Conception de logiciels Adaptés
Architecture d'une Application Mobile

Sébastien Mosser
INF600G - E20 - Séquence 2 - Partie 2

UQÀM | Département d'informatique

ace

1 Rétrospective L1

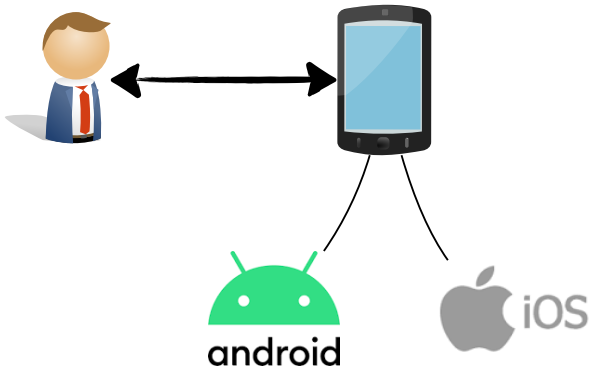
2 Architecture App. Mobile

3 Exposition de ressources (REST)

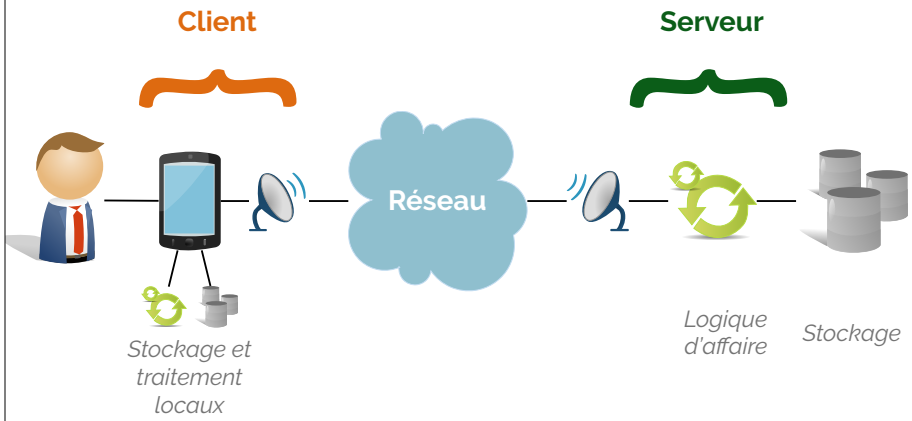
4 Intro. à Android

5 Travail à faire pour L2

Quels sont les constituants d'une app ?



