

JavaOne 2000

Topics

Keynotes

Talks overview

Interoperability, jini

Real-time and Embedded

XML and Java

Cool

Recommendations

First Keynote (McNeally, Jobs)

- 9 Gbits/s cross the Atlantic (m:m)
- Sun Startup Jumpstart programme
- J2EE server-side concentration, app built on server
- J2SE now free. 40% faster, 20% smaller than v1.2
McNeally says Sun working with Apple to make Mac platform of choice.
- J2ME, 60 companies. 1000:1 embedded to classical

Talks attended: see java.sun.com/javaone

- Keynotes
- XML Java Technology and Design Patterns (TS-555)
- The Future of Java Technology in Embedded & Real-time Systems Design (BUS-1132)
- Integrating Legacy Systems with Persistent EJBs.(TS-895)
- Java Data Objects (TS-700)
- Applied Ubiquitous Computing Using Jini Connection Technology (BUS 1228)
- XML Messaging and J2EE Technology (TS-1630)
- Jini Technology as a Foundation for Distributed Military command and control (TS-662)
- Developing OO, Distributed Web Applications with Java technology (TS-676)
- Redesigning a client-server Application to a 3-tier Component Based Application for Java (BUS-1579)
- Building Large Scale Server Applications with Java Beans Component Architecture (TS-1089)
- Interoperability of the J2EE components with CORBA and JMS (TS-1179)

Interoperability

Keystone of java: “Write-once, run anywhere”. Need to integrate enterprise data and Legacy systems.

Outline of technologies:

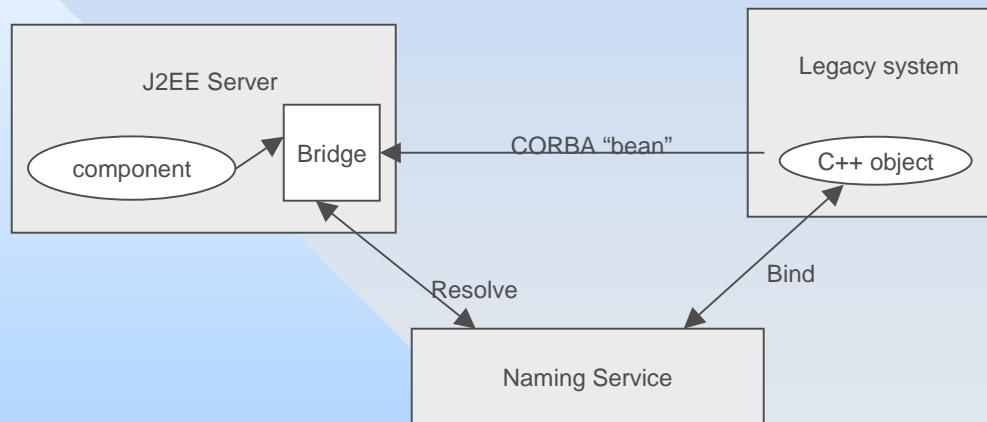
- J2 Enterprise Edition (J2EE). A number of technologies – in 1 architecture. An Application Programming Model and test suite for server side applications.
- JavaBeans. Model for middleware
- Enterprise JavaBeans (EJB). Model and a server for server-side component programming. “Transactional” objects and methods. Scalability & Portability.
- Application Server. Combines all of above + more. Like a web server but for applications. Includes development env. Debugging, source control.
- Jini. Agreed architecture for connecting devices “so they just work”. No drivers, no OS issues. Device self-discovers its name, connectivity. Manages transparency for any connectivity (ie CORBA, JMS, RMI etc).

Openwings: Military Command Control System

- Open framework for unifying (battlefield) control systems based on Jini.
- Open consortium led by Motorola. Whitepaper at www.openwings.org. Not secret.
- Connects different data acqu. and control sys (Mapping, Strategic Ops, Weather, Targeting, "Intel" etc) into a single sys with single configurable UI.
- Component based. Components live in "containers"
- Containers connected by "connector services" (CORBA. Also RMI, JMS etc) Components don't care which connector connects them.
- EJB compliant
- All "availability" modes supported: cold start, warm start, hot swap, failover
- Services:
 - Processor Service: manages availability.
 - Data Service: 3-tier model, using JDBC
 - Space Service: JavaSpaces.
 - Security Services: JAAS, JCE
- Demo: Streaming video to desktop and palm-pilot, sharing data and control.
- Reference implementation publicly available in October.

