

À la découverte des APIs

La tech amiénoise, Juin 2017

ITERACODE





Louis VIART
Développeur

ITERACODE

API everywhere

♥ ponyfac.es

 Microsoft Azure

MailChimp

facebook for developers 

stripe



Google Maps



Messenger

amazon
web services

API everywhere



Internet of things

Nouveaux Business models

Open Innovation / Open Data

ATAWAD (AnyTime, AnyWhere, AnyDevice)

Qu'est ce que c'est ?

API : Application Programming Interface



Qu'est ce que c'est ?

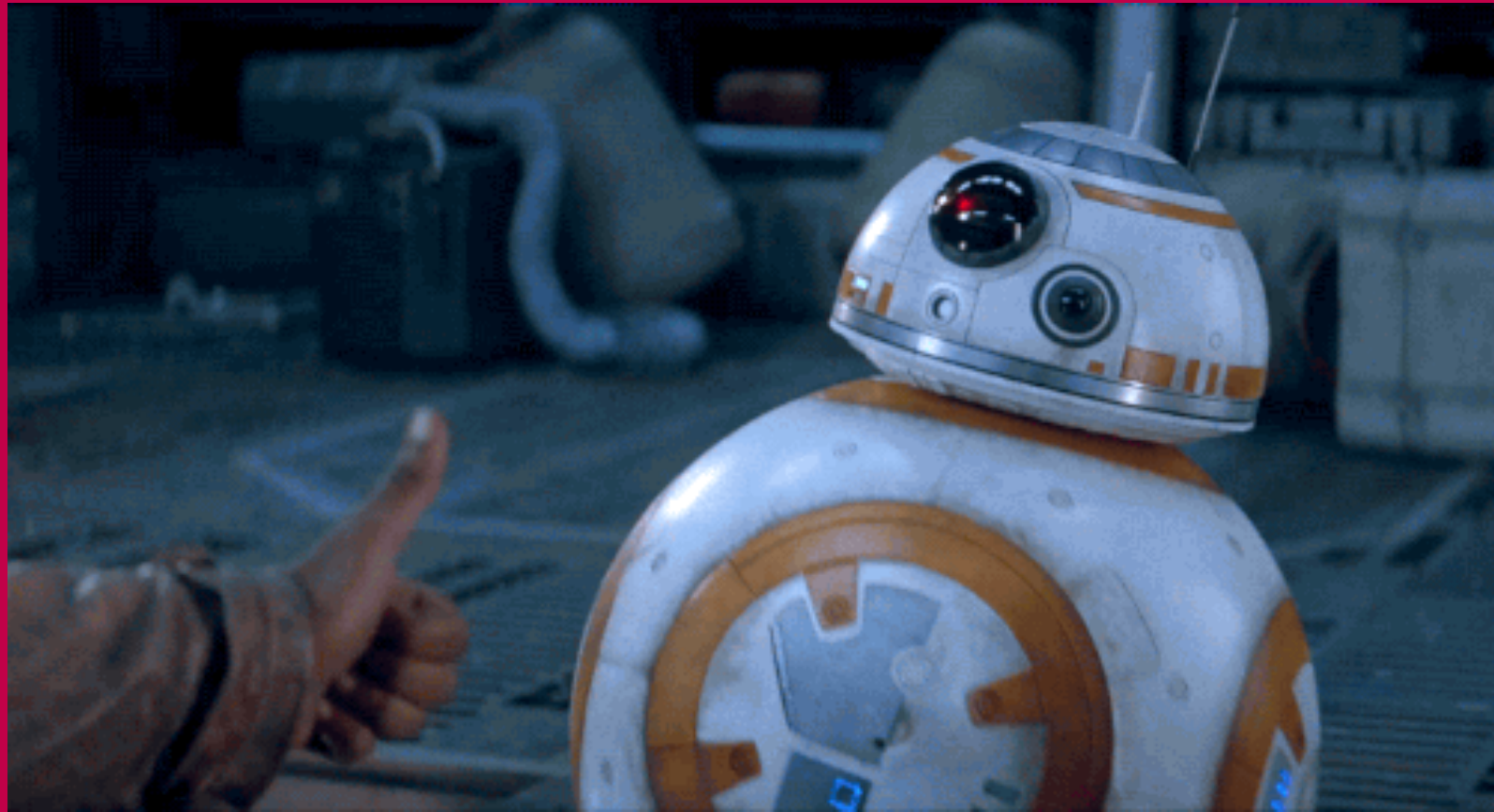
👉 Interface



Qu'est ce que c'est ?

