

## Are your friends already on Yelp?

Many of your friends may already be here, now you can find out. Just log in and we'll display all your contacts, and you can select which ones to invite! And don't worry, we don't keep your email password or your friends' addresses. We loathe spam, too.

Your Email Service



Your Email Address

(e.g. bob@gmail.com)

Your Gmail Password

(The password you use to log into your Gmail email)

[Skip this step](#)

**Check Contacts**

**commit**

@leomicheloni

Step 1  
Find Friends

Step 2  
Profile Information

Step 3  
Profile Picture

### Are your friends already on Facebook?

Many of your friends may already be here. Searching your email account is the fastest way to find your friends on Facebook.

 **Gmail**

Your Email:

Email Password:

[Find Friends](#)

 Facebook will not store your password.

 **Yahoo!**

[Find Friends](#)

 **Windows Live Hotmail**

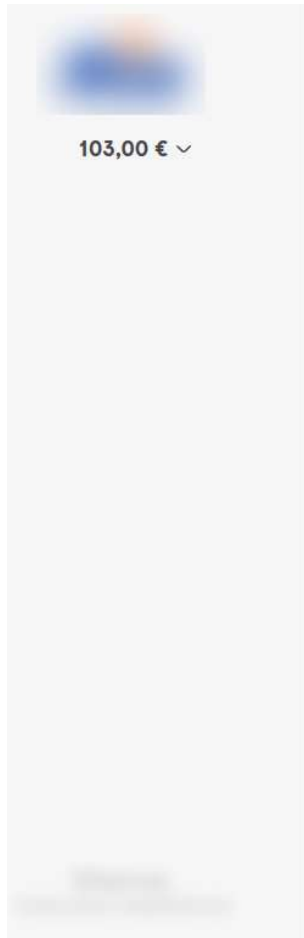
[Find Friends](#)

 **Other Email Service**

[Find Friends](#)

**commit**

@leomicheloni



Particulares / Private Empresas



Por favor, asegúrese de usar los mismos datos de acceso de su banca online

MODO DE IDENTIFICACIÓN:

Documento

Tipo de documento:

NIF

Número de documento

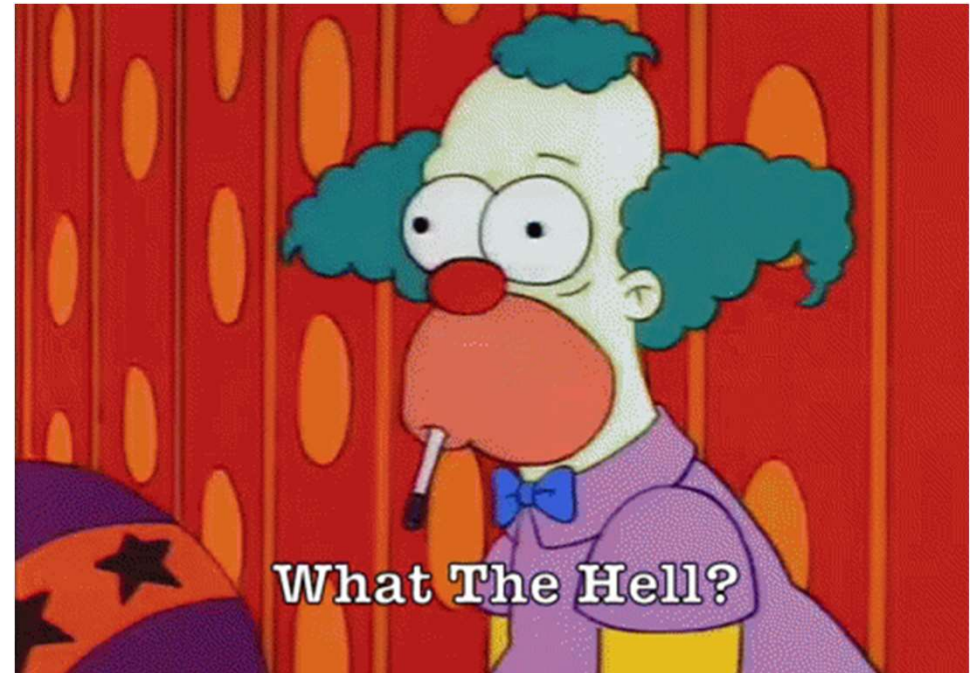
Clave de acceso:

••••••••

Ich akzeptiere die Verwendung des Zahlungsinittierungsdienstes oder des Kontoinformationsdienstes von Sofort, damit Sofort eine Zahlung initiieren oder auf mein Zahlungskonto zugreifen kann.

Se aplica nuestra [Política de protección de datos personales](#)

Continuar



commit

@leomicheloni



@leomicheloni

**commit**



# Autorización, Tokens, Flujos, oAuth y OIDC para todo el mundo

Sobre todo developers.

**TOKIOTA**



Leonardo Micheloni  
Developer

@leomicheloni



**commit**



# Objetivo

Comprender los conceptos principales

- Autorización
- Flujo
- Token
- Scope
- Client
- IDS
- Resource



**Disclaimer**

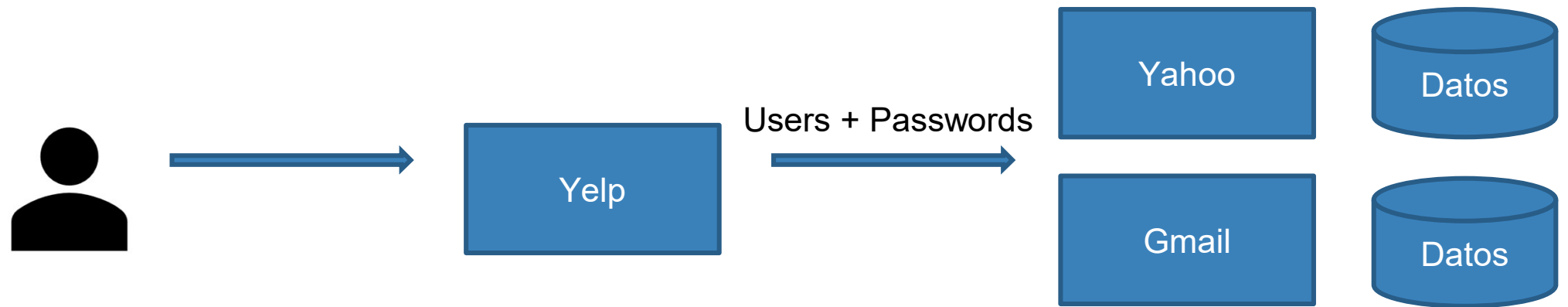


**commit**



**commit**

# Modelo de acceso a recursos legacy



@leomicheloni

**commit**



# Problemas

- Exponemos nuestra password
  - Y posiblemente la de otro sitio
- Damos acceso ilimitado a nuestros datos (recursos)
- Damos acceso por tiempo ilimitado
  - A menos que cambiemos nuestra password
  - Y el sitio web no lo haga por nosotros
- El Sistema opera en nuestro nombre

# OAuth2

**Authentication** is the process of ascertaining that somebody really is who they claim to be.

**Authorization** refers to rules that determine who is allowed to do what. E.g. Adam may be authorized to create and delete databases, while Usama is only authorised to read.

**OAuth** is an [open standard](#) for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords.<sup>[1]</sup> This mechanism is used by companies such as [Amazon](#),<sup>[2]</sup> [Google](#), [Facebook](#), [Microsoft](#) and [Twitter](#) to permit the users to share information about their accounts with third party applications or websites.

Generally, OAuth provides to clients a "secure delegated access" to server resources on [behalf of a resource owner](#). It specifies a process for resource owners to authorize third-party access to their server resources without sharing their credentials. Designed specifically to work with [Hypertext Transfer Protocol](#) (HTTP), OAuth essentially allows [access tokens](#) to be issued to third-party clients by an [authorization](#) server, with the approval of the resource owner. The third party then uses the [access token](#) to access the protected resources hosted by the resource server.<sup>[3]</sup>



