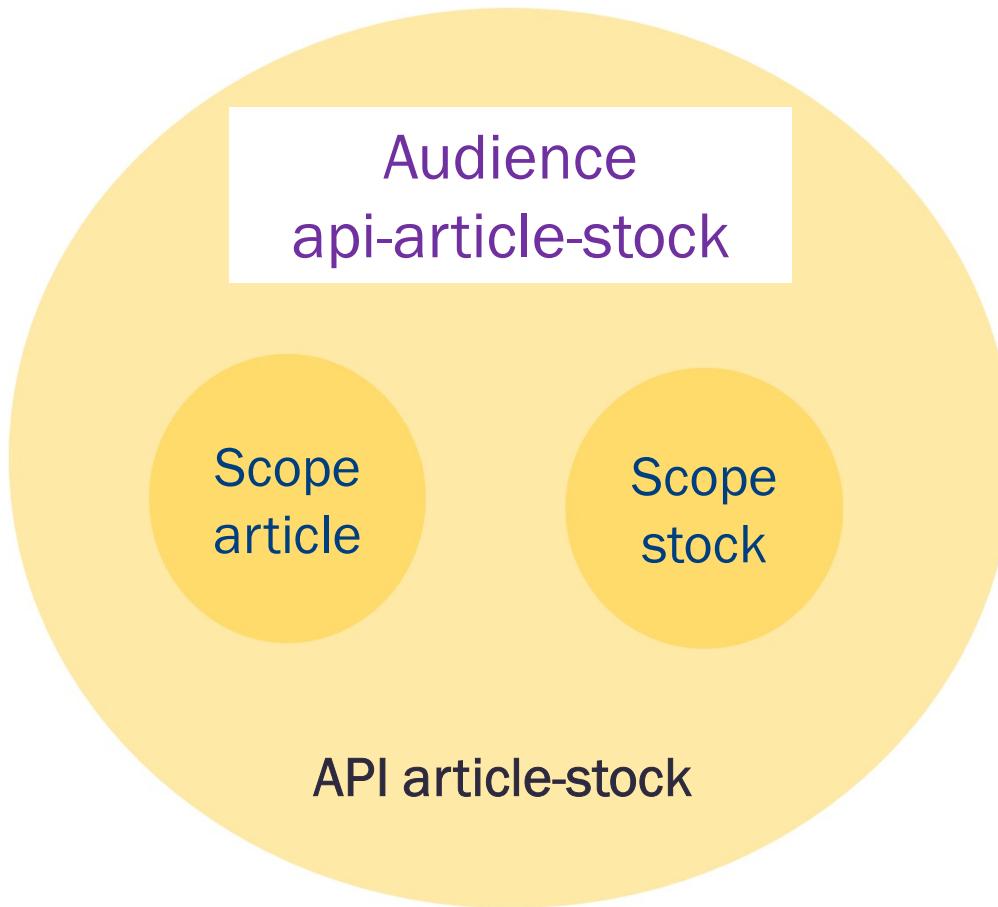
OpenFeedBack



OIDC configure a clientID Keycloak



OIDC Client Side



Third party Cookies

Set-Cookie: sessionId=abc123; **SameSite=None**; HttpOnly; Secure

Set-Cookie: sessionId=abc123; **SameSite=Lax**; HttpOnly; Secure

Set-Cookie: sessionId=abc123; **SameSite=Secure**; HttpOnly; Secure

Hash Function

`SHA256("Guillaume")`



`"d53afea4f412db7d0d3d45fdc
7fdffbedf267923915587d47e9
0d2d7bd21aec3"`

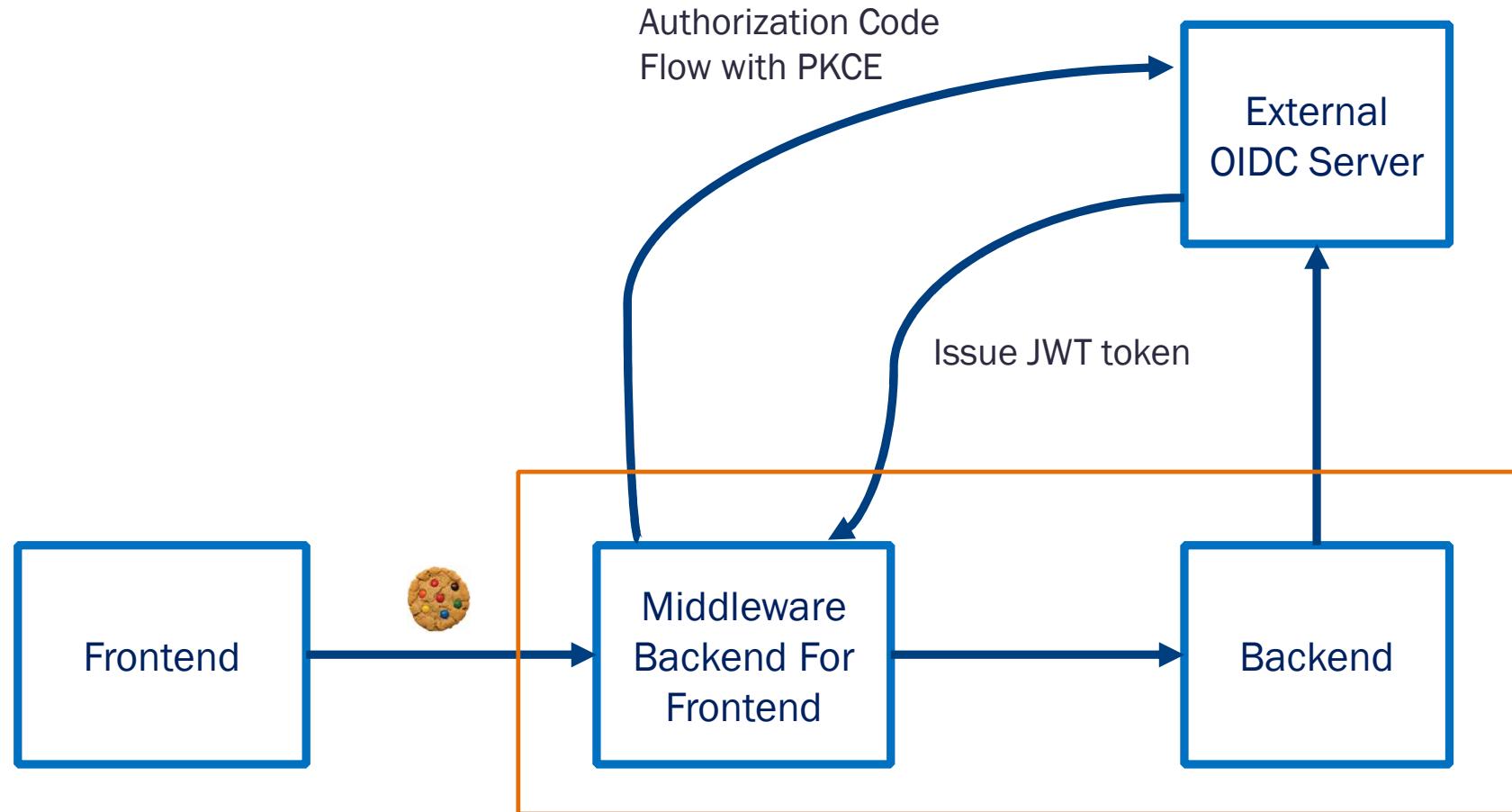
`SHA256("Amaury")`



`"271f83ef2b92c63dfd1bfd9be
b4294845f33210356dc40f646
154195aa126f01"`

Cross-site request forgery (CSRF or XSRF)

```
<body onload="document.forms[0].submit()">
  <form action="http://insurance.fr/transfer" method="POST">
    <input type="hidden" name="account" value="Vladimir"/>
    <input type="hidden" name="amount" value="50000"/>
    <input type="submit" value="Transfer"/>
  </form>
</body>
```



