

Projet eCOM IFO5 (ex RICM5) 2018

version provisoire 18/09/2018

Sybille Caffiau

Didier Donsez

Vincent Zurczak*

Université Grenoble Alpes, Polytech Grenoble & UFR IM2AG
Linagora

Sommaire

- Objectifs fonctionnels
- Objectifs pédagogiques
- Architecture globale
- Organisation et planning
- Résultats attendus

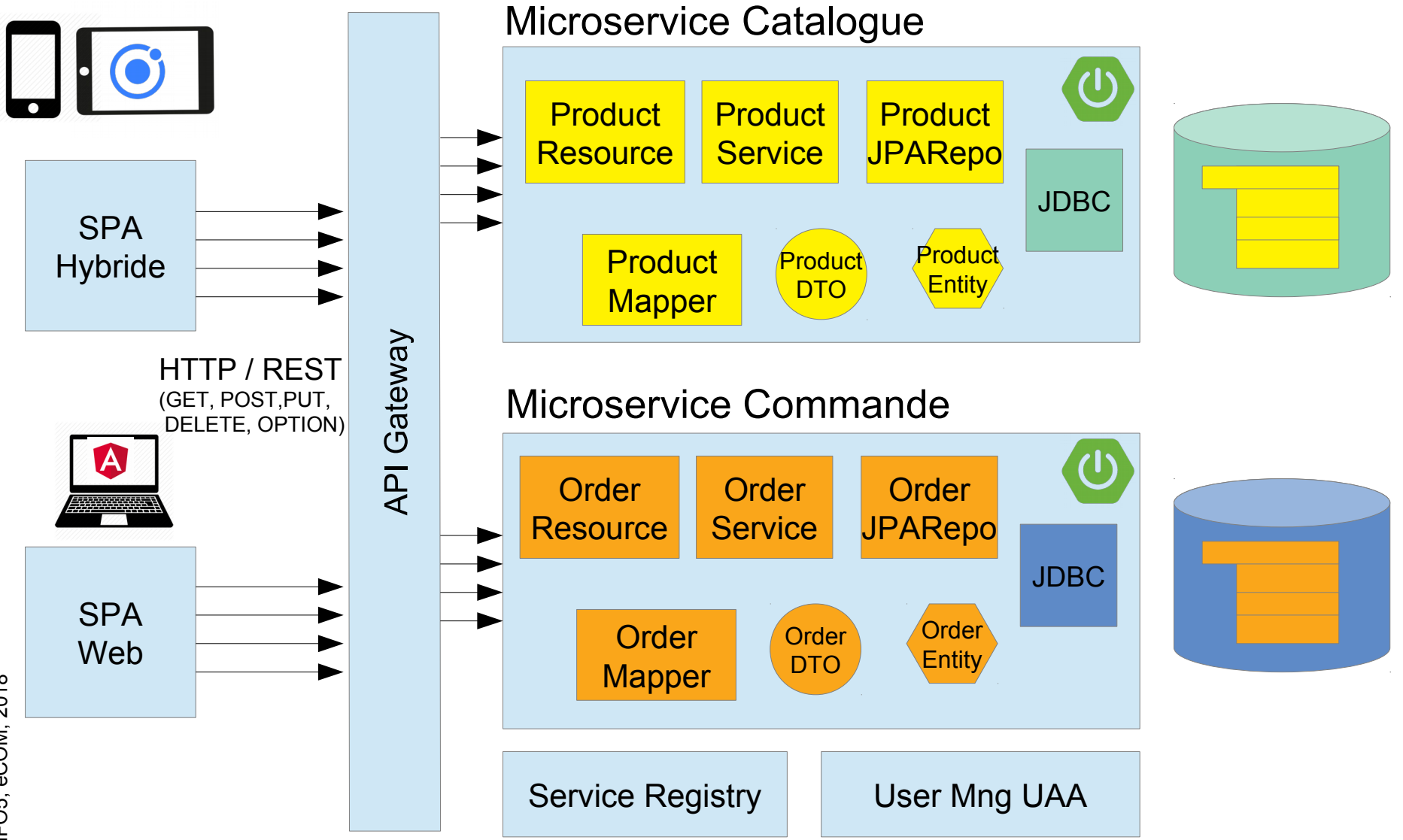
Objectifs fonctionnels du projet

- Développement d'une application simple de « prise de commande » par le Web
 - La nature des produits vendues est fixée
 - Le profil type des utilisateurs est à définir
 - Environnement matériel
 - Contexte
 - ...

