1st. Volokolamsky proezd, Bld.10
123060 Moscow • Russia
T +7 (495) 967-80-30 • F +7 (495) 967-80-32
www.luxoft.com • info@luxoft.com

Messaging systems

Choosing Messaging System for the Customer's Project

15 pages

Date: 21.02.2007
Version: 1.0

Table of Contents

1. History

1.1 Revision History

Version	Date	Description of Changes	Reason	Made by
0.1	09.02.2007	Created	Customer request for research	Elena Gomonova
0.2	16.02.2007	All sections were drafted		Oleg Moroz Vladimir Yeremin
0.4	20.02.2007	Summary section is developed		Andrey Afanasyev

1.2 Review History

Version	Date	Reviewer	Reference
0.3	16.02.2007	A. Ignatov	
0.5	20.02.2007	A. Ignatov	

1.3 Approval History

Version	Date	Approved by	Signature or reference
0.6	21.02	E.Povalyaev	

2. Introduction

2.1 Purpose

The main purpose of this document is to present the results of the research established by Luxoft to compare several messaging systems and find out the most appropriate solution, which is eminently suitable to business and technical requirements of Customer's project.

2.2 Summary

We evaluated and compared several solutions for notification systems. The choice of the solution depends on the approach:

- To buy the ready to use solution, that can contain a lot of useful components. But not all these components are necessary needed.
- To develop custom solution.

Independently from the chosen solution, you will still have to integrate it into your system.

As a representative of the **ready to use solution**, we recommend using Kenamea's Enterprise Notification System because this product covers all customer needs and has rich additional abilities. Kenamea's Enterprise Notification System (ENS) is best choice for Customer. ENS contains many ready to use modules. The ENS provides immediate visibility and awareness of critical events, enabling users to respond in real-time directly from the notification body, driving revenue-generating productivity. ENS can extremely low installation cost, because the system has all the necessary components and all these components are configurable. The main disadvantage of this system is its high cost. The rough estimate for less than 10,000 users in finance/banking industry is $150,000 per year.

If you want more flexible **custom solution**, we recommend to build a solution based on the Enterprise Service Bus (ESB). This solution should use enterprise messaging system (EMS) (the choice of ESB and EMS for such system is over the scope of current research). Such solution will allow sources to put their notification messages on the bus, providing the necessary information for the bus to perform routing, transformation, and delivery of notifications to the interested parties. Customer is already using the ESB in its systems architecture, so development of a custom notification system can be an interesting solution, because such solution can decrease Total Cost of Ownership (TCO).

We would not recommend creating notification systems as a full custom solution (without using standard ready components). There are many standards for communication between recipients. Depending on your needs, the Heterogeneous solution may contain many different components and there are two approaches to implement them:

- To buy and customize such components
- To develop such components.

The cost of both approaches is high.

2.3 Scope

The scope of this analysis is limited by the explicit list of messaging systems for evaluation, including the following products:

- Kenamea's Enterprise Notification System (ENS)
- JBoss ESB 4.0
- Luxoft In-house Solution

Solutions are evaluated basing on the list of requirement and criteria specified by the customer and listed in sections "Customer Needs" and "Estimation Criteria" correspondingly.

2.4 Definitions, Acronyms and Abbreviations

AJAX	Asynchronous JavaScript and XML
BAM	Business Activity Monitoring
EMS	Enterprise Messaging System is set of published Enterprise-wide standards that allows organizations to send semantically precise messages between computer systems. EMS systems promote loosely coupled architectures that allow changes in the formats of messages to have minimum impact on message subscribers
ENS	Kenamea's Enterprise Notification System
ESB	Enterprise Service Bus
JMS	Java Message Service
Notification System	Combination of software and hardware that provides a means of delivering a message to a set of recipients
SIP	Session Initiation Protocol
SMPP	Short Message Peer-to-peer Protocol
SMTP	Simple Mail Transfer Protocol
SOA	Service-Oriented Architecture
TCO	Total Cost of Ownership
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol
XSLT	Extensible Stylesheet Language Transformations

3. Customer Needs

Currently the conversation between users and notifications about events in the system happens by different ways (depends on stream). It may be mails via SMTP or set of refreshed tables in Oracle system or even need for users to call each other manually.