<https://my-frontend.axa.fr>

<https://my-frontend.axa.fr/api>

OIDC Server Side

Guillaume Chervet – novembre 2024

Cross-site request forgery (CSRF or XSRF)

```
<body onload="document.forms[0].submit()">
  <form action="http://insurance.fr/transfer" method="POST">
    <input type="hidden" name="csrf" value="NjJudPksuzNO"/>
    <input type="hidden" name="account" value="Vladimir"/>
    <input type="hidden" name="amount" value="50000"/>
    <input type="submit" value="Transfer"/>
  </form>
</body>
```



asymmetric cryptography



digital signature algorithm



Cross-site request forgery (CSRF or XSRF)

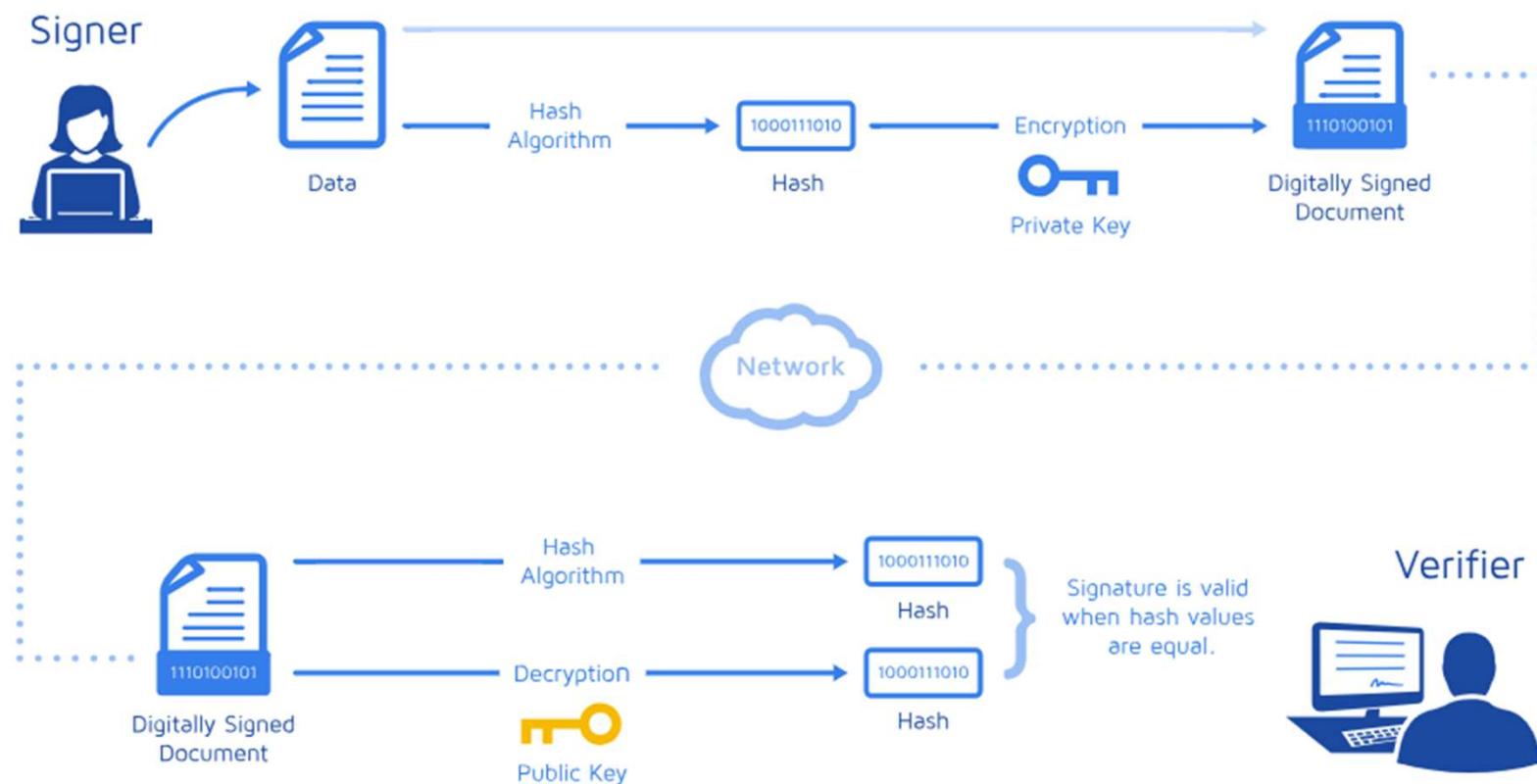
```
<body onload="document.forms[0].submit()">
  <form action="http://insurance.fr/transfer" method="POST">
    <input type="hidden" name="csrf" value="NjJudPksuzNO"/>
    <input type="hidden" name="account" value="Vladimir"/>
    <input type="hidden" name="amount" value="50000"/>
    <input type="submit" value="Transfer"/>
  </form>
</body>
```

Pourquoi dès aujourd'hui utiliser OpenID Connect côté client pour sécuriser vos fronts ?