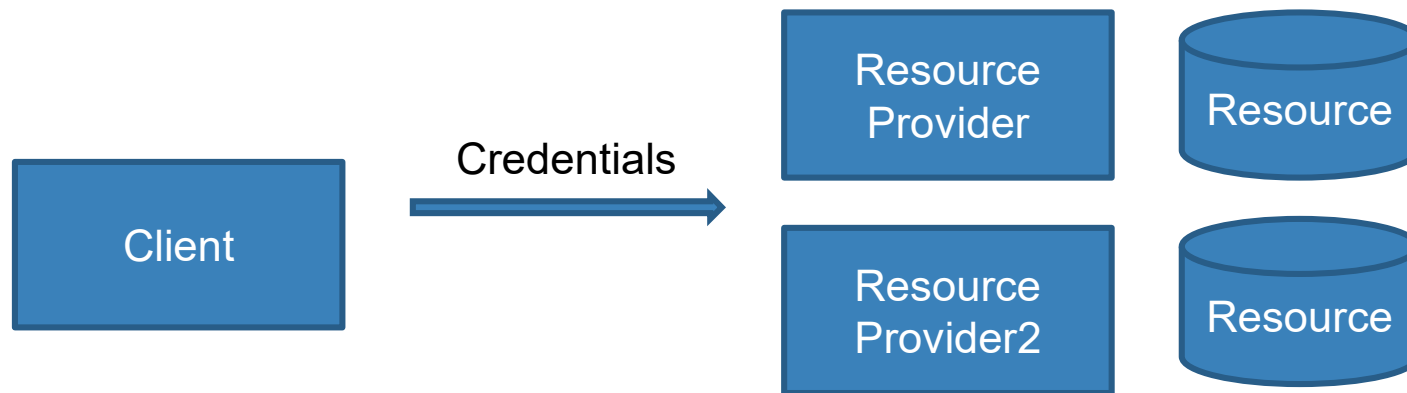
<https://oauth.net/2/>

**commit**

@leomicheloni



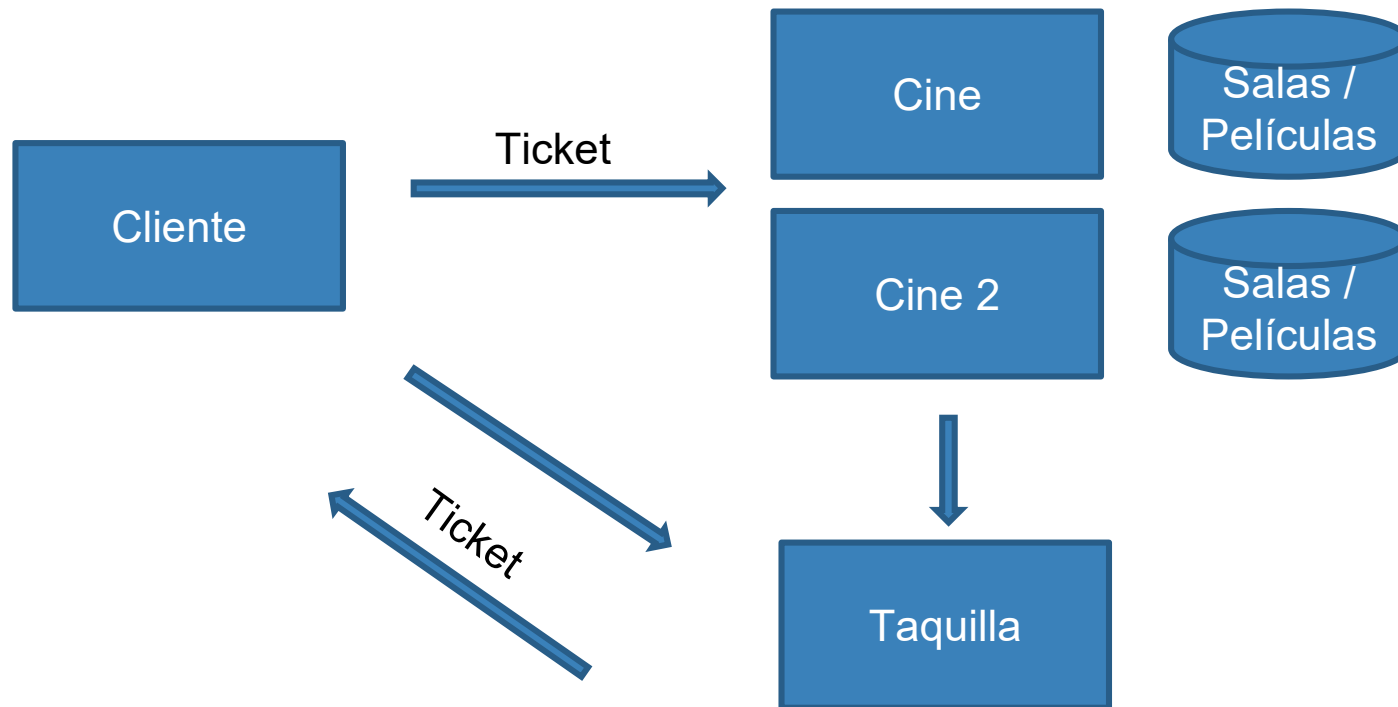
# Usando terminología OAuth



@leomicheloni

**commit**

# Mejora basada tickets

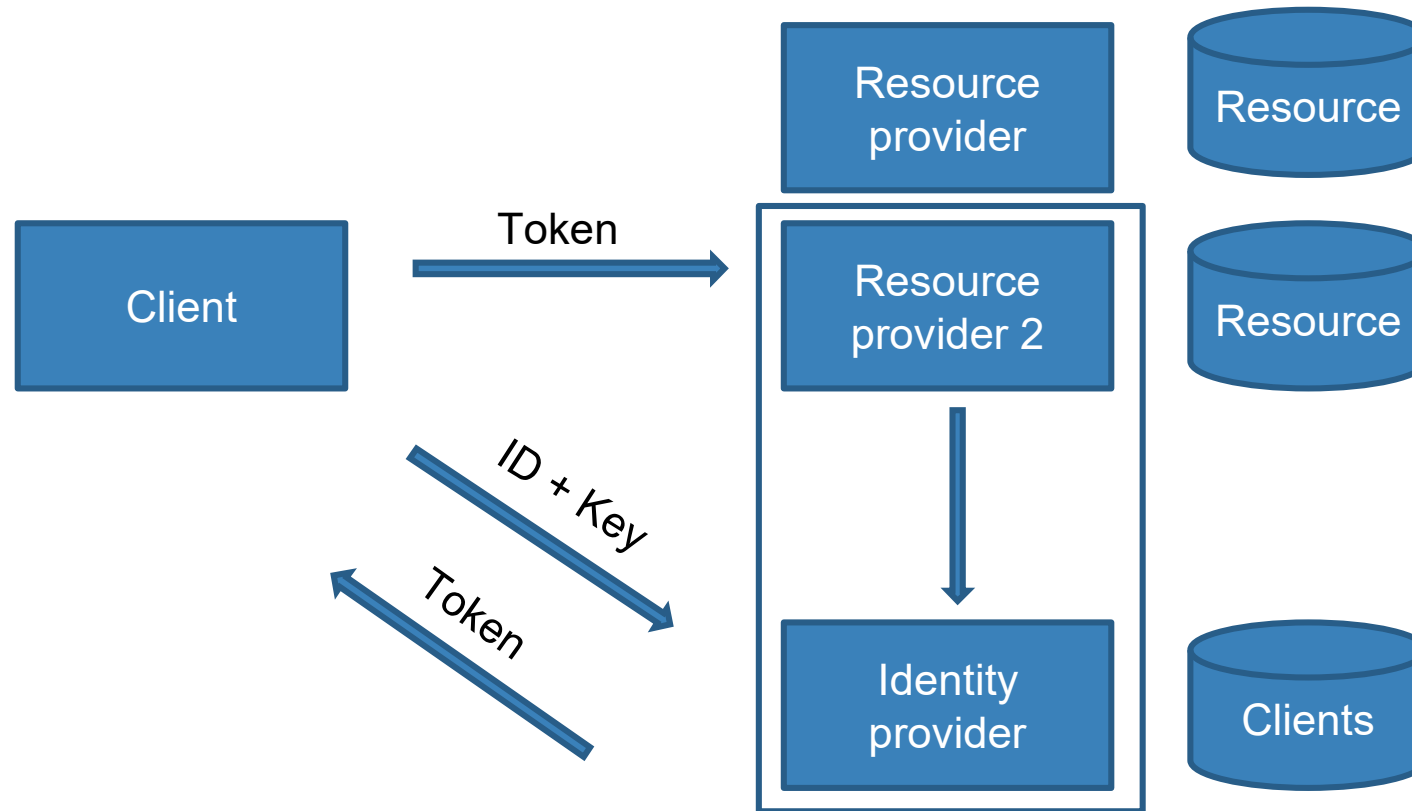


@leomicheloni

**commit**



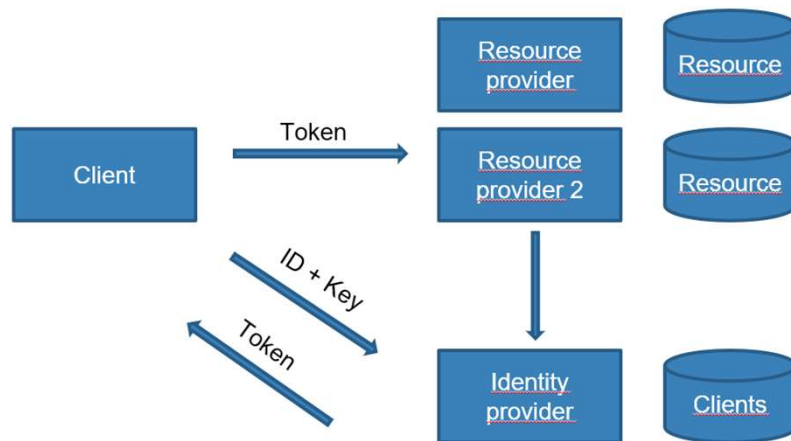
# Mejora basada en OAuth



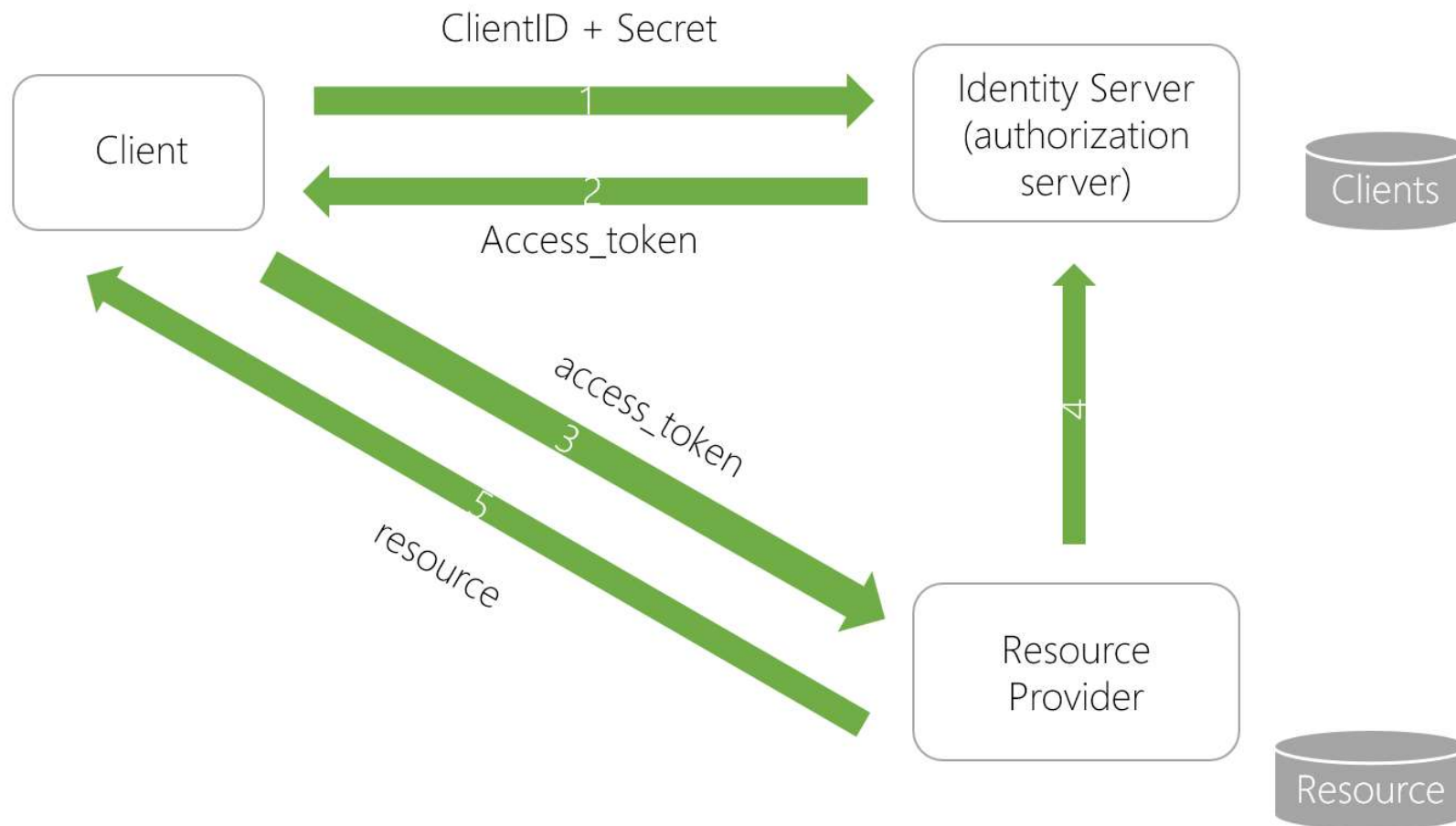
@leomicheloni

**commit**

# ¿Qué conseguimos con esto?

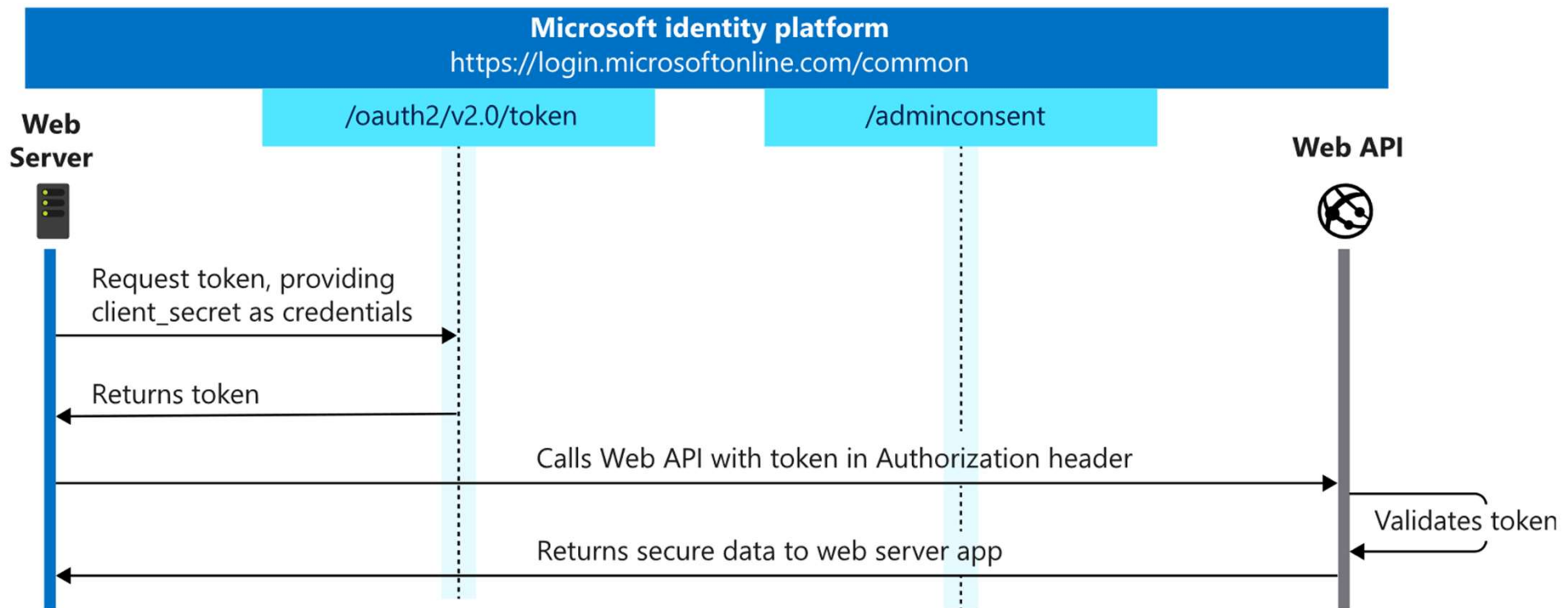


- No exponer nuestras credenciales
  - Otorgar un ticket (token) de acceso
  - Limitar el acceso a un recurso (scope)
  - Otorgar acceso por un tiempo limitado
  - Separar responsabilidades
- 
- El IDP (auth server) conoce a los clients
  - Los resource providers conocen al IDP
  - Centralizar el otorgamiento de accesos



@leomicheloni

# oAuth Client Credentials grant



# Ejemplo Client Credentials

@leomicheloni

**commit**




# JWT

## JSON Web Token

@leomicheloni

**commit**


Debugger Libraries Introduction Ask Get a T-shirt!
