

Projet eCOM RICM5 2017

version provisoire 17/09/2017

Sybille Caffiau

Didier Donsez

Thomas Ropars

Université Grenoble Alpes

Polytech Grenoble & UFR IM2AG

Sommaire

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

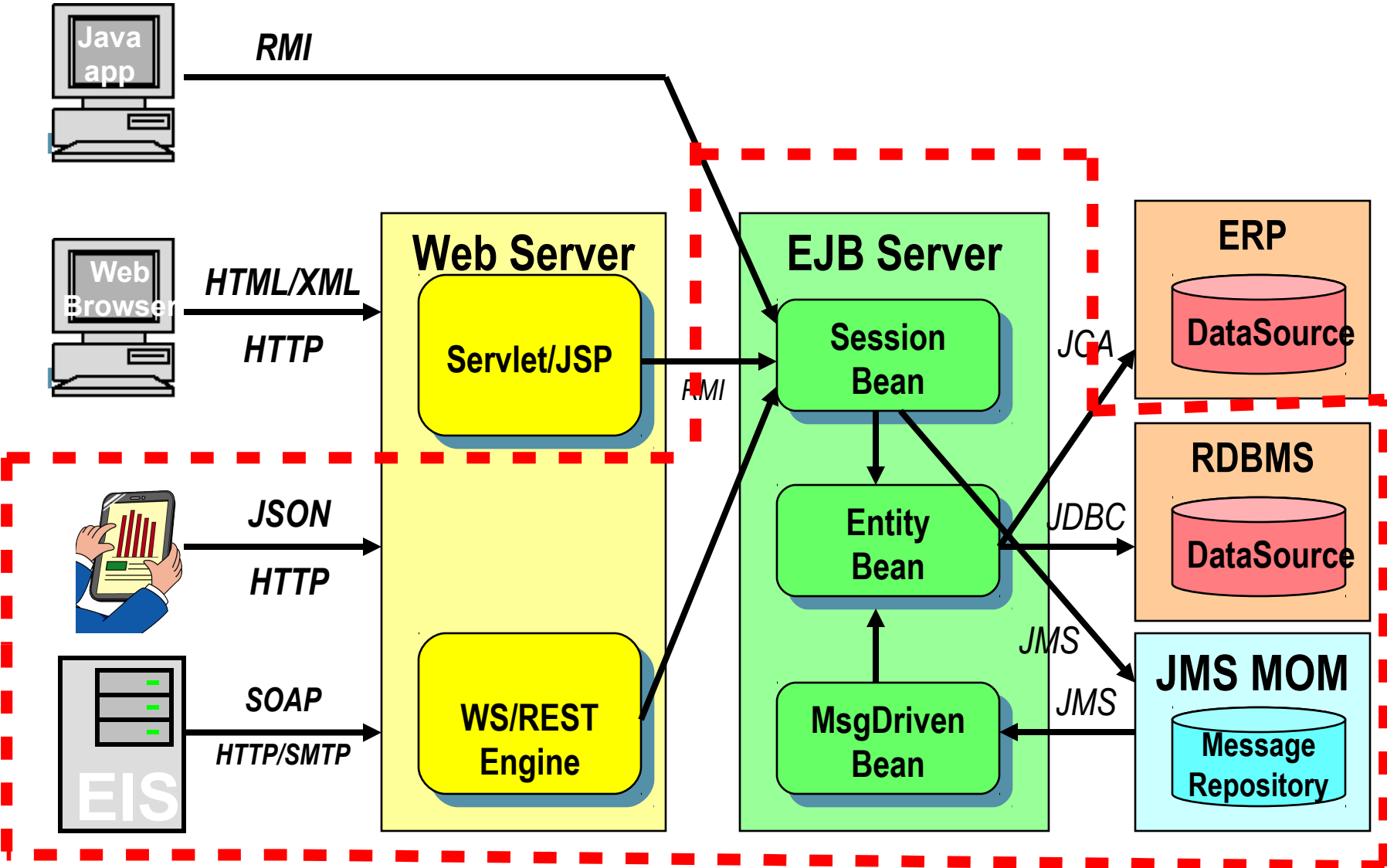
Objectifs fonctionnels du projet

- 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 preformantes** sur Internet
- Gestion d'un « Gros » projet → Scrum, Kanban (F2F, IRL)