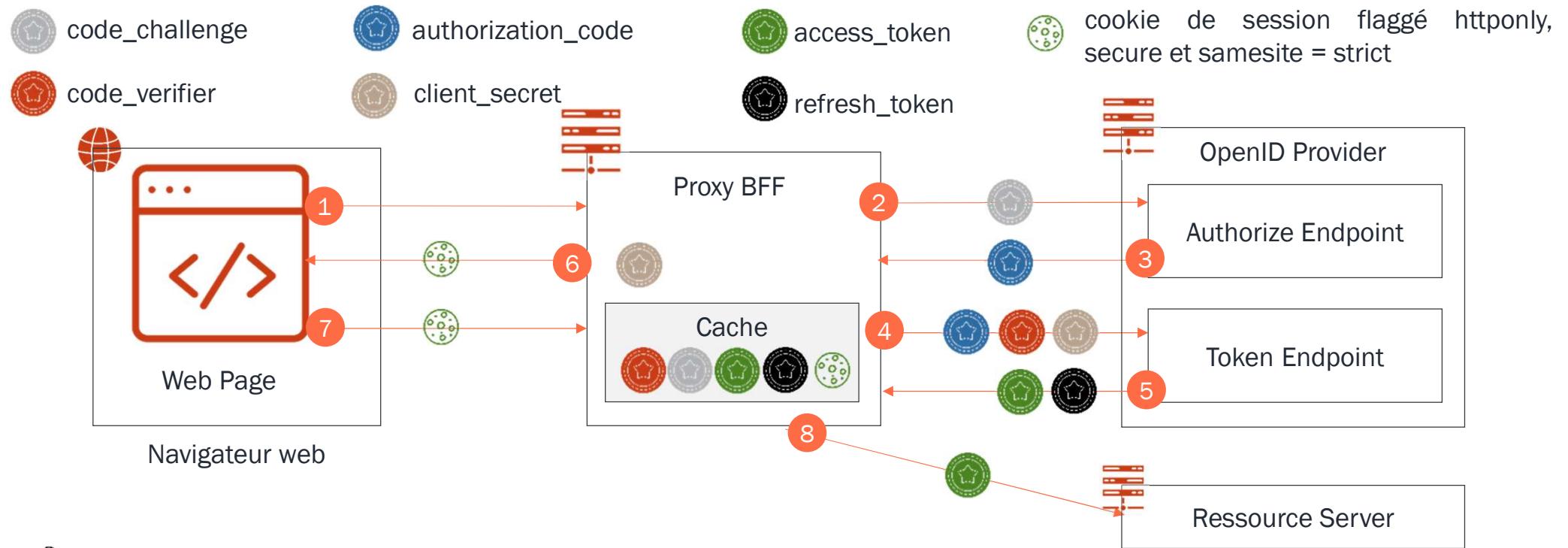
Vous créez à chaque fois un BFF (Back End For Front End) pour gérer l'authentification côté front ? Cela peut réduire votre « Time To Market », vous coûter plus cher que nécessaire et n'est aujourd'hui pas forcément plus sécurisé. Dans cette présentation qui sera assez technique, nous allons vous présenter le protocole OpenID Connect. L'architecture de l'OIDC côté client ainsi que son concurrent et ami côté serveur. Nous expliquerons avec beaucoup de démos comment fonctionnent les échanges de clés JWT. Nous décrirons les avantages et inconvénients de chaque mode. Ensuite, nous expliquerons toujours avec des démos pourquoi @axa-fr/oidc-client le mode avec l'utilisation du ServiceWorker est super sécurisé.

Finalement, nous terminerons par expliquer le concept de « Demonstrating Proof of Possession » (DPoP) qui est une "killer feature" et qui rend vos "tokens" inutilisables hors du contexte navigateur, tout cela grâce à l'API WebCrypto. Préparez vos cerveaux, ce sera pédagogique et progressif, mais restera très technique.

digital signature algorithm



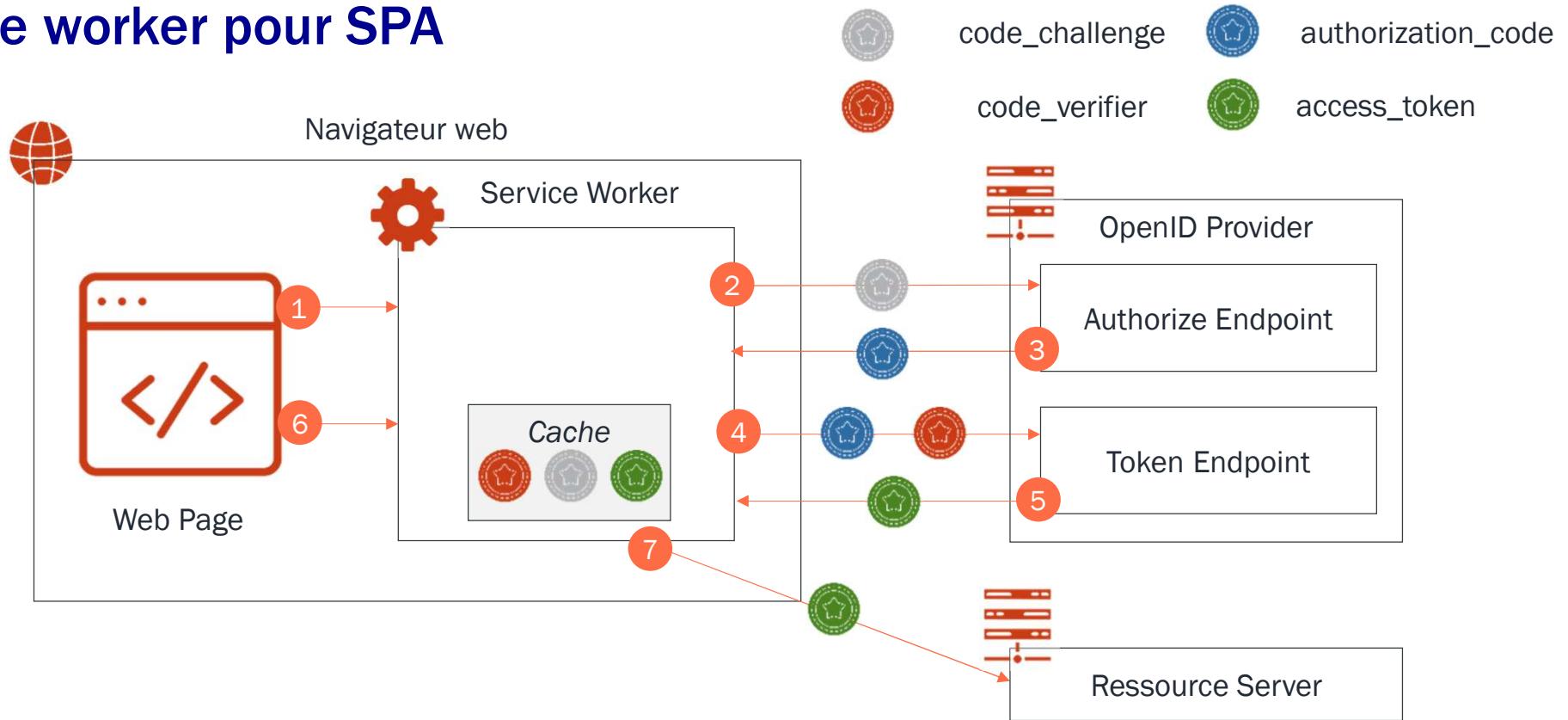
Proxy Backend for Frontend (BFF) pour SPA



Le rôle du proxy BFF est de mettre en cache (côté serveur) les jetons OIDC/Oauth2 et d'assurer la conservation des sessions côté navigateur via des cookies. Avec une architecture BFF, la SPA devient un client confidentiel. La sécurité des cookies doit par ailleurs être assurée par les tags `secure`, `httponly` et `samesite = strict` et le proxy BFF doit prévoir un mécanisme anti-CSRF



Service worker pour SPA



Le rôle du service worker est de conserver les jetons OIDC/Oauth2 dans un environnement sandbox du navigateur. Il doit initier la requête OIDC/Oauth2 en générant les codes PKCE, puis intercepter l'authorization code passé dans l'url et récupérer ensuite les jetons. Aucun des secrets ne doit être ni renvoyé ni rendu visible à la Web Page. Toute requête initiée depuis la Web Page doit être interceptée et bloquée par le service worker

Stockage et transport des jetons côté navigateur web : que choisir ?

