# IBM Inter-University Programming Contest 2012 Training

# Chapter 3: Application Server

# Agenda

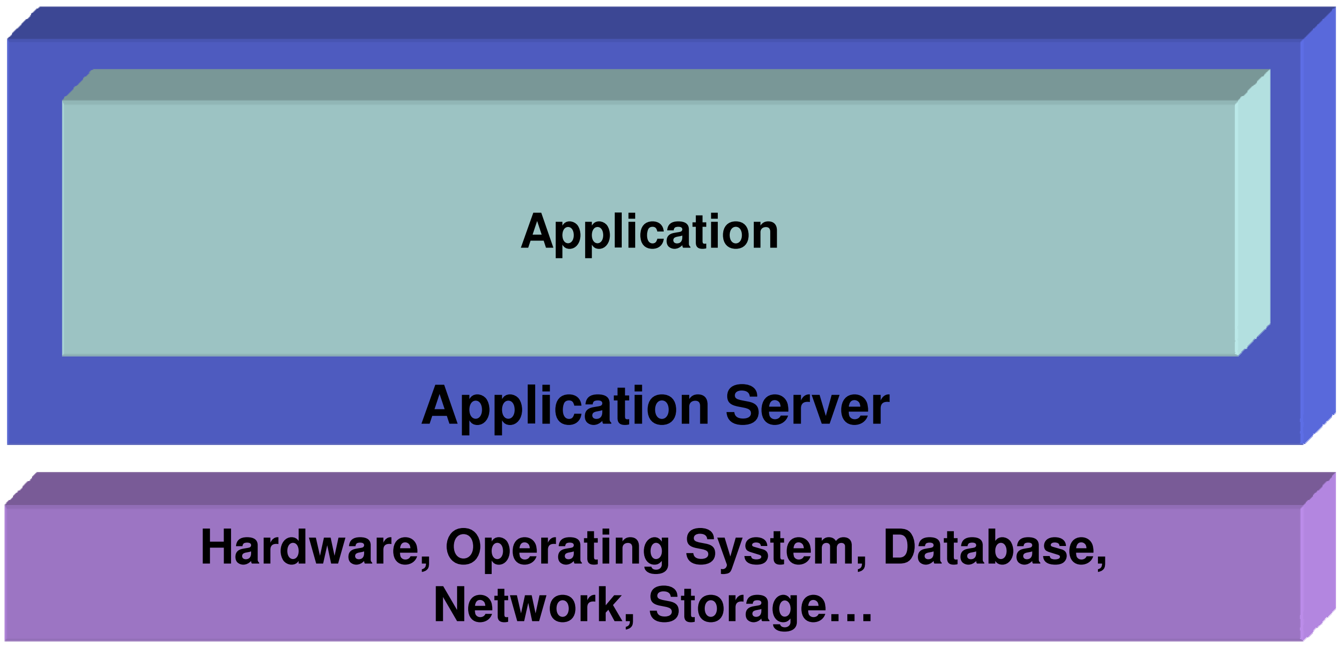
- What is an Application Server?
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- Application Server Architectural Overview
- Application Server in Cloud Computing
  - Topology Deployment
  - Infrastructure Virtualization
  - Health Management

# Agenda

- What is an Application Server?
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- Application Server Architectural Overview
- Application Server in Cloud Computing
  - Topology Deployment
  - Infrastructure Virtualization
  - Health Management