Objectifs pédagogiques

- Intégration de différentes disciplines
 - CAR, IHM, GL
 - Utilisation des principes et techniques vues dans les différentes formations
 - Approfondissement en options
 - Suivi / encadrement
 - Aller chercher les informations auprès des enseignants concernés (CAR / IHM / GL)
- Manipuler des technologies de pointe dans le développement d'applications **hautement disponibles et hautement performantes** sur Internet
- Gestion agile d'un « Gros » projet → Scrum, Kanban, Lean

Architecture Microservice 1 uS par (grande) fonction



Environnements/intergiciels Supports (i)

□ Jhipster

- Backend Spring Boot
- Bases de données
 - Dev : H2
 - Prod : Postgres, MySQL, MySQL répliqué
- Frontend SPA et/ou Hybride
- Docker et consors (k8s, ...)
- Hébergement Cloud
 - IaaS : Windows Azure, AWS, Google, Bluemix, Heroku
 - OVH (2,99 euros/mois)
 - Instances « Small » Linux gratuite
 - Déploiement multi-compte (4 à 5 comptes)
 - (Ne commitez pas les credentials sur un dépôt public)

Architecture globale

Les constituants à **générer** à l'étape 1

- La base de données relationnelle
 - 3 Tables Category, Product, *User*
 - + Tables optionnelles : ExtraUser, Customer, Order, OrderLine, Payment
- Le Backend Spring: Monolithe RESTFul
 - 3 Entities + Relationship + JPARepository
 - 3 DTO
 - 3 Resources+ 3 Services
 - + services additionnels (MailService, ...)
 - API OpenAPI (Swagger)
- 3 Frontends (générés avec JHipster)
 - 2x SPA Web : AngularX ou React (admin, users)
 - Mobile hybride : Ionic + plugins Cordova (plateformes iOS et Android)
- Un client de test basé sur cURL
 - Généré avec Swagger Codegen

Architecture globale

Les constituants à développer à l'étape 2

- Implémenter la logique métier dans les services
 - Par rapport aux ROLE des jhi_user
- IHM « responsive » ou IHM mobile first
 - en fonction des besoins clients
 - Canevas SPA (Angular X, ...), Ionic + Plugins
- Taches périodiques
 - Campagne de mailing
 - Envoi de rappel par mail
- Génération de PDF
 - Factures, Dossards, ...

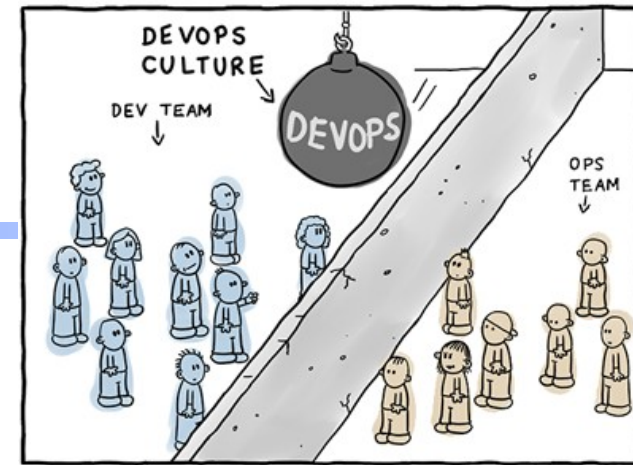
Architecture globale

Les constituants à développer à l'étape 2

- Optimisation des transferts des LOB / Collections
 - SPA ← → Backend
 - Backend ← → Base de données
 - LocalStorageService
- Services externes
 - Stripe.com (API de paiement) : utiliser le mode « test »
 - Gravatar, Social logins
- Séparation en 2 « microservices »
 - API Gateway
 - 1 seule base de données (répliquée)

DevOps

- Dev(elopment) – Op(eration)s
- Objectifs
 - TTM, MVP, Rapid Production Deployment
- Moyens
 - Continuous Integration (CI)
 - Continuous Delivery (CD)
 - Application Performance Monitoring (APM)



Environnements/intergiciels Supports (ii)

