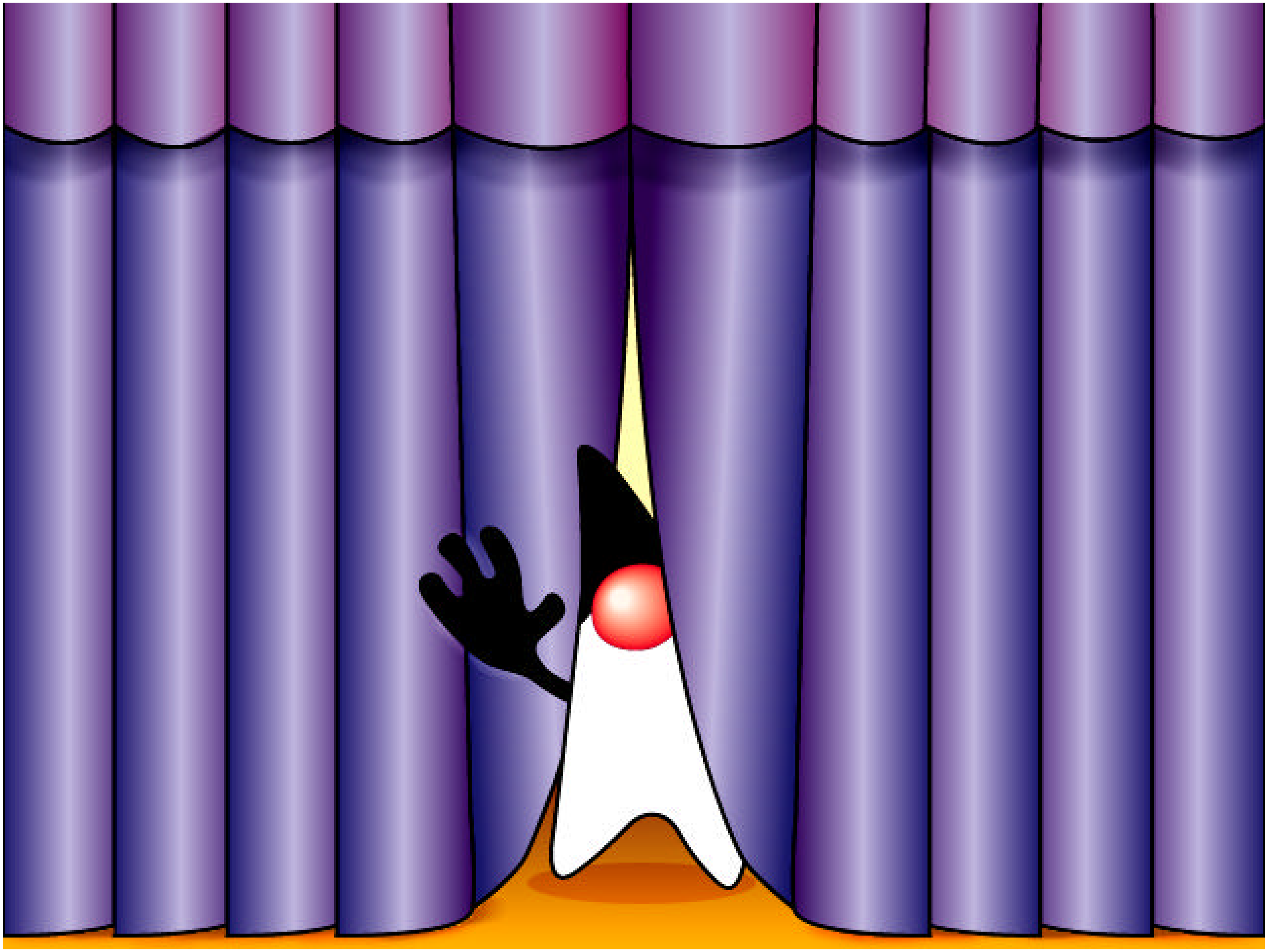


What's New In Java

Simon Ritter, Java Software

Sun Microsystems, Inc

Agenda

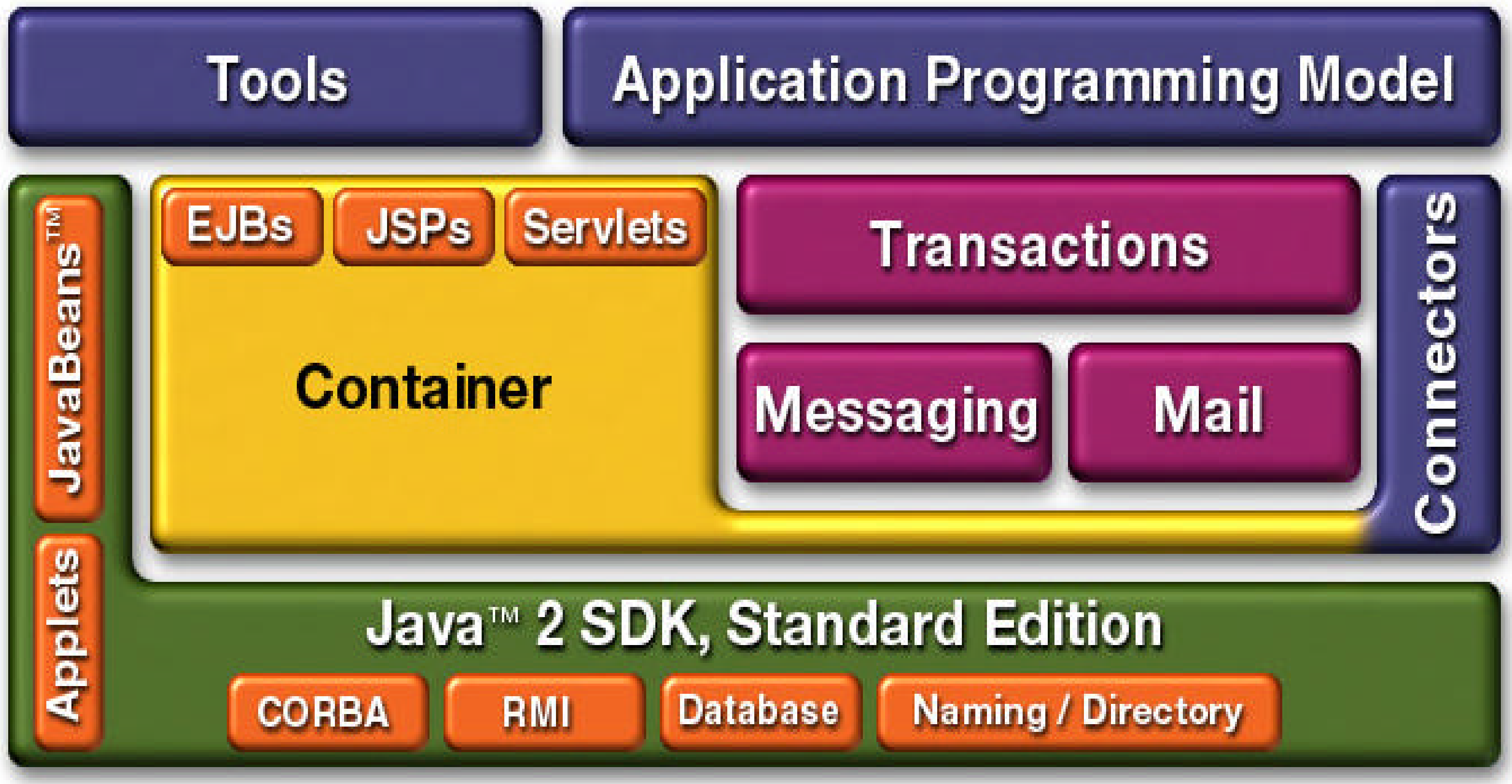
- New Java Model
  - Java 2 Platform, Standard Edition
  - Java 2 Platform, Enterprise Edition
  - Java 2 Platform, Micro Edition
- Hotspot VM
- Jini
- Summary

Java 2 Platform, Standard Edition

- New Security Model
- JFC
  - Swing
  - 2D API
  - Drag-and-drop
- Java IDL
  - JDK includes IDL Name Server
- Collections

More Information: http://java.sun.com/j2se

Java 2 Platform, Enterprise Edition



More Information: http://java.sun.com/j2ee

J2EE Features

- Enterprise Java Beans (EJB)
- Java Naming & Directory Services (JNDI)
- Java Messaging Services (JMS)
- JavaMail
- Java Database Connectivity (JDBC)
- Java Transaction Services (JTS)
- Servlets
- Java Server Pages (JSP)

What is EJB Technology?

