IDENTITY PROVIDERS

Do you know who I am?

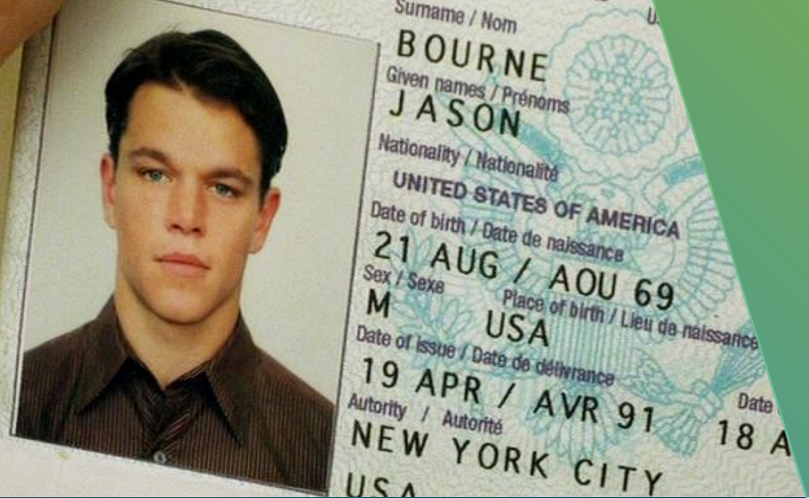
Identità sociologica

“... la concezione che un individuo ha di se stesso nell'individuale e nella società...” *

Identità filosofica

“...in filosofia, termine e principio filosofico che genericamente indica l'eguaglianza di un oggetto rispetto a sé stesso....” *

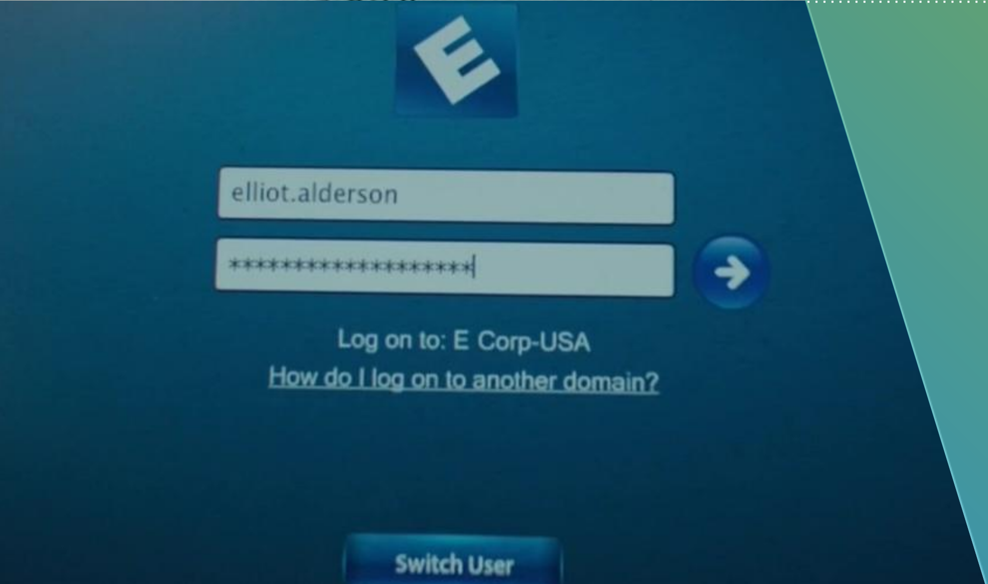
** fonte Wikipedia*



Surname / Nom
BOURNE
Given names / Prénoms
JASON
Nationality / Nationalité
UNITED STATES OF AMERICA
Date of birth / Date de naissance
21 AUG / AOU 69
Sex / Sexe
M
Place of birth / Lieu de naissance
USA
Date of issue / Date de délivrance
19 APR / AVR 91
Date
18 A
Autority / Autorité
NEW YORK CITY
USA

Identità anagrafica

"… insieme dei dati che identificano una persona nell'ambito delle istituzioni pubbliche..." *



elliot.alderson

****

Log on to: E Corp-USA

How do I log on to another domain?