Crafted by Auth0

ALGORITHM

RS256

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJSUzI1NiIsImtpZCI6IjVpc091MkV4dU11QmJleHU2ej1IS0EiLCJ0eXAiOiJhdCtqd3QifQ.eyJyYmY0IjE0TAZMTi1MjUsImV4cCI6MTU5MDMxNjE5N2l0eXNzIjoiaHR0cDovL2xvY2FsaG9zdDo1MDAwIiwiaXVkiOiYXBpMSIsImNsaWVudF9pZCI6ImNjLmNsaWVudC5vYXV0aCIsInNjb3BlIjpbImFwaTEiXX0.JHjJozdDaIi5SisxLGM1bUvAcwTxvqADzQpTd2obyMirocNeejfnBVRy1iFNKSK4o7uoTn7kAf_BwI0E4SF9NpJ7LV3826InL1EexPw09VwsCTN_Sq1u2fUzJdZ9FcAPS5tbb4R5R6VJpJzPoBONxBk5qc-1JjiK-uXT20g3huxScQxm1N_imQCTWP6m0-qRaNq-1rhwcPM5WUBkKncDXQqbzWzhaqy_Rs5lCm_PbE0hekMOVptaGq3Jikz1FvL6uaaRF1zAFsP5gh7jhyAjrZH-gXLQI1YohAqaR2c5t29etGW1G3eLf5yjkWhpW5iHXizw94yyh2uLyU48w
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "RS256",
  "kid": "51s0e2XuM8BbKXu6z9HKA",
  "typ": "at+jwt"
}
```

PAYLOAD: DATA

```
{
  "nbf": 1590312525,
  "exp": 1590316125,
  "iss": "http://localhost:5000",
  "aud": "api",
  "client_id": "cc.client.oauth",
  "scope": [
    "api"
  ]
}
```

VERIFY SIGNATURE

RSASHA256