Aperçu Général

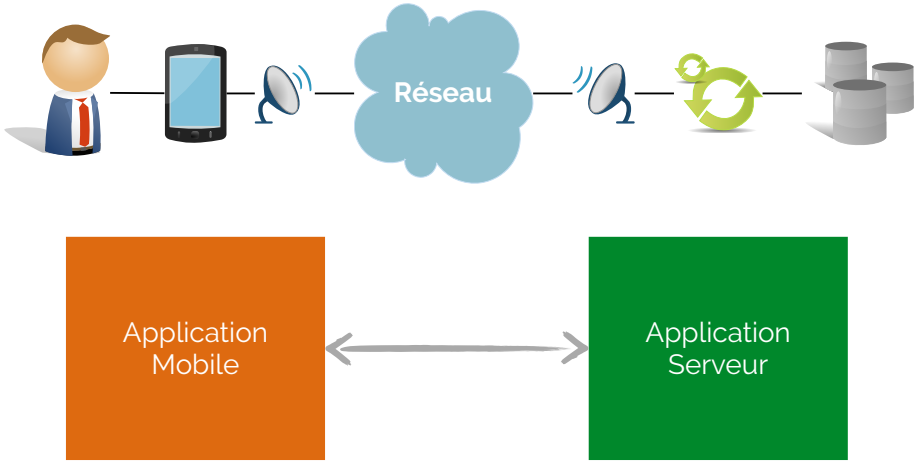


Grandes problématiques

+ Sécurité

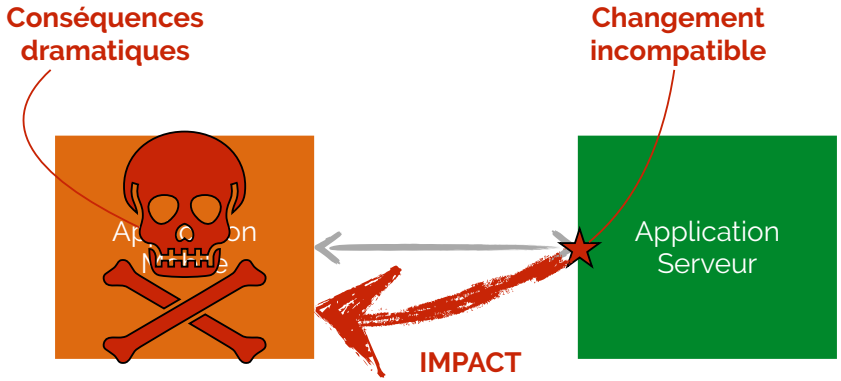
- Client (Téléphone, tablette)
  - Consommation énergétique, capacité de calcul
  - Variabilité du matériel
- Réseau
  - Lenteur, déconnexion, synchronisation
  - Transfert des données (p.ex., JSON, XML, …)
- Serveur
  - Passage à l'échelle ("*scalability*")
  - Exposition des API (souvent du REST, mais pas que)

Développement Monolithique



Problématique de maintenance

Généralement, l'impact d'un changement est transitif



Le Web, cette Jungle ...

API Name	Versions	# of Elements in Latter Version	# of Elements Changed	Proportion(%)
Twitter	v1-v1.1	109	51	47
Blogger	v1-v2	12	5	29
Blogger	v2-v3	33	33	100
Bitly API	v2-v3	74	74	100
MusicBrainz	v1-v2	36	36	100
Friendfeed	v1-v2	23	17	74
Tumblr	v1-v2	21	21	100
Sunlight Congress	v1-v2	12	12	100
OpenStreetMap	v0.3-v0.4	23	12	52
OpenStreetMap	v0.4-v0.5	35	20	57
OpenStreetMap	v0.5-v0.6	52	51	91
Groupon	v1-v2	8	8	100
Yelp	v1-v2	2	2	100
New York Times Article Search	v1-v2	1	1	100
Average				82

How Do Developers React to RESTful API Evolution? (Wang et al, ICSOC'14)

Types de changements identifiés [1/3]

Change Type	Explanation
Change Response Format	1) Entire Format Change: *e.g.*, from XML to JSON 2) Structure Change: add, remove or reorganize XML tags 3) Slight Modification: change XML tag or attribute name *e.g.*, OpenStreetMap API conducted practices 2) and 3).
Change Resource URL	Replace the old version number with new one in URLs *e.g.*, The domain name of Twitter changed from api.twitter.com/1 to api.twitter.com/1.1.
Change Authentication Model	Update existing authN model with new one *e.g.*, Twitter API v1.1 requires every request to be authenticated and client applications must use OAuth.
Change Rate Limit	1) Change Limit Window: change the length of window 2) New Headers and Resp Codes: update messages showing limit exceeded or status
Delete Response Format	Unsupport a format: *e.g.*, XML is not supported in Twitter API v1.1.
Add Response Format	Support a new format in new version: *e.g.*, NYT Article Search API added JSONP in Version 2.
Add Authentication Model	Support a new model but keep old ones: *e.g.*, MusicBrainz and Blogger API added more models.

How Do Developers React to RESTful API Evolution? (Wang et al, ICSOC'14)

Types de changements identifiés [2/3]

Change Type	Explanation
Change Method Name	*e.g.*, We observed this practice from Twitter, Blogger MusicBranz, FriendFeed, Yelp and NYT Article Search.
Change Response Format	The return format of a method can be changed, such as returning more values *e.g.*, Twitter method "GET friendships/lookup"
Change Rate Limit	A limit is usually set up on the number of data units can be retrieved per request. The rate limit can be changed.
Change Authentication Model	Different authentication models are set up for different methods. *e.g.*, to protect critical data, they update the authN model on methods modifying databases. *e.g.*, OpenStreetMap and MusicBrainz practiced this.
Change Domain URL	It is different from "Resource URL change" on, API level, because it is only applicable to very few methods. *e.g.*, the domain name of Twitter method "POST statuses/update_with_media" is changed from upload.twitter to api.twitter.com.
Delete Method	Unsupported methods in new version: *e.g.*, we observe every API practiced this, except for Blogger (v1 to v2), Yelp and NYT search
Add Method	Support new methods: *e.g.*, we observe every API practiced this, except for Blogger (v1 to v2) and NYT search
Add Error Code	Add more error codes to specific methods: *e.g.*, Twitter Blogger and OpenStreetMap.