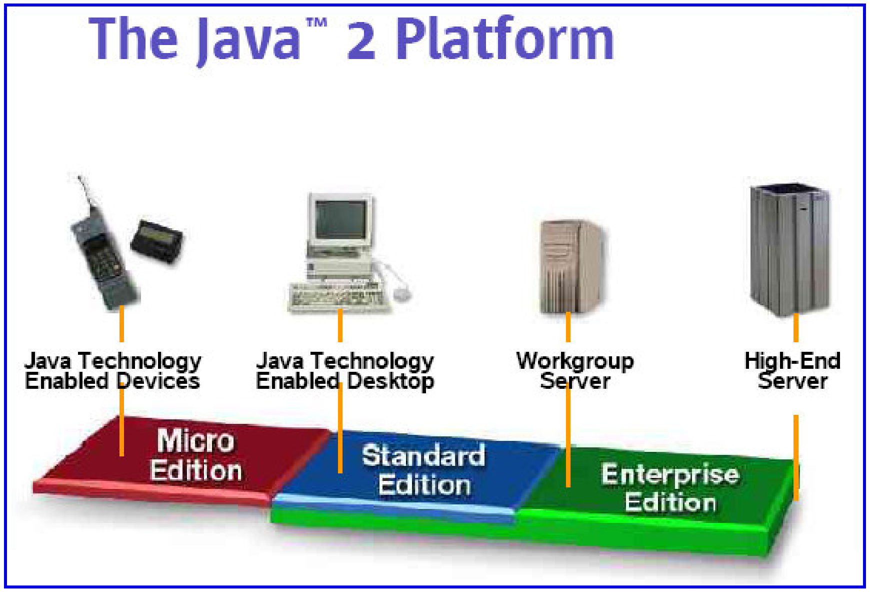
# What is an Application Server?

- Provides the infrastructure for running applications that run your business
  - **Insulates** applications from hardware, operating system, database, network…
  - Provides a common environment and programming model for applications
    - **Write once, run anywhere** (WORA)
    - Platform for developing and deploying **Web Services**
  - Provides a **scalable, reliable** transaction engine for your enterprise



# The Java Platforms

- **Java 2 Micro Edition (J2ME)**
  An optimized Java runtime environment for the consumer space. It covers the range of extremely tiny devices such as smart cards or a pager all the way up to the set-top box.




- **Java 2 Standard Edition (J2SE)**
  A complete environment for applications development on desktops & servers.
  - Java Runtime Environment, Standard Edition (JRE) - provides the Java APIs, Java virtual machine, and other components necessary to run applets and applications written in the Java programming language
  - Java Software Development Kit, Standard Edition (SDK) - A superset of the JRE that includes tools such as the compilers and debuggers necessary for developing applets and applications.

- **Java Enterprise Edition (JEE) – formerly J2EE**

# Java Enterprise Edition (JEE)

- Built upon J2SE, JEE provides the specifications for developing multi-tier enterprise applications with Java
- Accelerates and simplifies enterprise application development
  - With applications based on standardized, modular components
  - Providing a complete set of services to those components
    - Standard APIs for common services such as database access, transaction management, messaging, etc.
  - Handling many details of application behavior automatically, without complex programming (e.g. transaction management, security)
  - Providing a foundation for the third party component market (e.g. JCA, JMS, JDBC)
- Improves systems and operations management by providing:
  - Packaging, deployment, and management standards for enterprise applications
- Fulfills the promise of true portability
  - Hardware, operating system and vendor

# A Brief History of Enterprise Java

1998

EJB 1.0
Servlet 2.1

2000

**J2EE 1.2**
- EJB
- Servlet
- JSP
- JMS
- JavaMail

2001

**BPEL**

**Hibernate**

**J2EE 1.3**
- EJB
  - local EJBs
  - abs. CMP
  - MDB
- Servlet 2.3
  - Events
  - Filters
- JSP
  - XML
- JAXP
- Connectors
- JAAS

2003

**Portlets**

**SDO**

**Spring**

**J2EE 1.4**
- EJB 2.1
  - timers
  - pluggable JMS
- Web Services
  - Basic SOAP/HTTP
  - Registry
- JMX Mgmt
- J2EE Deployment
- JACC

**SCA**

2006

**Java EE 5**
- EJB 3
  - POJO components
  - POJO persistence
- Web Services
  - POJO components
  - protocol independence
  - JAXB
  - StAX
- JSF
- JSP
  - common EL
- Annotations
  - IoC

2009

**Java EE 6**
- EJB 3.1
  - No-interface view
  - Singletons
  - Async. session bean
  - Simplified packaging
  - EJB Lite
- Web Services
  - JAX-RS
- CDI
- Bean validation
- Servlet 3.0
- JSF 2.0
- Web profile

# JEE and WebSphere Application Server

- Application servers, based on standards, can be perceived as commodities

- WebSphere Application Server (WAS) delivers more value to customers than just standards compliance