Architecture globale (JavaEE)



Architecture globale

Les constituants à développer à l'étape 1

- La base de données (généré par le conteneur JPA)
 - 3 Tables ProductStores, Products, Accounts
 - + Tables optionnelles ? Customers, Orders, OrderLines, Payments
- Le Backend JavaEE
 - Microservice RESTFul
 - 3+3? Entity Beans + Relationship (JPA)
 - 2 Session Beans : Cart (*stateful*), Mailer
 - 1 Message Driven Bean pour fiabiliser l'envoi de mail
 - API OpenAPI2.0 (Swagger)
- 2 Frontends SPA
 - Techno JS Responsive + WebSocket ou RESTFul)
 - Customer
 - Administrateur
- Un client de test basé sur cURL
 - Généré avec Swagger Codegen

Architecture globale

Les constituants à développer à l'étape 2

- IHM « responsive » ou IHM mobile first (en fonction des besoins clients)
 - Canevas SPA (Angular X, ...)
- Message Driven Bean
 - Envoi des mails aux clients
 - Mail de retour en erreur
 - Campagne de mailing
- EJB Timer
 - Taches périodiques

Environnements/intergiciels Supports (i)

□ JavaEE (Java Enterprise Edition)

- Serveur JavaEE Wildfly
 - Intègrent SBGD Relationnel Java Pur + Conteneur Web
- Hébergement Cloud
 - IaaS :
 - Windows Azure, AWS, Google, Bluemix
 - 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)

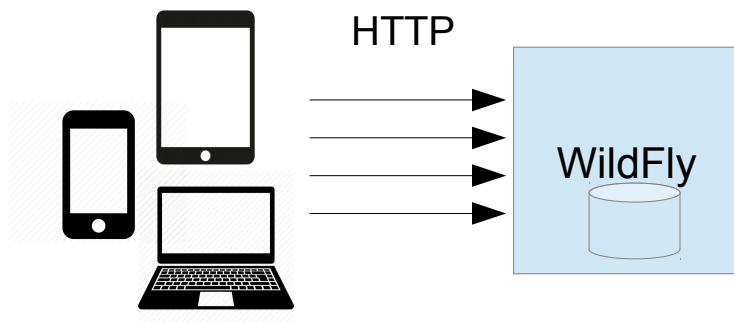
□ Exception : Groupe « JHipster »

- Backend : Spring

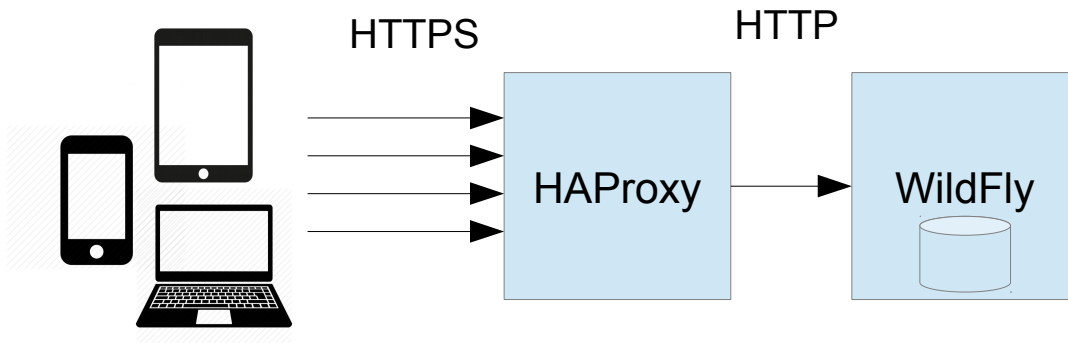
Environnements/intergiciels Supports (ii)