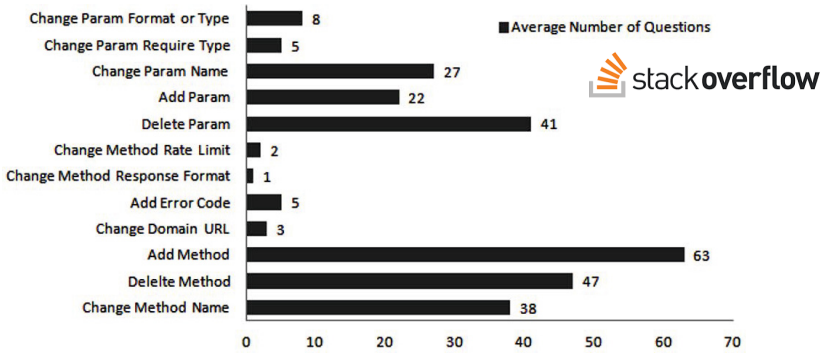
How Do Developers React to RESTful API Evolution? (Wang et al, ICSOC'14)

Types de changements identifiés [3/3]

Change Type	Explanation
Change	Rename parameters with a self-explanatory names
Change Format or Type	The return format of using a parameter can be changed. NYT Article Search practiced.
Change Rate Limit	The limit can be raised up or reduced. *e.g.*, OpenStreetMap raised up a limit.
Change Require Type	*e.g.*, require type of "cursor" of "GET friends/ids" is changed from optional to semi-optional.
Delete Parameter	Unsupported some functionalities of a method
Add Parameter	Support new functionalities of a method

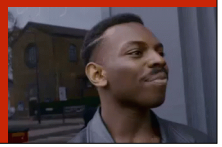
How Do Developers React to RESTful API Evolution? (Wang et al, ICSOC'14)

Fréquence des changements

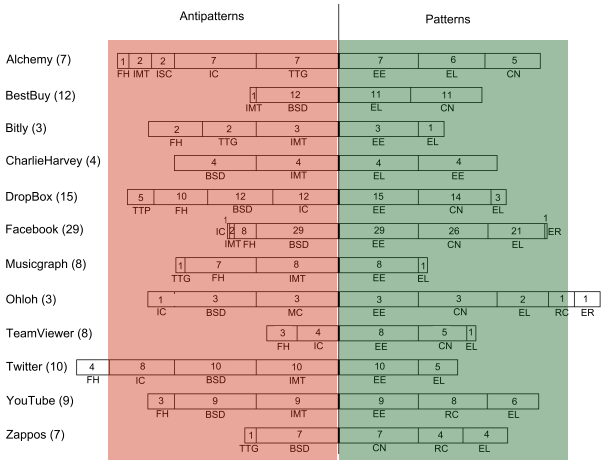


How Do Developers React to RESTful API Evolution? (Wang et al, ICSOC'14)

Il suffit de faire juste du premier coup !

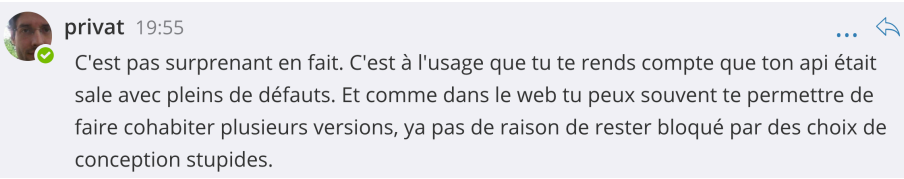


C'est difficile en fait ...



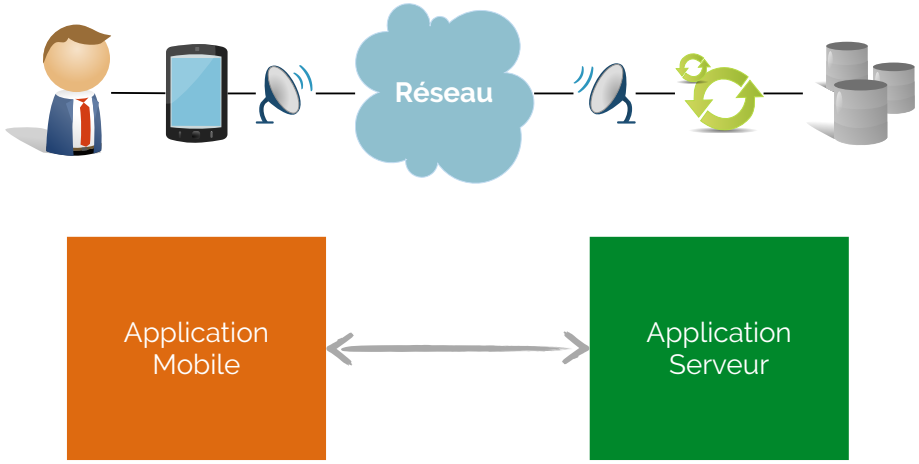
Are REST APIs for Cloud Computing Well-Designed? An Exploratory Study (Palma et al., ICSOC 2016)