```
base64UrlEncode(header) + "." +
base64UrlEncode(payload),
```

Public Key or Certificate. Enter it in plain text only if you want to verify a token

Private Key. Enter it in plain text only if you want to generate a new token. The key never leaves your browser.

**commit**

@leomicheloni

¿Cómo funciona la firma?



## {

}

- commit**



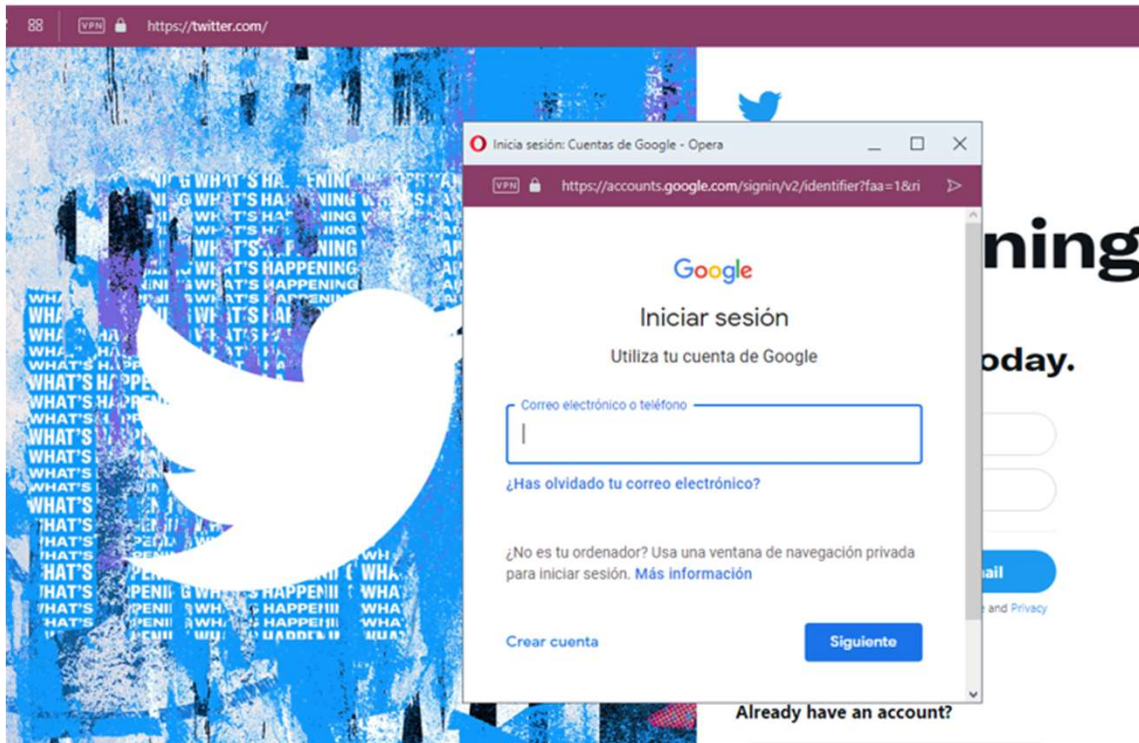
# Flujos / Grants

Cómo interactúa el cliente para obtener un token

@leomicheloni

**commit**

# ¿Y cómo funciona la autorización con Google?



Login

Login with password    Login with SMS


Mobile Number/Email


Password

[Forgot password?](#)

Next

Or

 Continue with Google

 Continue with Facebook

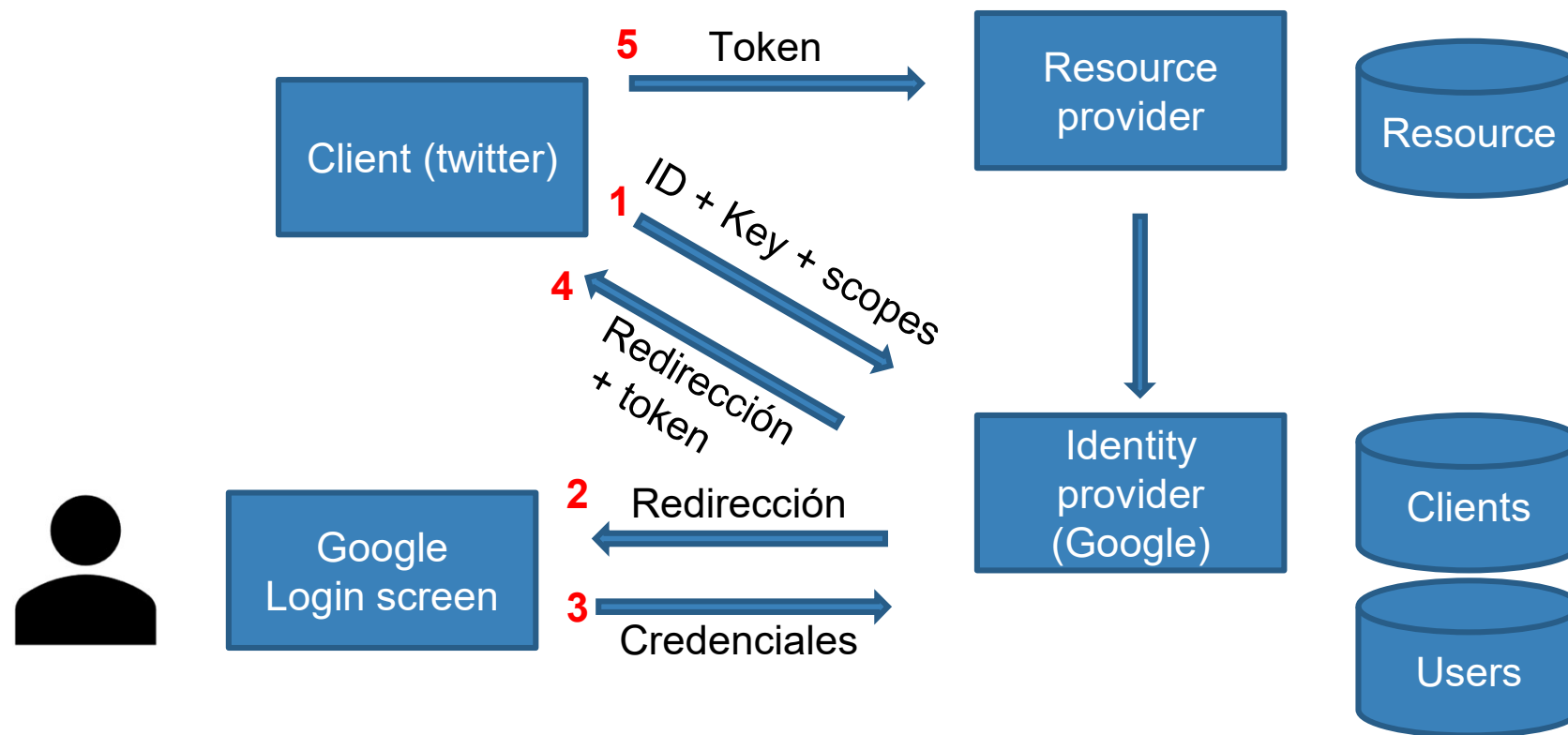
**commit**

@leomicheloni

# ¿Y cómo funciona la autorización con Google?

- Solicita nuestros datos (email, nombre) y nos registra en su sistema
- Yo doy acceso a sistema externo (cliente)
- El cliente es conocido previamente por el IDP
- El cliente establece previamente el scope (en este caso nuestro email, nombre, etc.)
- Nosotros somos resource owner (interacción del usuario)
- Siempre es necesario un navegador

# Flow con interacción del usuario



# oAuth flow / grants

“Los flujos o grants son las formas en que los clientes interactúan con el IDP para gestionar tokens”

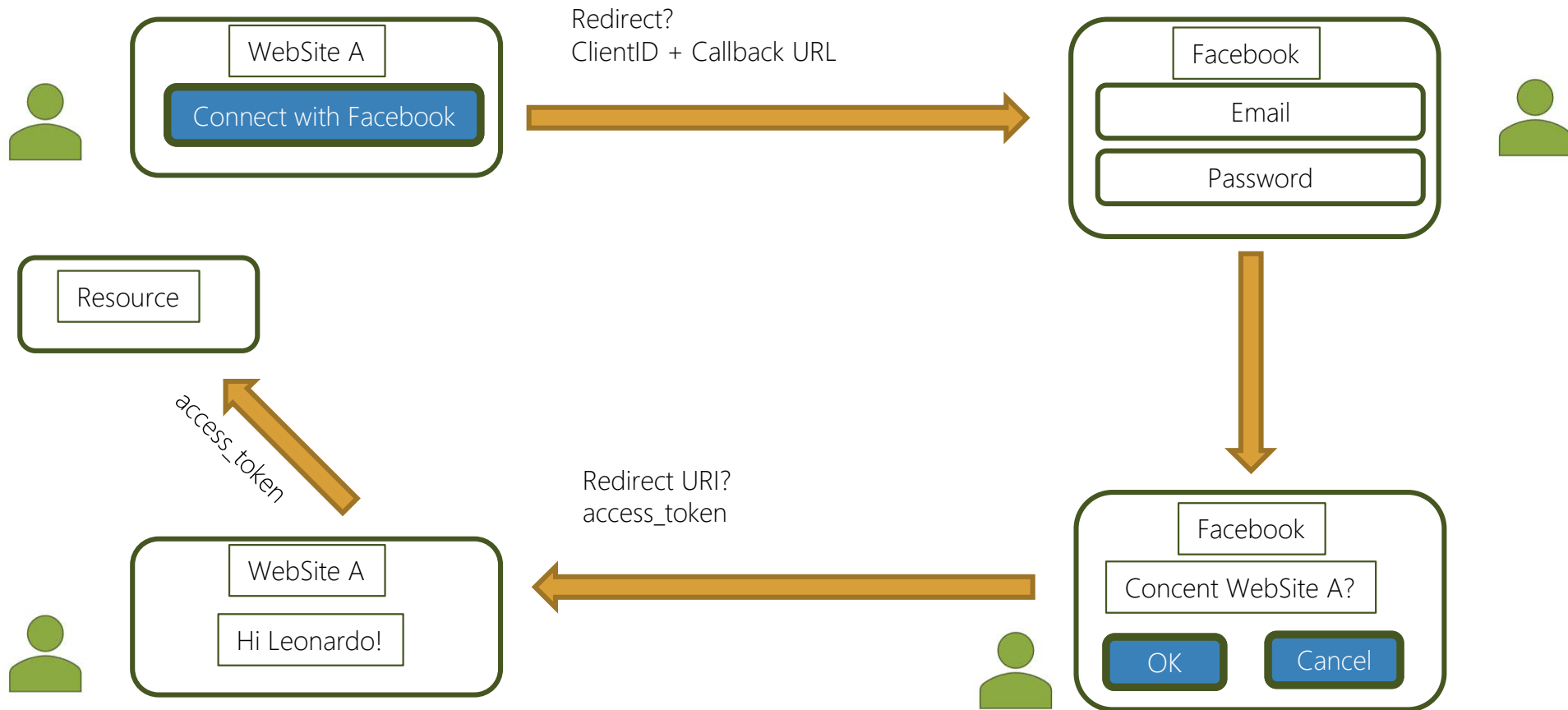
- Client credentials – Robots
- Implicit – Web SPA
- Code [PKCE] – Web
- Device – Devices
- Resource owner – Legacy
- Hybrid

@leomicheloni

**commit**



# Implicit flow

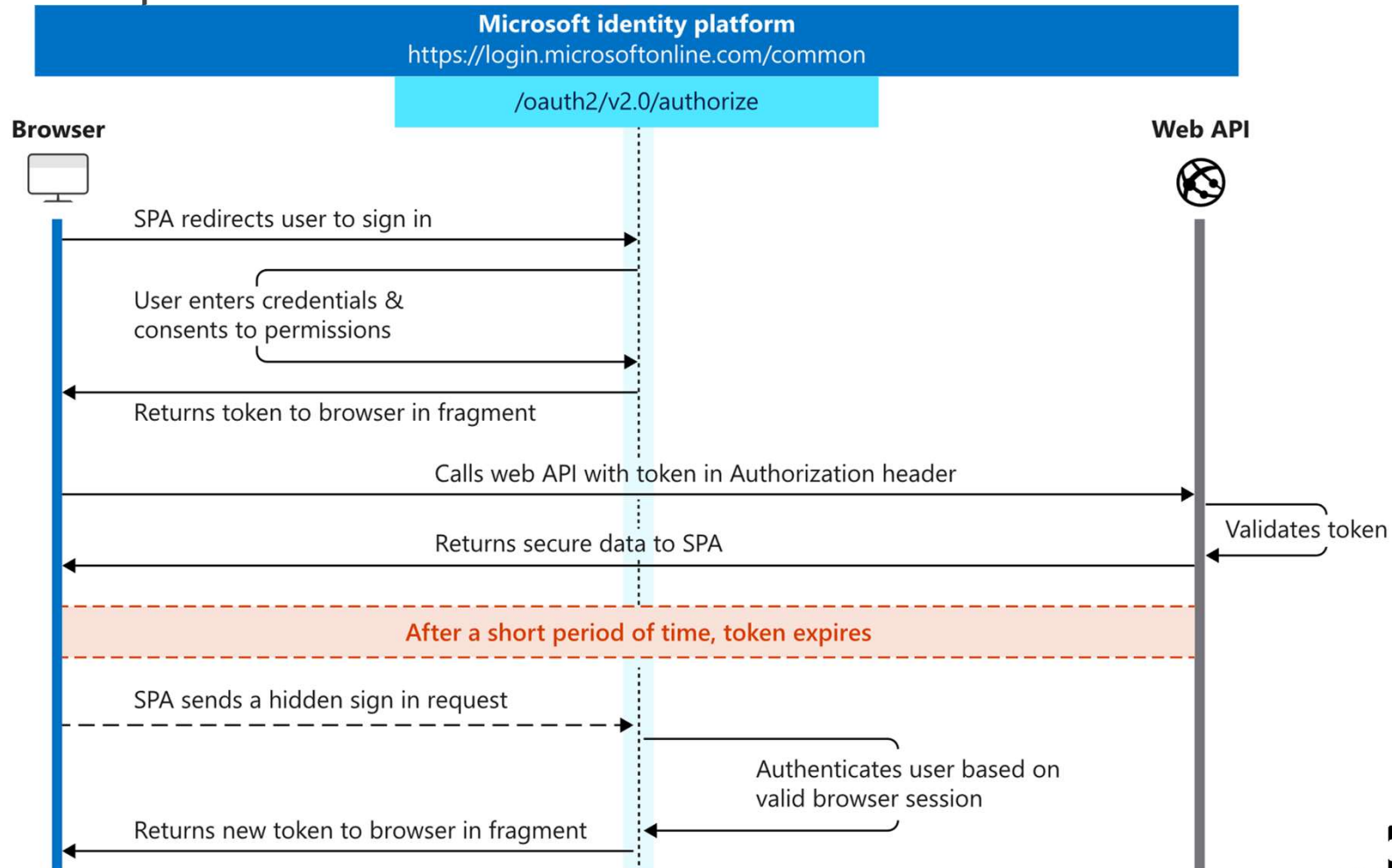


# Ejemplo Implicit flow

@leomicheloni

**commit**

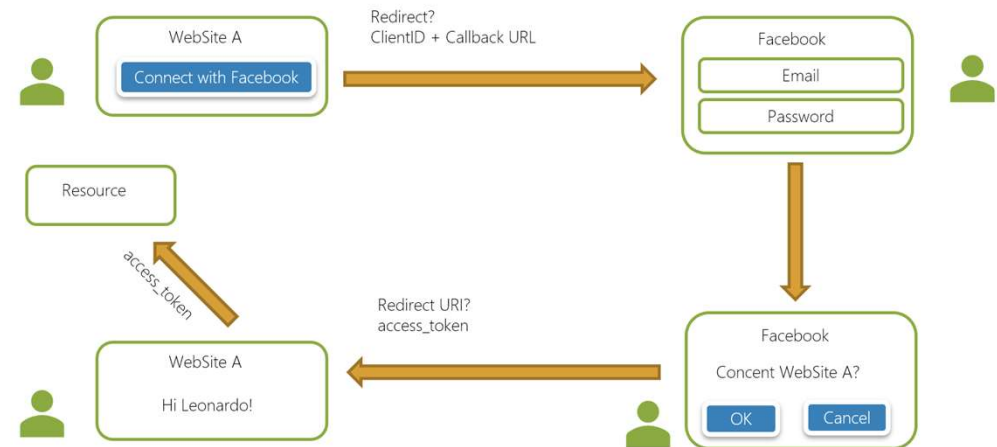
# oAuth Implicit flow



@leomicheloni

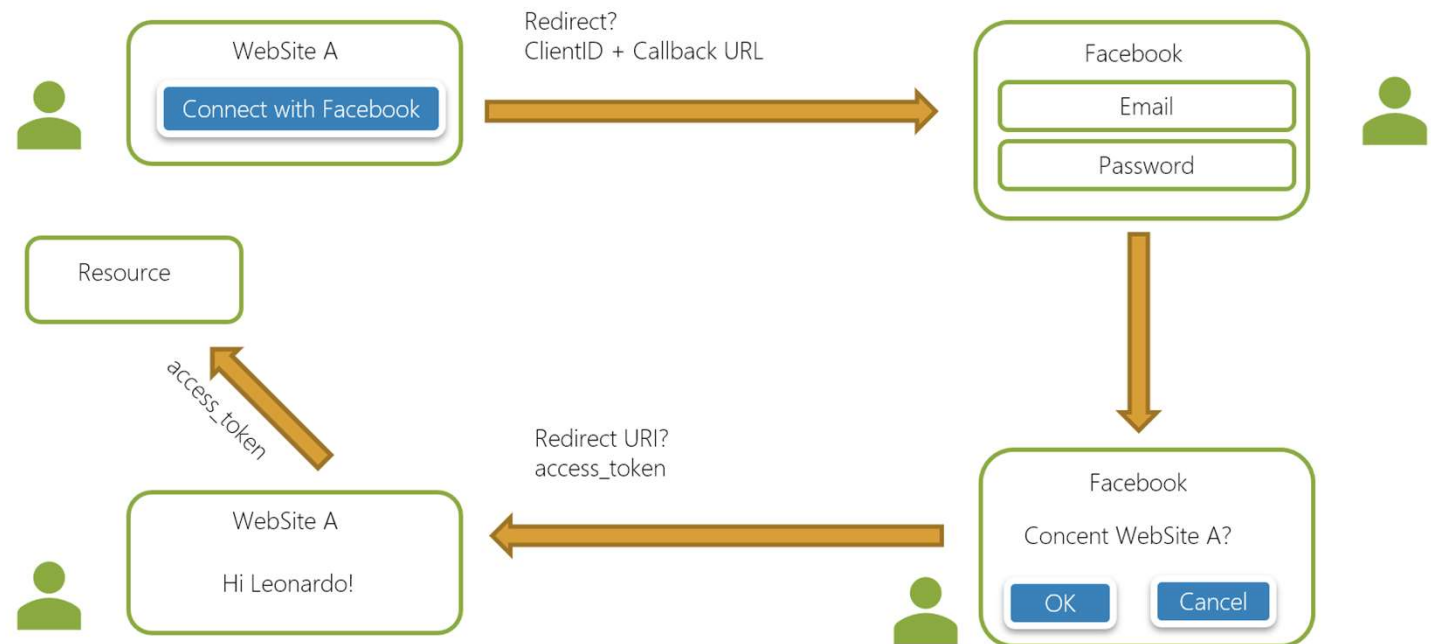
# Características

- El usuario ingresa sus credenciales en el sitio de confianza
- El cliente se encuentra registrado
- Se puede mostrar un “consent” para autorizar los acceso a scopes
- Se utilizan redirecciones (web)
- `callback_url` es un dato muy importante
- Todo ocurre en el frontend



# Problemas

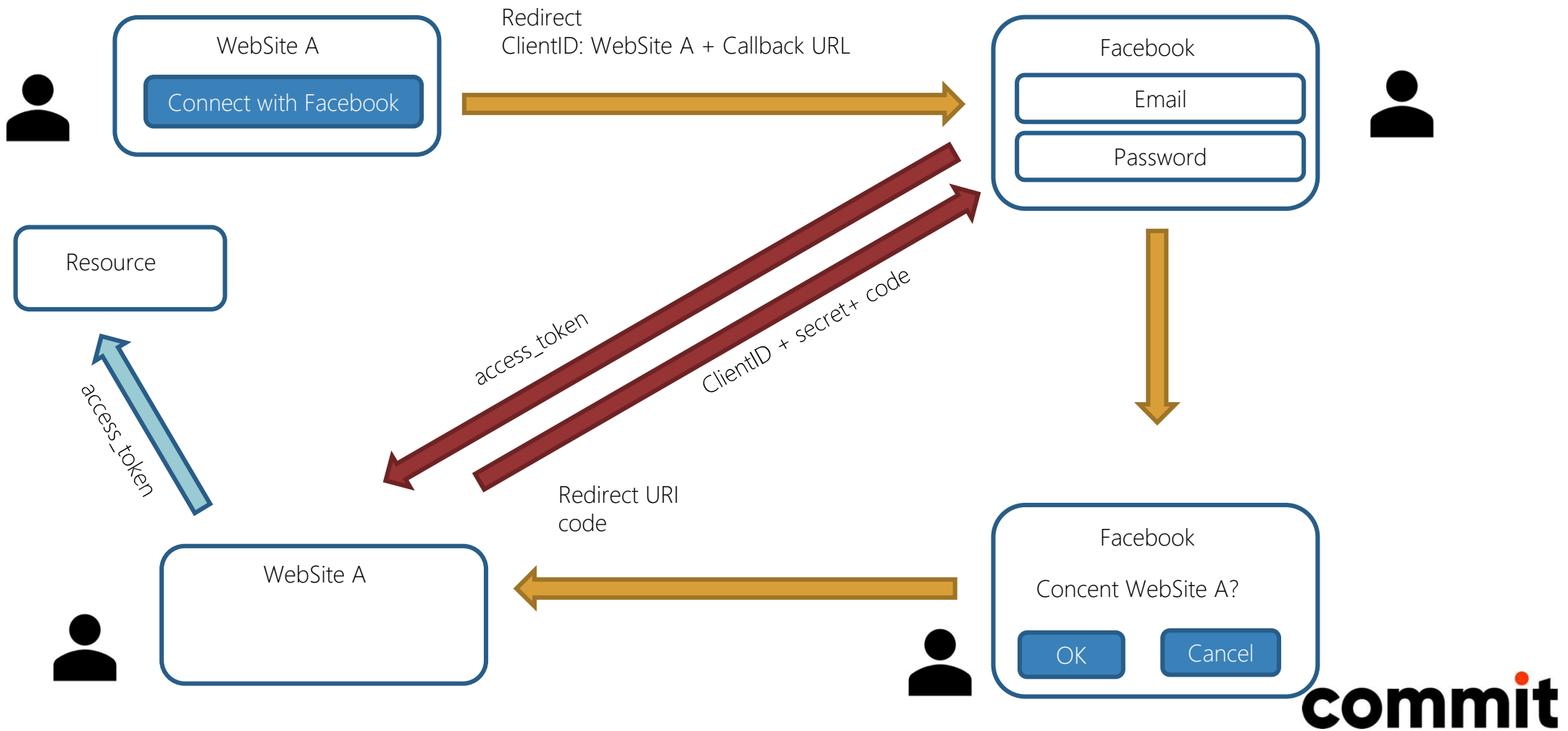
- Es posible interceptar el token
- El Token puede quedar almacenado en el browser



**commit**

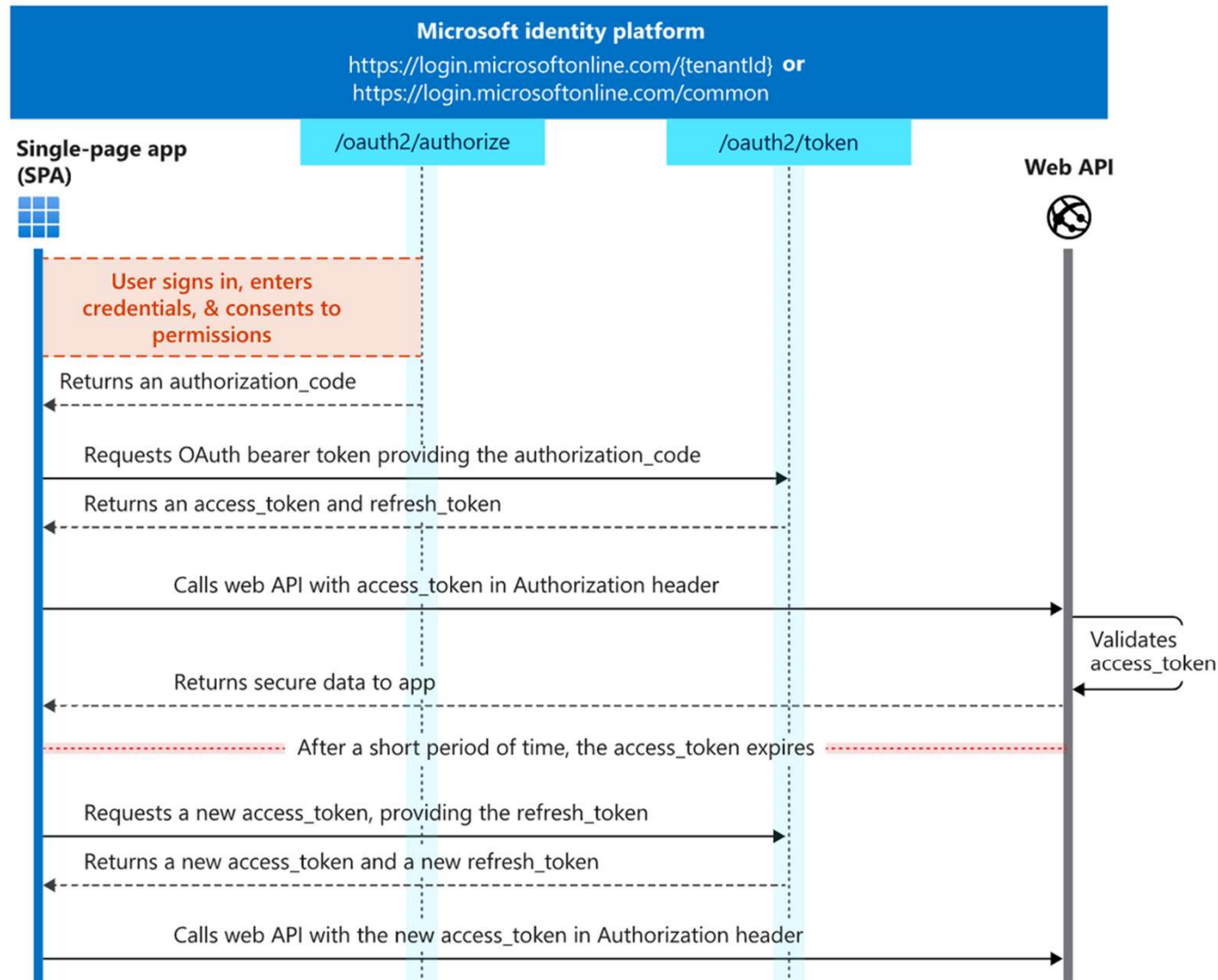
@leomicheloni

# Solucionando problemas del implicit Flow: code flow



@leomicheloni

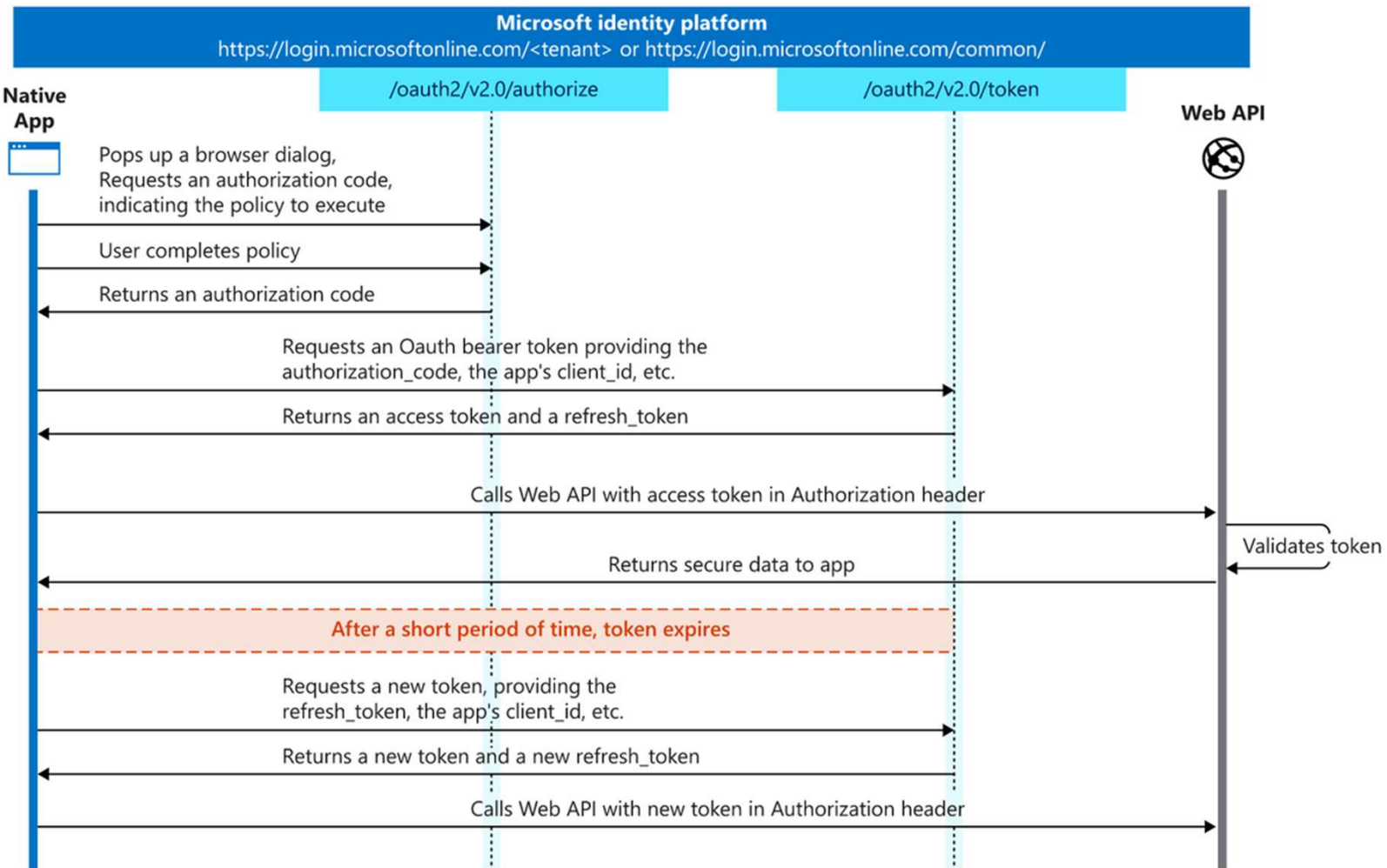
**commit**



@leomicheloni

**commit**

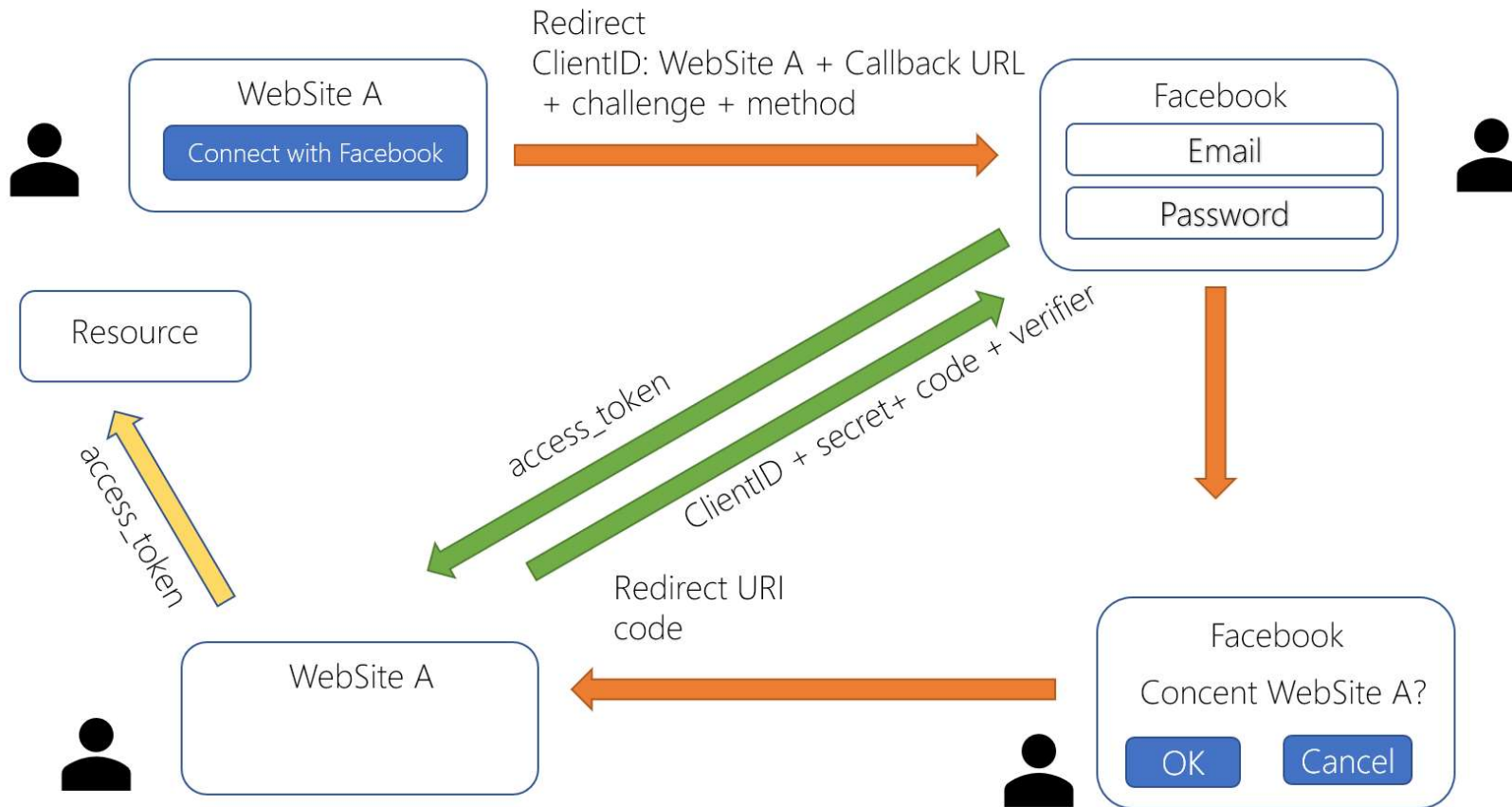