□ Sécurité

- HAProxy/Nginx en SSL Terminaison + Certificat Let's Encrypt
- IpTables, UFW
- Pen testing

□ Monitoring des instances de VM

- Hawkular, Prometheus, Telegraf, InfluxDB, Grafana

□ OAuth2

- Keycloak, OKTA
- Social Login

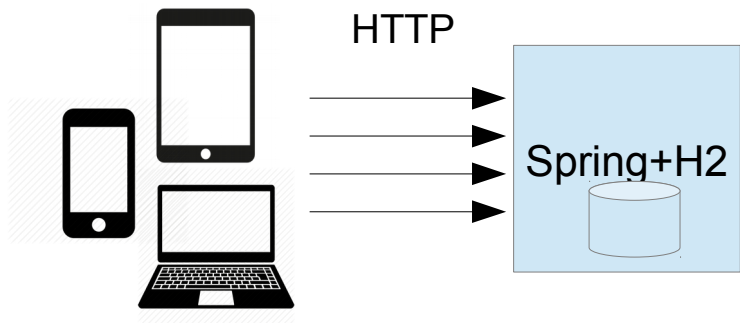
Exemple de rançon pour une base Mongo non protégée

- ▣ `$ db.PLEASE_READ_ME.find()`
- ▣ `{ "_id" : ObjectId("58a7287db7dc324adb249fdf"), "info" : "Don't panic. Your DB is in safety and backed up (check logs). To restore send 0.1 BTC and email with your server ip or domain name. Each 48 hours we erase all the data.", "amount" : "0.1 BTC", "data_we_have" : { "local" : ["startup_log"], "first_database" : ["users", "preferences"], "MyAppXXX" : ["emails"] }, "Bitcoin Address" : "1NSz9TRBGKHKFdjH2Gme3LwDi5", "email" : "xxxxx@xxxx.org" }`

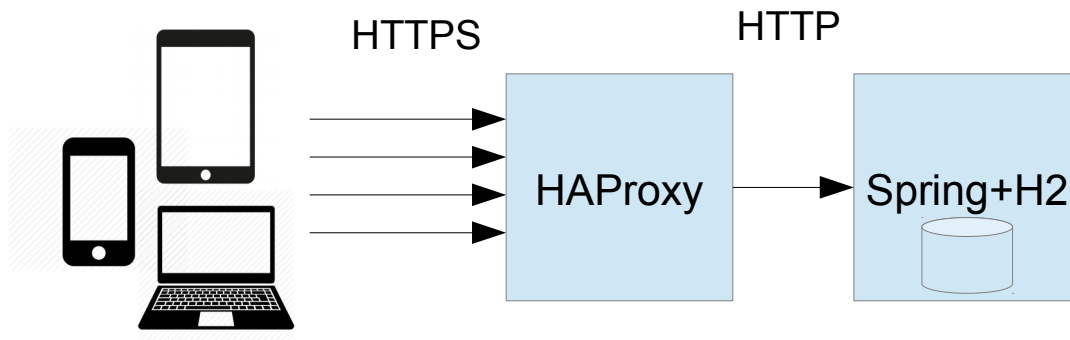
Environnements/intergiciels Supports (ii)

- DevOps
 - Docker, k8s (rolling update)
- Haute disponibilité
 - HAProxy/Nginx en Load Balancer
 - Spring en cluster
 - MySQL Server en replication
- Haute performance
 - Elasticité horizontale k8s (scale up/down)