- A server-side component architecture
- A specification from Sun
- Enables easy and efficient development and deployment of Java applications that are:
  - Transactional
  - Portable
  - Distributed
  - Multi-tier
  - Scalable
  - Secure

EJB Technology Design Goals

- Easy development & deployment of distributed applications
- The right expert for the right job
- Platform independent
- Middleware independent
- Protocol neutral
- Preserve IT investment
- Truly enable reuse

What EJB Means to Developers

- Faster, more productive development
  - Business logic, not low-level infrastructure
  - Reusable components in Java language
  - Declarative customization
- Leverage efforts & expertise across middleware
  - No proprietary API calls in code
  - Easily deployed into different servers
- Maximum component reuse: 3rd party & internal
- Increases quality and reliability

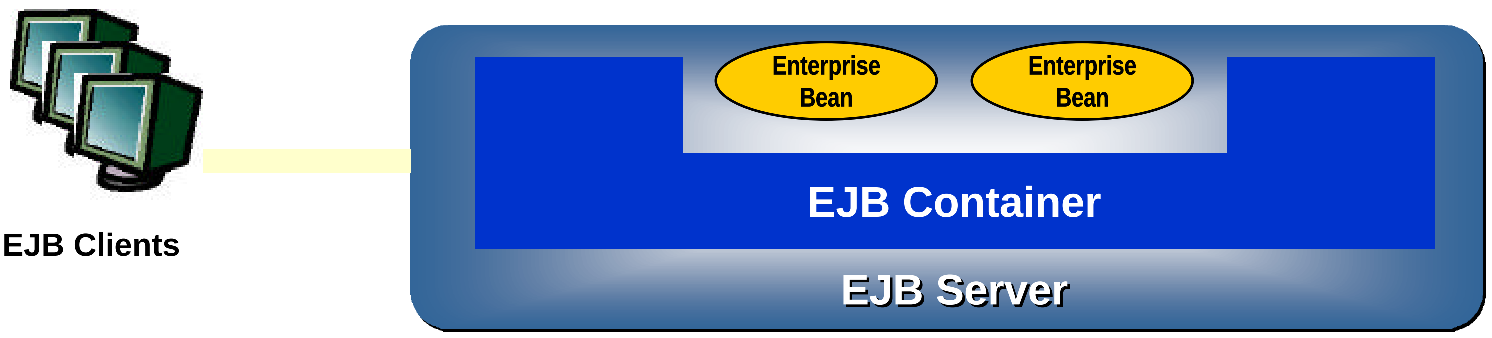
What EJB Means to IT Groups

- Cost reduction & Faster time-to-market
- Investment protection
  - Leverage existing middleware, database & back-end resources
  - Support multiple clients, Server OS and wire protocols
  - Integration with CORBA
- Safe choice
  - Broad industry support
  - Choice, not vendor lock-in
  - Choose vendor today, change freely tomorrow

Enterprise JavaBeans Architecture

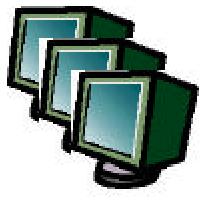
The EJB architecture specifies the responsibilities and interactions among EJB entities

- EJB Clients
- EJB Containers
- EJB Server
- Enterprise Java Beans



EJB Client

- Any application or program requesting a service
- Can be written in any language
- Access is controlled by the Container
- Use JNDI to instantiate or find existing instances of EJB components
- Protocol neutral (IIOP, JRMP,DCOM, etc.)
  - RMI is the standard method for invoking methods in EJB components

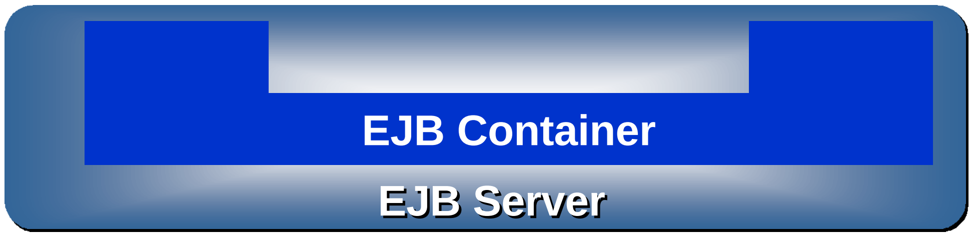

EJB Clients

EJB Server

- Can be designed for EJB from the ground-up
- Well-established servers easily adapted to support EJB (TP Monitors, ORBs, Database servers, etc.)
- Automatically manages the underlying “plumbing”
  - Transactions
  - Security
  - Naming
  - Threading
  - Resource pooling
  - Remote access
  - Persistence, etc.