API : Application Programming Interface

- Façade
- Propose des fonctionnalités
- Documentée



REST

Representational State Transfer



Roy Fielding

REST

Representational State Transfer

Architecture basée sur HTTP

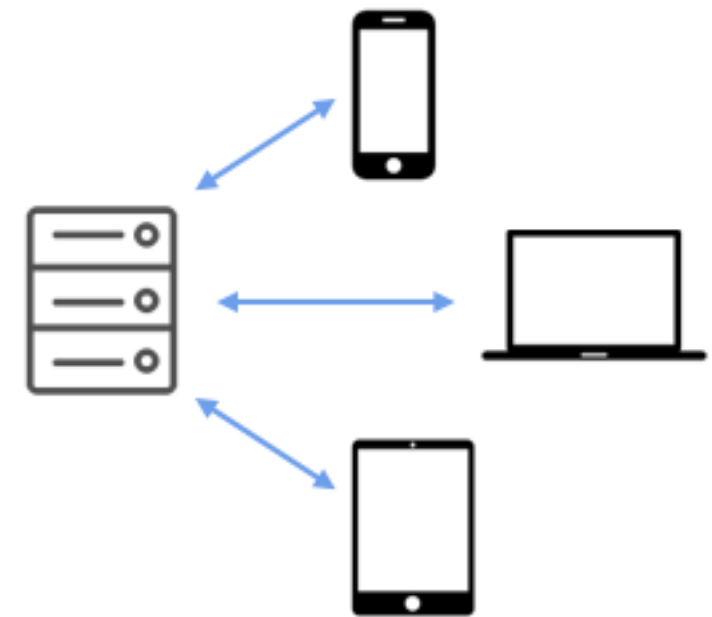
REST

Respect de 5 critères

- Client - serveur
- Sans état
- Mise en cache
- Interface uniforme
- Hiérarchisé par couche