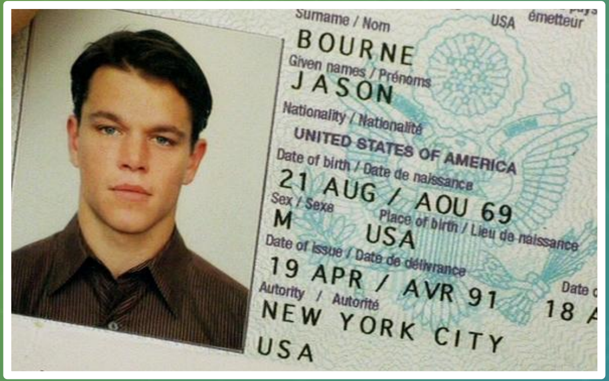
Identità virtuale

"... l'identità costituita da un utente presso comunità virtuali online..." *

** fonte Treccani e Wikipedia*

Real life

Virtual life



Surname / Nom USA émetteur pays

BOURNE

Given names / Prénoms

JASON

Nationality / Nationalité

UNITED STATES OF AMERICA

Date of birth / Date de naissance

21 AUG / AOU 69

Sex / Sexe Place of birth / Lieu de naissance

M USA

Date of issue / Date de délivrance

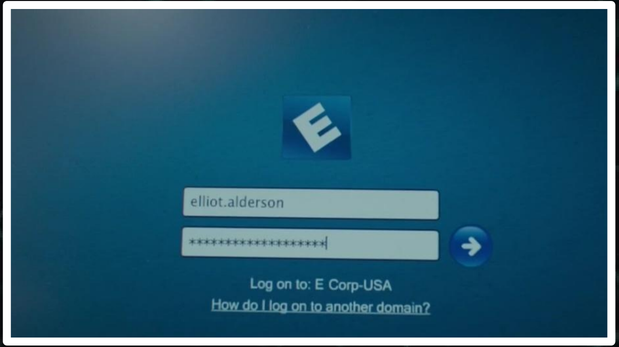
19 APR / AVR 91

Autority / Autorité Date

NEW YORK CITY 18 A

USA

VS

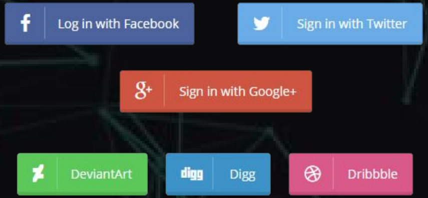


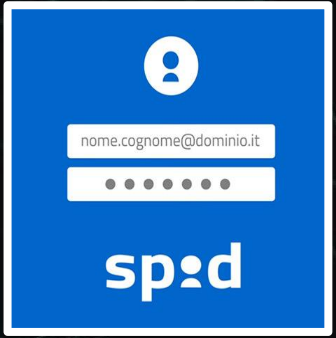
elliot.alderson

****

Log on to: E Corp-USA

How do I log on to another domain?





nome.cognome@dominio.it

●●●●●●●

What is an IdP??

An identity provider (abbreviated **IdP**) is a system entity that creates, maintains, and manages identity information for **principals.**

An identity provider offers **user authentication as a service.**

An identity provider is “**a trusted provider that lets you use single sign-on** (SSO) to access other websites or applications”.

Principals: A principal in computer security is an entity (people, computers, services, computational) that can be authenticated by a computer system or network.

Main IdP types

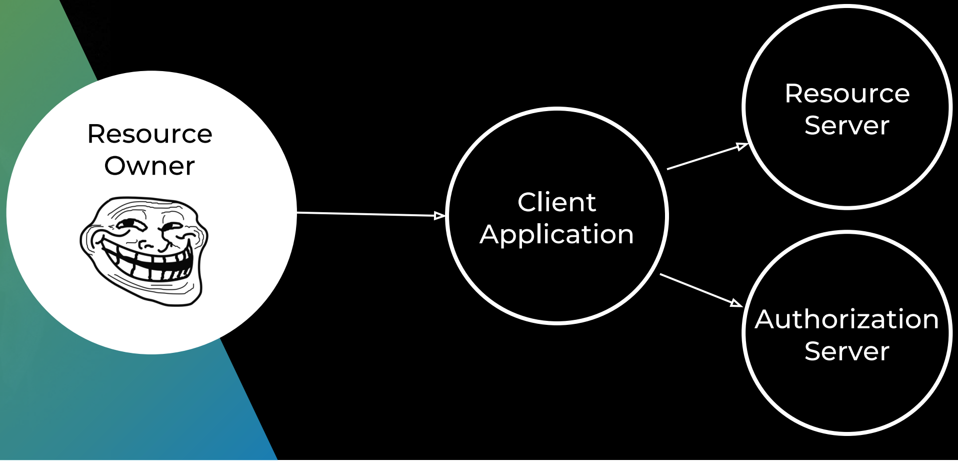
SAML v2.0
Security Assertion Markup Language