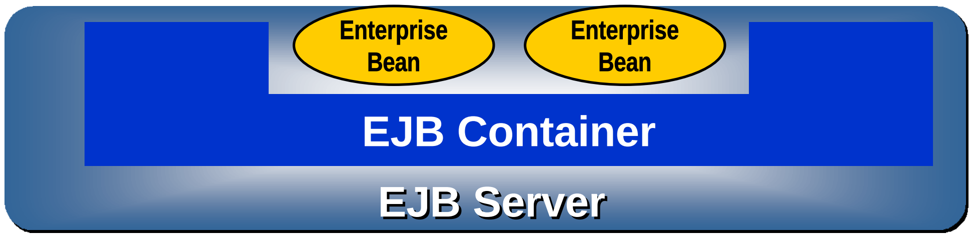
EJB Container

- Provides a Run-time Environment for EJB components
- Hosts EJB components
- Manages life cycle of EJB components
- Transparently delivers system-level services
  - Naming & Life cycle management
  - Persistence (state management)
  - Transaction Management
  - Security
  - Etc.



EJB Components

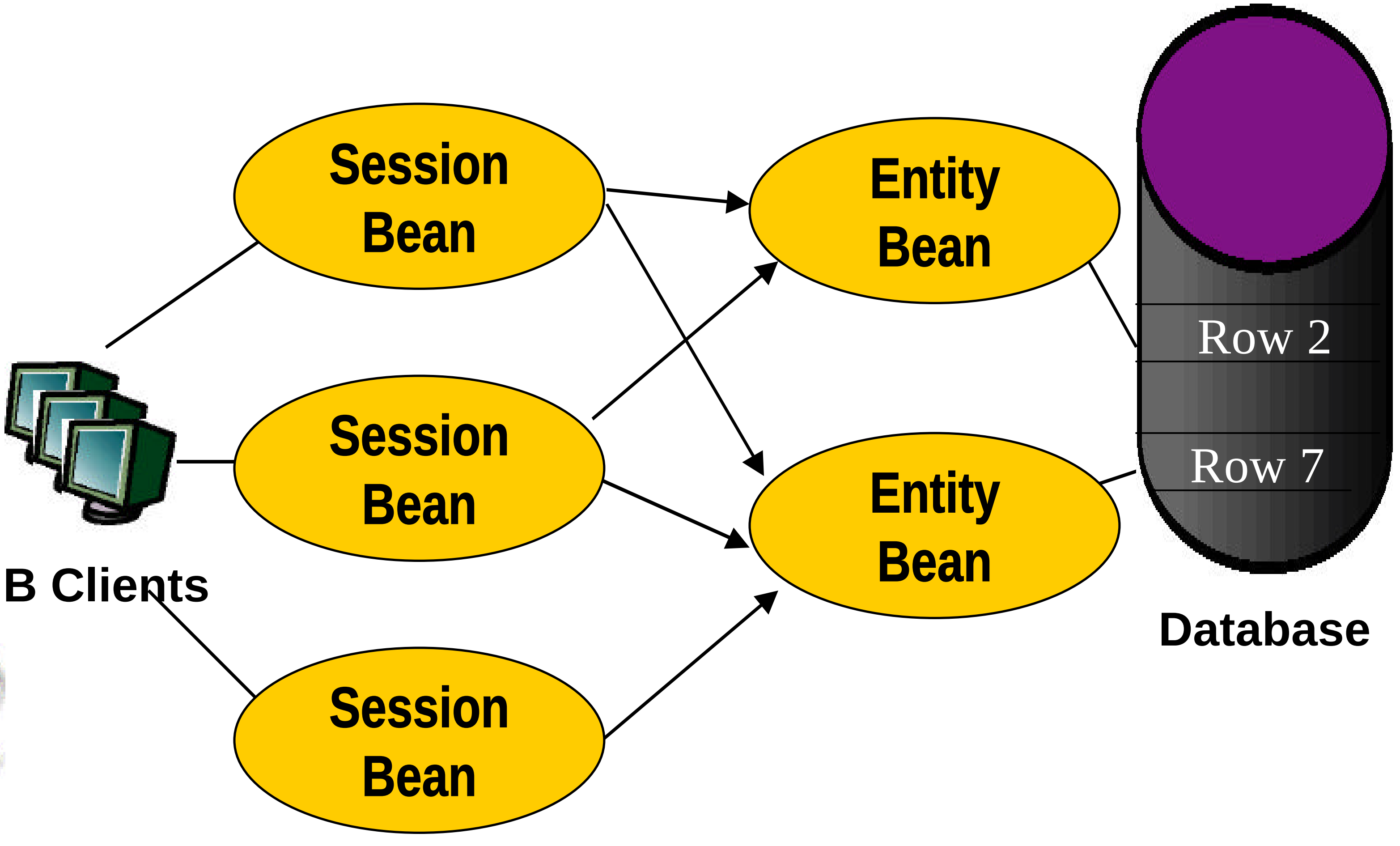
- A specialized Java class
- Contain business logic ONLY
- Distributed over a network
- Reusable across middleware servers
- Standard interfaces enable management and service delivery by EJB Server
- Two types:
  - Session Beans
  - Entity Beans