**commit**

@leomicheloni

# PCKE: Más seguridad a authorization code flow



@leomicheloni

**commit**

# Open ID Connect

Agregando autenticación a OAuth

@leomicheloni

**commit**

# OIDC



**OpenID Connect (OIDC)** is an [authentication](#) layer on top of [OAuth 2.0](#), an [authorization](#) framework.<sup>[1]</sup> The standard is controlled by the [OpenID Foundation](#).

OpenID Connect is a simple identity layer on top of the [OAuth 2.0](#) protocol, which allows [computing clients](#) to verify the identity of an end-user based on the authentication performed by an authorization server, as well as to obtain basic profile information about the end-user in an interoperable and [REST-like](#) manner. In technical terms, OpenID Connect specifies a [RESTful HTTP API](#), using [JSON](#) as a data format.

@leomicheloni

**commit**

# ¿Cómo funciona OIDC?

Ben pocas palabras, utilizando OAuth como base agregar scopes que representan elementos de la identidad del usuario

- Name
- Surname
- Address
- Etc.

Además define un nuevo endpoint (user\_info) para obtener un nuevo token (id\_token) con esta información de identidad

Y algunas cosas...

@leomicheloni

**commit**

# El Id\_token

ybjppZGVudGlmeTpyZXN0LFwFaTpyb2x1IjoiQWR  
 taW5pc3RyYXRvciIsInVybpbhbnlpZDpyb2x1Ijo  
 iQWRtaW5pc3RyYXRvciIsInJvbGU0IlsiQ2xhaW1  
 BZG1pbiIsIkNsYWltVHJhbnNmb3JtYXRpb25BZG1  
 pbiIsIkNvbmlkYXN0b3Rpb25BZG1pbiIsIkdyb3VwQWR  
 taW4iLCJZGVudGlmeSBTZXRJ2aWN1IiwiT3JnYW50  
 pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR1cEFkbWl  
 uIiwiVXNlckFkbWluIl0sImVtYWlsIjpbInRlc3R  
 AZW1haWw529tIiwiZGVudGlmeSBTZXRJ2aWN1Ii  
 wibmFtZSI6ImFkbWluIiwiYXN0b3Rpb25BZG1pbi  
 IsIkdyb3VwQWRtaW4iLCJZGVudGlmeSBTZXRJ2a  
 WN1IiwiT3JnYW50pemF0aW9uQWRtaW4iLCJTeXN0  
 ZW1TZXR1cEFkbWluIiwiVXNlckFkbWluIl0sImVt  
 YWlsIjpbInRlc3RlZGVudGlmeSBTZXRJ2aWN1Ii  
 wiT3JnYW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1T  
 ZXR1cEFkbWluIiwiVXNlckFkbWluIl0sImVtYWls  
 IjpbInRlc3RlZGVudGlmeSBTZXRJ2aWN1IiwiT3J  
 nYW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR1c  
 EFBZG1pbiIsIkNsYWltVHJhbnNmb3JtYXRpb25B  
 ZG1pbiIsIkNvbmlkYXN0b3Rpb25BZG1pbiIsIkdy  
 b3VwQWRtaW4iLCJZGVudGlmeSBTZXRJ2aWN1Ii  
 wiT3JnYW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1T  
 ZXR1cEFkbWluIiwiVXNlckFkbWluIl0sImVtYWls  
 IjpbInRlc3RlZGVudGlmeSBTZXRJ2aWN1IiwiT3  
 JnYW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR1  
