

Laboratoire du cours de Systèmes distribués

Année académique 2017 - 2018

*3^{ème} année Informatique et Systèmes -
Informatique de Gestion*



*Mounawar Madani
François Caprasse*

Dossier final

Mise en situation

Le contexte de ce dossier final est la création d'une solution distribuée permettant la gestion de prises de sang.

L'application fera intervenir trois types d'intervenants :

- Des médecins qui vont pouvoir se loguer sur le système et, pour un patient donné, prescrire une liste d'analyse à effectuer.
- Un laborantin qui encodera les résultats des analyses dans le système.
- Des patients à qui on a prescrit des analyses. Ils pourront consulter leurs résultats et effectuer le paiement de la prestation via PayPal.

Exemple d'utilisation concret

1. Le médecin se connecte l'application.
2. Le médecin sélectionne un patient (l'ajoute si nécessaire).
3. Le médecin choisit les analyses à effectuer.

<input checked="" type="checkbox"/> Leucocytes	5.300 /mm3	4.000 à 10.000
<input type="checkbox"/> Hématies	4.730.000 /mm3	4200000 à 5800000
<input checked="" type="checkbox"/> Hémoglobine.	14,3 g%	13,0 à 18,0
<input type="checkbox"/> Hématocrite.	42,8 %	40,0 à 54,0
<input type="checkbox"/> V.G.M.	90 u3	80 à 100
<input type="checkbox"/> C.C.M.H.	33,4 %	32,0 à 36,0
<input checked="" type="checkbox"/> T.C.M.H.	30,2 uug	26,0 à 33,0
<input type="checkbox"/> RDW	13.5 %	inf. à 15.0

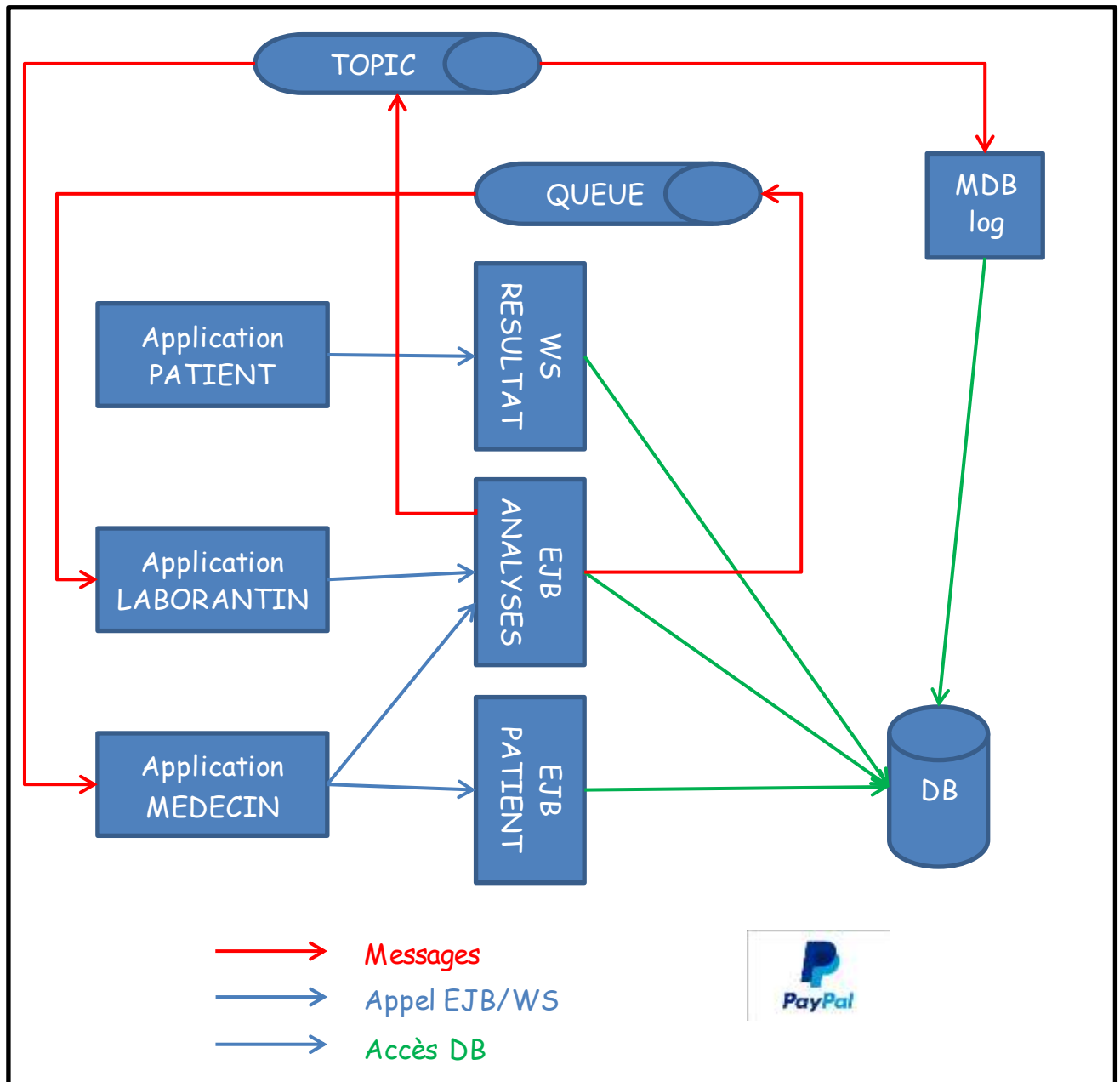
4. Le médecin envoie la demande via l'EJB analyse.
5. L'EJB analyse effectue les insertions nécessaires dans la DB, communique en retour au médecin la référence de la demande et envoie un message sur la QUEUE destination d'un laborantin.
6. Le médecin donne oralement la référence de la demande au patient (pour la consultation des résultats)
7. Le laborantin reçoit la demande via la QUEUE.
8. Le laborantin encode les résultats via l'EJB analyse.
9. L'EJB analyse effectue les mise à jours nécessaires dans la DB et envoie un message sur le TOPIC de manière à ce que le médecin soit averti de manière asynchrone que des résultats urgents sont disponibles (dans le cas d'une demande urgente) et à ce que le MDB de log puisse effectuer son calcul de durée de traitement et insérer cette information dans la table de log.
10. Le patient via l'application patient (et donc le WS) consulte ses résultats grâce au numéro de demande. Il récupère également les informations nécessaires pour le paiement.