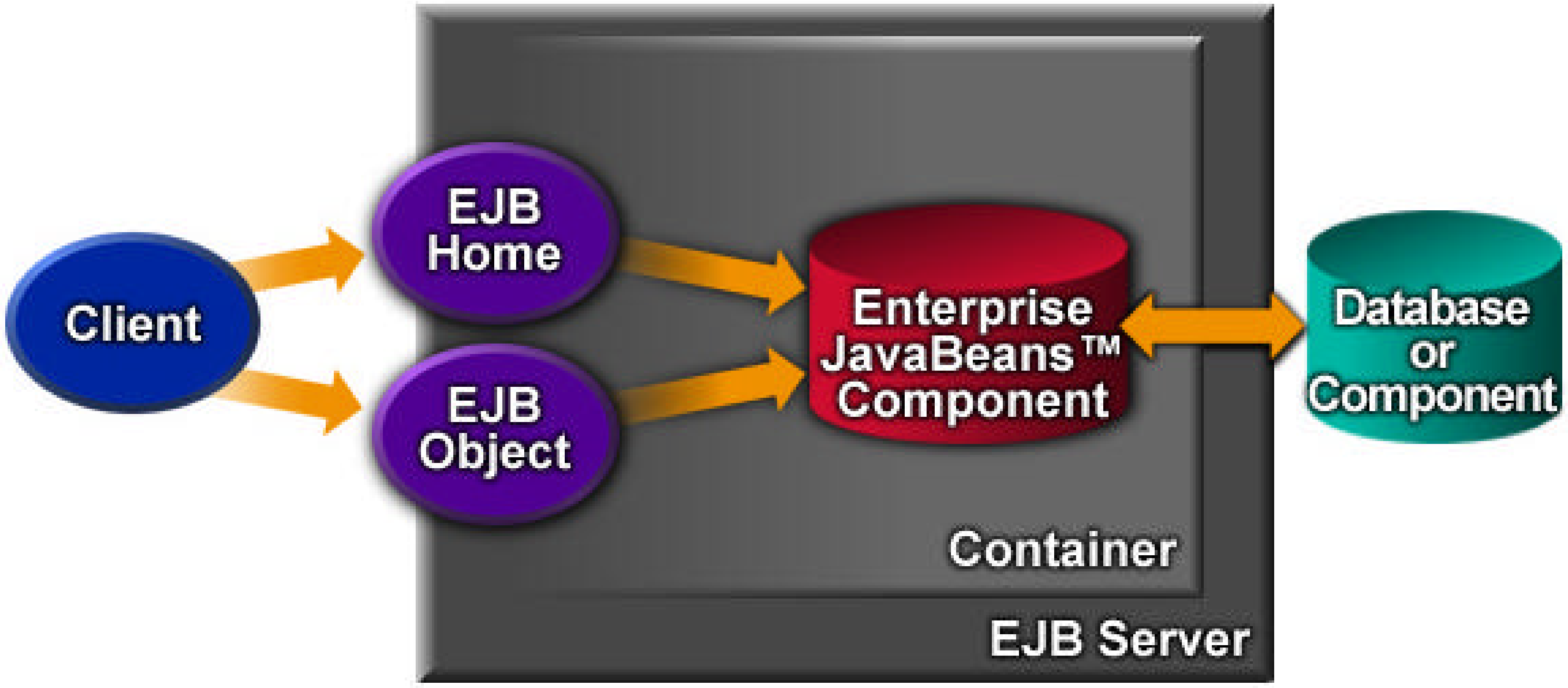
Session and Entity Beans

Session Beans	Entity Beans
Represents a task	Represents underlying data
One instance per client	Instance shared by clients
Short-lived	Long-lived
Transient	Persistent
Can be any Java class	A Java class that maps to persistent data
May be transactional	Always transactional
Mandatory for EJB 1.0	Optional for EJB 1.0

Enterprise Java Beans - Typical Scenario



EJB Architecture



The Deployment Descriptor

- Gives the container instructions on how to manage the EJB component
- Allows declarative customization
- Controls behaviors for:
  - Transaction
  - Security
  - Life cycle
  - State management
  - Persistence
  - Other...

Responsibilities: Developer & Server

Developer Provides

- EJB component --
  - Business logic (EJB class)
- Home Interface
  - Create/Find instances
- Remote Interface
  - Call business methods
- Deployment Descriptor
  - Customization and control

EJB Server Provides

- EJB Container
  - Runtime and management
- EJBHome
  - Implements Home Interface
- EJBObject
  - Implements Remote Interface
- Tools for creation of DD
  - Optional
- May provide tools for generating interfaces (Home & Remote)

How it all works together

- Server instantiates EJBHome & places in JNDI name space
- Client finds EJBHome using JNDI
- Client creates a new EJB component instance (Session) using Create method

OR

- Client finds an existing EJB component instance (Entity) using Finder methods
- Client gets reference to EJBObject instance
- Client calls methods via EJBObject

Java Naming and Directory Services (JNDI)

- Provides naming/directory services
- Can be used to find files, printers, objects, etc. on a network
- Provides a common API on top of any directory service product
  - JNDI is partially implemented by Java Software
  - Directory service products (LDAP, DNS, NDS, etc) implement most of the specification