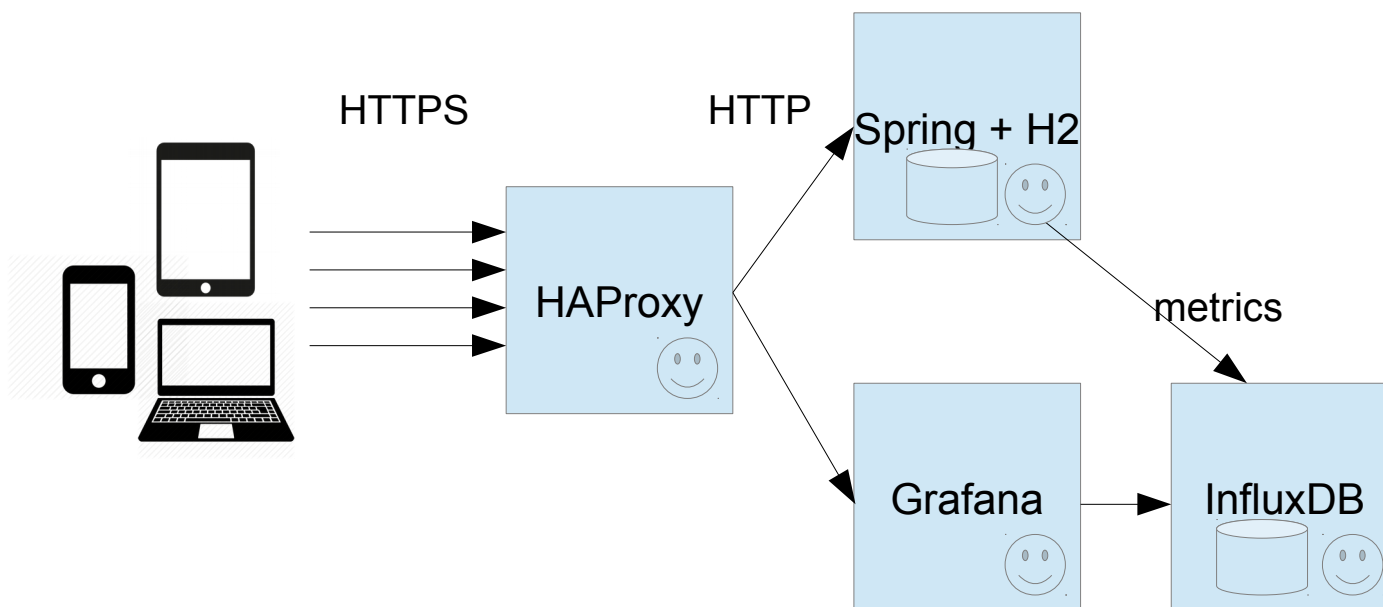
Etapes de développement (i)



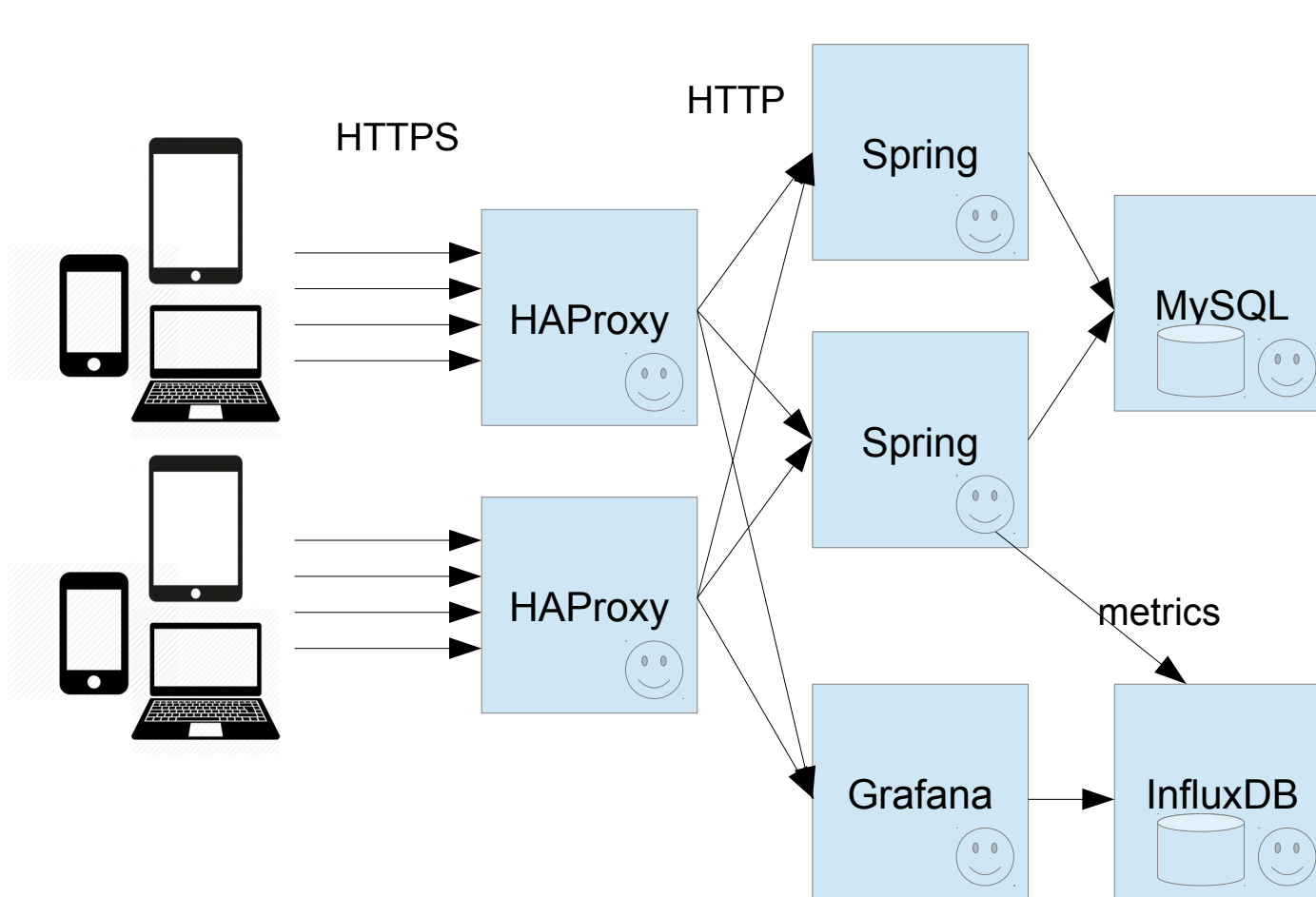
Etapes de développement (ii)