Solution	Avantage / inconvénient
Cookie	<p>Les avantages :</p> <ul style="list-style-type: none">- Permet une protection XSS via le tag « httponly »- Permet un premier niveau de protection contre les CSRF via l'attribut samesite = strict- Permet de garantir un transport chiffré via le tag secure <p>Les inconvénients :</p> <ul style="list-style-type: none">- Les jetons sont stockés sur la machine de l'utilisateur et persistent à la fermeture du browser- L'application doit prévoir l'implémentation d'un mécanisme anti-CSRF complémentaire au samesite = strict- Les jetons peuvent être interceptés par une extension web malveillante
Service Worker	<p>Les avantages :</p> <ul style="list-style-type: none">- Les jetons sont seulement traités en mémoire- Nativement robuste contre les attaques CSRF (ne nécessite pas de prévoir un mécanisme supplémentaire côté serveur) et XSS- Garantit nativement le transport chiffré <p>Les inconvénients</p> <ul style="list-style-type: none">- N'est pas supporté par tous les browsers (ex : Internet Explorer)- Les jetons peuvent être interceptés par une extension web malveillante
Web Storage	<p>Les avantages :</p> <ul style="list-style-type: none">- Nativement robuste contre les attaques CSRF (ne nécessite pas de prévoir un mécanisme supplémentaire côté serveur) <p>Les inconvénients</p> <ul style="list-style-type: none">- Les jetons sont stockés sur la machine de l'utilisateur et persistent à la fermeture du browser dans le cas d'utilisation du local storage- Les jetons sont accessibles au code JS, donc peuvent être récupérés par du code tiers (ex : script Google Analytics) ou suite à une attaque XSS- Les jetons peuvent être interceptés par une extension web malveillante



Le service worker apporte nativement plus de bénéfices sécurité que le cookie. Il est donc à privilégier.
Le Web Storage, quant à lui, n'est plus autorisé



Man in the middle



Guillaume



Samuel

<https://fr.wikipedia.org/wiki/Coronavirus>

https://fr.wikipedia.org/wiki/Attaque_de_l%27homme_du_milieu

Classification : Interne





1. OpenId Connect
2. OIDC Client Side
3. OIDC Server Side
4. DPOP

Summary



Introduction	03
Part .1	05
Part. 2	06
Part. 3	08
Conclusion	10



Open ID Connect

Standardisation des informations utilisateurs

Une API (User Info endpoint)

Utilisation du scope ID Token

Standardisation de l'authentification

Gestion de la session SSO (ex: le Single Logout)

Système de découverte du serveur OpenID afin de permettre aux clients de s'enregistrer par eux-mêmes

OAuth 2 définit 4 rôles distincts

Resource Owner

Un humain ou une machine

Resource Server

Héberge les données dont l'accès est protégé

Client Application

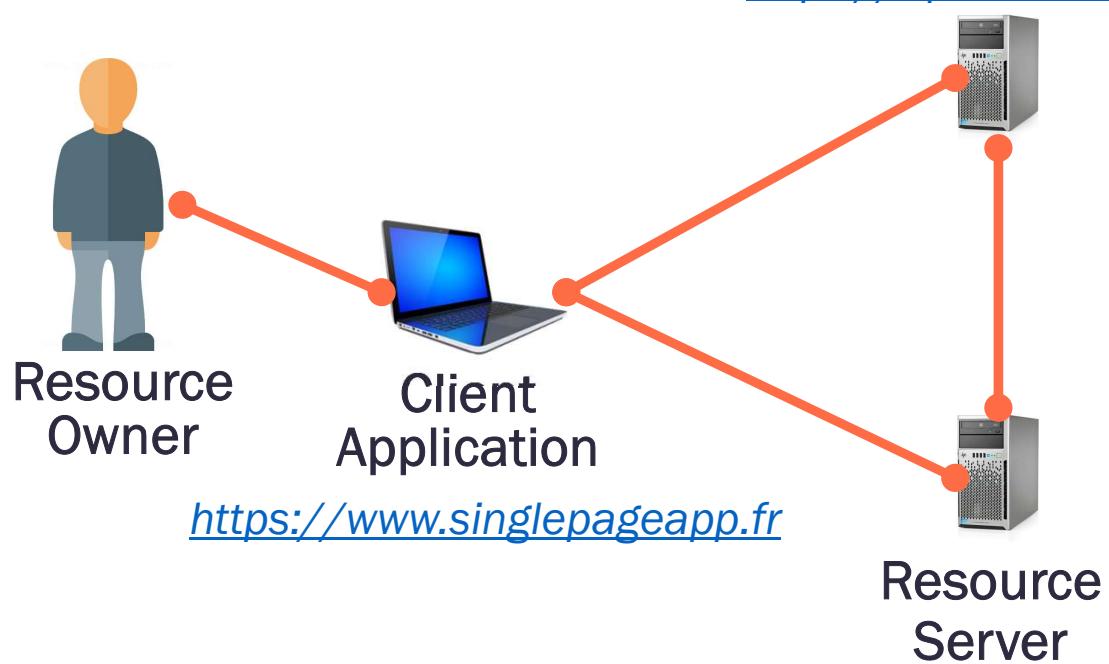
Application demandant des données au serveur de ressources.

Authorization Server

Délivre des jetons d'accès (tokens en anglais) au client.

Authorization Server

<https://api.authorization.fr>



OpenId Connect

OpenID Connect : endpoints

authorization : pour authentifier un utilisateur

token : pour demander un token (access / refresh / ID)

user info : pour récupérer des informations sur l'utilisateur (son identité, ses droits)

revocation : pour supprimer un token (access / refresh)

introspection: pour valider un token (access / refresh)

Open ID Connect

Cela permet de changer de fournisseur
sans changer votre code

Oauth 2 et les tokens

Access Token, le token d'accès

Permet au serveur de ressources d'autoriser la mise à disposition des données d'un utilisateur.
Ce token est envoyé par le client (l'application) dans la requête vers le serveur de ressources.
Il a une durée de vie limitée qui est définie par le serveur d'autorisation: Par exemple 20 minutes.