- Monitoring des instances de VM
 - Hawkular, Prometheus, Telegraf, InfluxDB, Grafana
- Sécurité
 - HAProxy en SSL Terminaison + Certificat Let's Encrypt
 - IpTables, UFW
- Haute disponibilité
 - HAProxy en Load Balancer
 - JavaEE en cluster
 - MySQL Server en replication
- Haute performance
 - Elasticité horizontale
- DevOps
 - Docker, Swarm, OpenShift

Etapes de développement (i)

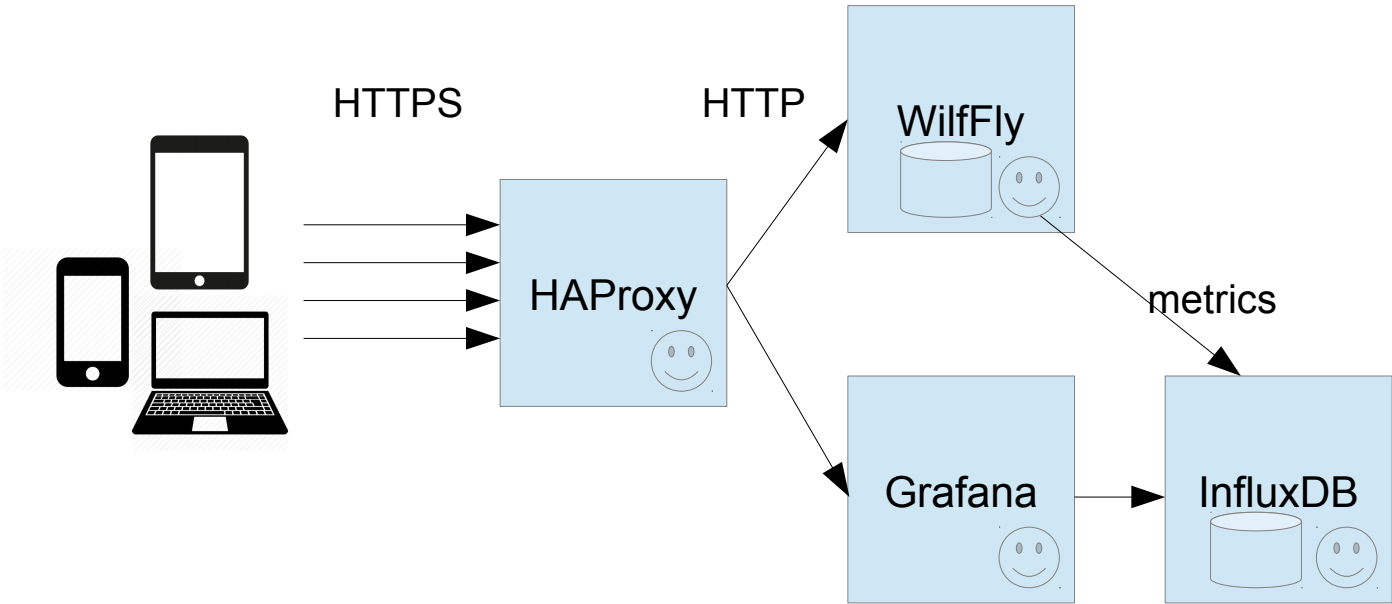


Etapes de développement (ii)

