From Open Source to Scalability and Beyond!

Venkata Achanti, PhD, Senior Consultant

Chris Beal, Senior Manager

Lie Jian Li, Senior Consultant

Cap Gemini Ernst & Young, Atlanta, GA

HP WORLD 2003
Solutions and Technology Conference & Expo

Agenda

- Introduction
- Problem Statement
- Implementation Drivers
- Development Tools & Logistics
- Open Source for Development & BEA for Deployment
- Migration from Open Source to BEA
- Summary of Experience Gained
- Questions?

Introduction

Speakers from CGE&Y will share their experiences, the pros and cons and some techniques for leveraging open-source application server technology for development while planning and executing a migration to the BEA Systems application platform suite.

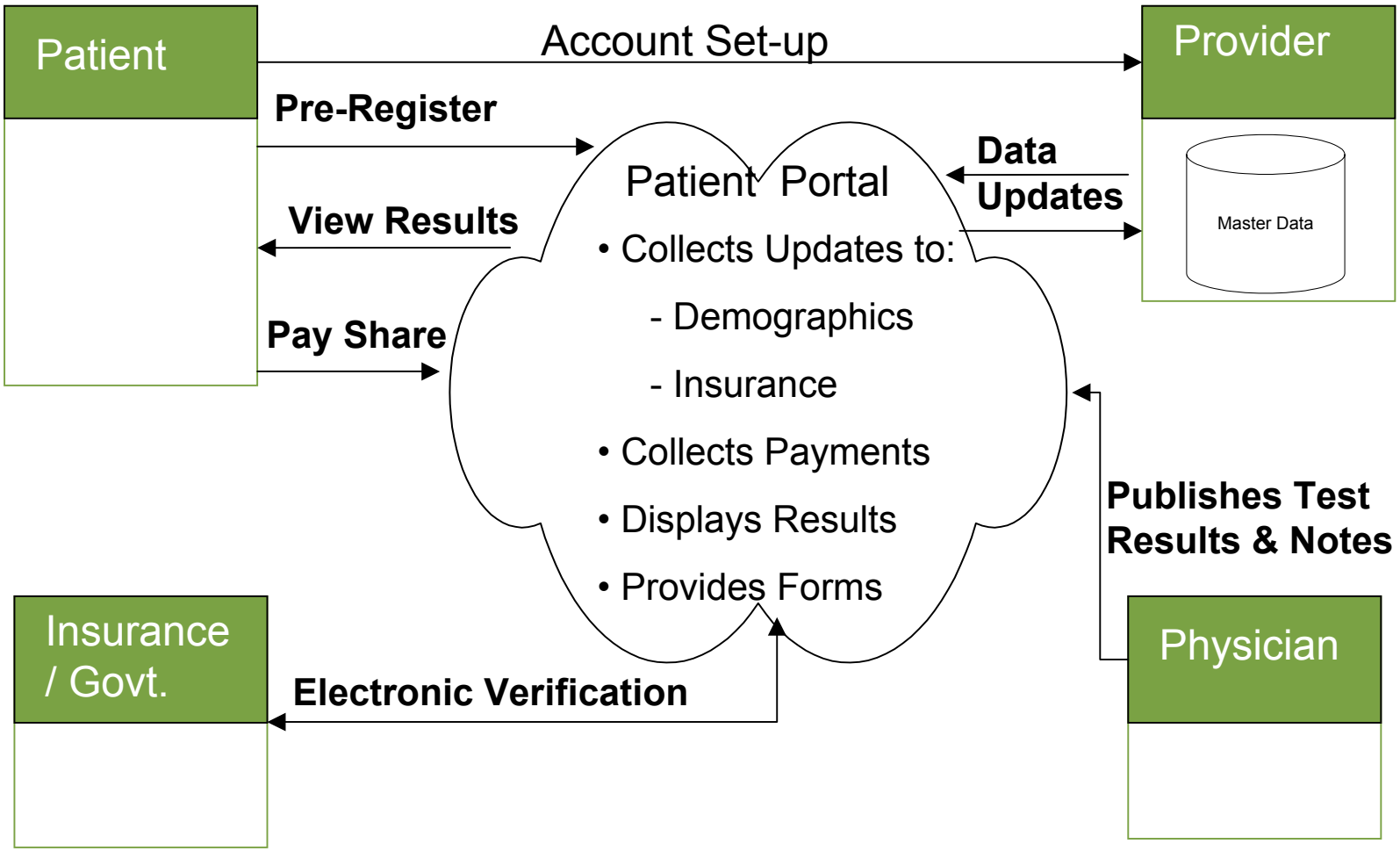
- Venkata Achanti
  - *Project Role: Development Lead & Offshore Liaison*
- Chris Beal
  - *Project Role: Engagement Director*
- Lie Jian Li
  - *Project Role: Team Lead*