According to Customer request, the needed messaging system should meet the following requirements:

- Ability for users to be subscribed for receiving information about necessary events
- Ability to create and manage groups of users, including opportunity to put user into and remove user from a group
- Ability for users to send message to the specific user or to the whole group.

There are also some additional, but advisable abilities that messaging system should have:

- Ability to provide information about presence of the users
- Ability to provide client authentication
- Ability to provide convenient way for integration with the LEMG UI tier(HTML/JSP/Struts), for example Ajax system ticker
- Ability to provide Client User Agent or be integrated with existed solutions (Skype, ICQ, Windows Messenger, Jabber, FIDO, RSS)
- Ability to provide a way for integration with 2G networks (via SMPP to SMSC) or 3G/IMS networks via SIP/SIMPLE based solutions
- Ability to provide guaranteed delivery.

4. Messaging Systems Comparison

4.1 Kenamea's Enterprise Notification System (ENS)

http://www.kenamea.com/products_ENS.html

Kenamea's Enterprise Notification System "ENS" is a configurable turnkey notification management system that integrates and interoperates with existing business applications and back office systems to provide a single, consistent and actionable view of all revenue-impacting alerts. The ENS provides immediate visibility and awareness of critical events, enabling users to respond in real-time directly from the notification body, driving revenue-generating productivity. Multi-device delivery ensures immediate visibility, no matter where your users are located. All activity, including collaboration and task resolution, is tracked for reporting purposes.

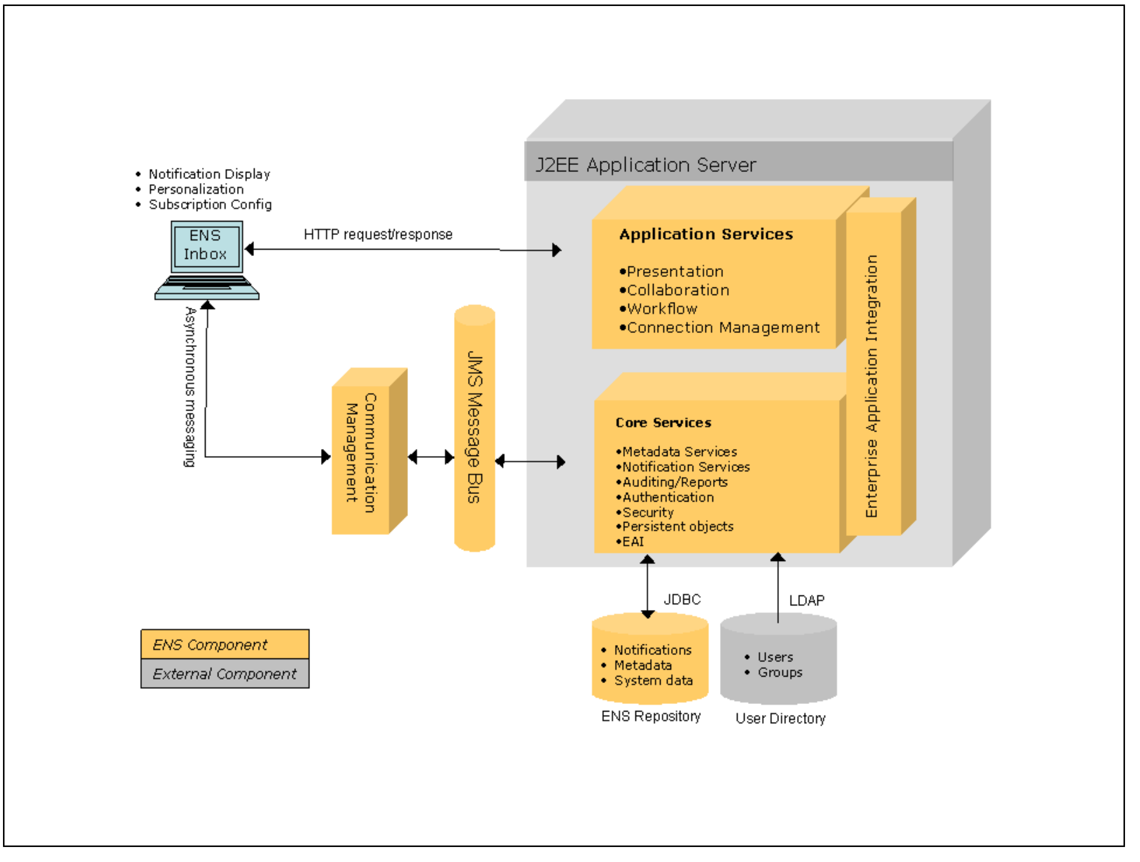
User and group control	Excellent
GIU and interaction	Excellent
Notification channels	Excellent
Price	Poor
Total Evaluation	★★★★★