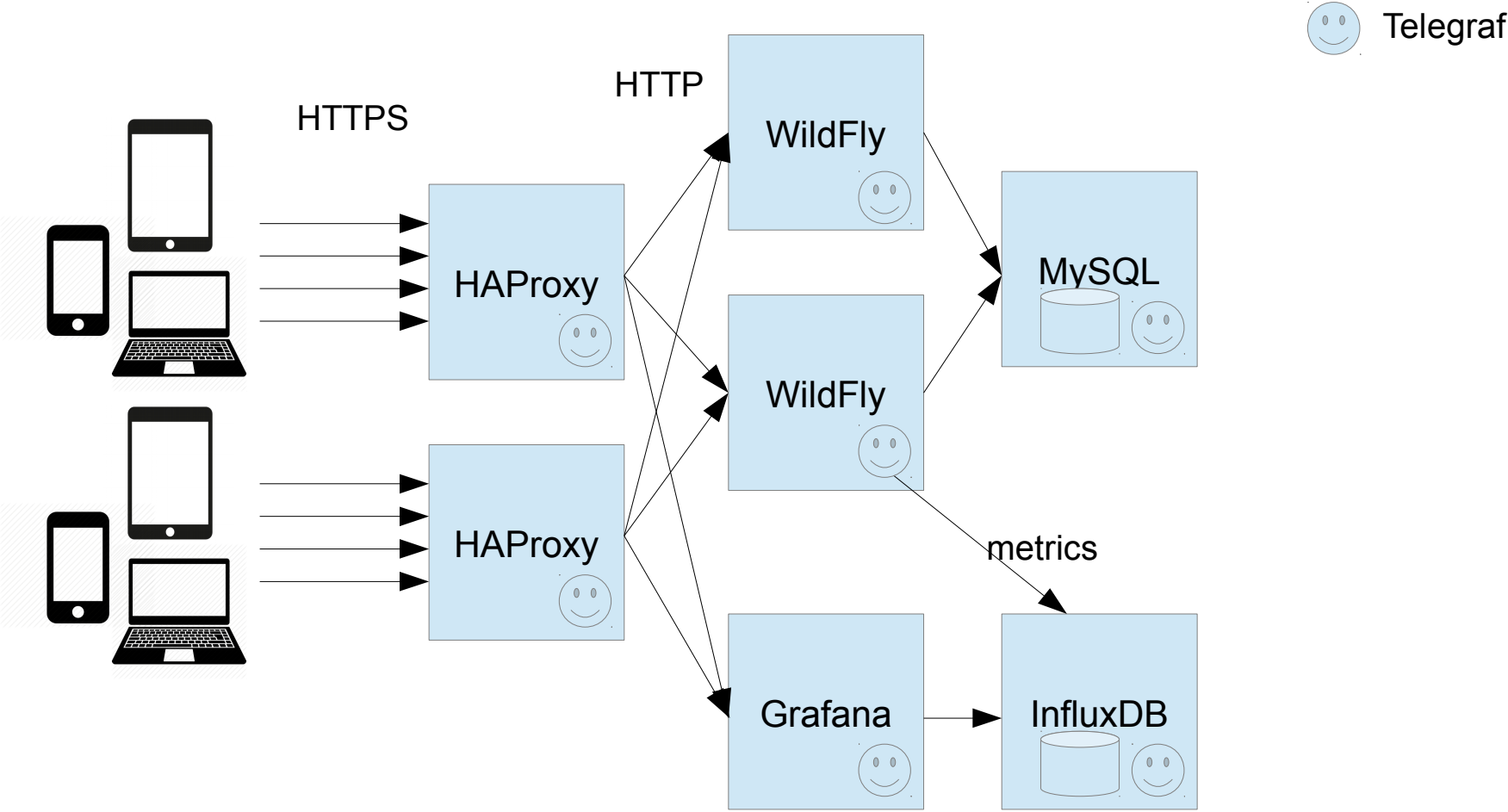


Etapes de développement (iii)