Problem Statement

A large healthcare client's need to develop and implement an Internet-based application with three primary functions that enabled their patients to:

- Pre-register for services
- View results of diagnostic tests
- Pay outstanding amounts due

Problem Statement Domain



While Problem is Well Defined, Presented Several Challenges…

Application development under:

- Tight Budget
- Aggressive Timeframes
- Rapid Deployment Needs
- Offshore Development

Migrate seamlessly from Development Environment to a Production Platform

(while production platform preparations were ongoing, client wanted to jump start with development in open source environment and migrate the application seamlessly to production environment)

Implementation Drivers

Our approach consisted of J2EE based application framework that leveraged:

- Component Based Application Development
  - 45 Use Cases, over 250 Objects & 300 JSP's
- Part Offshore, Part US-based
  - to reduce investment in development effort
  - to leverage expanded development cycle time (20 Hrs Vs 10 Hrs a day)
- Develop, Test, Integrate on Open Source and Migrate to BEA Platform for Deployment

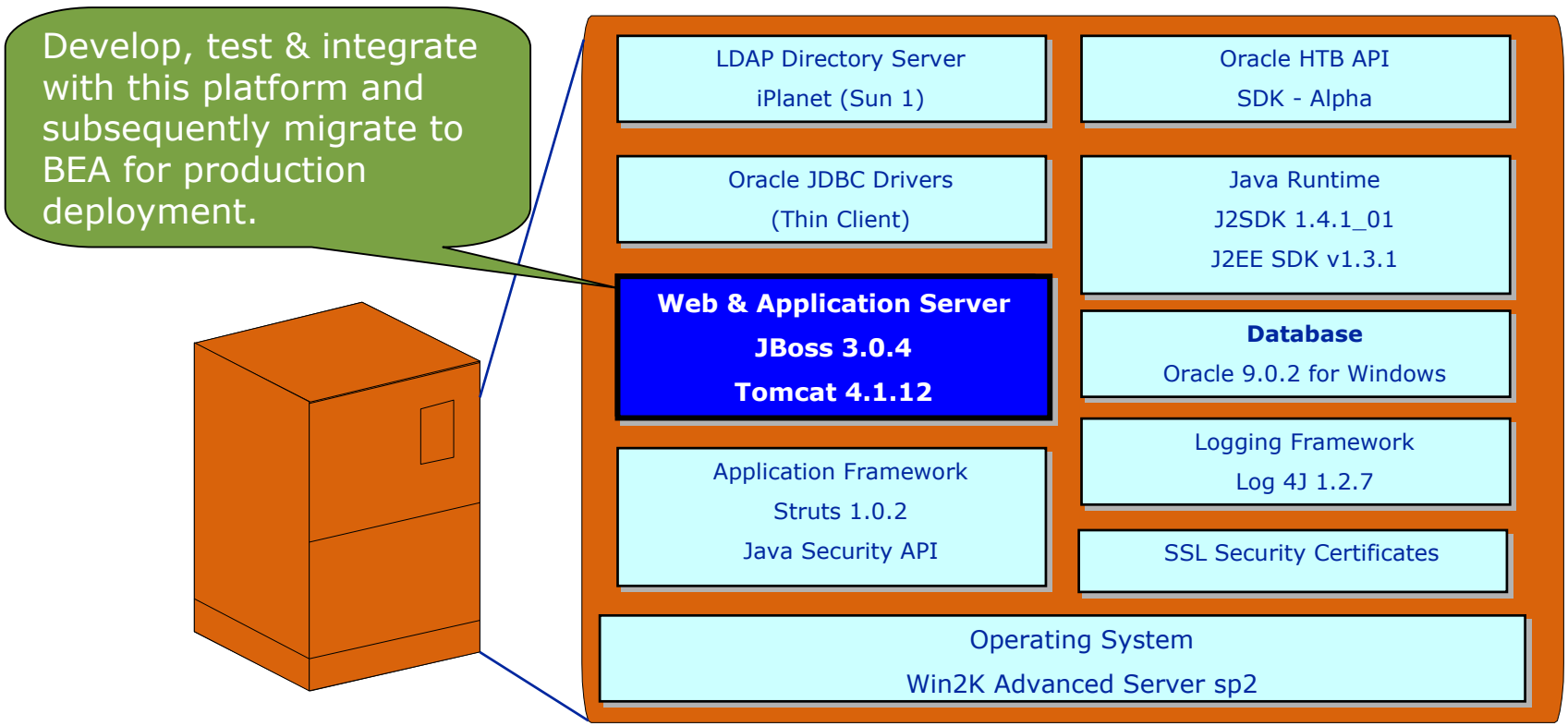
Development Tools

Our approach consisted of J2EE based application framework utilizing:

- Tomcat
- JBOSS
- Oracle 9i
- Log4J
- Ant
- J2EE SDK
- Struts
- Oracle JDeveloper
- Toad

Application Logistics

Our development environment leveraged free tools at our disposal & avoided new investments.



JBoss Open Source

JBoss is a freely downloadable application server software and supports following functionality:

- JBoss/Server–EJB Container & JMX Infrastructure
