



# Architecture @ MyCoach

Clément Agarini

# **Clément Agarini**

Tech lead Backend & Infrastructure.



@imjustclem



Clément Agarini

# My Coach Sport, créateur de solutions numériques

- Q Société fondée en **2011**.
- Q Applications pour **les coachs** et **les pratiquants**.
- Q Sport **loisir, amateur et professionnel**.
- Q Partenaire de **6 fédérations françaises**.



# **Notre objectif**

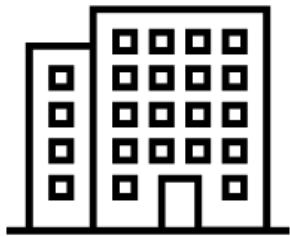
Accompagner les acteurs du sport  
dans leur transformation digitale.



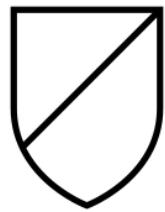
1.

## MY COACH SPORT

# Nos clients



Fédérations



Clubs



Coachs



Grand Public

## Le modèle My Coach Sport

- Q Éditeur de logiciel en mode SAAS
- Q Co-conception avec
  - Les fédérations
  - Les clubs et centres de formation



2.

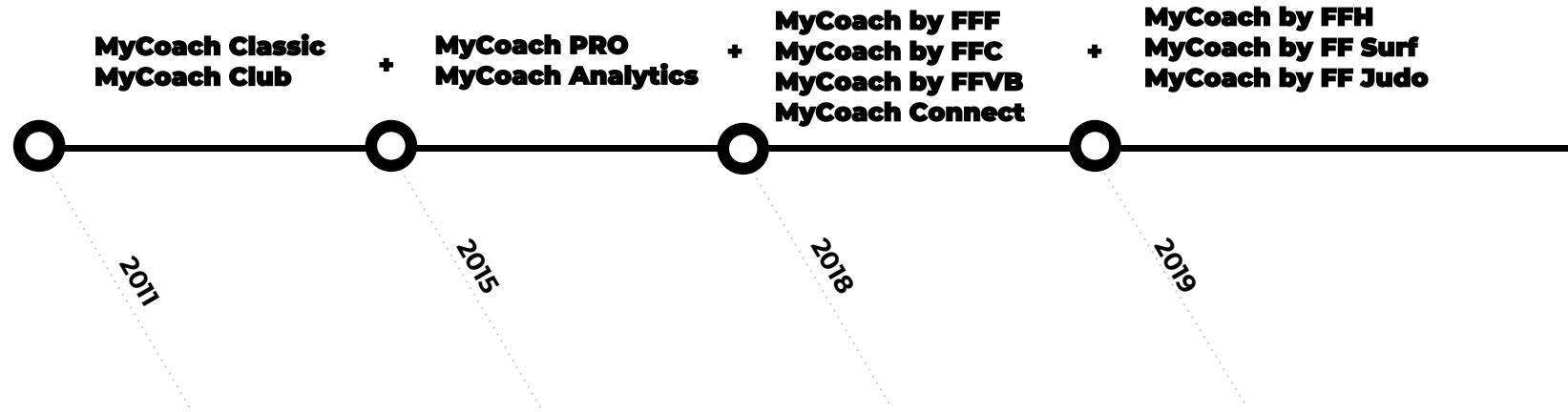
DÉMO

The background of the image is a blurred photograph of an office environment. It features several desks arranged in rows, each equipped with a computer monitor. The office is well-lit by overhead fluorescent lights. In the foreground, there are several potted plants of various types and sizes, adding a touch of greenery to the workspace.

**3.**

## **ARCHITECTURE & BESOIN**

## Nos produits et leur date de livraison



# **LE BESOIN**

**L 'architecture est pilotée par le besoin.**

2011 - 2015

Nos coachs et nos clubs ont besoin de consulter et saisir des informations via leur ordinateurs.

2015 - 2018

Nos coachs, nos clubs, et nos joueurs ont besoin de consulter et saisir des informations via leur ordinateurs, leurs tablettes ou leurs smartphones.

2018 - Aujourd'hui

Les acteurs du football, rugby, volleyball, vélo, judo, surf ont besoin de consulter et saisir des informations via leurs ordinateurs, leurs tablettes ou leurs smartphones.