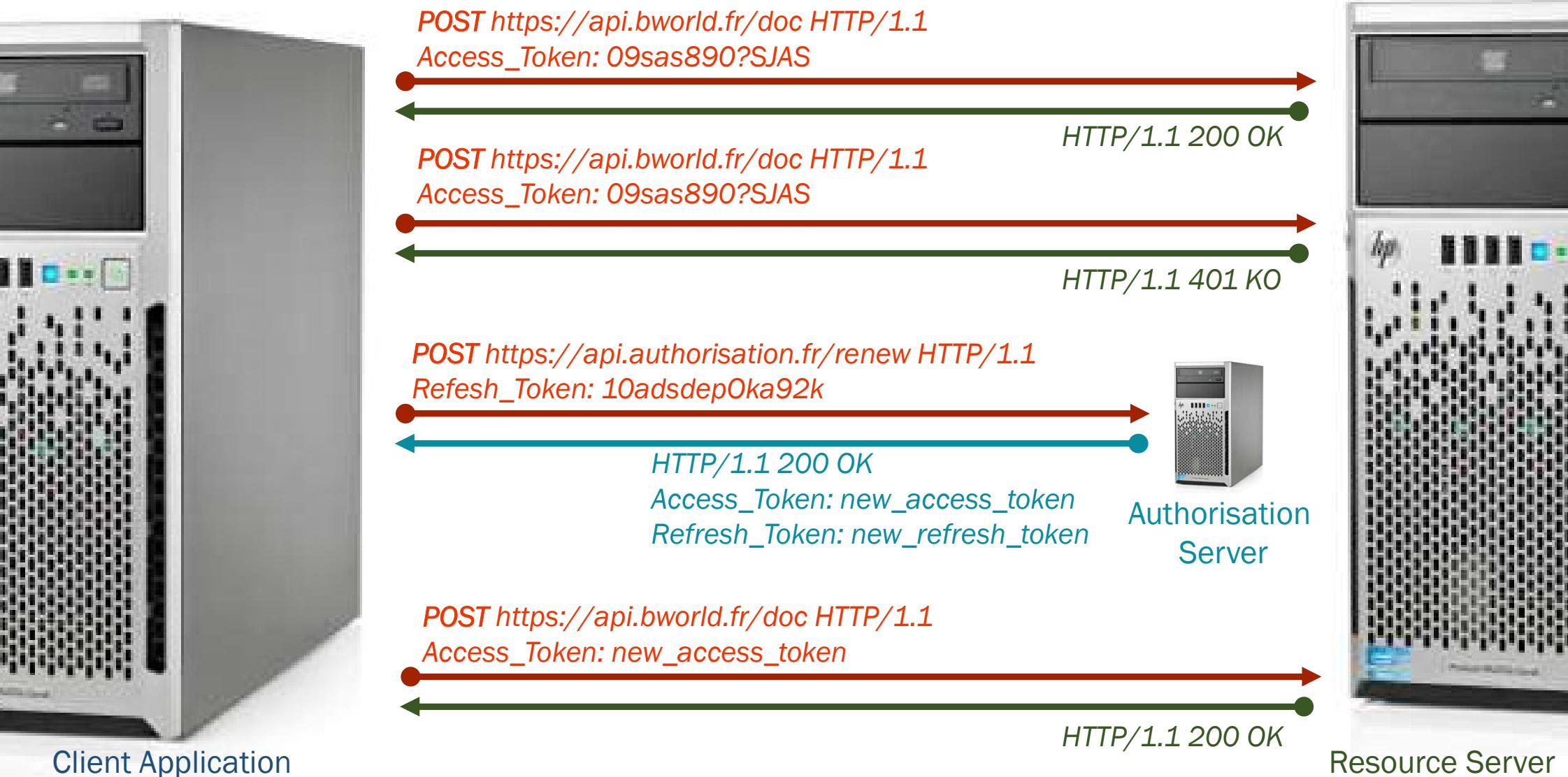
Refresh Token, le token de renouvellement

Ce token est délivré au même moment que le token d'accès.
Il sert à renouveler l'access token quand celui-ci est expiré.
Il a une durée de vie limitée, mais plus longue que l'access token: Par exemple 1 semaine.

OpenId Connect

<http://www.bubblecode.net/fr/2016/01/22/comprendre-oauth2/>

Utilisation et renouvellement des tokens



```
{  
    "sub": "S607718",  
    "cei": "e8cb4bff",  
    "iss": "https://openid.bworld.fr",  
    "client_id": "7fc411cb",  
    "aud": [  
        "https://openid.bworld.fr",  
        "api-bworld"  
    ],  
    "acr": "1",  
    "rlm": "OpenId from api bworld",  
    "scope": "openid profile email api-bworld",  
    "custome_info": "youhou",  
    "exp": 1585672871,  
    "member_of": [  
        "CN=ADV_ADMIN,CN=IAM_ADV,OU=applis,O=bworld,DC=REWACAD,DC=fr"  
    ],  
    "iat": 1585669271,  
    "jti": "8e20b896-252e-40a8-80a2-78b2a5237177"  
}
```

Classification :

access_token payload

HTTPS

OAuth2 impose l'utilisation de HTTPS pour les échanges.

<http://www.bubblecode.net/fr/2016/01/22/comprendre-oauth2/>

Classification : Interne

Scope

Sert à limiter les droits d'accès.

Le serveur d'autorisation définit la liste des scopes disponibles.

Le client doit envoyer le ou les scopes qu'ils souhaitent utiliser lors de la demande d'autorisation.

Exemple : openid email profile account-read account-payment

Audience

Spécifier le ou les API que vous cibler.

Le serveur d'autorisation définit la liste des audiences disponibles.

Le client doit envoyer le ou les scopes qu'ils souhaitent utiliser lors de la demande d'autorisation.

Exemple : api-bank api-payment



Sécurité

Votre serveur de « resource » doit contrôler les scopes nécessaire à chaque actions/routes

Votre serveur de « resource » doit contrôler que son audience est bien positionné

Oauth 2.0

Un client ne peut utiliser le protocole OAuth sans être connu du serveur d'autorisation. Pour cela il lui faut donc s'enregistrer auprès du serveur d'autorisation. Pour cela il doit fournir un ensemble de données :

Nom de l'application

Url du site

Url de retour

Etc.

En échange, le serveur fournira un identifiant et un code secret (**client id** et si nécessaire une **client secret**) sous forme de chaîne de caractères, qui permettra au client de s'identifier.

Types d'autorisation

Authorization Code Grant, l'autorisation via un code

Le client est un serveur web. Permet d'obtenir un token d'accès de longue durée qui pourra être renouvelé via un token de renouvellement.



Implicit Grant, l'autorisation implicite

L'application se trouve côté client (Javascript, application mobile). Il ne permet pas d'obtenir de token de renouvellement.



Authorization Code Flow with Proof Key for Code Exchange (PKCE)

L'application se trouve côté client (Javascript, application mobile) ou serveur. Permet d'obtenir de token de renouvellement.

Client Credentials Grant, l'autorisation serveur à serveur

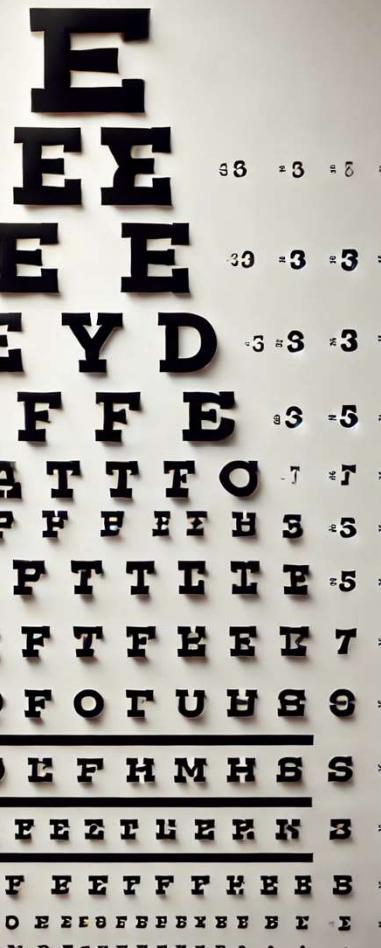
Utilisé par les clients pour obtenir un jeton d'accès en dehors du contexte d'un utilisateur.

