

# **eMobile Application Demonstrates End-to-End J2EE™ 2 Interoperability**

**By the MDE Enterprise Java Team  
Sun Microsystems**



Copyright 2000 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303 U.S.A. All rights reserved.

This document and the product to which it pertains is distributed under licenses restricting its use, copying, distribution, and decompilation. No part of the product or of this document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Sun, Sun Microsystems, the Sun logo, Java, the Java Coffee Cup, J2EE, Enterprise JavaBeans, EJB are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries.

Federal Acquisitions: Commercial Software -- Government Users Subject to Standard License Terms and Conditions.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2000 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, Californie 94303 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Sun, Sun Microsystems, le logo Sun, Java, le Java Coffee Cup, 2EE, Enterprise JavaBeans, EJB sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



Please  
Recycle



Adobe PostScript

# Introduction

The Internet has changed the rules of developing applications. Enterprises want to develop and deploy quality applications quickly and at low cost to create new business opportunities and solve business problems. Typically, these applications are constructed by combining existing services to perform higher-level, more complex tasks, and are often designed using a multi-tier, distributed object architecture.

Today, access to the web is not only limited to desktops like PCs or workstations, but can be accessed from any hand held devices like the Psion, Palm Pilots, cell phones, etc. In the Internet age, information is critical. Employees, customers, and suppliers need access to data 24 x 7, no matter where they are located.

The J2EE™ architecture provides a standard framework for enterprises to develop and deploy end to end e-commerce applications more rapidly, using any J2EE Application server.

The eMobile application, based on J2EE technologies, can be accessed from cell phones or Java™ technology-enabled devices, like Psion or Palm Pilot, and breathes new life into existing systems, such as CORBA services.

## About eMobile Application

eMobile application is based on J2EE technologies like Enterprise JavaBeans™ (EJB™) technology (version 1.1), JavaServer Pages™ (JSP™) technology (version 1.1), and servlet (version 2.2), and can be deployed on any J2EE-compliant servers including the J2EE Reference Implementation.

eMobile is a sample application that demonstrates the use of J2EE technology by modeling an online car-buying service. It allows an end user to:

- Search for cars based on preferences
- Select and buy a car
- Obtain financing information for it
- View videos for specified cars.

The eMobile application leverages CORBA-based system, and integrates new technologies such as Java™ Messaging Service and XML-based technologies: XSL/XSLT, SAX, and DOM parsers.

## eMobile Application Modules

The eMobile application is architected as a multi-tier modular model, where each module is responsible for a specific functionality. Diagram 1 show an overview of the eMobile architecture.

The core components are responsible for the major functionality of the application: browsing and buying cars. The Content Tracking System and Notification System represent the business-to-business functionality where email is outsourced to another application.

The eMobile application can be accessed from multiple clients, including browser-based clients, such as a set-top box, cell phones, or Java technology-enabled devices like Psion or Palm Pilot.

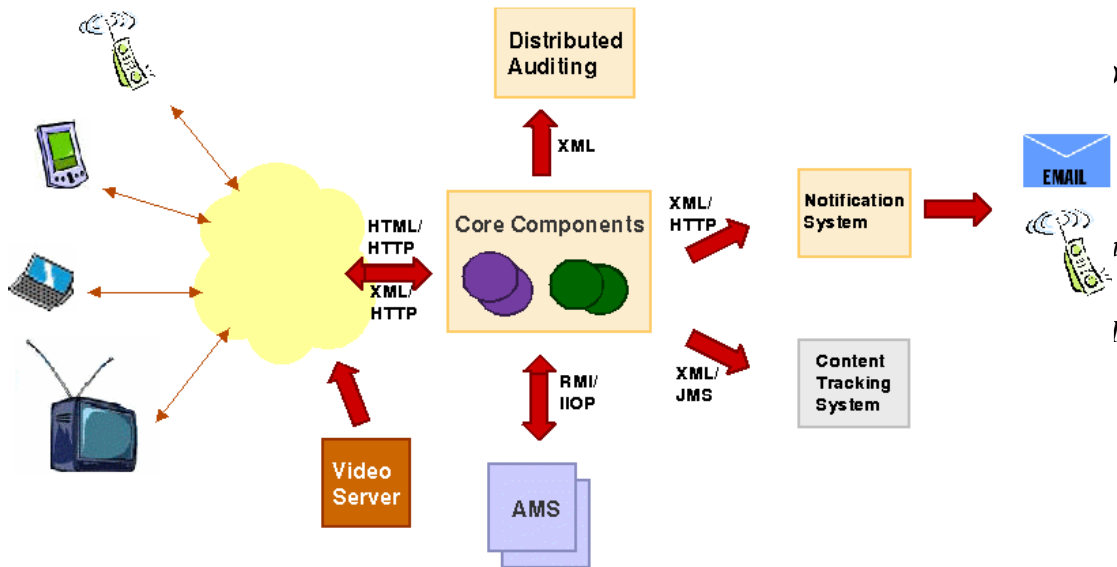


Diagram - High-level overview of the eMobile application

## Core Components

The eMobile core components are made up of Enterprise JavaBeans technology, Java-Server Pages technology, and servlets.