- Used in conjunction with RMI to locate EJBs on a server

Java Messaging Services

- Asynchronous Communications
- Publish & Subscribe
- Reliable Queues
- Guaranteed Delivery
- Open And Cross Platform
- Support From:
    - Tibco, IBM, Modulus, Active

Java Mail

- Abstract APIs that model a mail system
- Current protocol support
  - IMAP
  - SMTP
  - POP3
- Flexible interface allows 3rd party “plugin”
  - NNTP
  - Lotus

JDBC - Database Access

- Java interface to relational databases
- May be incorporated into EJB components for Database calls
  - EJBload and EJBstore methods
- JDBC 2.0 Extensions
  - Scrollable Cursors
  - Support for SQL3 types
  - Full support for storing & retrieving Java object type
  - Character streams (Unicode, etc)
  - Fully backward compatible with JDBC 1.0

Java Transaction Services

- Based on CORBA Object Transaction Servcices
- Distributed Transaction Processing
- Access to Transaction Monitors
  - JTS is not a TP monitor
- Industry Support
  - IBM, Inprise, Bull, WebLogic

Servlets/Java Server Pages

- Dynamic Creation Of HTML
- Servlets Replace CGI Scripts
  - Small fixed HTML, large dynamic content
- JSPs Embed Java in HTML Page
  - Small dynamic content, large fixed HTML
  - XML Integration

Java Platform, Micro Edition

- Targeted at Consumer & Embedded
- Different Size Virtual Machines
  - Standard Virtual Machine
  - K Virtual Machine
  - Java Card Virtual Machine
- Different Profiles
  - Personal Java
  - Embedded Java
  - Java Card

More Information: http://java.sun.com/products/j2me

The HotSpot Virtual Machine

- Significantly Improved Performance
- Faster thread synchronisation
- Adaptive compiler technology (“Hot Spot”)
- Method inlining
- Garbage Collector
  - Incremental (“pauseless”)
  - Generational
  - Mark-compact eliminates memory fragmentation

More Information: http://java.sun.com/products/hotspot

Jini™ Technology

Simply connect.

