Oracle Email Server

Rule Engine White Paper

Introduction

As the Internet population grows at an astounding rate and ecommerce becomes a more popular way of doing business, more and more businesses rely on the Internet to attract, retain and communicate with online customers. Electronic mail is one of the most reliable, convenient and familiar tools for the Internet community. The increasing amount of information available on the Internet dramatically changed the way eMail is being used. From a tool of exchanging personal conversation online, eMail is becoming a medium for everyday business transactions.

It is becoming important for an eMail server to be able to not only store and transport large amount of messages efficiently, but also to manage, analyze and process these mail messages intelligently using business logic. Oracle Email Server ™ is a powerful and flexible mail platform that meets these demanding requirements by providing a built-in rules engine. Powered by the industry leading RDBMS Oracle8i, the rules engine can perform customized actions on eMail messages based on pre-defined rules. With eMail messages stored as relational data, this mechanism allows other database applications to easily integrate with the Email Server and apply consistent database transaction logic to both mail data and external data.

Email Life Cycle

An eMail message moves through several stages in its life cycle. From the point of view of one eMail server, an inbound eMail message goes through the following states during its lifetime in that particular mail system:

- **Arrived**: when an eMail message is first transported from a remote mail server to the local system, it stays in a queue awaiting delivery to its designated recipients.
- **Delivered**: the eMail message is delivered by the local delivery agent to each of its possibly many recipients, so that each recipient gets a copy of the mail in his/her inbox.
- **User processed**: once a mail message stays in a user's mailbox, it is under the complete control of the user. During this time it can be filed, tagged or otherwise manipulated by the user.
- **Deleted**: eventually if and when the eMail message is deleted by the user, the system cleanly removes all of the data associated with that eMail.

An outbound eMail message goes through the following two states:

- **Composed**: a user composes the message using a mail client and hands the message to a mail transport agent by pressing "Send Mail" or some similar action.
- **Departed**: the transport agent then compiles a list of target mail hosts where the recipients of the eMail are located and transports the message to each mail host.

As illustrated in Figure 1, messages go through the system in the order labeled on the paths, starting from its composition on the client side all the way to finally being deleted into the system trash can.

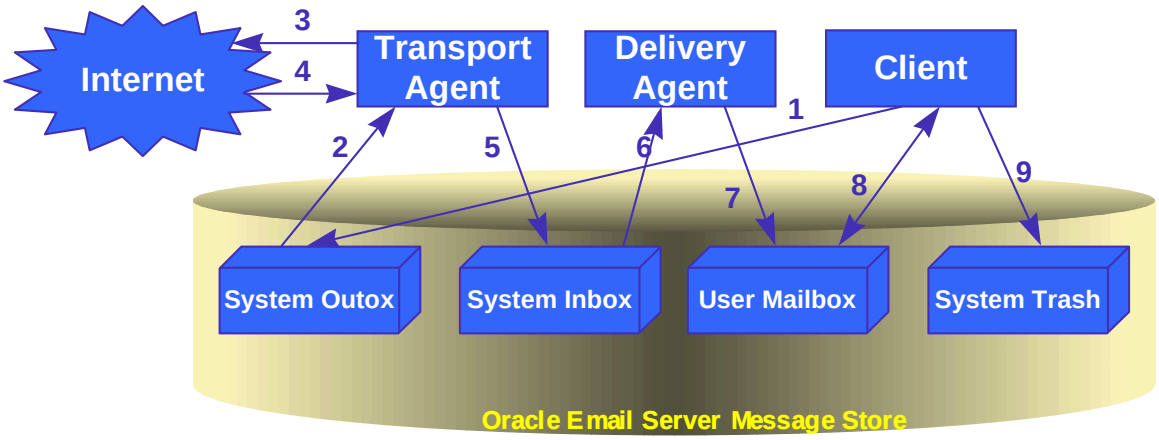

Figure 1

Where Rules Engine Fits in

When an eMail message changes from one state to another during its life cycle, it is called an "event." Rule actions are triggered by events during the life cycle of eMail messages. When an event happens, the rules engine will start evaluating rules defined for that particular event. For example, when a message state changes from “Arrived” to “Delivered”, the subsystem that is responsible for the change (in this case, the delivery agent) will call the rules engine to perform rule checking and rule execution for rules defined for the “Delivery” event.