11. Le patient effectue le paiement via PayPal.

Remarque : Le médecin peut à tout moment rafraichir la liste de résultats.

Architecture de la solution

L'architecture de la solution est **imposée** et est la suivante :



Nous retrouvons dans cette architecture trois applications clientes distinctes, deux EJBs, un WebService, un topic, une queue, un MDB et une base de données.

La base de données

La base de données élémentaire nécessaire à notre application contiendra les tables suivantes :

- Patient (Id, Nom, Prénom, Login)
- Medecin(Id, Nom, Prénom, Login)
- Demande (Id, RefPatient, RefMedecin, DateHeureDemande, Urgent)
Remarque : Urgent (True ou False)
- Analyses (Id, Item, Valeur)
- Logs (Id, Infos)

L'authentification

Une authentification sera nécessaire pour les applications médecin et laborantin. Celle-ci sera gérée par Glassfish (cfr. manipulation 6). Dans le cas d'un laborantin celle-ci sera suffisante. Pour un médecin, la correspondance entre le médecin connecté et son identité dans la base de données sera réalisée en utilisant la méthode `getCallerPrincipal()`.

Détail des fonctionnalités

- **L'application Médecin :**

Après s'être identifié, un médecin va pouvoir :

- Gérer les patients (via l'EJB patient) :
 - Récupérer la liste des patients déjà enregistrés sur le système.
 - Ajouter un patient.
 - Modifier les informations d'un patient.
- Prescrire des analyses (via l'EJB analyse) :
 - Le médecin sélectionne un patient.
 - Il peut ensuite choisir les items pour lesquels il souhaite une analyse.
 - Il envoie la demande.
- Consulter les résultats (via l'EJB analyse ou message sur le topic) :
 - En effectuant un rafraîchissement de la liste des résultats.
 - De manière asynchrone pour les résultats urgents (via un filtrage des messages sur le topic).

- **L'application Laborantin :**

Après s'être identifié, un laborantin sera en attente d'une nouvelle demande d'analyse. Lors de la réception d'une nouvelle demande il va la traiter et encoder les résultats. Il traite les demandes une à une.

- **Le MDB log**

Le MDB log a pour fonction de logger la durée de traitement de chaque demande d'analyses. Chaque fois qu'un laborantin aura terminé de traiter une demande il enverra un message sur le topic après avoir stocké les résultats dans la base de données via l'EJB analyse. Lors de la réception du message le MDB de log insèrera un nouveau tuple dans la table de log. Celui-ci aura la forme suivante : « La durée de traitement de l'analyse X a été de XhYmin ».

- **L'application patient :**

Cette application permettra à un patient de consulter les résultats de ses analyses. Il devra pour ce faire encode la référence d'analyse qui lui aura été fournie oralement par le médecin au moment de la demande.

Lors de la visualisation de ses résultats il pourra également obtenir les informations nécessaires à la réalisation d'un paiement PayPal (Communication et montant à payer).

Il devra ensuite réaliser le paiement via une page web développée à cet effet. La Vérification de réception du paiement sera effectuée en se connectant à PayPal avec le compte facilitator.

- **L'EJB Client**

L'EJB Client permet la gestion des patients :

- Obtenir la liste des patients.
- Ajout d'un patient dans le système.
- Mise à jour des données d'un patient.
- ...

- **L'EJB Analyses**

L'EJB Analyses permet aux médecins et laborantins de respectivement soumettre des demandes d'analyses et d'encoder les résultats d'analyses pour une demande correspondante.

Remarques générales

Les différents messages envoyés sur le topic ne sont pas destinés à tout le monde. Il faudra donc mettre en place un système de filtrage.

Pour la présentation de votre solution, les applications des laborantins seront toujours démarrées avant que les médecins commencent à effectuer des demandes. Il est bien entendu que si l'application devait fonctionner en production, cette contrainte n'est pas tolérable et qu'il faudrait mettre en place un système permettant la rétention des informations nécessaires aux différents intervenants.