More Information: http://www.sun.com/jini

Sun's Java™ Technology Strategy for the Networked Age

- Distributed object-oriented systems
  - Each device/service is just an object
- Enabling technologies

Jini Technology	Distributed Computing
Java Virtual Machine	Hardware/Platform Independent
Java Platforms	Best Object Language/Platforms
Java Chips	For Great Systems on a chip

Jini Technology Enables You to Simply Connect

Introduces Simple, Powerful New Concepts

- **Instant On**
  - Plug it in and it just works, no fuss, hassle free
- **Impromptu community**
  - Create your personal community of devices and services — at home, in the office, or on the road — and interact with other communities quickly and easily
- **Resilient**
  - Your Jini community maintains itself & adapts to change
  - Your Jini community is always available
  - The Service Age allows the system to be more tolerant and redundant
- **Special delivery**
  - Services are available on demand

The Philosophy Behind Jini Technology

- Simplicity: *Less is more*
  - Small code base
  - No complicated OS
  - Everything is an object
  - Use RMI to extend objects to remote resources
- Self-healing networks
  - System restores state after failures
  - Resilience
- Community
  - Easy access to Jini technology
  - Anyone can join the Jini community

"We've taken the time to make it simple"

Bill Joy

Jini Technology

	Infrastructure	Programming model	Services
Base Java Technology	•JVM •RMI •Java Security	•Java APIs •Java Beans •etc...	•JNDI •EJB Components •JTS •etc...
Java + Jini Technology	•Discovery & Join •Distributed Security •Lookup Services	•Leasing •Transactions •Distributed Events	•Printing •Transaction Manager •JavaSpaces Service

Jini Technology Infrastructure: *Discovery & Join*

- Discover (find) and join a community of Jini technology-enabled devices
- Advertise its capabilities
- Provide any required software and attributes – no drivers required
- Requires only one Java Virtual Machine on the network
  - Send out a multicast packet with reference to yourself
  - Receive a RMI reference to the Lookup Service

Jini Technology Infrastructure: *Lookup Service*

Binds the Jini Community Together

- Repository of available services
- Stores service as extensible set of Java application objects
  - ID, interface, GUI's, attributes, drivers...
- Service objects downloaded as required
- May be federated with other lookup services
- Lookup Service interface
  - Registration, Access, Search, Removal

Jini Technology Programming Model: *Leasing*

Provides Method of Managing Resources in a Networked Environment

- Protocol for managing resources using a renewable, duration based model
- Contract between objects
- Resources can be shared or non-shared

Jini Technology Programming Model: *Distributed Events*

Addresses Peculiarities of Messages in the Networked Environment

- Extends Java platform event model to allow it to work in a distributed network
- Register interest, receive notification
- Allows for easy use of event managers
- Can use numerous distributed delivery models
  - Push, pull, filter ...
- Uses leasing protocol

Jini Technology Infrastructure: *Distributed Security*

Builds on the Java Virtual Machine

- Jini distributed security adds notion of principal and access control lists
- Jini services are accessed on behalf of a principle which traces back to a particular user/device
- Access to a service depends on the access control list associated with that service

Jini Technology Services: *JavaSpaces™ Technology*

May be Used to Implement a Large Number of Distributed Computing Patterns

- Shared, “dynamic memory” for networked Java Virtual Machines
- Helps federate the network of Java Virtual Machines
- Provides simple, dynamic object persistence
- Facilitates alternative messaging patterns
  - async, store and forward, routed, filtered...
- Service interface of JavaSpaces technology
  - Writing, finding, reading, removing, event

Jini Technology Adoption is Accelerating

- Jini community is increasing daily
- Current development by Jini community members
  - Computer devices (Printers and storage)
  - Consumer devices (Camera, DVD, VCR, settop)
  - Mobile devices (Pager, cell phone, PDA)
  - Automotive devices (GPS, sound, embedded control)
  - Networked devices (Routers, switches)
- Expect commercial devices and services to ship by 1/1/00

Jini Technology: *Innovation for the Future*

- Powerful, yet simple technology & licensing
  - Enables mobile behavior and computing
- Drives emerging networks of devices/services
  - Catch system-on-a-chip wave
  - No bloated fragile OS with complex configuration
- Appropriate software for the networked age
  - Platform independent via Java Virtual Machine
  - Object-oriented via best language
  - Allows dynamic systems that can easily evolve

Simply connect.

Summary

- Java 2 Platform
  - Complete
  - Stable
  - Secure
  - Fast
- Jini
  - Instant On
  - Plug and Work

microsystems

THE NETWORK IS THE COMPUTER™