# **ÈRE 2011 - 2015**

**2 Applis web**

PHP (Symfony) + MySQL

Requêtes en **DB** depuis PHP

# ÈRE 2015 - 2018

**1 REST API**

Scala (Akka HTTP + Slick)

MySQL

MongoDB

**4 Applis mobiles**

Swift (IOS)

Java 7 (Android)

**4 Applis web**

PHP (Symfony)

ReactJS (un tout petit peu)

# ÈRE 2018 - Aujourd'hui

**1 REST API**

Scala (Akka HTTP + Slick)

MySQL

MongoDB

**> 10 Applis mobiles**

Swift (IOS)

Java 7 (Android)

**> 5 Applis web**

PHP (À terme plus du tout)

ReactJS (Beaucoup)

# ÉVOLUTIONS

Le besoin évolue, l'architecture doit suivre **mais** doit rester compatible avec l'existant.



**4.**

## **ÉVOLUTION DE L'ARCHITECTURE**

# **NOUVEAU BESOIN**

On veut un système d'événements génériques, pour pouvoir :

- Réserver un emplacement
- Inviter une ou plusieurs personnes
- les différents types sont : Entrainements, Matchs, Cours, Rendez-vous avec le staff, ...

# **PREMIÈRE APPROCHE**

# **DEUXIÈME APPROCHE**

## L'architecture évolue, mais les erreurs se répètent

- Récupérer des informations pour une liste de joueurs
  - PHP + Mysql
    - Une boucle for (une itération/joueur)
    - Une requête SQL / joueur
    - Pas optimal du tout mais performance OK
  - PHP + API
    - Une boucle for (une itération/joueur)
    - Un appel à l'api/joueur
    - Une connexion TCP / appel
    - Toujours pas optimal et un petit moins performant Serveur <-> Serveur
  - ReactJS + API
    - Une boucle for (une itération/joueur)
    - Un appel à l'api/joueur
    - Une requête AJAX, une connexion TCP / appel
    - Toujours pas optimal et un petit moins performant

## Solution

- Récupérer des informations pour une liste de joueurs
  - ReactJS + API
    - Un appel à l'API avec la liste de joueurs pour lesquels on veut des informations
    - Une seule requête AJAX -> une seule connexion TCP -> Une seule requête en DB

## Transformer un modèle très orienté football en modèle multi-sport

- Un match de football a une durée
- Un match de volleyball n'en a pas
- Au foot premier carton jaune -> pas d'exclusions
- Au hockey premier carton jaune -> exclusion de 5 minutes minimum
- Les règles varient en fonction des catégories
- Difficulté
  - Avoir le moins de différences possible au niveau API
  - Ca permet d'accélérer les développements web et mobile en factorisant au maximum
  - Ca permet aussi d'avoir moins de code à maintenir



**5.**

**COMPATIBILITÉ**

## Préserver la compatibilité

- **Un seul backend pour n applications mobile et web**
- Changer une route peut impliquer casser la compatibilité en prod
- Les changements qui peuvent casser la compatibilité
  - Changer le chemin d'une route
  - Changer le body entrée/sortie
    - Ajouter un nouveau champ obligatoire en entrée
    - Supprimer un champ en sortie

## Solution

- Un langage commun pour définir les "payloads"
- Le langage est compilé en Scala, Swift, PHP, TypeScript
- Si jamais on fait un changement incompatible, on va casser la compilation des applications dépendantes

The background of the image is a blurred photograph of an office interior. It features several wooden desks arranged in rows, each equipped with multiple computer monitors. The office is well-lit by overhead fluorescent lights. In the foreground, there are several large, healthy green plants in various pots, some on the desks and others on shelving units. A person's arm and shoulder are visible on the left side, suggesting they are sitting at one of the desks.

**6.**

## **INFRASTRUCTURE**

# **MICRO SERVICES**

**12 Services backend**

**16 Services frontend**

**+**

**~ 5 service additonels**

# **AVANT**

**3 Machine virtuelles**

**Rien n'est redondé**

**Le déploiement d'un service entraîne une interruption de service**

**Aucun failover en cas de panne**

# **NOTRE BESOIN**

Les services critiques doivent être redondés.

Tous les services doivent redémarrer tout seul en cas d'anomalies.