- JBossMQ for JMS messaging
- JBossMX for mail
- JBossTX for JTA/JTS transaction
- JBossSX for JAAS based security
- JBossCX for JCA connectivity
- JBossCMP for CMP persistence

Using Open Source for Development

Open Source based development enabled us with:

- Flexible & Rapid Development Environment
  - Each developer could download and install an instance of application server on their own desktops/laptops
  - No interdependencies for starting & restarting application server (JBOSS) instance
- Work around for Licensing window of 60 to 90 days
  - At the time of our project inception, this approach provided our client with additional timeframe in making financial decisions & production infrastructure preparations
- Almost ***NO*** Learning Curve
  - Freely Downloadable - both offshore & US
  - No Cost to Client!

…but Proven Technology for Production

While Open Source based development has its advantages, proven Application Server technology such as BEA Weblogic Platform comes with added advantages:

- Application Scalability
  - Our experience suggests use of BEA in production for mid to high range applications
  - Clustering requirements
  - Client environment is already using BEA for other applications in their production environment

…but Proven Technology for Production Cont'd

- Application Stability
  - JBoss is relatively new and maturing, BEA stabilized over the years
- Proven Integration Ability
  - BEA Platform-based products allow seamless and efficient integration to additional BEA platform components that are already in use or planned as future purchases.

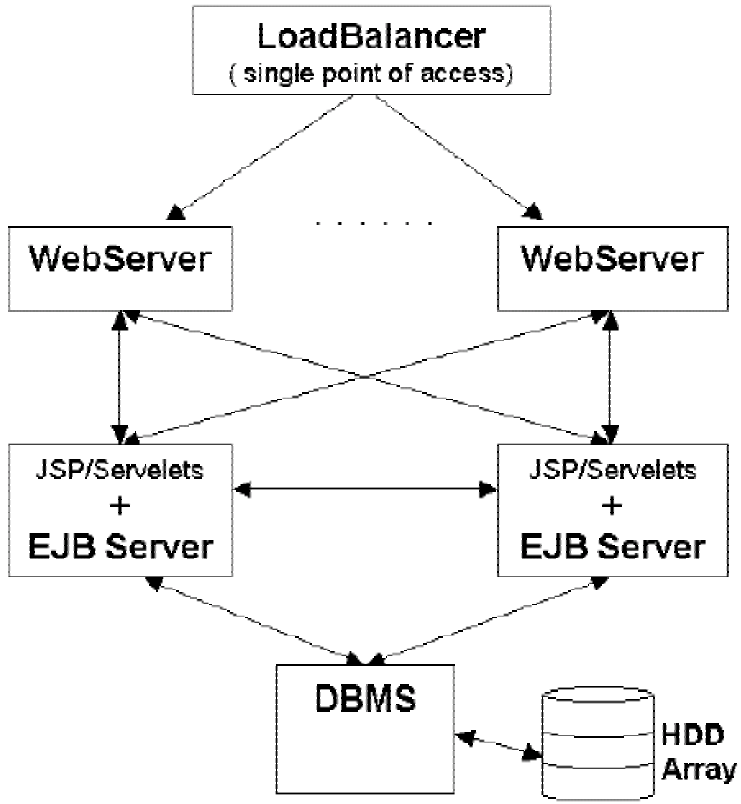
…but Proven Technology for Production Cont'd