OAUTH 2

OpenID Connect

https://aaronparecki.com/oauth-2-simplified/
http://tutorials.jenkov.com/oauth2/index.html (use PKCE extension “pixy”)
http://openid.net/2015/05/26/enhancing-oauth-security-for-mobile-applications-with-pkse/
https://connect2id.com/learn/openid-connect

OAuth2 Roles



The Third-Party Application: "Client"

The client is the application that is attempting to get access to the user's account.
It needs to get permission from the user before it can do so.

The API: "Resource Server"

The resource server is the API server used to access the user's information.

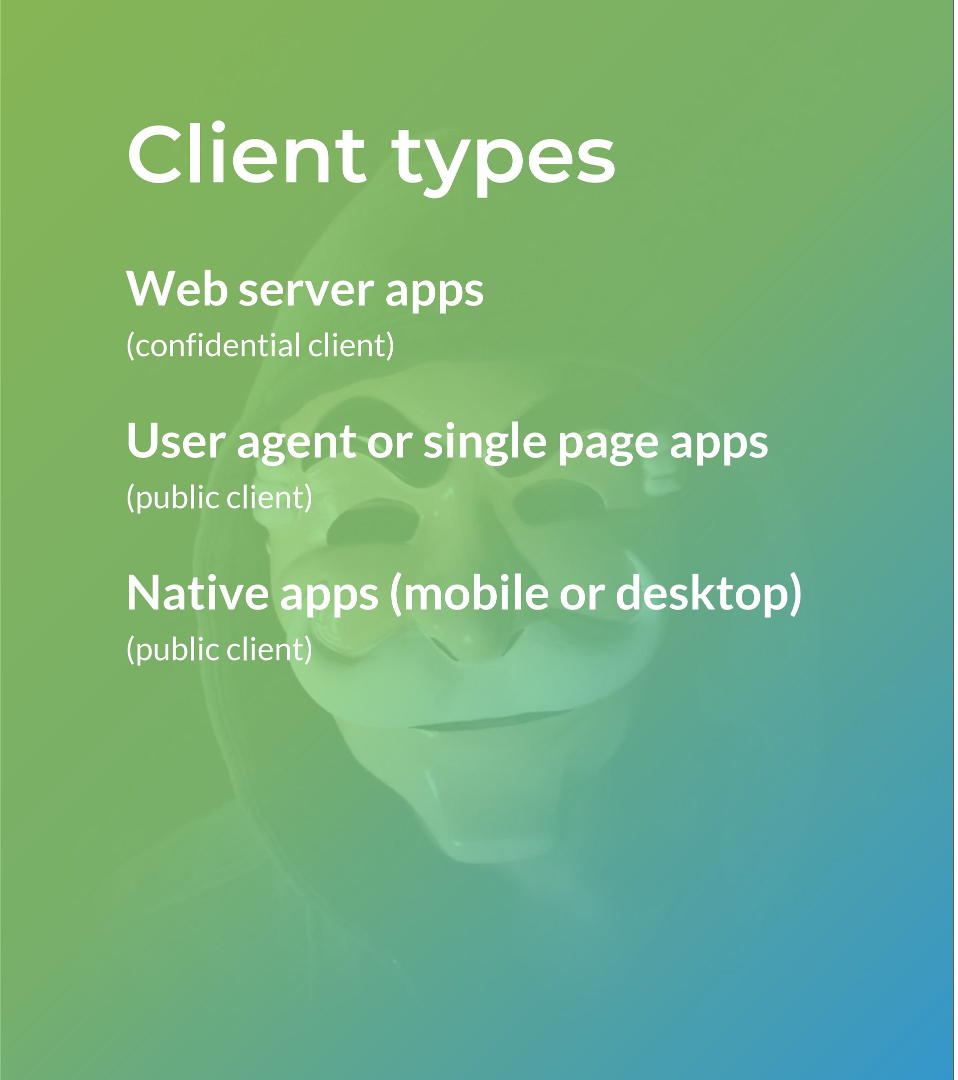
The Authorization Server

This is the server that presents the interface where the user approves or denies the request. In smaller implementations, this may be the same server as the API server, but larger scale deployments will often build this as a separate component.