In the rule execution stage, message attributes and header values are retrieved and matched against predefined conditions in order to determine if certain action needs to be taken. Conditions are logical combinations of questions such as “Is this message bigger than 10MB?”, “Does the subject of this message contain ‘get rich fast’?” or “Is the sender from ‘aol.com’?”. If the incoming message matches the specified condition(s) of the rules for that event, then associated action will be carried out before the message appears in its new state.

“Server Side” vs. “Client Side”

The advantage of the Oracle Email Server rules engine is 100% server side functionality. Various mail clients provide features called “mail filtering” to mail users. Mail filters allow users to set up criteria so that certain eMails are automatically filed or deleted based on attributes like mail subject or sender. Mail filtering is totally implemented on the client side. While useful to some extent, mail filtering does not have the following advantages of server side rules processing:

- **Client independence**: in today’s mobile computing age, more and more people need to check their eMail from multiple clients. While a desktop mail client is ideal for the office, PDAs, pagers or the Web become more convenient mail tools when a user is on the go. Client side mail filtering cannot transfer from client to client, and many clients do not have this feature at all. Therefore the usefulness of mail filtering is limited. On the other hand, Oracle Email Server's server side rules engine applies rules independently of which client a user chooses to use.
- **Continuous execution**: Client side filters are only executed when a user is actively checking email or at least keeps the client connected. Therefore there is no filter execution when the user is disconnected. Oracle Email Server's server side rules execute so long as the mail server is up and running
- **Increased functionality**: the rules allowed in client side filtering are significantly limited by the fact that clients are owned by individual users and not the system. It is not possible to set up rules that can be applied to multiple users, or rules that are triggered by an event other than new mail arrival. Oracle Email Server's server side rules provides flexible mechanisms to integrate with external applications, which is not possible with client side rules.

