Intégration d’Applications d’Entreprise (EAI) – Projet

Gestion de pré conventions de stages

Dossier technique

|  |  |
| --- | --- |
| Mahdi HENTATI - Lucas LUZAN | M2 MIAGE 2018-2019 |

Table des matières

[1 Rappel du besoin 2](#_Toc526642732)

[2 Technologies utilisées 2](#_Toc526642733)

[3 Vue d’ensemble 2](#_Toc526642734)

[4 Architecture 2](#_Toc526642735)

[4.1 Objectifs 2](#_Toc526642736)

[4.2 Application internes (bureau des stages, scolarité, service juridique, départements d’enseignement) 3](#_Toc526642737)

[4.2.1 Distribution des applications 3](#_Toc526642738)

[4.2.2 Communication 3](#_Toc526642739)

[4.2.3 Bases de données 3](#_Toc526642740)

[4.2.4 Accès aux données externes 3](#_Toc526642741)

[4.3 Application de demande de conventions (étudiant) 3](#_Toc526642742)

[4.4 Open Data interne 3](#_Toc526642743)

[5 Exemples de scénarios de déploiement physique possibles 4](#_Toc526642744)

[5.1 Déploiement distribué 4](#_Toc526642745)

[5.2 Déploiment centralisé 4](#_Toc526642746)

[5.3 Déploiement entièrement géré par le service des stages 5](#_Toc526642747)

# Rappel du besoin

Le projet à développer est un système de gestion de convention de stages étudiants au sein d’une université.

Les échanges de messages entre les parties prenantes de ce système devront s’appuyer sur JMS (Java Message Service).

En plus, il est demandé de réaliser un service de publication interne des données relatives aux conventions validées qui va être utilisé par d’autres acteurs de l’université.

Enfin, il faudra effectuer lors de ce processus des appels à des ressources externes via HTTP.

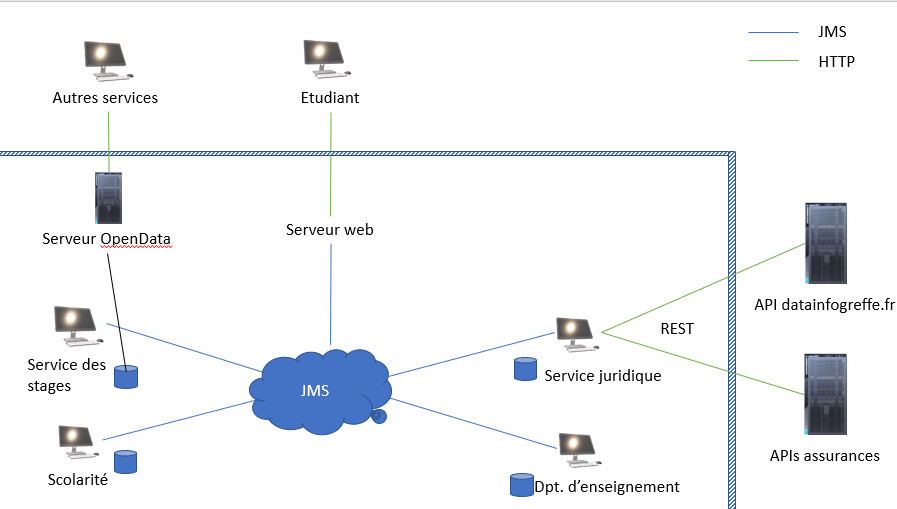
# Technologies utilisées

Glassfish 5 : serveur d’application JEE

openmq : MOM implémentant l’API JMS (installé par défaut avec Glassfish)

A définir : sgbd

# Vue d’ensemble



# Architecture

## Objectifs

La solution proposée a été pensée afin d’optimiser les facteurs suivants:

* La disponibilité
* La performance
* L’évolutivité
* La scalabilité
* La sécurité

## Application internes (bureau des stages, scolarité, service juridique, départements d’enseignement)

### Distribution des applications

Nous avons choisi de développer une application JEE par acteur afin de permettre une distribution maximale de la solution. Ainsi, il sera possible de déployer chaque application sur une instance de serveur d'application JEE séparée.

### Communication

La communication entre les différentes applications est un élément central du SI à développer. Cette communication s’effectue via un MOM.

Afin d’améliorer la fiabilité du MOM et faciliter son évolutivité et son administration (droits d’accès, distribution des brokers..), nous avons choisi de configurer les différentes instances du serveur d'application JEE pour utiliser un serveur MOM distant.

Ce choix permet aussi de découpler le fournisseur MOM du serveur d'application JEE. Ainsi, même une application non-JEE pourra accéder au MOM.

### Bases de données

Nous avons choisi d’établir une base de données par service dans l’optique d’améliorer l’autonomie de chaque service et la confidentialité des données qu’il gère.

Un schéma de base de données par service sera crée.

### Accès aux données externes

Le service juridique doit accéder à des API HTTP REST externes afin de vérifier l’exactitude de certaines informations. L’application dédiée au service juridique fera ces appels sans intermédiaire pour des raisons de performance.

## Application de demande de conventions (étudiant)

Une application web permettant aux étudiants de faire une demande de convention de stage redirigera les informations entrées par l’étudiant directement au bureau des stages via le MOM.

Etant donné que cette application est directement accessible depuis internet, il nous semble adéquat qu’elle soit déployée sur une instance de serveur d'application JEE séparée.

## Open Data interne

Certaines données collectées par le bureau des stages seront publiées en interne via une API HTTP REST.

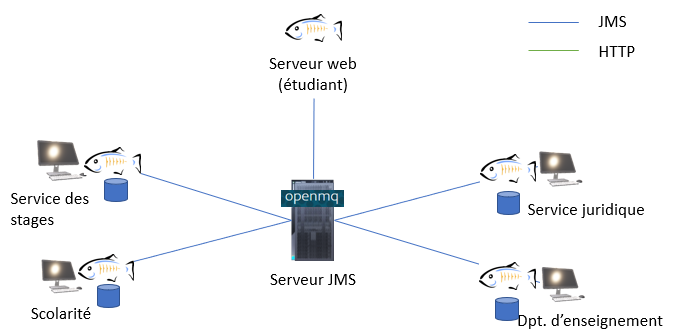
Les APIs HTTP REST permettent de publier les ressources sans être liées à une technologie particulière. Ces ressources peuvent être exploitées par n’importe quel client HTTP.

Pour extraire les données, ce service se connectera directement à la base de données du bureau des stages en mode lecture. Eventuellement, il peut les cacher dans son propre entrepôt de données pour réaliser des traitements décisionnels.

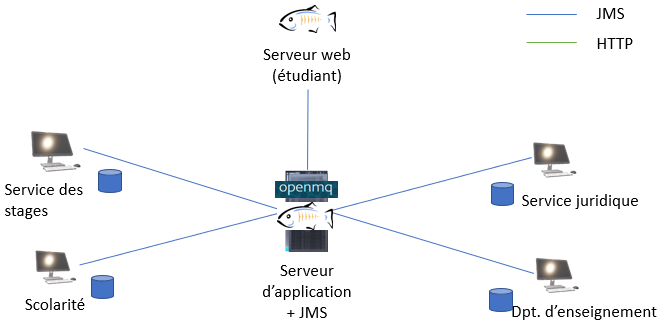
# Exemples de scénarios de déploiement physique possibles

Le projet sera développé dans l’optique de permettre plusieurs scénarios de déploiement :

## Déploiement distribué



## Déploiment centralisé



## Déploiement entièrement géré par le service des stages