Resource Owner Password Credentials Grant, l'autorisation via mot de passe



EYE CHART

EYE CHART



Identification

Authentification

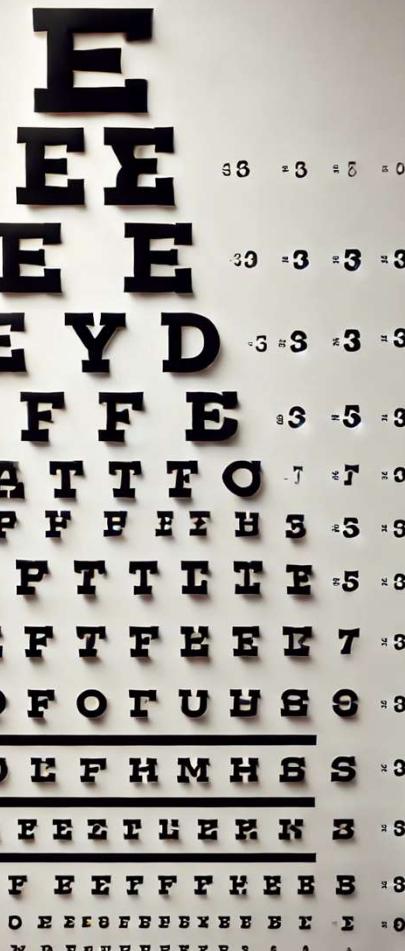
Authorisation

Guillaume Chervet – novembre 2024

Introduction

EYE CHART

EYE CHART



Identification

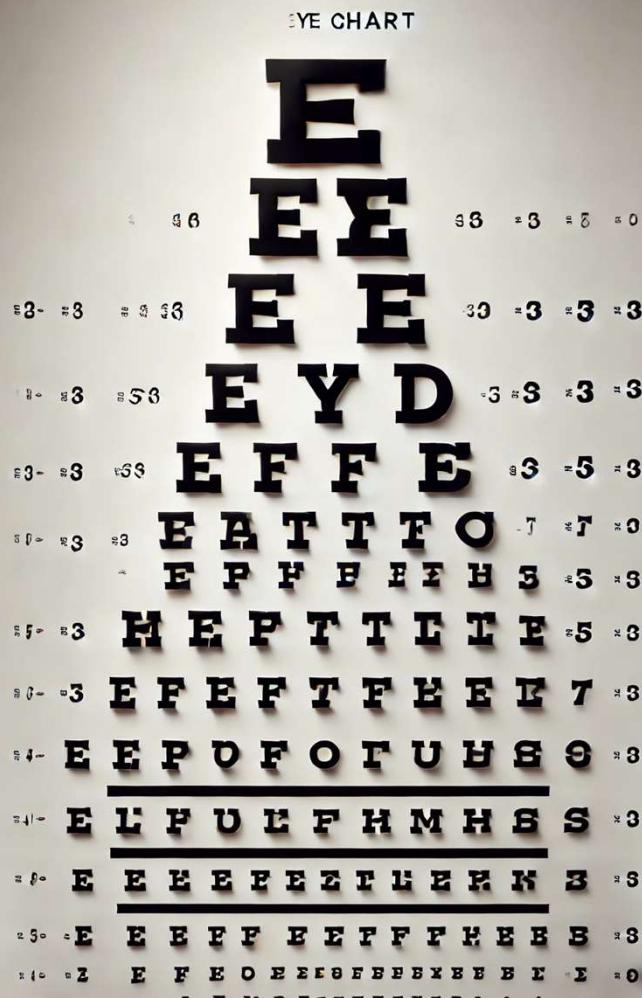
Authentification

Authentification

Guillaume Chervet – novembre 2024

Introduction

EYE CHART



Identification



Authentification

Authorisation

Identification Who are you ?

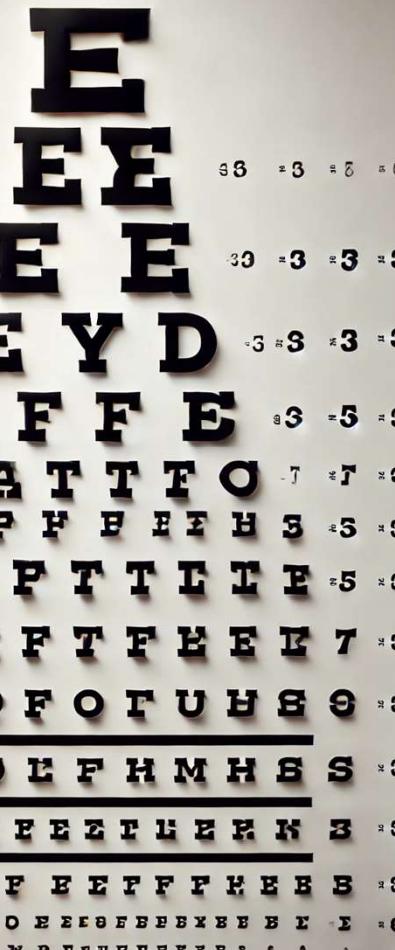
Example: Login
Who can be autenticated ?
One human, one machine

Guillaume Chervet – novembre 2024

Introduction

EYE CHART

EYE CHART



Identification

Authentification

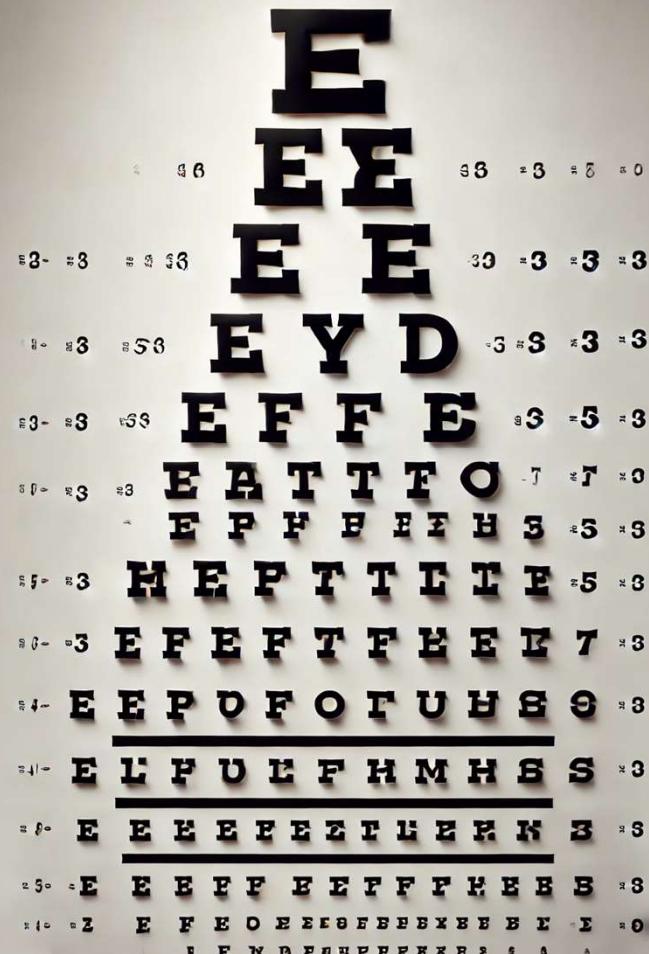
Authorisation

Guillaume Chervet – novembre 2024

Introduction

EYE CHART

EYE CHART



Identification

Authentification



Autentification

Authentification

Are you really that
person/machine ?