The User: "Resource Owner"

The resource owner is the person who is giving access to some portion of their account.

* https://aaronparecki.com/oauth-2-simplified/



Client types

Web server apps

(confidential client)

User agent or single page apps

(public client)

Native apps (mobile or desktop)

(public client)

Auth grant types

Authorization Code

for apps running on a web server, browser-based and mobile apps

Password

for logging in with a username and password

Client credentials

for application access

Implicit

was previously recommended for clients without a secret, but has been superseded by using the Authorization Code grant with no secret.

Create an app

Client ID

The `client_id` is a public identifier for apps.

Client Secret

The `client_secret` is a secret known only to the application and the authorization server.

Redirect URI

`RedirectURL`(s) are a critical part of the OAuth flow. After a user successfully authorizes an application, the authorization server will redirect the user back to the application with either an authorization code or access token in the URL.

```
https://app.example.com/auth
myapp://callback
fb00000000://
```

`client_id` it's best that it isn't guessable by third parties

`client_secret` should be provided only depending on app client type

`redirect_uri` should be mandatory

`PKCE` (pixie, Proof Key for Code Exchange) should be used for public clients to mitigate the threat of having the authorization code intercepted - *RFC7636*

* *https://www.oauth.com/* | *https://www.oauth.com/oauth2-servers/pkce/*

Auth-code mode on web server apps

3

```
https://authorization-server.com/auth?response_type=code&client_id
=CLIENT_ID&redirect_uri=REDIRECT_URI&scope=email&state=1234zyx
```

4

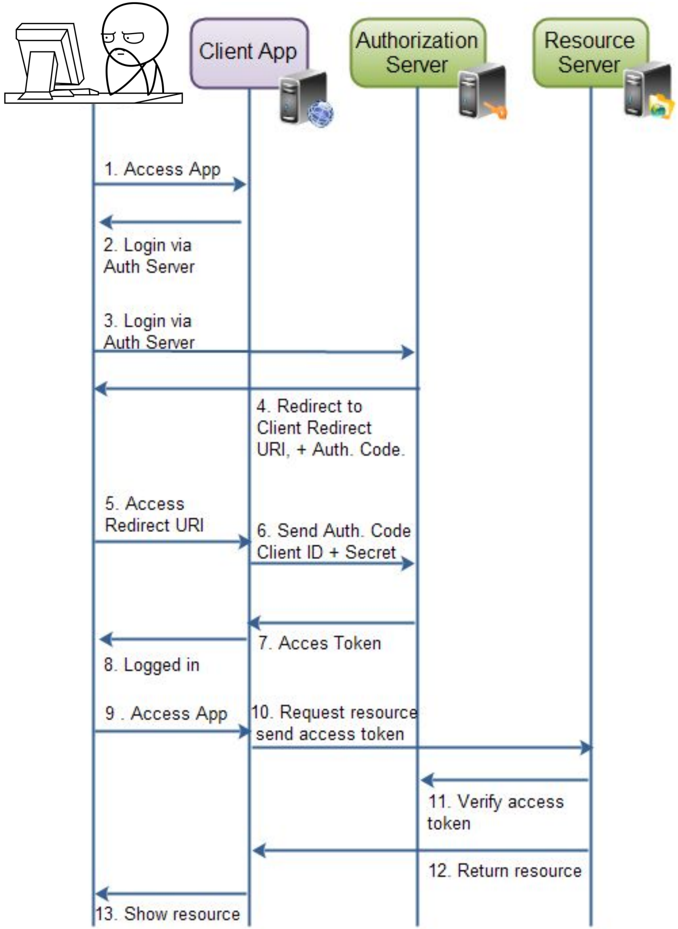
```
https://example-app.com/cb?code=AUTH_CODE_HERE&state=1234zyx
```