eMobile: interoperability with CORBA

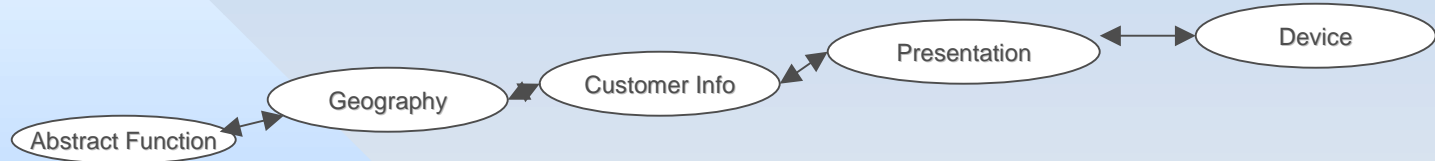
- Managed by J2EE server



Bridge object options: Client side stubs or Dynamic Invocation (DII)
DII is better: Stubless run-time binding, Client stubs not required by EJB

“Universal Interaction” Project

- Development Environment for “highly interactive” applications
- Ongoing IBM research project. Now in use (Openwings)
- Assumes data from multiple “channel” types (db, device, streaming etc) in pier-to-pier architecture.
- Unifies these for application development through chain of XSL stylesheets.



- Dialog Management markup Language (DMML) – Legal dialogs
- Interaction Markup Language (IML) – Presentation
- Issues:
 - Performance help up by XML and XSL
 - Stability of standards

Real-time and embedded

- Hard RT spec. by JCP (IBM, aGile, James Gosling, Doug Jensen). Canvassing for volunteers.
- Close to reference implementation (needed for JSR)
- Commands at 20ms, 2ms distributed proc synch. Max RT thread cycle every 5us, 500ns latency.
- Objectives: “temporal predictability”, incl “trans-node”. Garbage Collection now truly transparent. Built in concurrency and networking. Write-once run anywhere.
- Timeliness constraints travel with the data!
- Map memory by 1) instantiate at given address. 2) Container class.
- aGile RT Java Chip: context switching in <1us. Power <1mW/MHz. All but 2 JVM byte codes supported in HW. www.agile.com
- 5 Myths: Sun dominated, Too Big, Too Slow, Hard to start, Not mature.

Misc Recommendations for Large, Component based system development

Design so components are reusable and customizable. Encapsulate functionality

Database interactions are transparent to developer – Oracle BC4J

Infobus & JFC/Swing based, gives rich UI controls, slice and dice, drill-down etc.

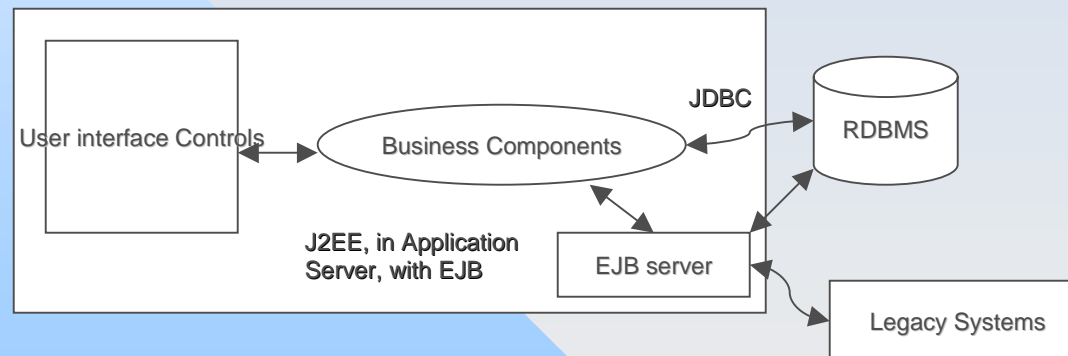
Use IDE capable of debugging on both tiers

Interactions with external systems are the bottlenecks

DB Schema is critical

Choose an App server that handles as many connection protocols (CORBA, RMI, JNI, HTTP, XML, sockets etc) as possible

Avoid table scans. Use good indexes and optimized, cached queries. Application tools which separate SQL from “business logic”.



Cool

- **Java Data Objects**

Do transaction management for you (2-phase commit, roll-back)

- **Zaplets**

Common applet frameworks similar to JavaSpaces. Eg, voting through email

- **Sega hockey game in Java (you had to be there).**

Recommendations

- Careful: there may be dragons
 - Cache, reserve enough memory, avoid serialization, avoid db
- Server side is where its at. Careful of client side.
- Cost out usual tools
 - J2EE, EJB
 - Application Server (BEA WebLogic, Silverstream, Apache?)
- Research our suppliers: Windriver etc.
- Breadth first, not depth first: Look at interoperability architectures, frameworks and tools, eg JMS, RMI, XML, JDO, JNI as well as CORBA. Look at abstract connectors. Familiarize with different JVMs.
- Look at and hook up with suitable JCPs (openwings, RT)
- Get and look at reference architectures and whitepapers (eMobile, openwings).
- Cost/Benefit Analysis.