  – Quality of service, such as
    - Workload management
    - High availability and failover
    - Performance

  – Tooling support
    - Runtime embedded in Application Developer products

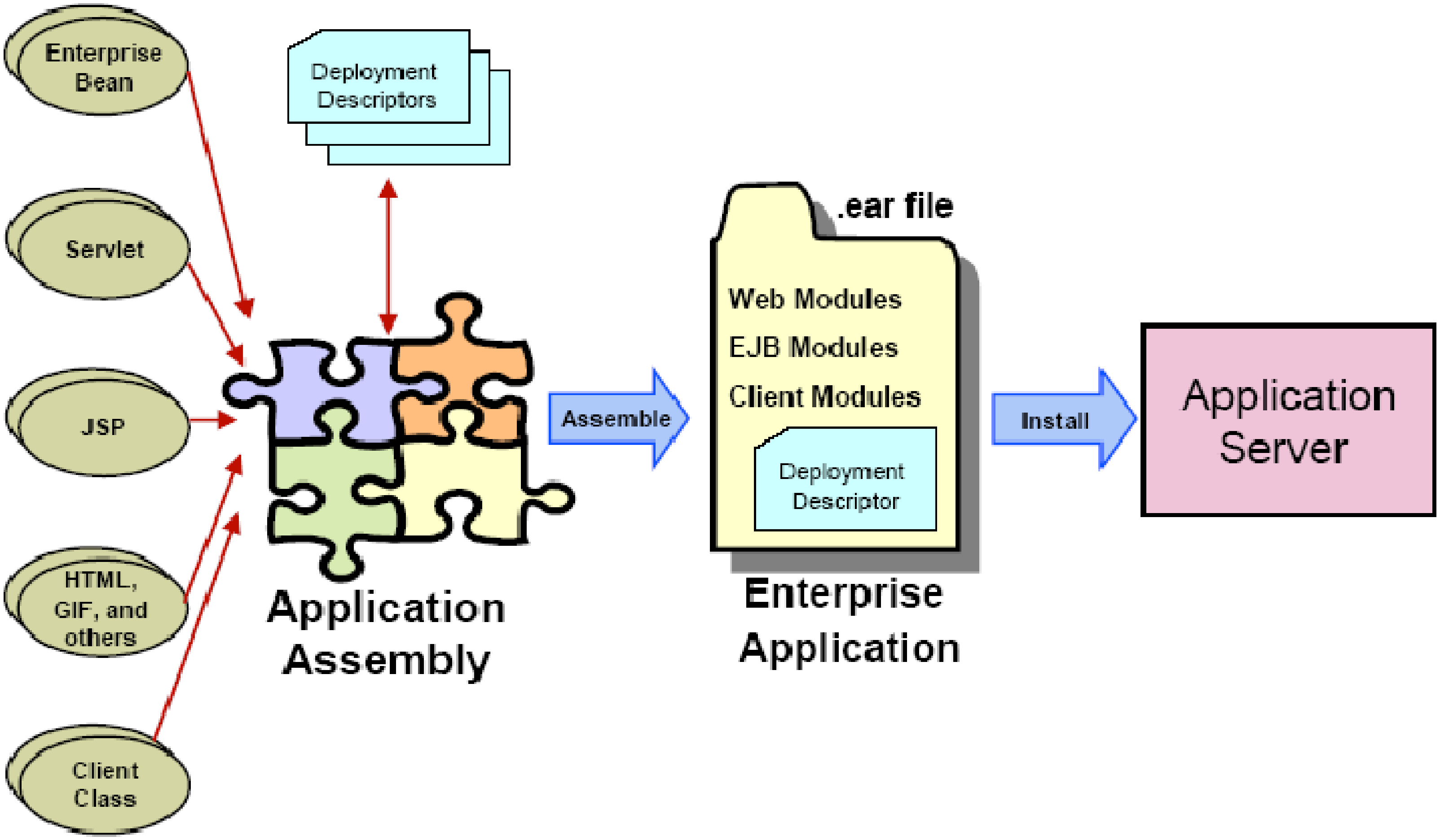
  – Programming model extensions
    - Future standards
    - Customer identified critical functions