L'évolution des APIs est un phénomène naturel

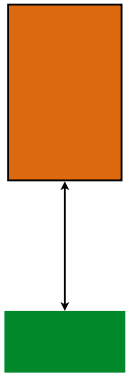


Il faut faire attention à se protéger pour qu'un changement unilatéral de la part du fournisseur d'API ne mette pas en péril votre propre application

Développement en couche à la rescousse



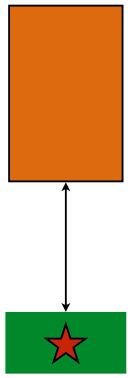
Exemple : Renommage de méthode



```
public void displayMessages(Display d) {
  JSONArray data = Http.get("http://..../getMessages");
  for(JSONObject jobj: data.getContents()) {
    Widget w = new Widget();
    w.setTitle(jobj.readString("msgTitle"));
    w.setContent(jobj.readString("msgData");
    d.add(w);
  }
  d.render();
}
```

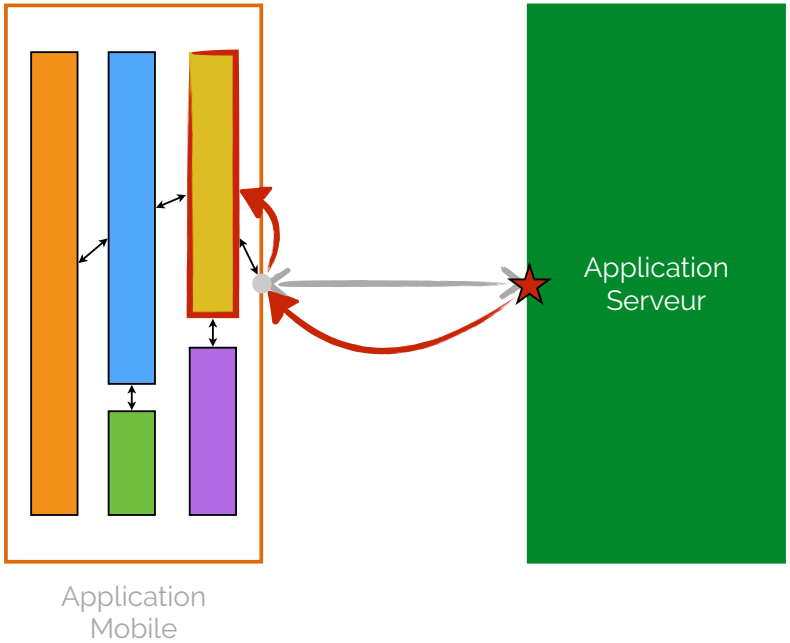
http://..../getMessages

Exemple : Renommage de méthode

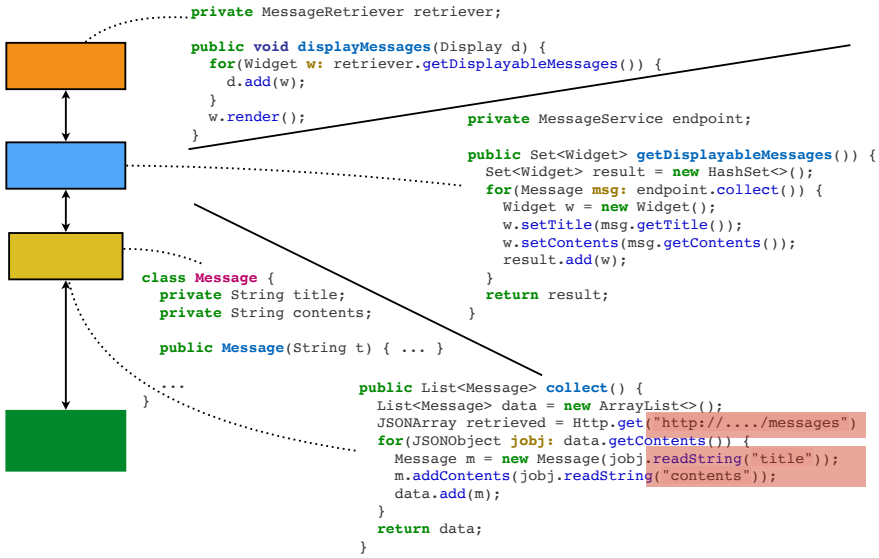


```
public void displayMessages(Display d) {
  JSONArray data = Http.get("http://..../messages");
  for(JSONObject jobj: data.getContents()) {
    Widget w = new Widget();
    w.setTitle(jobj.readString("title"));
    w.setContent(jobj.readString("contents");
    d.add(w);
  }
  d.render();
}
```