6

```
POST https://authorization-server.com/token
  grant_type=authorization_code&
  code=AUTH_CODE_HERE&
  redirect_uri=REDIRECT_URI&
  client_id=CLIENT_ID&
  client_secret=CLIENT_SECRET
```

7

```
{
  "access_token":"RsT5OjbzRn430zqMLgV3Ia",
  "expires_in":3600
}
```



Auth-code mode on mobile apps with PKCE

3

```
//native app|webbrowser|SFSafariViewController|CustomTabsService
fbauth2|https://authorization-server.com?response_type=code&clien
t_id=CLIENT_ID&
redirect_uri=REDIRECT_URI&scope=email&state=1234zyx&
code_challenge=XXXXXXX&
code_challenge_method=S256
```

4

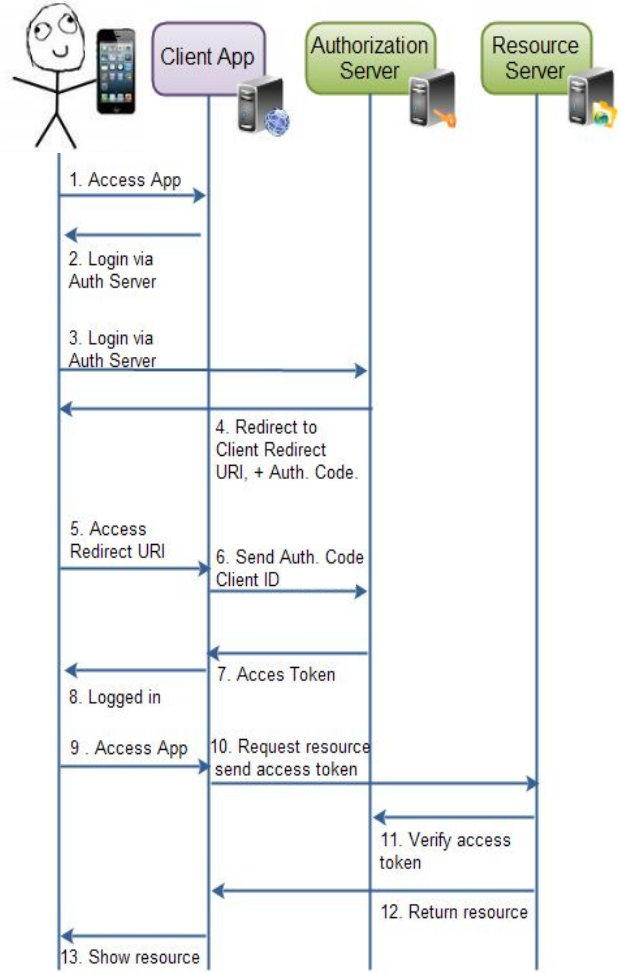
```
fb00000000://authorize?code=AUTH_CODE_HERE&state=1234zyx
```

6

```
POST https://authorization-server.com/token
  grant_type=authorization_code&
  code=AUTH_CODE_HERE&
  redirect_uri=REDIRECT_URI&
  client_id=CLIENT_ID&
  client_secret=&
  code_verifier=VERIFIER_STRING
```

7

```
{ "access_token":"RsT5OjbzRn430zqMLgV3Ia",
  "Expires_in":3600 }
```



Client App

Authorization Server

Resource Server

1. Access App

2. Login via Auth Server

3. Login via Auth Server

4. Redirect to Client Redirect URI, + Auth. Code.

5. Access Redirect URI

6. Send Auth. Code Client ID

7. Acces Token

8. Logged in

9 . Access App

10. Request resource send access token

11. Verify access token

12. Return resource

13. Show resource

* https://tools.ietf.org/html/draft-ietf-oauth-native-apps-12

OpenID Connect

OpenID Connect is a simple identity layer built on top of the **OAuth 2.0 protocol**.
OpenID Connect implements authentication as an extension to the **OAuth 2.0 authorization process.**

3

```
https://authorization-server.com?response_type=code&client_id=CLIENT_ID&
state=1234zyx&
redirect_uri=REDIRECT_URI&
scope=openid
```