- **Controller** is modeled as a stateful session bean and controls the flow of data on behalf of the user. Upon its creation, Controller finds the nearest Agent to be used for the car browsing/purchasing, and finds (and binds to) the AMS applications.
- **Agent** is modeled as an entity bean and acts as the agent for all dealers, located in a certain geographical criteria, such as state.
- **Finance** is modeled as a stateless session bean and calculates the payments for given car(s). The Finance bean gets the best interest rate by searching specific URIs specified in the deployment descriptor, and then calculates the quotes for 2, 3, and 4 years given a specified down payment.
- **Dealer** is modeled as an entity bean and represents a car dealer, including its inventory. This coarse-grained object allows its client to get a collection of cars that meet a certain criteria, buy one car, and to get the dealer information.
- **JavaServer Pages and servlets** present the user interface of the application to the customer. They allow the user to browse for cars that meet a certain criteria, create an interest list, and purchase a car. The JSPs also let the user get the videos, the financial information, and the dealer information for the cars in the interest list.
- **Service** is a stateful session bean that is used by the management console.
- **CustomerInfo** is a stateful session bean used to maintain the customer information. When the state of a car changes, the CustomerInfo bean sends out notification to the customer.

## Distributed Auditing

Keeping track of the flow of data is a complex task in distributed applications. eMobile application demonstrates the use of a distributed auditing system, which is made up of two parts:

- **EJBLog** is the helper object used by the enterprise beans to log the method calls, and to trace the data flow along the application. EJBLog is modeled as a JavaBean, and send XML messages over HTTP to a servlet for each event to be logged.
- **EJBLogServlet** is the servlet that receives and processes the XML message from the eMobile components.

## Asset Management System (AMS)

The Asset Management System (AMS) is used by the Controller to obtain information about the video content available for a specific car. The AMS keeps track of all the available video content. The video content is stored in a video server. The AMS stores information about the car(s) along with information about the video content, so that the information about the video content for a particular car can be easily looked up.

## Notification System

The Notification System is responsible for notifying the customer of changes to his order. The notification system is implemented as a servlet and receives XML messages over HTTP for each change of state for a car. The servlet then creates and sends a notification message to the customer, via email, phone alerts, and/or regular mail.

## Content Tracking System

The Content Tracking System (CTS) keeps track of which videos are viewed by the user. The CTS, along with the eMobile application, is a B2B system. The two applications are loosely coupled, and the eMobile application communicates to the CTS using asynchronous XML messages sent using JMS.

## Clients

The clients interact with the application via JavaServer Pages or servlets. The content, which is dynamically generated by the JSP or servlets, must be in a format supported by the requesting client. The identification of the capabilities of a requesting client means, in our case, specifically identifying the proper format in which to respond. The selection of the proper content type is made according to a configuration table, which maps a given category of clients or an advertised or accepted MIME types to a particular MIME type, that MIME type being used as the response content type. Technology used in eMobile Application

## Technology used in eMobile Application

The eMobile application uses the following:

- J2EE platform: EJB technology, servlets, JSP technology, JavaMail™ technology, JavaIDL
- Java Messaging Service
- Java API for XML: SAX and DOM parsers
- Java™ 2 Standard Edition (J2SE)
- Java™ 2 Micro Edition (J2ME)
- CORBA C++

## Conclusion

The J2EE platform provides a prerequisite set of Java APIs and services to support the development and deployment of enterprise applications. The J2EE platform is based on proven, open standards to deliver the broadest adoption and highest level of portability. The J2EE platform allows dramatically shorter product cycles, so that new ideas go from concept to reality in days and weeks, not months and years.

The eMobile Application demonstrates that it is possible now to create an end-to-end web-based ecommerce application based on open standards, which breathes new life into existing investments. New technologies, such as Java Messaging Service and XML, and existing technologies such as CORBA, can be seamlessly integrated into a J2EE application to create an enterprise-class, complex application.

Please come and see the eMobile application at JavaOne<sup>SM</sup>. eMobile is shown in the Market Development Engineering booths at JavaOne: booths #1039 and #342.