http://..../messages



Exemple : Renommage de méthode

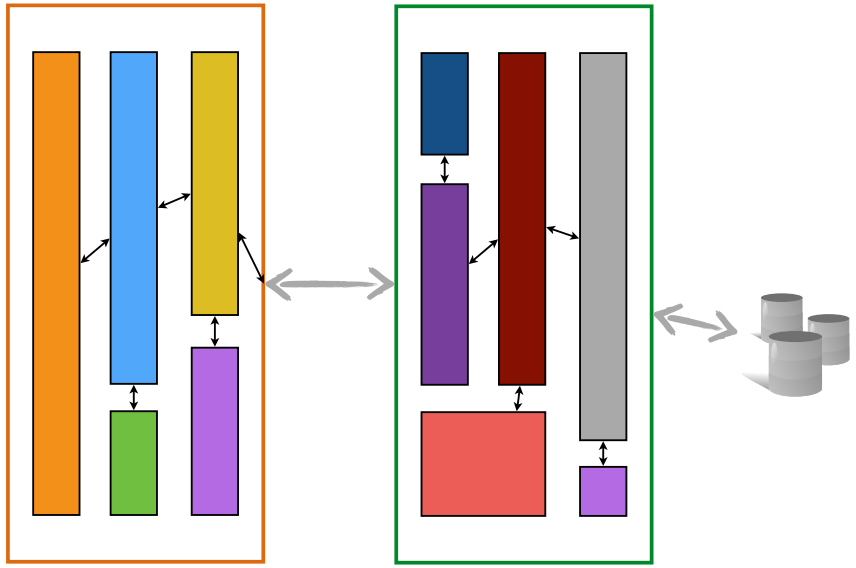


Le ticket d'entrée est plus cher

- **Identifier les différentes couches**
  - En théorie, on parle d'architecture à N couches
  - En pratique, on a souvent N=3
    - Accès aux données, Logique d'affaire et Exposition
- **Définir les abstractions** entre les différentes couches
  - Attention à ne pas faire d'*over-engineering*
    - *Il faut accepter qu'on se trompe souvent au démarrage*
- *Sur le long terme, il est crucial de mettre en place cette modularité*

Du patron architectural à la réalisation

- Dans une Interface Personne-Machine :
  - On va parler de "Modèle - Vue - Controleur" (MVC)
    - Ou de Modèle - Vue - VueModèle (MVVM)
- Dans une partie arrière :
  - On va parler d'architectures 3-tiers :
    - Presentation, Logic and Data tiers
  - Attention, un tiers est au sens de partie, pas de 1/3



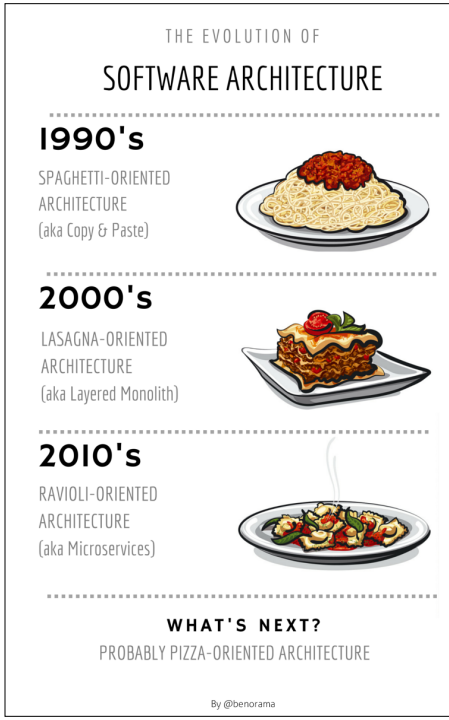
Application Mobile

Application Serveur

Stockage

Digression : Architectures Micro-services

- Si vous avez entendu parler de "**micro-services**"
  - TL;DR: Un micro-service est une architecture 3-tiers autonome
- Plutôt que de faire 3 couches qui contiennent "X" domaines d'affaire
  - **On fait "X" micro-services** qui contiennent chacun les 3 couches (On parle d'architectures hexagonales)
- **Les modes passent, mais les concepts restent**
  - C'est pour ça qu'on n'enseignent pas de "technos"



Venez à la maîtrise si le sujet vous intéresse !

ace