The ENS provides the following features:

- All notifications from all source systems are managed and controlled in a dedicated Corporate Inbox, personalized for each user
- Transformation of flat paper and electronic alerts into actionable notifications for immediate response
- All activity on each notification is tracked and stored on the notification body and a central repository for compliance reporting
- Each notification has a unique lifecycle and stages from inception to resolution, as dictated by the client. The ENS provides real-time visibility into the state/lifecycle of each notification
- Guaranteed notification delivery
- Secure, encrypted delivery
- Multiple device delivery - dedicated inbox, PDA, email, phone
- Real time collaboration in context of notification topic
- Publish/subscribe and point-to-point delivery
- Task Management
- Flexible Formatting: Handles Any Message/Alert/Notification
- Simple integration to any source
- Zero client, AJAX-style browser user interface

- Collaboration and productivity features:
  - Notes, reminders, and attachments
  - Status, priority, and user addressing
  - Embedded content and applications

The high-level system architecture can be depicted as follows:



Kenamea ENS Architecture

ENS is a complete solution that provides most of the necessary functionality out of the box (some small development effort may be needed for integration with JSP/Struts/AJAX-based UI). It can be deployed either as an in-house system or on the premises of the vendor as an ASP.

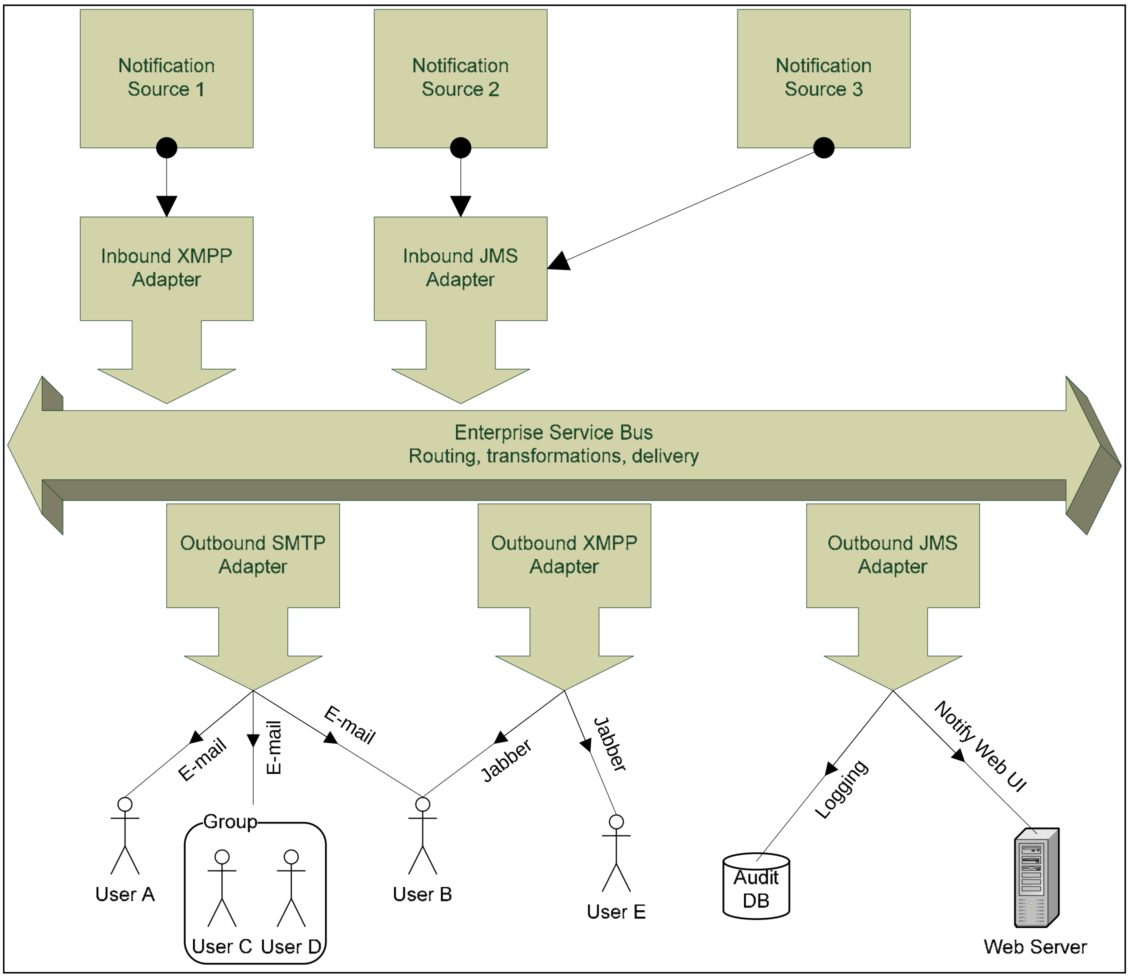
ENS pricing varies with the total number of users, integration points, customer's industry, and many other factors; the rough estimate for less than 10,000 users in finance/banking industry using integration points mentioned in the business requirements is $150,000 per year (hosting – if needed – and SOW work are separate).

4.2 Custom development based on ESB

Enterprise Service Bus (ESB) is one of the core concepts of Service-Oriented Architecture (SOA). ESB connects, mediates and controls all communications and interactions between services and provides error and exception processing. Its design permits rapid change in services and in the connections among services, and provides the management visibility into services and processes across a distributed environment.