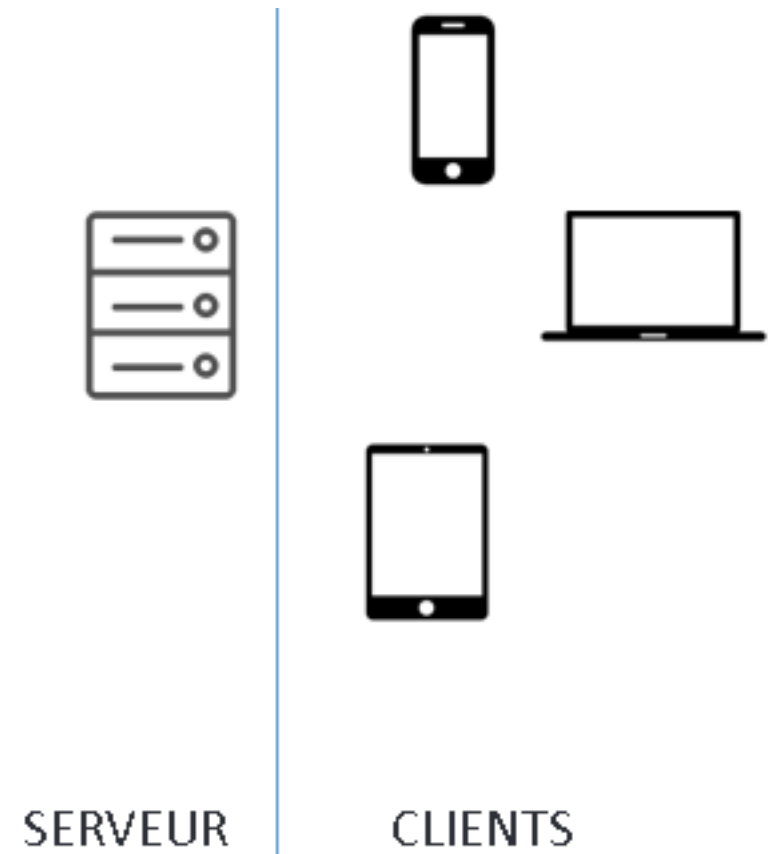
REST

Client - Serveur



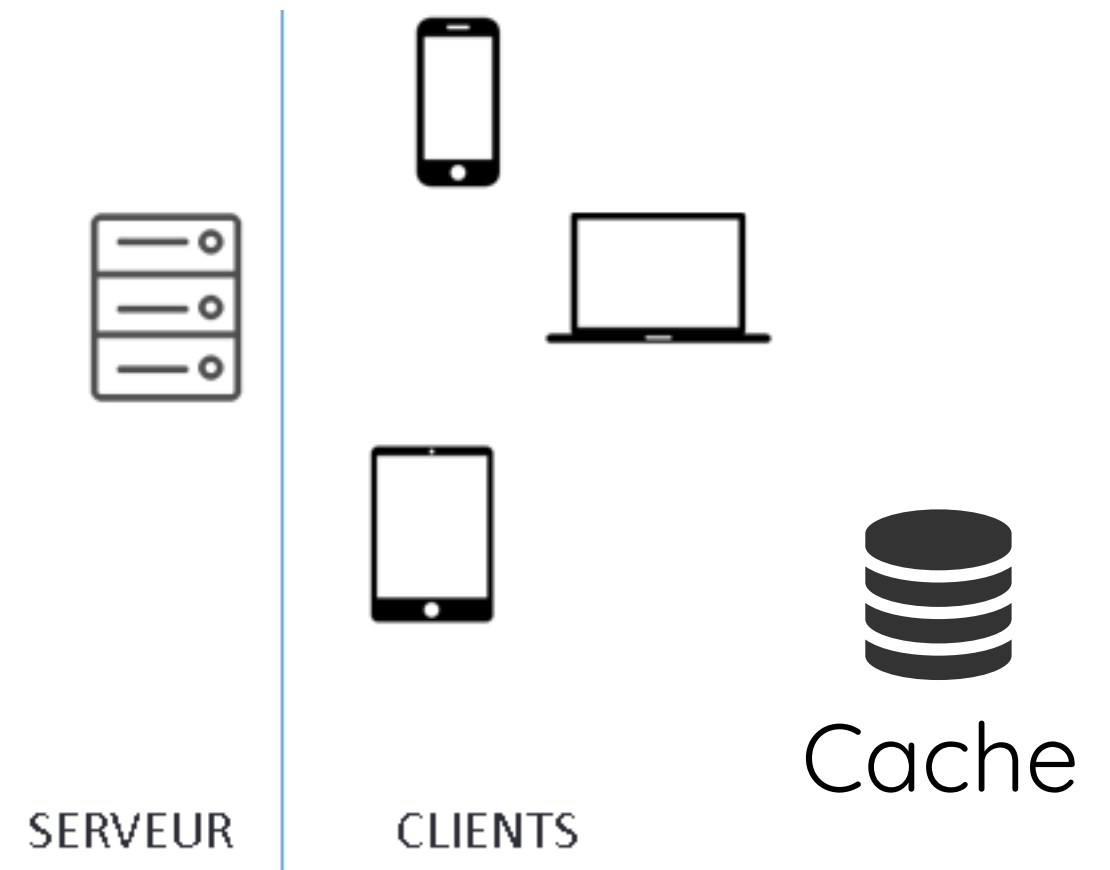
REST

Sans état



REST

Mise en cache



REST

Interface Uniforme

- Identification des ressources
- Représentation définie
- Message auto-descriptif
- Hypermedia

REST

Interface Uniforme

- **Identification des ressources**
- Représentation définie
- Message auto-descriptif
- Hypermedia

<code>delete</code>	<code>DELETE /playlists</code>
<code>list</code>	<code>GET /playlists</code>
<code>insert</code>	<code>POST /playlists</code>
<code>update</code>	<code>PUT /playlists</code>