Built-in Functionality

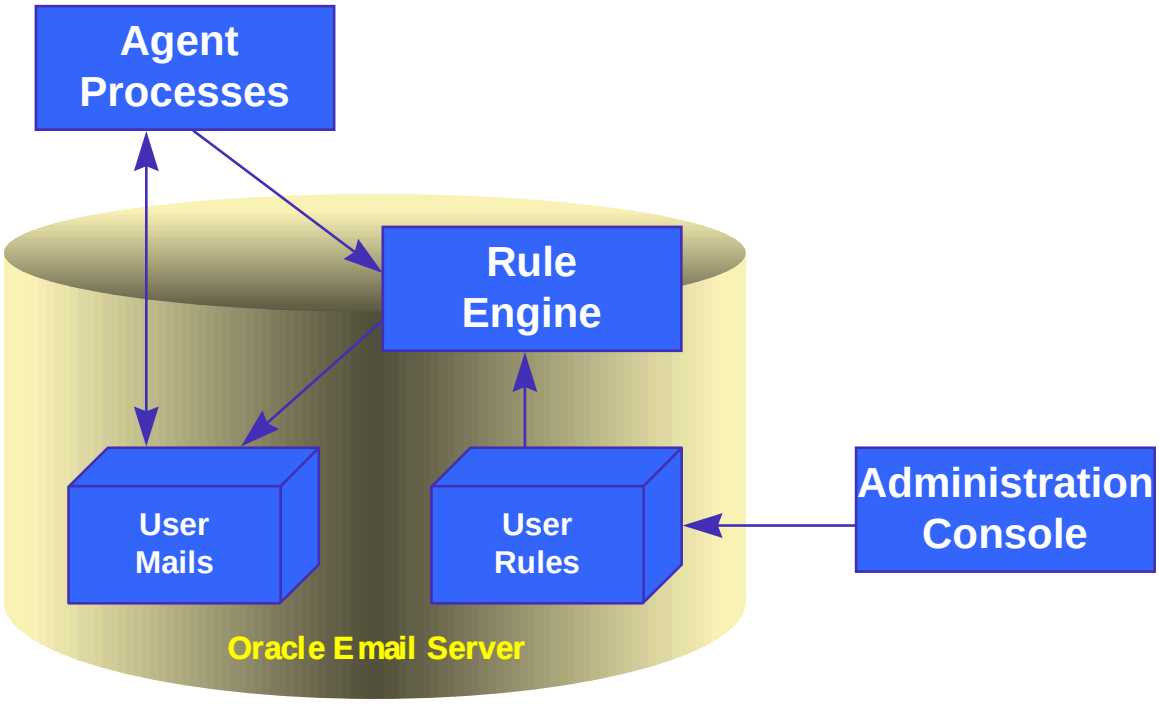

Figure 2

As illustrated in Figure 2, rules are stored as user-specific data along with users' mails, Oracle Email Server's rules engine performs rule logic and execution when called upon by agent processes such as a delivery agent. The rules are maintained by administrators over an administration console.

The rules engine provides a configuration interface that allows rules to be set up at either the user level or the system level. Administrators or application developers can use this interface to define complex conditions and associate a variety of actions with them.

Conditions are defined as either a simple comparison between a message attribute and a value or logical combinations of multiple attribute matches. Attributes of a messages that can be compared include sender address, sender's real name, "TO" and "CC" recipients, subject, date, priority, and size.

The Oracle Email Server rules engine provides a set of built-in actions that can be associate with conditions defined, these actions can be setup to process the incoming message on behalf of the rule owner. The list of actions includes, but is not limited to:

- Blind copy the message to a another recipient
- Forward the message to a new address
- Generate auto-reply to the message sender
- Mark the message high (or low) priority
- Delete the message
- Move the message to a specific folder
- Copy the message to a specific folder while retaining the original message in the INBOX

Aside from built-in actions, the rules engine also provides a mechanism to plug in third party PL/SQL procedures as actions residing in a separate schema space or even a separate database. This allows for even greater application integration flexibility. By using external customized action features in Oracle Email Server ™, any Oracle database application can support transactions or queries triggered by eMail.

Applications of Rule Engine

The rules engine feature of Oracle Email Server can be used in many ways to satisfy a variety of business requirements. For example:

- A Corporation with stringent eMail security requirements can use Oracle Email Server to monitor and blind copy eMail traffic selectively, using the rules engine, so that all suspicious eMails can be logged into one special account for later reference.
- A Company with a multi-level eMail approval process can set up rules that routes all eMails to the requestor's supervisor before emailing the note to the final approver. Another set of rules can be set up so that mails are automatically sent to the original sender once approval has been obtained.
- An Internet service provider with anti-spamming requirements can set up system-wide rules that messages with offending senders or subjects are rejected right away.
- A mailing list server provider that must enforce various degrees of security and moderation requirements based on sender address can use the rules engine to set up a public mailing list server to enforce those rules as well as forward mails to an expanded list of recipients.
- A customer support center that needs to handle huge amounts of eMail support inquires can set up rules to dispatch customer emails to different service representatives based on eMail subjects.
- An e-commerce company can process email based order status request by setting up rules to automatically generate status notification eMail back to the requester.

Example Application

To illustrate the use of the Oracle Email Server rules engine in a specific application, let's design a eMail-based resume repository application for a company named Resumes-R-Us. The company has the following requirements:

- The system accepts resume submissions via eMail from potential job seekers. The resume is then stored in the database with key information extracted and stored separately such as candidate qualification, geographical preference, salary demand, etc. Upon receiving a resume submission, the sender is notified to acknowledge the reception via eMail.
- The system also accepts resume inquiries via eMail from potential employers asking about candidates with specific job qualifications along with other criteria. Upon receiving such requests, the system generates a list of resumes based on the employer criteria and sends the resume list back to the requester. At the same time, every candidate on the list is notified via eMail about which employer has just requested his/her information.