 cEFkbWluIiwiVXNlckFkbWluIl0sImVtYWlsIjpb  
 InRlc3RlZGVudGlmeSBTZXRJ2aWN1IiwiT3JnYW  
 50pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR1cEFk  
 bWluIiwiVXNlckFkbWluIl0sImVtYWlsIjpbInRl  
 c3RlZGVudGlmeSBTZXRJ2aWN1IiwiT3JnYW50pe  
 mF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR1cEFkbWl  
 uIiwiVXNlckFkbWluIl0sImVtYWlsIjpbInRlc3R  
 lZGVudGlmeSBTZXRJ2aWN1IiwiT3JnYW50pemF0  
 aW9uQWRtaW4iLCJTeXN0ZW1TZXR1cEFkbWluIi  
 wiVXNlckFkbWluIl0sImVtYWlsIjpbInRlc3RlZG  
 VudGlmeSBTZXRJ2aWN1IiwiT3JnYW50pemF0aW  
 9uQWRtaW4iLCJTeXN0ZW1TZXR1cEFkbWluIiwiV  
 XNlckFkbWluIl0sImVtYWlsIjpbInRlc3RlZGVu  
 dGlmeSBTZXRJ2aWN1IiwiT3JnYW50pemF0aW9u  
 QWRtaW4iLCJTeXN0ZW1TZXR1cEFkbWluIiwiVXN  
 lckFkbWluIl0sImVtYWlsIjpbInRlc3RlZGVudG  
 lmeSBTZXRJ2aWN1IiwiT3JnYW50pemF0aW9uQW  
 RtaW4iLCJTeXN0ZW1TZXR1cEFkbWluIiwiVXNl  
 ckFkbWluIl0sImVtYWlsIjpbInRlc3RlZGVudGl  
 meSBTZXRJ2aWN1IiwiT3JnYW50pemF0aW9uQWR  
 taW4iLCJTeXN0ZW1TZXR1cEFkbWluIiwiVXNlck  
 FkbWluIl0sImVtYWlsIjpbInRlc3RlZGVudGlme  
 SBTZXRJ2aWN1IiwiT3JnYW50pemF0aW9uQWRta  
 W4iLCJTeXN0ZW1TZXR1cEFkbWluIiwiVXNlckFk  
 bWluIl0sImVtYWlsIjpbInRlc3RlZGVudGlmeSB  
 TZXRJ2aWN1IiwiT3JnYW50pemF0aW9uQWRtaW4  
 iLCJTeXN0ZW1TZXR1cEFkbWluIiwiVXNlckFkbW  
 luIl0sImVtYWlsIjpbInRlc3RlZGVudGlmeSBTZ  
 XRJ2aWN1IiwiT3JnYW50pemF0aW9uQWRtaW4iL  
 CJTeXN0ZW1TZXR1cEFkbWluIiwiVXNlckFkbWlu  
 Il0sImVtYWlsIjpbInRlc3RlZGVudGlmeSBTZXR  
 J2aWN1IiwiT3JnYW50pemF0aW9uQWRtaW4iLCJ  
 TTeXN0ZW1TZXR1cEFkbWluIiwiVXNlckFkbWluI  
 l0sImVtYWlsIjpbInRlc3RlZGVudGlmeSBTZXRJ  
 2aWN1IiwiT3JnYW50pemF0aW9uQWRtaW4iLCJTe  
 xN0ZW1TZXR1cEFkbWluIiwiVXNlckFkbWluIl0s  
 ImVtYWlsIjpbInRlc3RlZGVudGlmeSBTZXRJ2aW  
 N1IiwiT3JnYW50pemF0aW9uQWRtaW4iLCJTeXN0  
 ZW1TZXR1cEFkbWluIiwiVXNlckFkbWluIl0sImV  
 tYWlsIjpbInRlc3RlZGVudGlmeSBTZXRJ2aWN1I  
 iwiT3JnYW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1  
 TZXR1cEFkbWluIiwiVXNlckFkbWluIl0sImVtYW  
 lsIjpbInRlc3RlZGVudGlmeSBTZXRJ2aWN1Iiwi  
 T3JnYW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZ  
 XR1cEFkbWluIiwiVXNlckFkbWluIl0sImVtYWls  
 IjpbInRlc3RlZGVudGlmeSBTZXRJ2aWN1IiwiT3  
 JnYW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR  
 1cEFkbWluIiwiVXNlckFkbWluIl0sImVtYWlsIj  
 pbInRlc3RlZGVudGlmeSBTZXRJ2aWN1IiwiT3Jn  
 YW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR1c  
 EFBZG1pbiIsIkNsYWltVHJhbnNmb3JtYXRpb25B  
 ZG1pbiIsIkNvbmlkYXN0b3Rpb25BZG1pbiIsIkdy  
 b3VwQWRtaW4iLCJZGVudGlmeSBTZXRJ2aWN1Ii  
 wiT3JnYW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1T  