  – Foundation for evolving WebSphere Software Platform

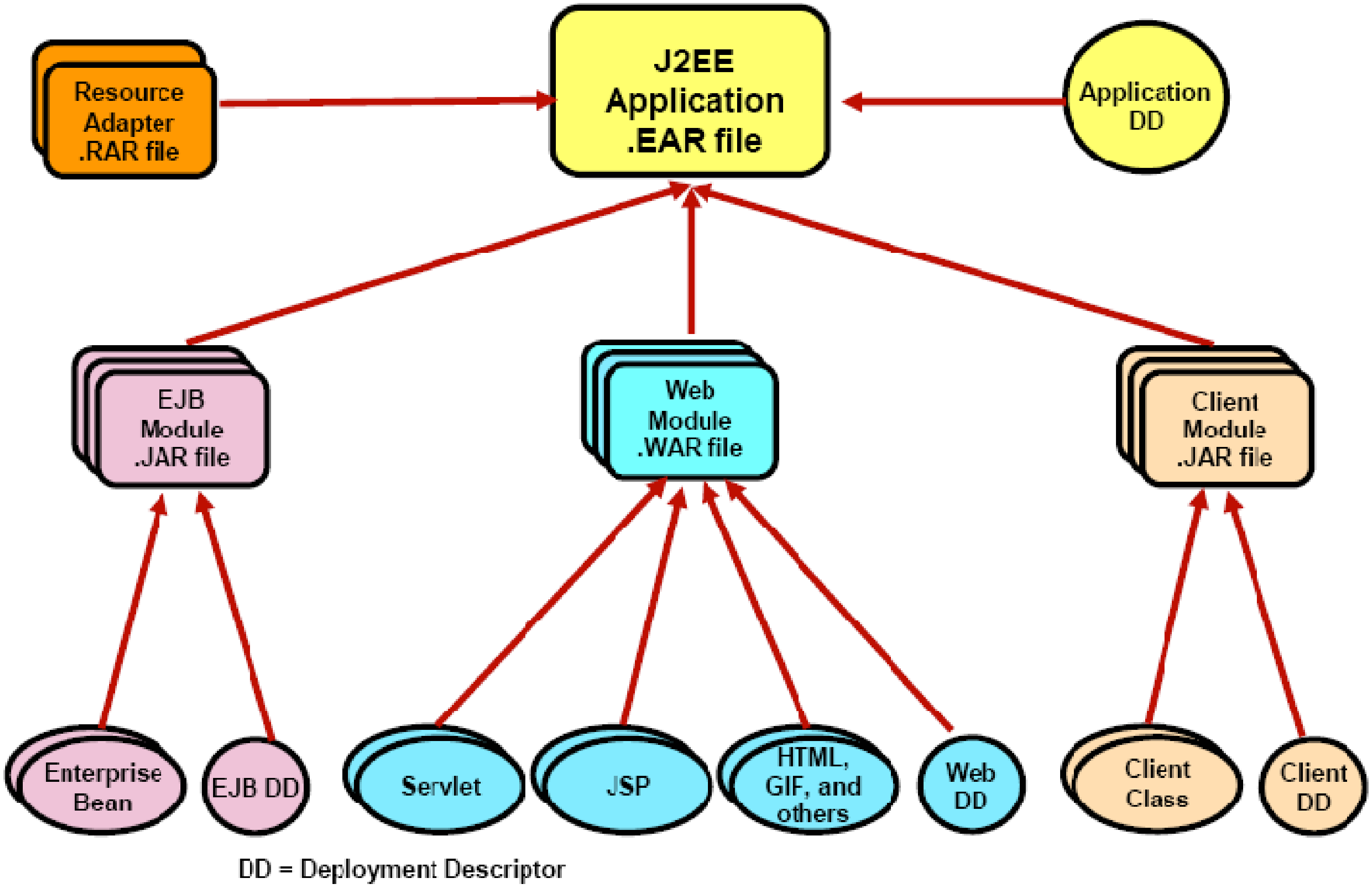
# Agenda

- What is an Application Server?
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- Application Server Architectural Overview
- Application Server in Cloud Computing
  - Topology Deployment
  - Infrastructure Virtualization
  - Health Management