Déployer n'importe quand, sans coupure.

Déploiement en 1 Clic.

# KUBERNETES !

Orchestration de conteneur.

Très facile de redonner des conteneurs "stateless".

Round robin natif.

En cas d'anomalies, les conteneurs redémarrent tout seuls.

Déploiement sans coupures.

Déploiement "safe".

# **CHIFFRES**

**1064** mise en production depuis Février 2018.

Déploiement en moins d'1 minute.

**42** conteneurs qui tournent 24/7.



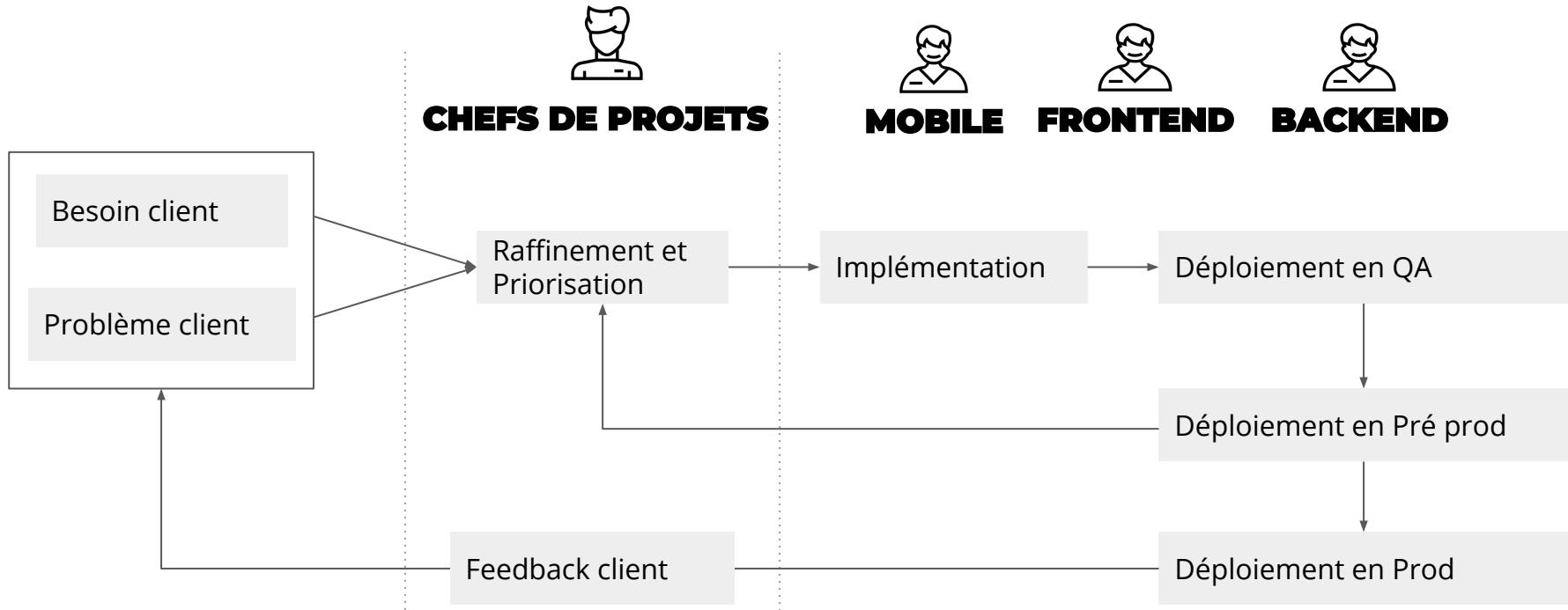
# 7. QUOTIDIEN

# L'ÉQUIPE

8 Développeurs  
+  
3 chefs de projets  
=  
8 produits. (Et + à venir)



# LE PIPELINE MY COACH SPORT



# AUTOMATISATION

- Q On **rationalise** au maximum.
- Q Nombreux outils, **open source et privés**.
- Q Tout est **automatisé**.
- Q **Construire un livrable pour une release** = 1 Clic
- Q **Déployer en production** = 1 Clic
- Q **Déployer une application** sur un app store = 1 Clic

# DES QUESTIONS ?



My Coach Sport



@MyCoach\_fr