- Admin Console is feature rich
  - In JBOSS, deployment is not as straight forward as BEA WLS. Tryst with descriptor files.
  - Developer friendly UI for deployment in BEA
- Clustering for mission critical applications
  - Higher availability via fail-over
  - Increased scalability through load balancing

JBOSS's clustering capabilities, though rumored to be sufficient, are relatively new and documentation is sparse whereas BEA platform has proven track record for critical applications.

…but Proven Technology for Production Cont'd

- WebLogic Clustering Functionality
  - Fail Over for stateless as well as stateful session beans
  - Smart caching, locking mechanisms, and connection pooling lowers demand on DBMS lowering the requirements to have more than one DB Server in a high performance cluster (BEA caching estimated to have 60 fold performance increase over non-cached servers)



Migration from Open Source to BEA Weblogic Server

Although changes are minimal, some deployment descriptors need to be verified & changed during migration.

	From JBoss	To BEA WLS
To start with, create environment variables	JBOSS_HOME	WL_HOME
Classpath should include:	All JBoss Specific jars, e.g., jboss.jar e.g., C:\jboss-3.0.3_tomcat-4.1.12\server\default\lib	weblogic.jar
Deployment descriptors may require formatting:	JBoss parser requires begin and end tags on the same line for ejb definitions in ejb-jar.xml	The ejb-jar.xml file is already well formatted, so no changes required for migration!
Some deployment descriptors are specific:	• Jboss.xml • Jboss-odbc-cmp.xml • Oracle-service.xml • Mail-service.xml • Jboss-mq-destinations.xml • Log4j.xml	• weblogic.xml • config.xml • application.properties • weblogic-cmp-rdbms-jar.xml

Migration from Open Source to BEA Weblogic Server Cont'd

[JBOSS] JNDI Descriptors: **jboss.xml** in <JBOSS HOME>\server\default\deploy folder

```
<jboss>
 <enterprise-beans>
  <entity>
   <ejb-name>UserProfile</ejb-name>
   <local-jndi-name>com/hospital/atlanta/infrastructure/dataobject/UserProfileLocal</local-jndi-name>
  </entity>
  <session>
   <ejb-name>PatientAgent</ejb-name>
   <jndi-name>com/hospital/atlanta/common/service/PatientAgent</jndi-name>
   <local-jndi-name>com/hospital/atlanta/common/service/PatientAgentLocal</local-jndi-name>
  </session>
  <message-driven>
   <ejb-name>HL7ProviderMessageBean</ejb-name>
   <destination-jndi-name>HL7ProviderMessageBean</destination-jndi-name>
  </message-driven>
  ...
 </enterprise-beans>
</jboss>
```

[BEA] JNDI Descriptors: *weblogic-ejb-jar.xml*

```
<weblogic-enterprise-bean>
  <ejb-name>HR</ejb-name>
  <stateless-session-descriptor>
   <pool>
    <max-beans-in-free-pool>10</max-beans-in-free-pool>
    <initial-beans-in-free-pool>0</initial-beans-in-free-pool>
   </pool>
   <stateless-clustering>
    <stateless-bean-is-clusterable>true</stateless-bean-is-clusterable>
    <stateless-bean-load-algorithm>RoundRobin</stateless-bean-load-algorithm>
   </stateless-clustering>
  </stateless-session-descriptor>
  <jndi-name>com.bea.wlxt.sample.HR</jndi-name>
 </weblogic-enterprise-bean>
```

