

conception/programmation d'applications & services web - Serveur

gérôme canals
gerome.canals@univ-lorraine.fr

Objectifs du module

- comprendre la notion de service / micro-service et les principes REST
- concevoir et architecturer le backend d'une application web moderne basée sur des services et concevoir les apis REST
- développer les services et implanter les api
- produire du code validé et conforme aux bonnes pratiques en php

organisation et évaluation

- séances cours/tds/tps + projet
- 1 projet fil rouge : les td contribuent à ce projet
- évaluation :
 - 1 projet fil rouge : le bon sandwich
 - 1 projet avec évaluation individuelle
 - Projet fil rouge pour les alternants
 - Projet atelier pour les non alternants

contenu

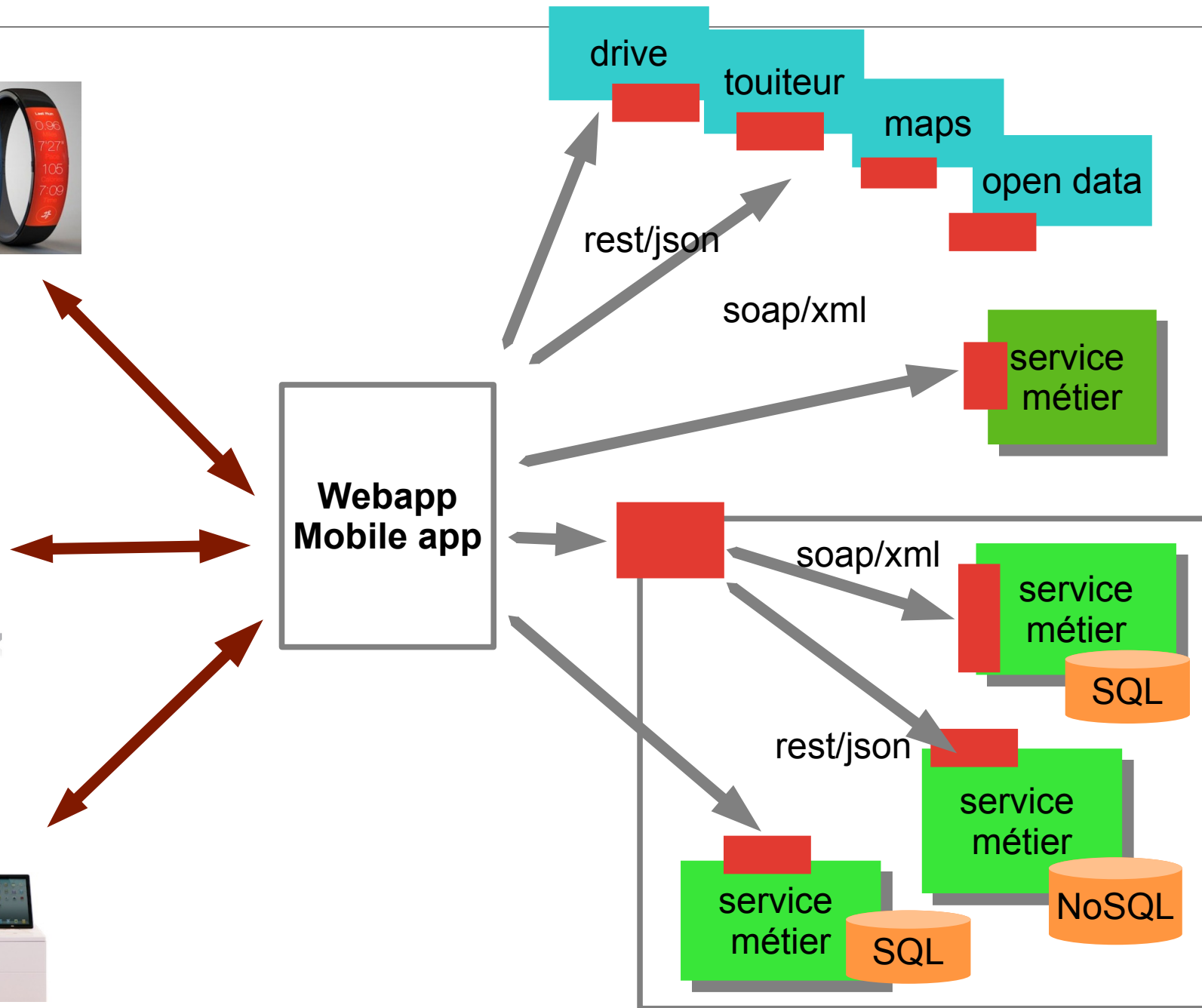
- Utiliser un micro-framework (php ou node) pour développer une application web
- Concevoir et implanter un service web exposant une api conforme aux principes REST
- Organiser et déployer un backend web dans une architecture à base de services et micro-services
- Gérer les données, les erreurs, la communication entre services et garantir la sécurité

contexte

le web est devenu une des plateformes applicatives majeures :