 ZXR1cEFkbWluIiwiVXNlckFkbWluIl0sImVtYWls  
 IjpbInRlc3RlZGVudGlmeSBTZXRJ2aWN1IiwiT3  
 JnYW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR  
 1cEFkbWluIiwiVXNlckFkbWluIl0sImVtYWlsIj  
 pbInRlc3RlZGVudGlmeSBTZXRJ2aWN1IiwiT3Jn  
 YW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR1c  
 EFBZG1pbiIsIkNsYWltVHJhbnNmb3JtYXRpb25B  
 ZG1pbiIsIkNvbmlkYXN0b3Rpb25BZG1pbiIsIkdy  
 b3VwQWRtaW4iLCJZGVudGlmeSBTZXRJ2aWN1Ii  
 wiT3JnYW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1T  
 ZXR1cEFkbWluIiwiVXNlckFkbWluIl0sImVtYWls  
 IjpbInRlc3RlZGVudGlmeSBTZXRJ2aWN1IiwiT3  
 JnYW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR  
 1cEFkbWluIiwiVXNlckFkbWluIl0sImVtYWlsIj  
 pbInRlc3RlZGVudGlmeSBTZXRJ2aWN1IiwiT3Jn  
 YW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR1c  
 EFBZG1pbiIsIkNsYWltVHJhbnNmb3JtYXRpb25B  
 ZG1pbiIsIkNvbmlkYXN0b3Rpb25BZG1pbiIsIkdy  
 b3VwQWRtaW4iLCJZGVudGlmeSBTZXRJ2aWN1Ii  
 wiT3JnYW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1T  
 ZXR1cEFkbWluIiwiVXNlckFkbWluIl0sImVtYWls  
 IjpbInRlc3RlZGVudGlmeSBTZXRJ2aWN1IiwiT3  
 JnYW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR  
 1cEFkbWluIiwiVXNlckFkbWluIl0sImVtYWlsIj  
 pbInRlc3RlZGVudGlmeSBTZXRJ2aWN1IiwiT3Jn  
 YW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR1c  
 EFBZG1pbiIsIkNsYWltVHJhbnNmb3JtYXRpb25B  
 ZG1pbiIsIkNvbmlkYXN0b3Rpb25BZG1pbiIsIkdy  
 b3VwQWRtaW4iLCJZGVudGlmeSBTZXRJ2aWN1Ii  
 wiT3JnYW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1T  
 ZXR1cEFkbWluIiwiVXNlckFkbWluIl0sImVtYWls  
 IjpbInRlc3RlZGVudGlmeSBTZXRJ2aWN1IiwiT3  
 JnYW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR  
 1cEFkbWluIiwiVXNlckFkbWluIl0sImVtYWlsIj  
 pbInRlc3RlZGVudGlmeSBTZXRJ2aWN1IiwiT3Jn  
 YW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR1c  
 EFBZG1pbiIsIkNsYWltVHJhbnNmb3JtYXRpb25B  
 ZG1pbiIsIkNvbmlkYXN0b3Rpb25BZG1pbiIsIkdy  
 b3VwQWRtaW4iLCJZGVudGlmeSBTZXRJ2aWN1Ii  
 wiT3JnYW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1T  
 ZXR1cEFkbWluIiwiVXNlckFkbWluIl0sImVtYWls  
 IjpbInRlc3RlZGVudGlmeSBTZXRJ2aWN1IiwiT3  
 JnYW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR  
 1cEFkbWluIiwiVXNlckFkbWluIl0sImVtYWlsIj  
 pbInRlc3RlZGVudGlmeSBTZXRJ2aWN1IiwiT3Jn  
 YW50pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR1c  
 EFBZG1pbiIsIkNsYWltVHJhbnNmb3JtYXRpb25B  
 ZG1pbiIsIkNvbmlkYXN0b3Rpb25BZG1pbiIsIkdy  
 b3VwQWRtaW4iLCJZGVudGlmeSBTZXRJ2aWN1Ii  
 wi