Migration from Open Source to BEA Weblogic Server Cont'd

Deploying entity beans:

- For deploying entity beans in BEA WLS, ensure that entity field mapping Indicator is turned on. Or else, we need to run scripts explicitly for entity beans field mapping.
- JBoss does this (field mapping entities in db) automatically.
- If the mapping indicator is turned off, for example, a script similar to the following needs to be run explicitly for all entity beans (CMP) in BEA WLS.

```
Create table PATIENT_PROFILE (
    ID NUMBER(10) NOT NULL,
    PROFILEID VARCHAR2(255) NOT NULL,
    FNAME VARCHAR2(255),
    LNAME VARCHAR2(255)
);
```

Migration from Open Source to BEA Weblogic Server Cont'd

We have discussed examples of some descriptors and their differences here.

Defining Database Connections:

[JBOSS] Oracle-Service.xml <JBOSS HOME>\server\default\deploy

```
<attribute name="JndiName">OracleDS</attribute>

<attribute name="ManagedConnectionFactoryProperties">
   <properties>
     <config-property name="ConnectionURL"
           type="java.lang.String">jdbc:oracle:thin:@MyServer:1521:MyDatabase</config-property>
     <config-property name="DriverClass" type="java.lang.String">oracle.jdbc.driver.OracleDriver</config-
property>
     <!--set these only if you want only default logins, not through JAAS -->
     <config-property name="UserName" type="java.lang.String">appuser</config-property>
     <config-property name="Password" type="java.lang.String">remember</config-property>
    </properties>

</attribute>
```

[BEA] Edit config.xml in myapplication domain

All in one place: **Most of the descriptors are in config.xml.**

- **JDBC Connection Pool**
- **JDBC DataSource**
- **JMS Connection Factory**
- **JMS JDBCStore**
- **JMS Queues**
- **Mail Service Definitions**
- **Application EJB Definitions**
- **Server Settings**
- **Define Oralce DBSERVER, DBPORT, SID, USER, PASSWORD, etc. in myapplication.properties file**

Migration from Open Source to BEA Weblogic Server Cont'd

[JBOSS] JNDI Descriptors: **ejb-jar.xml** in myapplication\META_INF

```
<ejb-jar>
 <enterprise-beans>
  <session>
   <description>Session Bean ( Stateless )</description>
   <display-name>PaymentAgentBean</display-name>
   <ejb-name>PaymentAgent</ejb-name>
   <home>com.hospital.atlanta.preregistration.service.PaymentAgentHome</home>
   <remote>com.hospital.atlanta.preregistration.service.PaymentAgent</remote>
   <local-home>com.hospital.atlanta.preregistration.service.PaymentAgentLocalHome</local-home>
   <local>com.hospital.atlanta.preregistration.service.PaymentAgentLocal</local>
   <ejb-class>com.hospital.atlanta.preregistration.service.PaymentAgentBean</ejb-class>
   <session-type>Stateless</session-type>
   <transaction-type>Container</transaction-type>
  </session>
  ...
 </enterprise-beans>
</jboss>
```

[BEA] One difference of note

While enterprise beans in JBOSS are defined in one place, for BEA WLS, EJB descriptors are for each component deployed as a part of the application.

No change in the way beans are defined but during migration, need to divide up descriptors and ensure there is a EJB descriptor for each component deployed as a part of the application.

Summary

We gained valuable experience with this approach.

- JBoss provided a rapid development platform with free downloads and environment flexibility
- Open Source based development suits well for accelerated development needs
- Migrating applications to BEA is fast and simple
- In our example, while licensing and infrastructure needs were being worked out, starting development on open source application server proved meaningful
- Approach should be guided by project/client needs

Acknowledgements

Our special thanks to:

- Christine White, BEA Systems Inc.
- Kishore Sannidhanam, BEA Systems, Inc.
- Kiran Emani, Cap Gemini Ernst & Young US

Questions?