- systèmes d'information de grande taille
- accès universel, médias multiples
 - mobilité, systèmes embarqués, objets connectés
- software as a service (SaaS), déploiement dans le cloud :
 - applications à disposition à distance au travers d'une interface web
- très grande échelle

Contexte : les nouveaux besoins



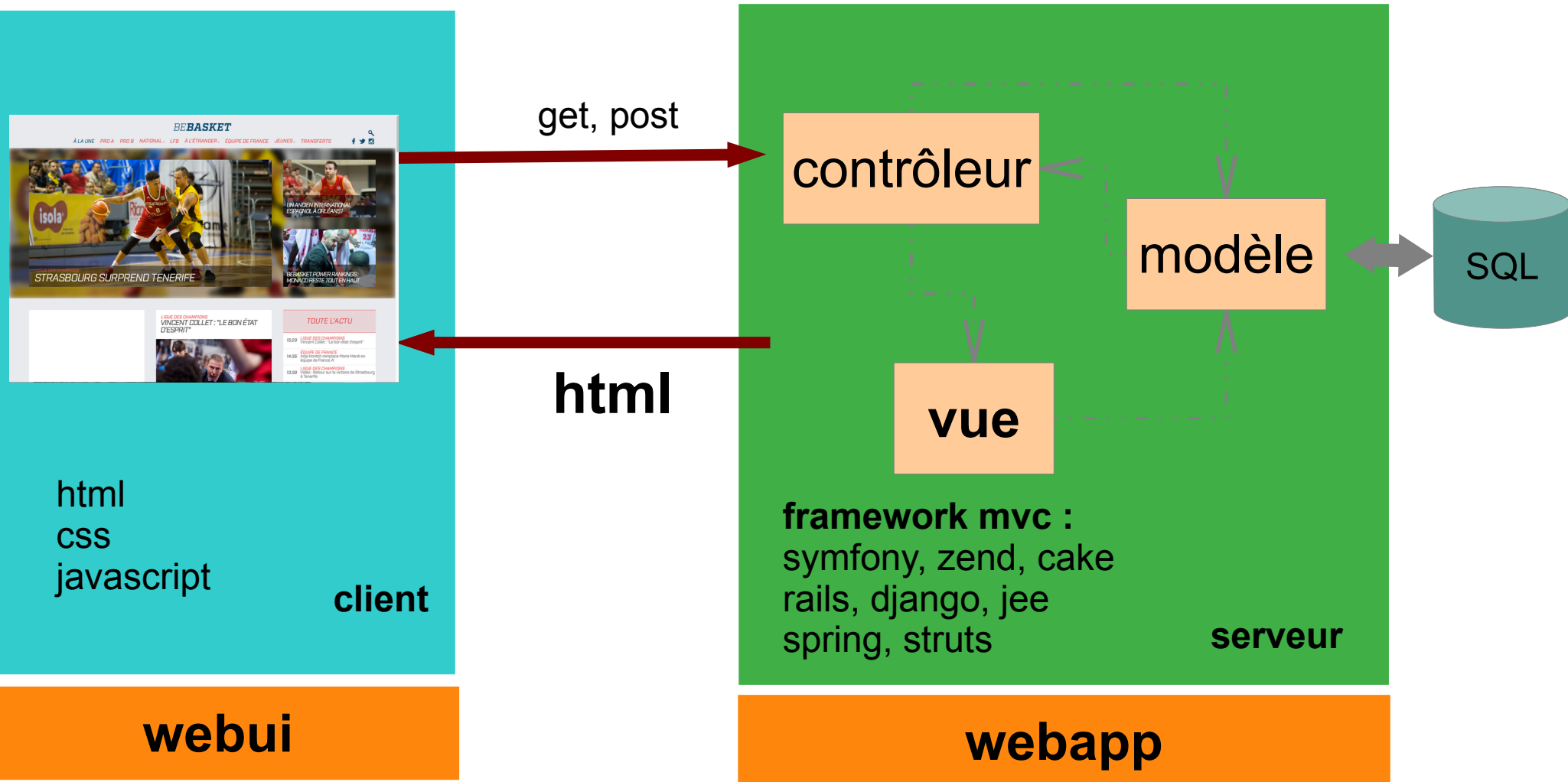
contexte

Sur le plan technique :

- webapp sur le client, single-page apps
- agrégation de services web ouverts : stockage, communication, maps, commentaires ...
- technos web, web mobile, web standalone