REST

Interface Uniforme

- Identification des ressources
- **Représentation définie**
- Message auto-descriptif
- Hypermedia

```
{
  "kind": "youtube#comment",
  "etag": etag,
  "id": string,
  "snippet": {
    "authorDisplayName": string,
    "authorProfileImageUrl": string,
    "authorChannelUrl": string,
    "authorChannelId": {
      "value": string
    },
    "channelId": string,
    "videoId": string,
    "textDisplay": string,
    "textOriginal": string,
    "parentId": string,
    "canRate": boolean,
    "viewerRating": string,
    "likeCount": unsigned integer,
    "moderationStatus": string,
    "publishedAt": datetime,
    "updatedAt": datetime
  }
}
```

REST

Interface Uniforme

- Identification des ressources
- Représentation définie
- **Message auto-descriptif**
- Hypermedia

```
HTTP/1.0 200 OK
Content-Type: application/json;charset=utf-8
{
  "films": "http://swapi.co/api/films/",
  "people": "http://swapi.co/api/people/",
  "planets": "http://swapi.co/api/planets/",
  "species": "http://swapi.co/api/species/",
  "starships": "http://swapi.co/api/starships/",
  "vehicles": "http://swapi.co/api/vehicles/"
}
```


REST

Interface Uniforme

- Identification des ressources
- Représentation définie
- Message auto-descriptif
- **Hypermedia**

```
{
  "links": {
    "self": "http://example.com/articles",
    "next": "http://example.com/articles?page[offset]=2",
    "last": "http://example.com/articles?page[offset]=10"
  },
  "data": [{
    "type": "articles",
    "id": "1",
    "attributes": {
      "title": "JSON API paints my bikeshed!"
    },
    "relationships": {
      "author": {
        "links": {
          "self": "http://example.com/articles/1/relationships/author",
          "related": "http://example.com/articles/1/author"
        },
        "data": { "type": "people", "id": "9" }
      },
      "comments": {
        "links": {
          "self": "http://example.com/articles/1/relationships/comments",
          "related": "http://example.com/articles/1/comments"
        },
        "data": [
          { "type": "comments", "id": "5" },
          { "type": "comments", "id": "12" }
        ]
      }
    }
  }],
  "links": {
    "self": "http://example.com/articles/1"
  }
}
```

REST

Hiérarchisé par couche

create	POST	/files/ <i>fileId</i> /comments/ <i>commentId</i> /replies
delete	DELETE	/files/ <i>fileId</i> /comments/ <i>commentId</i> /replies/ <i>replyId</i>
get	GET	/files/ <i>fileId</i> /comments/ <i>commentId</i> /replies/ <i>replyId</i>
list	GET	/files/ <i>fileId</i> /comments/ <i>commentId</i> /replies
update	PATCH	/files/ <i>fileId</i> /comments/ <i>commentId</i> /replies/ <i>replyId</i>

REST

Verbes HTTP (en théorie)

GET

POST

PUT

PATCH

DELETE

REST

Verbes HTTP (dans la vraie vie)

/comments/commentId

GET

POST

PATCH

DELETE

REST

Verbes HTTP (dans la vraie vie)

/comments

GET

POST

PATCH

DELETE

REST

Verbes HTTP (dans la vraie vie)

/comments/commentId

GET

POST

PATCH

DELETE

REST

Verbes HTTP (dans la vraie vie)

/comments

GET

POST

PATCH

DELETE

REST

Verbes HTTP (dans la vraie vie)

/comments/commentId

GET

POST

PATCH

DELETE

REST

Verbes HTTP (dans la vraie vie)

/comments/commentId

GET

POST

PATCH

DELETE

Authentication



When you ask the landlord to install a lock on your door

JSON Web Token

- Standard ouvert (RFC7519)
- Échanger des informations
- JSON object



JWT

Comment ça marche ?



JWT

Comment ça marche ?