This application can be built using Oracle Email Server with rules engine functionality in the following steps:

- Design resume storage schema to store individual resumes and their extracted key information using Oracle RDBMS.
- Install Oracle Email Server and create two mail accounts submit@resumesrus.com and inquire@resumesrus.com.
- Create a rule for submit@resumesrus.com so that for every message with an attachment named "resume", call up an external procedure and hand the mail over to that procedure. This procedure applies business logic to the mail body in order to extract and store the resume in the schema.
- Create another rule for submit@resumesrus.com such that for every message with a resume attachment, generate a auto-reply and notify the sender that the company has received the resume.
- Create a rule for inquire@resumesrus.com that for every message, call up another external procedure and hand the mail to the procedure. This second procedure first extracts inquiry criteria, matches them against the resume database, then generates a list of resumes and associated email addresses. The procedure then combines the resumes and sends the list back to the inquiring employer, and also sends eMail notifications to each of the resume email addresses on the list.

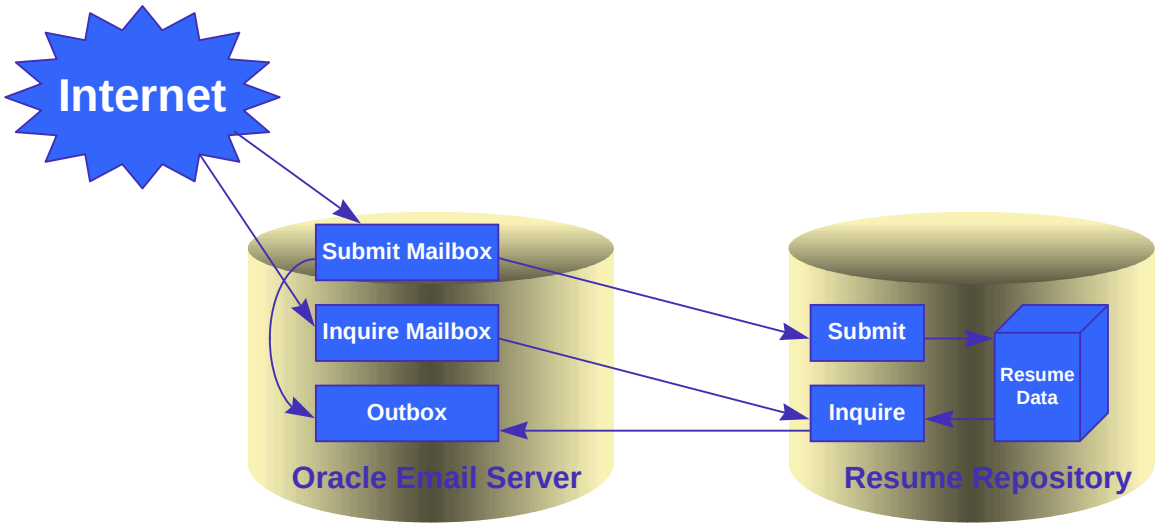

Figure 3

As illustrated in Figure 3, the resume repository application may reside on a separate database. When eMail arrives at the Oracle Email Server, the rules engine communicates with the resume repository application via external procedure calls to update and retrieve from the resume database.

Conclusion

The information explosion on the Internet demands intelligent data processing. Email is a proven, reliable and convenient medium to transport information on the Internet. Because eMail is increasingly used to perform business online, intelligent, automated processing of eMail messages is a necessity for e-commerce. Oracle Email Server ™, offering the open connectivity of standard based eMail and the flexibility of the Oracle Email Server rules engine, is the ideal eMail solution for today's Internet business.