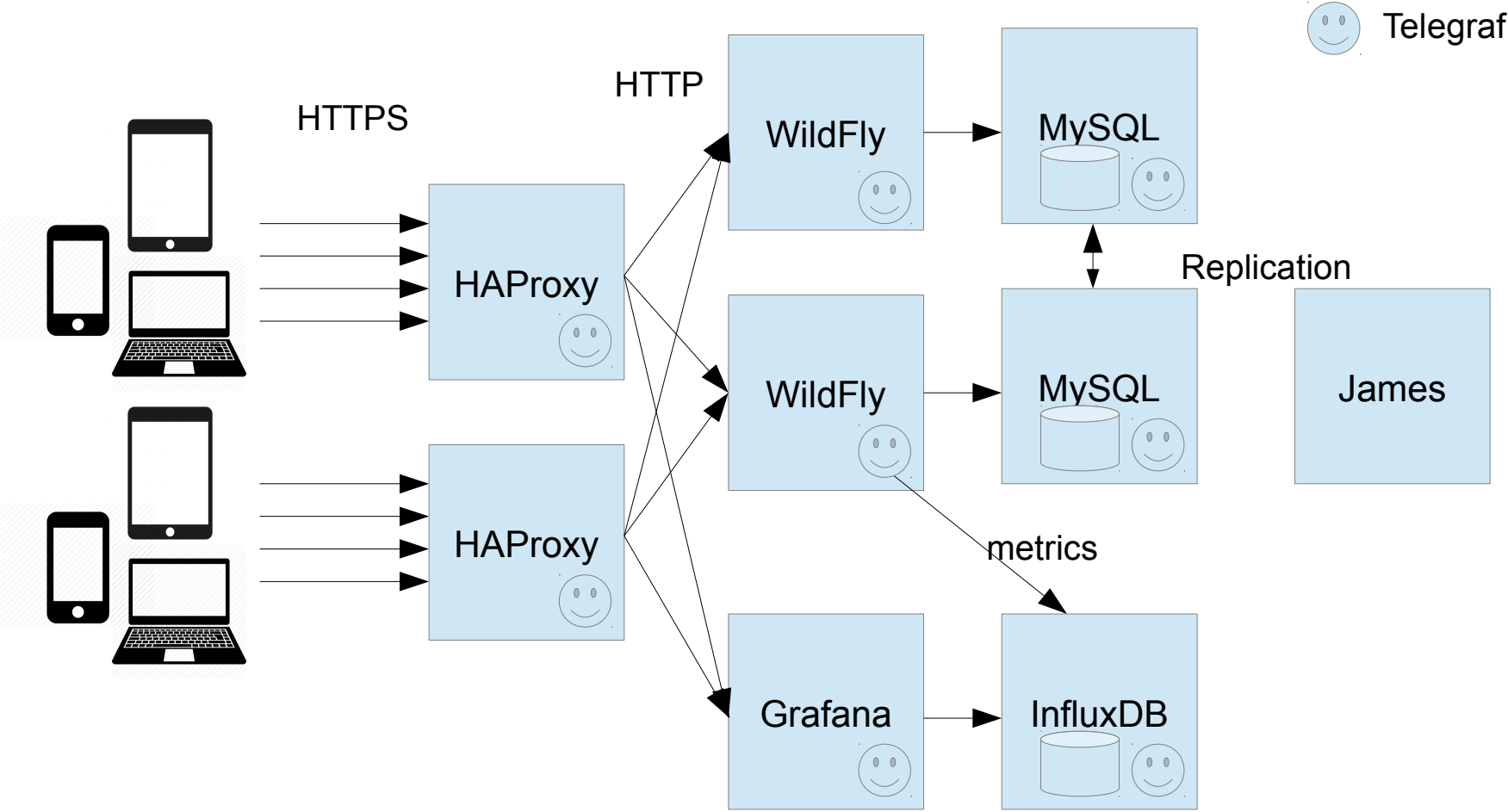
 Telegraf



Etapes de développement (iv)



Etapes de développement (v)



Livraison en Continue

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

Qualité du logiciel produit

□ *MetaWare (Java et Javascript)*

- IDE
 - Eclipse JavaEE ou NetBeans JavaEE ou JetBrains
 - Plugins Azure ou AWS
- Builder
 - Maven 3, bower, ...
- Forge privée (github student pack, bitbucket, ...)
 - Intègre Git, Jenkins, Sonar ...
- Mesure de performance
 - Apache JMeter, Gatlin, CLIF
- Test unitaire
 - Cactus, Junit, DBUnit, Mockito ...

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
- Attribuer des rôles/spécialités
- Définir l'échéancier (# de sprints, durée, ...)
- Auto-formation
- Gestion des risques

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*, ergonome, 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 Sonar)
- objectifs Système
 - Expliquer les principaux choix de conception
 - Montrer l'aspect multi-utilisateurs du logiciel
 - Montrer les performances du logiciel (Rapport JMeter)
 - Montrer l'administration du logiciel (observation, déploiement multi-instances (serveur JavaEE 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>