Structure d'un JWT :

XXXX.YYYY.ZZZZ

Header

Payload

Signature

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

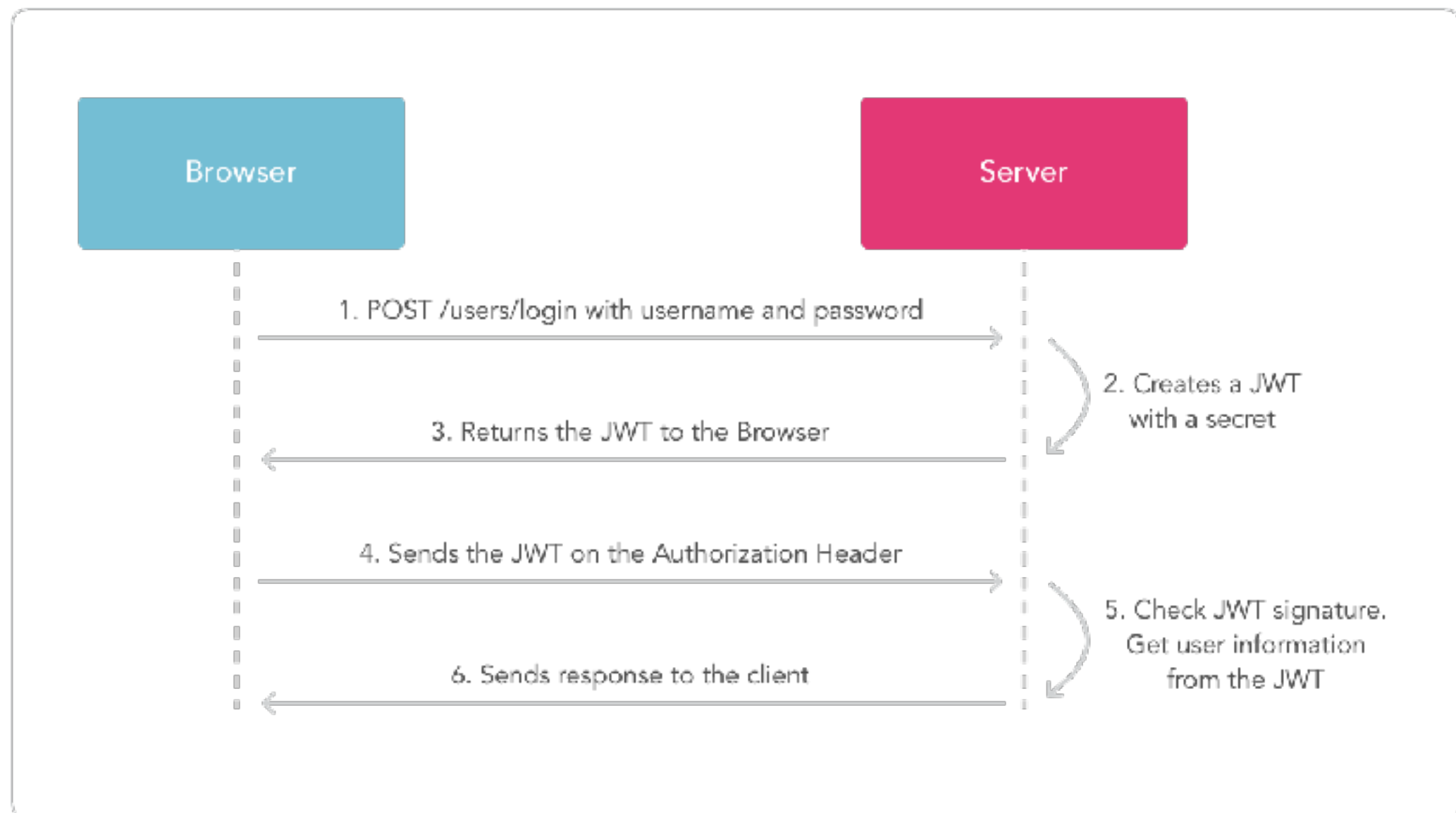
```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "admin": true  
}
```

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  secret)
```

JWT

Comment ça marche ?

Étapes de l'authentification :



Un peu de code ?



Merci !

@Petrucheqa