ESB can be used as the core of enterprise-wide notification system, where sources put their notification messages on the bus, providing the necessary information for the bus to do the routing, transformation, and delivery of notifications to the interested parties, including gateways to e-mail, IM, and phone messaging for actual delivery to intended recipient, as well as web UI notification subsystem, logging subsystem, Business Activity Monitoring (BAM) subsystem, etc.

The high-level architecture of the proposed solution will probably look like the following diagram:



This approach needs much more development effort than the use of off-the-shelf notification solutions, because ESB typically provides only the underlying message bus and some of the adapters and transformers (such as JMS input adapters and XSLT transformers), but nevertheless it can be more cost-effective, especially compared to high-end solutions (ENS).

The following components will need to be implemented for this alternative:

1) Standard formats / schemas for notification-related messages on the bus (notification, acknowledgment, status monitoring, administrative, etc.)

2) Inbound adapters for all protocols that can be used by notification sources (such as Jabber XMPP), except JMS and other standard MOM protocols, and transformers for input message formats where the standard format cannot be used.

3) Java library / component to be used for uniform notification source access to the bus (optional).

4) User / group management component for use in outbound adapters.

5) Outbound adapters for all user notification protocols, including SMTP (e-mail), XMPP (Jabber), SMPP (SMS), SIP (3G network messaging).

6) User inbox component (optional).

7) Web UI components for access to the inbox and displaying real-time notifications (Struts/AJAX, portlet).

8) Logging / auditing component (optional)

9) Message rendering and template management (optional).

