

Enterprise eMail Integration System

Table of contents

1 Introduction.....	2
2 Fonctionnalités.....	2
3 Technologies.....	3
3.1 J2EE.....	3
3.2 Intégration.....	3
3.3 Interface Utilisateur.....	4

1. Introduction

eMis (Enterprise eMail Integration System) est une série d'outils permettant aux entreprises de gérer un nombre important de demandes faites par emails.

Aujourd'hui nous devons parfois traiter un volume important d'emails à titre personnel. Ceci est d'autant plus vrai pour les entreprises, surtout si celles-ci disposent d'une vitrine sur le web, ou si elles proposent un service d'assistance. Cependant, les outils de gestion des emails internes à l'entreprise ne sont pas adaptés à ce genre de contexte car ils ne fournissent pas les fonctionnalités nécessaires aux traitements de masse (affichage de l'historique des clients, gestion de 'workflow', statistiques...)

Dans ce type de contexte, l'entreprise doit notamment pouvoir répondre aux problématiques suivantes :

- la distribution : comment être certain qu'un type de demande particulier sera dirigé vers le bon interlocuteur (ou le bon groupe d'interlocuteur).
- la qualité des réponses : comment être certain que les réponses apportées auront le même niveau de qualité, comment permettre une gestion partagée des réponses types.

2. Fonctionnalités

Le but du projet est de proposer un système simple mais efficace qui doit permettre d'assurer les tâches suivantes :

- retirer régulièrement les emails reçus sur une liste de comptes.
- permettre le filtrage des emails avec la possibilité d'envoyer une réponse automatique en cas de refus. Ceci permet, par exemple de refuser les emails avec pièces jointes en demandant à l'expéditeur de réitérer sa demande, faire un filtrage sur l'adresse de l'expéditeur (anti-spam), faire une réponse automatique en demandant à l'expéditeur de contacter une autre adresse (changement d'adresse) ...
- rassembler les emails d'un même échange en conversations (suite de demandes et de réponses sur un même sujet).
- associer un email à un utilisateur référencé dans une ou plusieurs bases de données existantes.
- qualifier la demande et la ventiler vers le bon interlocuteur.
- proposer une interface ergonomique aux personnes chargées de rédiger les réponses. Cette interface leur permet de construire leur réponse en utilisant une série de templates, un véritable système d'assistance à la création de contenu, le but étant de fournir des réponses efficaces, précises, personnalisées aux demandes formulées.
- fournir un système de validation par un tiers (superviseur ...) permettant de s'assurer de la

qualité du contenu.

- fonctionnalité d'externalisation : pour certains types de requêtes particulières, les demandes peuvent être redirigées vers une adresse email externe au système (pôles d'experts par exemple). La réponse est apportée par le destinataire et renvoyée par email vers eMis. La réponse est alors réintégrée dans le système et suit son parcours comme si elle avait été construite avec le système.
- assurer l'envoi des réponses.
- référencer, dans un but statistique, un large panel d'événements survenant durant toutes ces opérations.

3. Technologies

3.1. J2EE

Pour la richesse des API disponibles, le nombre important de projets déjà existant, la puissance des outils, et notre propre compétence affirmée dans le domaine, nous avons choisi la technologie Java/J2EE.

Les développements du middleware sont faits pour le serveur d'application JBoss. Dans la mesure du possible, le code restera un code J2EE portable.

Le noyau du système est composé de Message Driven Beans et de Queues JMS : chaque service est un MDB qui écoute une Queue JMS et envoie des messages sur une autre queue quand son travail est terminé. C'est une implémentation de type MOM (Middleware Orienté Message).

Plusieurs services délèguent une partie de leur travail à une série de modules. Le but de cette architecture est de fournir un système modulaire hautement personnalisable.

Le stockage des données est assuré par une base de données (PostgreSQL), un annuaire LDAP et le système de fichiers.

3.2. Intégration

Dans la mesure où ce système est prévu pour s'interfacer avec un système d'information existant (plateforme CRM ...), nous ne stockons pas les clients en base de données. Seuls sont stockés :

- un identifiant unique interne (pour les clefs étrangères dans les autres tables).
- l'identifiant du client dans le système d'information tiers.
- le nom que le client présente dans le header de l'email (le nom qu'il paramètre dans son logiciel de messagerie).

- l'ensemble des adresses de messageries détectées pour ce client.

Quand un nouveau mail est récupéré depuis le serveur POP/IMAP, si l'adresse de l'expéditeur n'est pas trouvée dans la base locale, eMis requête votre système d'information par le moyen que vous définissez (HTTP, JDBC, Webservice). Vous devez juste écrire votre propre implémentation du CustomerHook (un SSB) et le déployer.

Les autres informations concernant les clients sont aussi récupérées depuis votre système d'information pour l'affichage dans l'interface de réponse, pour les intégrer dans les réponses fournies.

3.3. Interface Utilisateur

L'IHM utilisateur est un point important de ce projet. Son ergonomie est déterminante dans la productivité et la qualité des réponses fournies. Nous avons préféré nous orienter vers une interface cliente de type 'client riche' pour disposer de fonctionnalités d'affichage avancées. La plateforme RCP Eclipse nous apporte la modularité : L'application est composée de 'features', chacune étant dédiée à un cœur de métier du périmètre fonctionnel de l'application. Un utilisateur, selon son 'skill', verra son interface s'enrichir des fonctionnalités dont il a besoin pour travailler. Le niveau d'accès des utilisateurs est défini dans un annuaire LDAP.