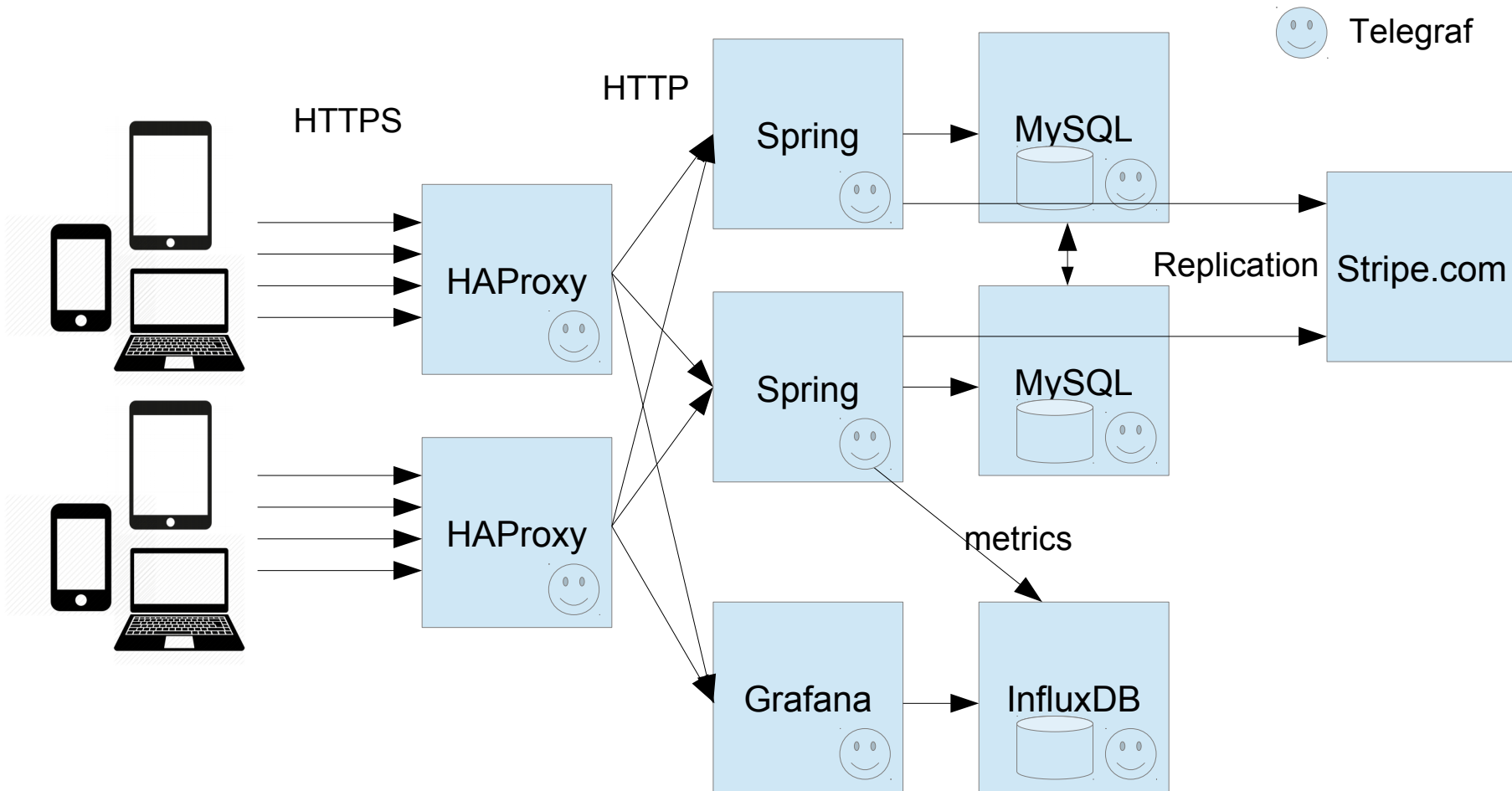
Etapes de développement (iii)