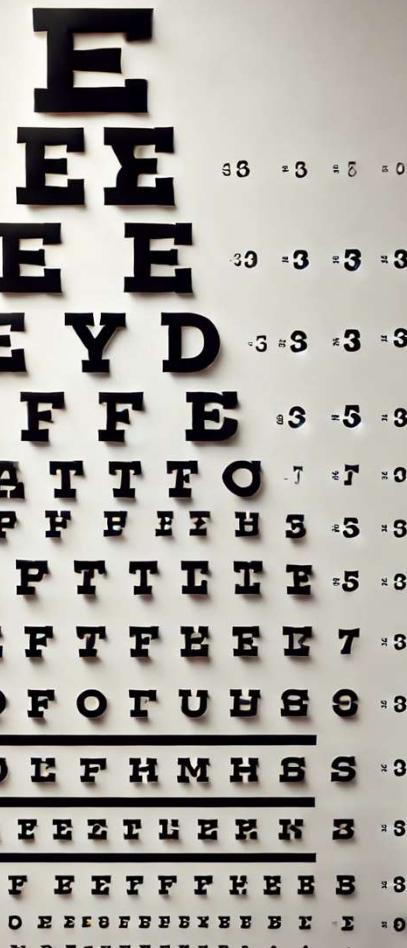
Example : Password

Guillaume Chervet – novembre 2024

Introduction

EYE CHART

EYE CHART



Identification

Authentication

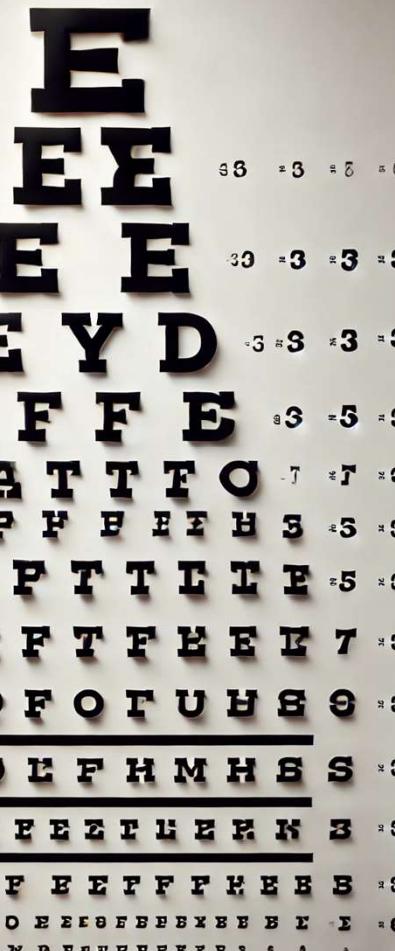
Authorization

Introduction

Guillaume Chervet – novembre 2024

EYE CHART

EYE CHART



Identification

Authentification



Authorization

Authorization

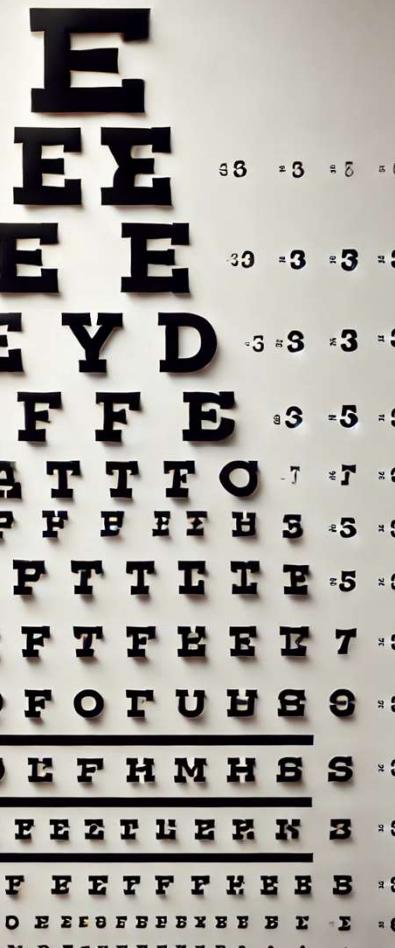
Does this
personne/machine have
the right to access this
resource ?

Guillaume Chervet – novembre 2024

Introduction

EYE CHART

EYE CHART



Identification

Authentication

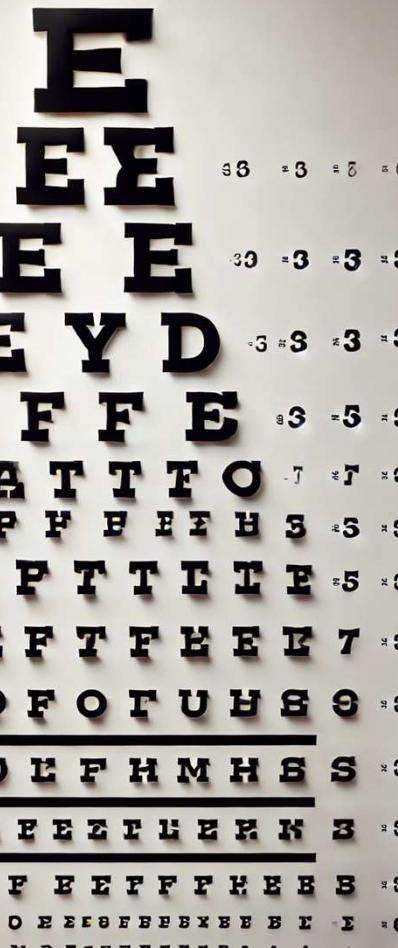
Authorization

Introduction

Guillaume Chervet – novembre 2024

EYE CHART

EYE CHART



Identification

Authentification

Authorisation



HTTP 401

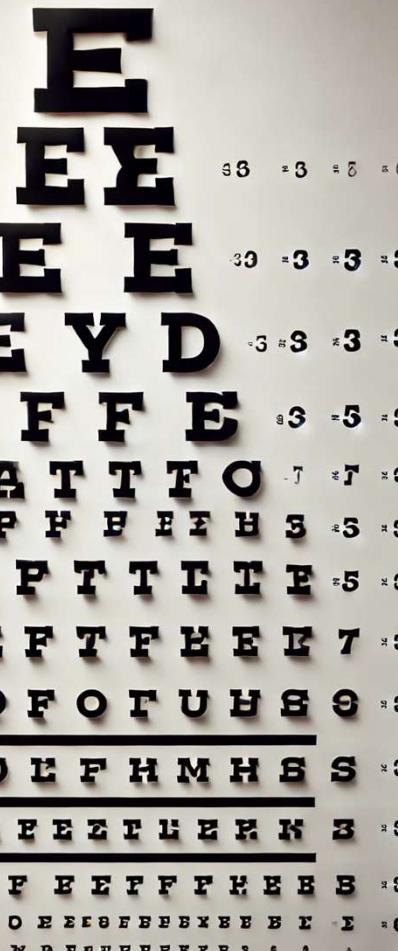
I'm not authenticated and trying to access a private resource