4.2.1 JBoss ESB-based solution (JBoss ESB 4.0)

http://www.jboss.com/products/esb

JBoss ESB 4.0 is a good example of open-source J2EE-based ESB is JBoss ESB 4.0. The key features of this product include:

- A pluggable architecture that enables all JBoss ESB subsystems such as messaging and transformation to be swapped with other alternatives, which gives customers flexibility and choice.
- Support for a variety of messaging services, including secure FTP, HTTP, e-mail, and JMS (JBossMQ, JBoss Messaging, IBM MQSeries, and ActiveMQ).
- Transformation engine that bridges data formats for seamless communication, supporting XSLT and Smooks, a flexible alternative.

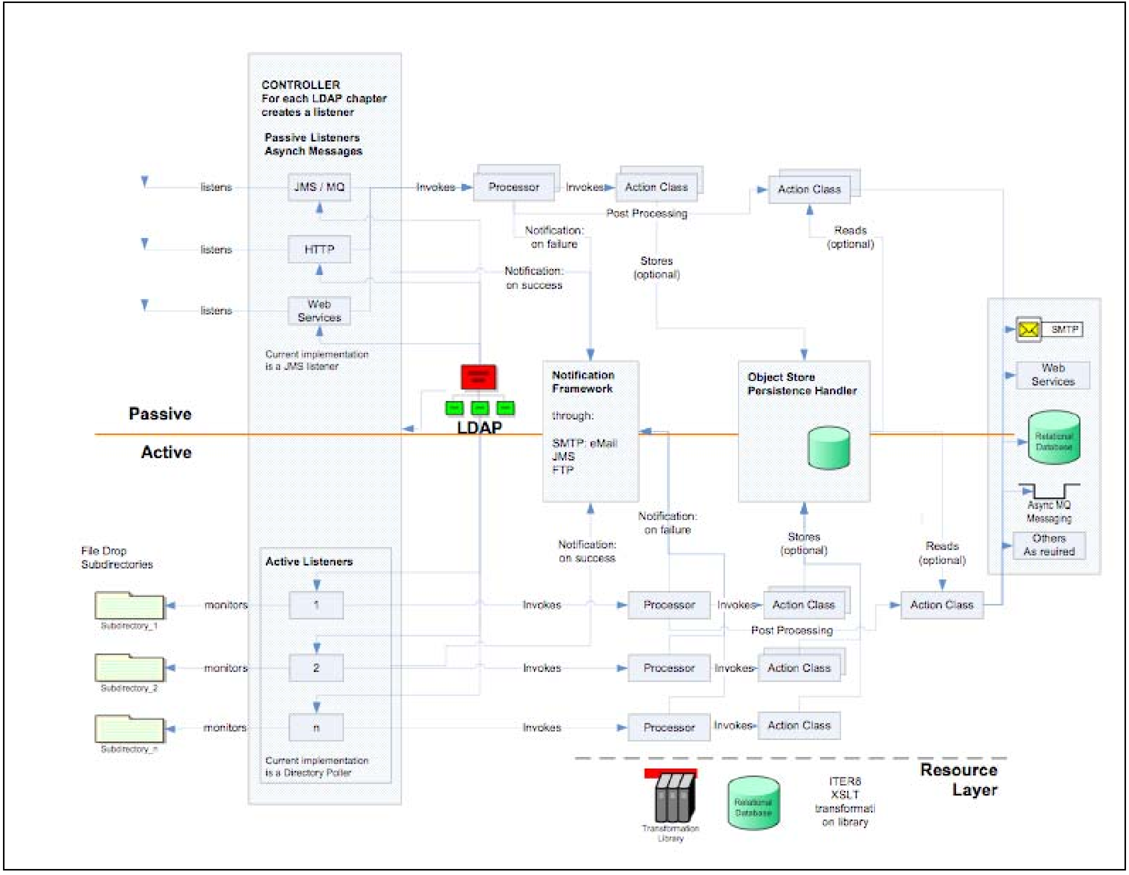
User and group control	Poor
GIU and interaction	Poor
Notification channels	Poor
Price	Excellent
Total Evaluation	★★

- Service registry for service discovery and integration, using JAX-R and UDDI.