Etapes de développement (iv)



Etapes de développement (v)



Livraison en Continue

- Principe « Minimal Viable Product »
- Service immédiatement en production
 - Après un staging « court »
- Blue – Green Deployment
 - Rolling Update
 - Fast rollback
- En option : Elasticité horizontale (scaling)
 - Ajout et Retrait de VM en cours d'exécution
- → scripts avec awscli, ssh, curl (Swagger), git, Docker, Docker Compose ...

Intégration en Continue

- Test unitaire
- Test d'intégration
- Test e2e
- Test de charge
- Test de résilience
 - Panne d'une machine jusqu'à datacenter outage
- Test de pénétration
- Software Quality Assurance
- Automatisation
 - Jenkins, Travis CI

Qualité du logiciel produit

- IDE
 - Eclipse JEE ou NetBeans JEE ou JetBrains
 - Plugins Azure, AWS, ...
- Builder
 - maven 3, yarn, webpack ...
- Forge (Gitlab Gricad, ...)
 - Intègre Git, Jenkins, Sonar ...
- Software Quality Assurance
 - Rapport SonarQube
 - Code review

Modalité pédagogique

□ Organisation

- Groupe mixte de 4 à 5 des 2 options SR et CM
 - Composition imposée.
- Sujet imposé
 - Scénarii à analyser
- Elire un chef de projet et un scrum master identifiés du début à la fin
 - Contact avec les enseignants
 - Contact avec le client
- Attribuer des rôles/spécialités
- Définir l'échéancier (# de sprints, durée, ...)
- Auto-formation
- Gestion des risques (technique et humain)

Modalité pédagogique (2)

□ Planning

- Voir site
- <http://air.imag.fr/index.php/ECOM-RICM>

Cahier de Charge (CDC) « allégé »

- membre de l'équipe
 - rôle (chef de projet, *scrum master*, ergonomiste, graphiste, navigation)
- objectif du site
- bénéfice attendu quantifié
- description des utilisateurs cibles avec priorité
- plate-forme informatique (client, serveur, ...)
- modèle de tâches : ensemble des fonctions du système
- requis non fonctionnels
 - ex : accomplissement d'une tâche < 10s, ludique, ...
- définition de scénarios types (utilisé par la recette)
- jeu de données
- analyse de la concurrence

Modalités d'évaluation (résultat)

1) Démonstration finale

- objectifs GL
 - Méthodologie Scrum
 - Qualité du logiciel (Rapport SonarQube)
- objectifs Système
 - Expliquer les principaux choix de conception
 - Montrer l'aspect multi-utilisateurs du logiciel
 - Montrer les performances du logiciel (Rapport Gatling)
 - Montrer l'administration du logiciel (observation, déploiement multi-instances (backend Spring répartis), reprise après arrêt...)
- objectifs / IHM
 - (Voir présentation semaine prochaine)

Grille d'évaluation

- Sur le site web
- A produire à chaque audit

Liens

□ Vers le sujet

- <http://air.imag.fr/index.php/ECOM-RICM>