Guillaume Chervet – novembre 2024

Introduction

EYE CHART

EYE CHART



Identification

Authentication

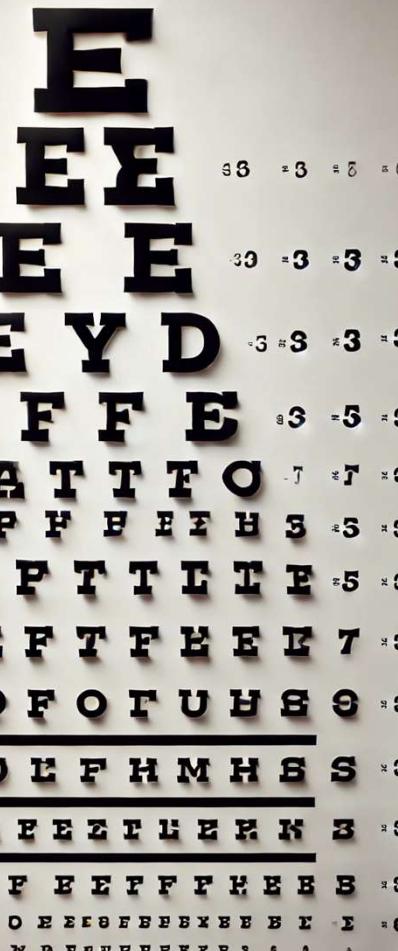
Authorization

Introduction

Guillaume Chervet – novembre 2024

EYE CHART

EYE CHART



Identification

Authentification

Authorisation



HTTP 403

I am authenticated but I
do not have the right to
access the resource

1. OpenId Connect
2. OIDC Client Side
3. OIDC Server Side
4. DPOP

Focus OAuth 2.0

Guillaume Chervet – novembre 2024

OpenId Connect



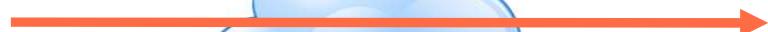
Why OAuth ?

Domains :

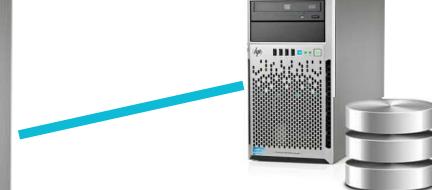
- Commands
- Authentification/authorisation
- Send SMS/mail
- Stock/Article managment
- Images



client



server monolith



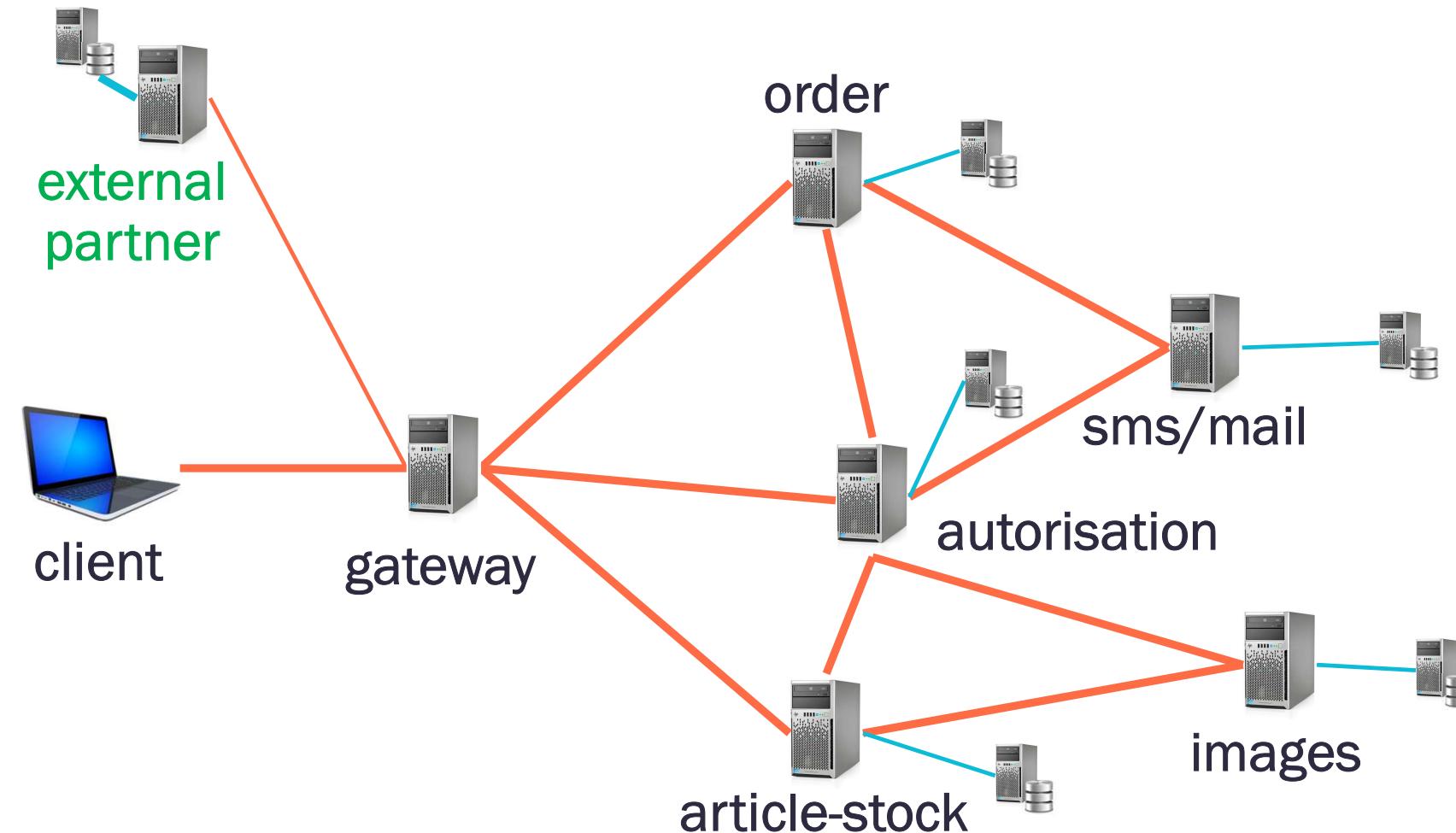
database

Why OAuth ?

OpenId Connect

Guillaume Chervet – novembre 2024

Why OAuth ?



OpenId Connect



Why OAuth ?

OpenId Connect

Why OAuth ?

Au revoir.
Guillaume Chervet – novembre 2024

OpenId Connect



Audience
api-market-square

Why OAuth ?

Scope
pig
payment

Scope
armor
payment

OpenId Connect

Guillaume Chervet – novembre 2024

Open ID Connect



Guillaume Chervet – novembre 2024

OpenId Connect