- Persisted event repository to support governance of the ESB environment.
- Notification service to allow the ESB to register events and signal subscribers.
- Content-based routing based on XPath and JBoss Rules for a more flexible and dynamic alternative to publish-subscribe.
- Gateways that allow non-ESB aware clients to interact with services deployed within the JBoss ESB environment.

JBoss ESB also supports some notification-specific functionality, such as outgoing SMTP adapter, XSLT, and notification interfaces and configuration infrastructure in the core. In JBoss ESB 4.0 these interfaces are limited and mainly aligned with the use case for notifying a list of targets by e-mail or JMS in case of specific events taking place on the bus (such as exceptions and important lifecycle occurrences) though they can be easily generalized to support customer's requirements.

The components and interactions in the typical JBoss ESB deployment are depicted in the following diagram.



Typical JBoss ESB Deployment

JBoss ESB requires JBoss application server 4.0.x to run. The open-source license is free and commercial support can be bought from JBoss / RedHat with JBoss Subscription program.

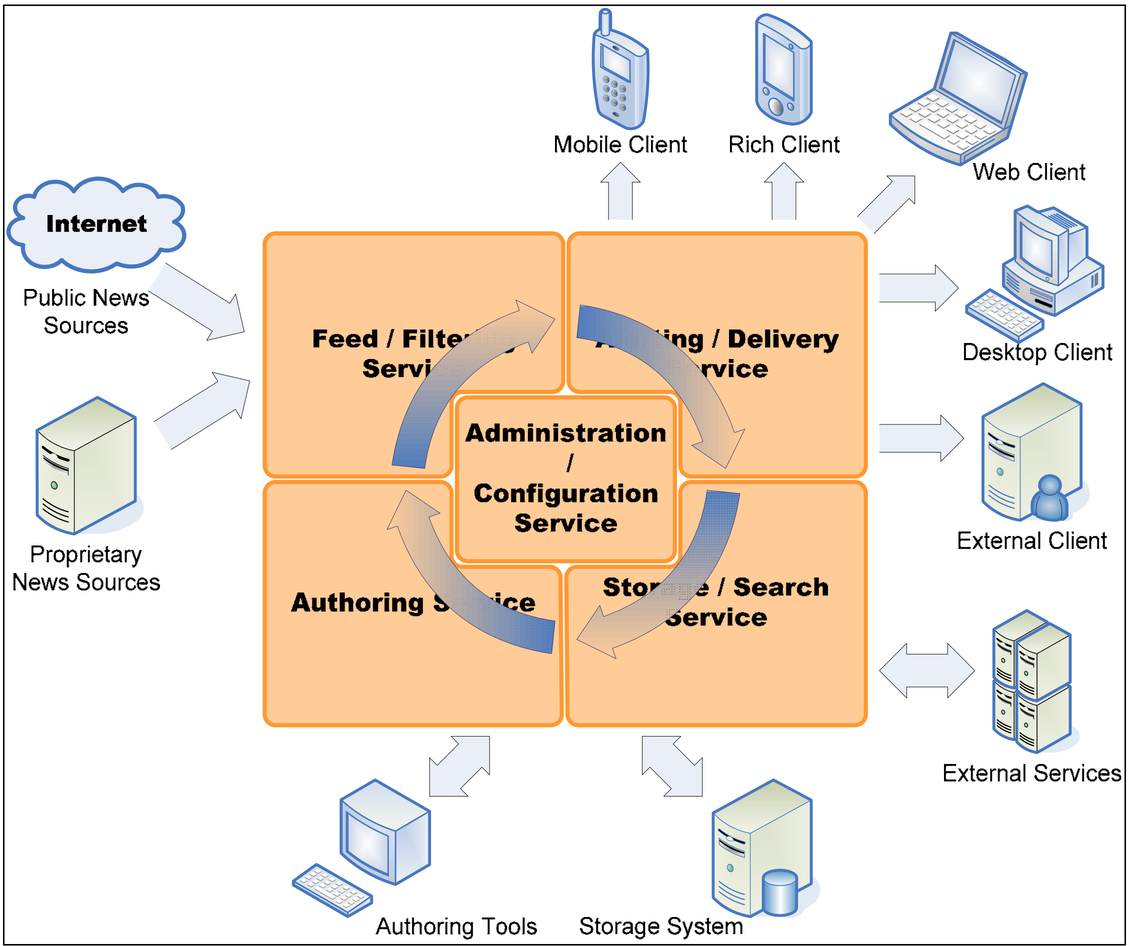
4.3 Luxoft In-house Solution

Luxoft solution (code name “Confeedence”) is a configurable information aggregation system that includes both “pop” and “push” models to deliver business critical information to the customer.

This solution includes special services that receive information from the external sources (via RSS/Atom feeds, or using custom adapters to external data sources), processing engine (that filters and spread information using configurable rules), then its information can be routed, published (as RSS/Atom feed) and delivered to users using different methods (e-mail, IM, SMS, etc) and delivery rules.

User and group control	Good
GIU and interaction	Satisfactory
Notification channels	Satisfactory
Price	N/A
Total Evaluation	★★

The following picture illustrates the main idea of the solution:



“Confeedence” project is designing as an enterprise solution. It means that system will allow integration with existing account management systems (using LDAP). It will include a set of adapters to integrate with existing data/information sources. It has extendable architecture (using plug-ins and API) to fit into existing enterprise infrastructure seamlessly.

"Confeedence" features include:

- System can use any source of data that publish information using RSS/Atom feeds. Additional adapters can be used to convert non-syndicating information (like changes in the relational database tables) to XML feeds.