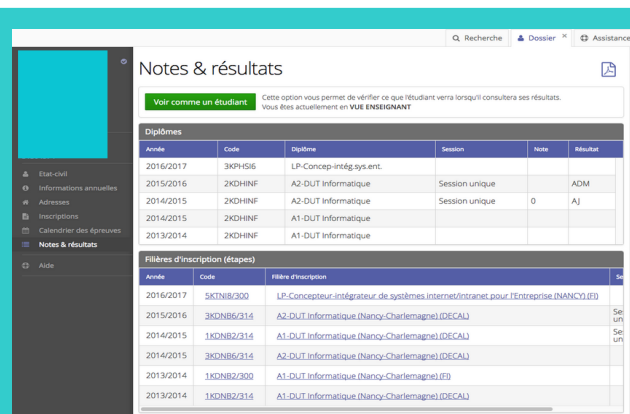
architecture classique mvc *server-side rendering (ssr)*

L'interface est *générée* par le serveur, *exécutée* sur le client



les nouvelles architectures *single-page application (SPA)*

L'interface est **générée et exécutée sur le client**
le serveur réalise des **services** et retourne des données en format neutre, la webapp est une Single-page application (SPA)



Notes & résultats

Voir comme un étudiant

Cette option vous permet de vérifier ce que l'étudiant verra lorsqu'il consultera ses résultats.
Vous êtes actuellement en VUE ENSEIGNANT

Année	Code	Diplôme	Session	Note	Résultat
2016/2017	3KPHS16	LP-Concept-intég.sys.ent.			
2015/2016	2KDHN16	A2-DUT Informatique	Session unique		ADM
2014/2015	2KDHN15	A2-DUT Informatique	Session unique	0	AJ
2014/2015	2KDHN15	A1-DUT Informatique			
2013/2014	2KDHN14	A1-DUT Informatique			

Année	Code	Filière d'inscription
2016/2017	5KTNB/300	LP-Concepteur-intégrateur de systèmes internet/intranet pour l'entreprise (NANCY IUT)
2015/2016	3KDNB/214	A2-DUT Informatique (Nancy-Charlemagne) (DECAL)
2014/2015	1KDNB/214	A1-DUT Informatique (Nancy-Charlemagne) (DECAL)
2014/2015	3KDNB/214	A2-DUT Informatique (Nancy-Charlemagne) (DECAL)
2013/2014	1KDNB/214	A1-DUT Informatique (Nancy-Charlemagne) (DECAL)
2013/2014	1KDNB/214	A1-DUT Informatique (Nancy-Charlemagne) (DECAL)

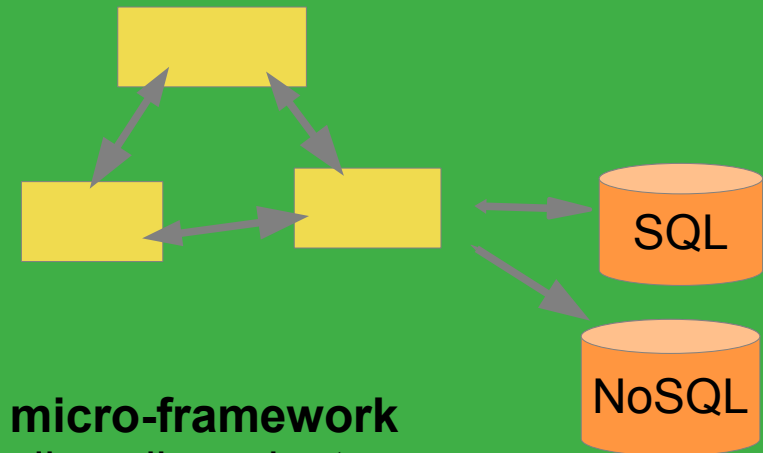
html, css, javascript
mvc js : Vue, Angular, React
framework UI : jQuery, zepto, Yui
plateforme mobile : cordova, ionic

webapp

REST, Soap

**json, xml,
html**

**A
P
I**



micro-framework
slim, silex, sinatra, express
+ composants
orm, auth, log ...

backend

API & Services

une API : une **interface** pour accéder à un **service backend** par **programme** et récupérer des **données** utilisables et non pas des pages web

- déporter l'application et son interface vers le client
- Le backend expose des services qui sont utilisables par différentes applications (web et mobile front-office, backoffice, B2B ...)
- éventuellement en complément d'une app. classique, pour rendre cette application **programmable**
 - fb, twitter, flickr, dropbox, github, google-* ...
- Interface facile d'emploi et d'utilisation par des **programmeurs**
- Retourne des données dans un format neutre exploitable par n'importe quelle application :
 - **xml, json**