```
{
  "sub": "admin",
  "uiId": "427a74fb-7e47-4bed-b92e-0d09e6f1b479",
  "unique_name": "admin",
  "auth_time": 1578286467,
  "urn:identify:rest-api:role": "Administrator",
  "urn:anyid:role": "Administrator",
  "role": [
    "ClaimAdmin",
    "ClaimTransformationAdmin",
    "ConnectionAdmin",
    "GroupAdmin",
    "Identify Service",
    "OrganizationAdmin",
    "SystemSetupAdmin",
    "UserAdmin"
  ],
  "email": [
    "test@email.com",
    "test@email.com"
  ],
  "name": "admin",
  "profile": "profile value",
  "family_name": "Family Name",
  "given_name": "Given Name",
  "birthdate": "28/10/1982",
  "nickname": "ctt",
  "website": "https://ctt.com",
  "gender": "Male",
  "email_verified": "True",
  "phone_number_verified": "False",
  "urn:internal:userId": "1d684918-b8ac-4060-be4f-9dda04a1e5fb",
  "token_usage": "identity_token",
  "jti": "21f12442-8582-45f2-a9d8-bf7398ae27ef",
  "aud": [
    "webmvc_codeflow_id",
    "https://dev56.safewhere.local/runtime/"
  ]
}
```

**commit**

@leomicheloni

# Ejemplo OIDC

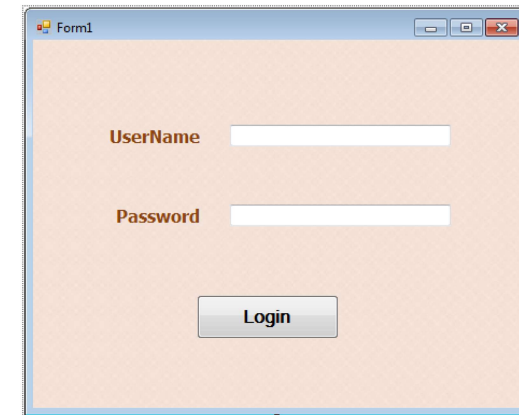
@leomicheloni

**commit**



# Otros flujos

- Resource owner: Para aplicaciones legacy o sin acceso a un navegador (user y password se envían al Auth Server)
- Device: Para dispositivos donde necesitamos que el usuario haga login pero el dispositivo no tiene acceso a un browser (ej. az login)



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Leonardo Micheloni> az login
A web browser has been opened at https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize. Please continue the login in the web browser. If no web browser is available or if the web browser fails to open, use device code flow with 'az login --use-device-code'.
```

**commit**

@leomicheloni

# ¿Preguntas?

## Referencias

- [JWT.io](https://jwt.io/)
- [oAuth](https://oauth.net)
- [Keycloak](https://www.keycloak.org/)
- [oAuth Playground](https://www.oauth.com/playground)



@leomicheloni

**commit**

¡Gracias!

**commit**