- All subscriptions to notification are managed inside single administrative application.
- The system uses two models: role-based and object's level securities to secure information and share responsibilities.
- The system supports public and private feeds to restrict access to secure information.
- The system makes it easy to find feeds with a customizable taxonomy (hierarchy of feeds), based on article ratings and current subscriptions. Employees can collaborate by clipping and annotating items and allowing others to subscribe to that feed
- Users can subscribe to particular information as well as an authorized user (Administrator) can subscribe other users / groups in order to send them notifications.
- The system supports event driven feeds with publish / subscriber model.
- Notifications can be delivered to different type of devices (desktop, PDA, Blackberry, mobile phone, etc)
- The system support guaranteed delivery on predefined set of channels (IM, SMS).
- The system will have different clients (desktop, reach, web) to work with information feeds. It will support full synchronization between them.

The system is implementing as a J2EE application using Open Source products (MySQL, Apache TomCat, Apache Active MQ, Hibernate, etc). It allows reduce TCO for large deployments.

4.4 Messaging Systems Description Matrix

We reviewed the messaging systems according to the Customer needs (described in section "Customer needs") and created the comparison matrix (see Appendix) to facilitate the analysis.

Appendix. Messaging Systems Description Matrix

Features	Kenamea ENS	JBoss ESB-based solution	Luxoft In-house Solution
Subscription for events	Yes	Yes	Yes
User groups, group management	Yes	Needs development	Yes
Sending to user or to group	Yes	Yes	Yes
User presence information	Yes	Needs development	Planned
Client authentication	Yes	Yes	Yes
UI integration with JSP/Struts/AJAX	Needs development	Needs development	Planned
UI integration with JSR-168 portlet	Yes	Needs development	Planned
Separate Client User Agent	Yes	No	Planned
Notification by e-mail	Yes	Yes	Yes
Notification by IM	Jabber	Needs development	Planned
RSS support	Yes	Needs development	Yes
Notification by phone (SMS / 3G)	SMS, SIP integration	Needs development	Planned
Guaranteed delivery	Yes	Yes	Planned

Features	Kenamea ENS	JBoss ESB-based solution	Luxoft In-house Solution
Additional requirements			
Integration with GA	Needs development	Needs development	Needs development
Logging for BAM and audit	Yes	Yes, possible modifications	Planned
Sequentially processing notifications from single source	No	Needs development	Planned
Template-based message rendering	Yes	Needs development	Planned
Price	$150,000/year for less than 10000 users	Open-source, free	N/A