6

```
POST https://authorization-server.com/token
  grant_type=authorization_code&
  code=AUTH_CODE_HERE&redirect_uri=REDIRECT_URI&
  client_id=CLIENT_ID&client_secret=CLIENT_SECRET
```

7

```
{"id_token":
"eyJhbGciOiJSI6IjFlOWdkazcifQ.ewwbGUuIiOiAiMjJ1zZwgjxqGByKHiOtX7TpdVq026
g6EJbOEoRoSLl0nx7RkKU8NXvKMzqq.026g6I6IjFlOWdkEJbOiMjJ1zZwgjxqGBy",
"access_token": "SlAV32hkKG",
"token_type": "Bearer",
"expires_in": 3600}
```

Advantages

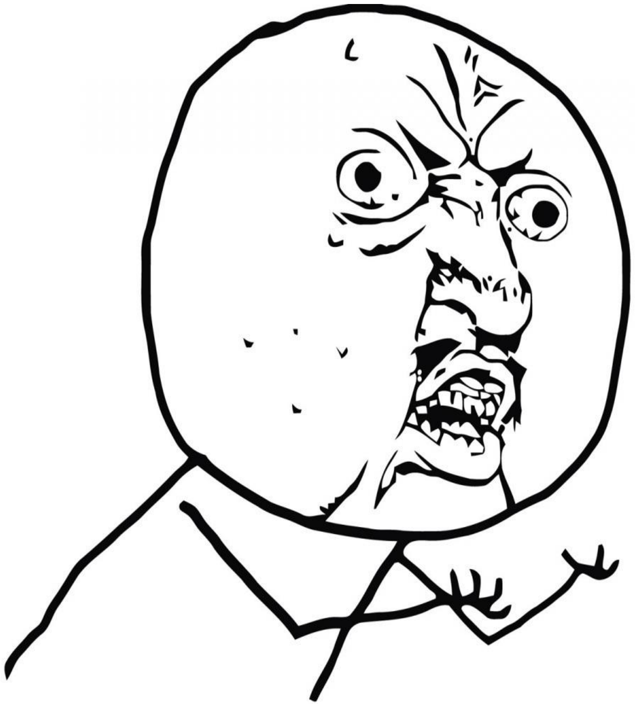
- Easy to consume identity tokens
- Based on the OAuth 2.0

ID_token uses

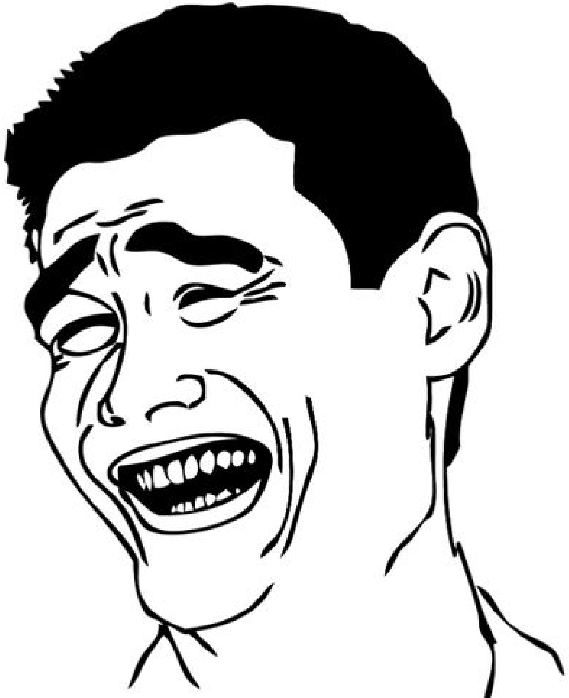
- Stateless sessions
- Passing identity to 3rd parties

* https://connect2id.com/learn/openid-connect

“So what’s next??”



“No more passwords”



UNIFYID

UnifyID combines implicit authentication
with machine learning to uniquely identify.
The service can authenticate you based on unique factors
like the way you walk, type, and sit.

Be authentic, be yourself.

https://unify.id/

someone.id

- no more passwords and sign-up
- as secure as your email is
- multiple auth channels -> 1 digital identity
  (including Telegram, Messenger, Skype, Slack, mobile app)
- user privacy: no user's data collected (email only)
- nothing to install, you are your ID
  (several apps already know your ID)
- sniffer proof

https://someone.id

https://github.com/someoneid