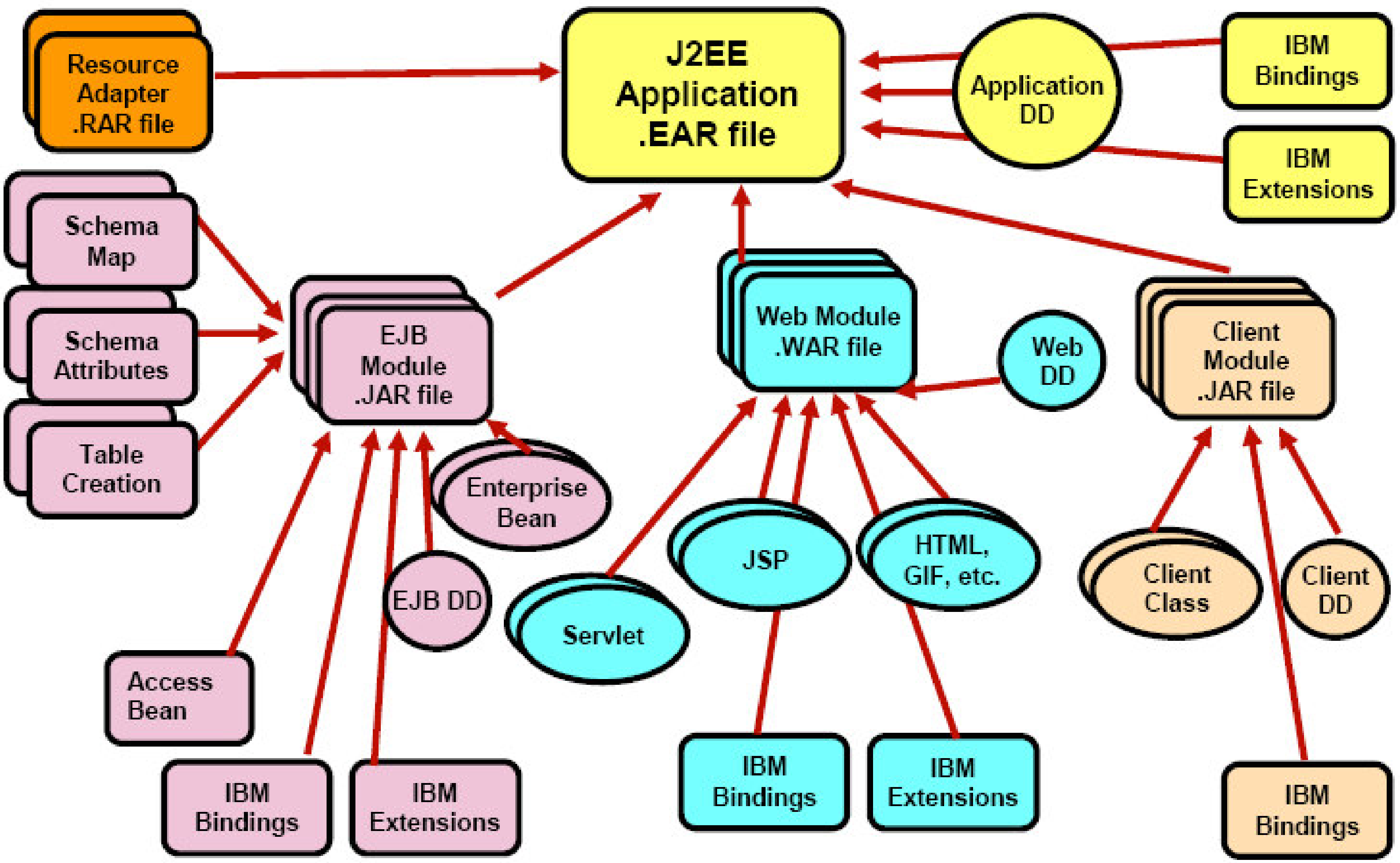
# Overview of Application Assembly and Installation



# JEE Packaging

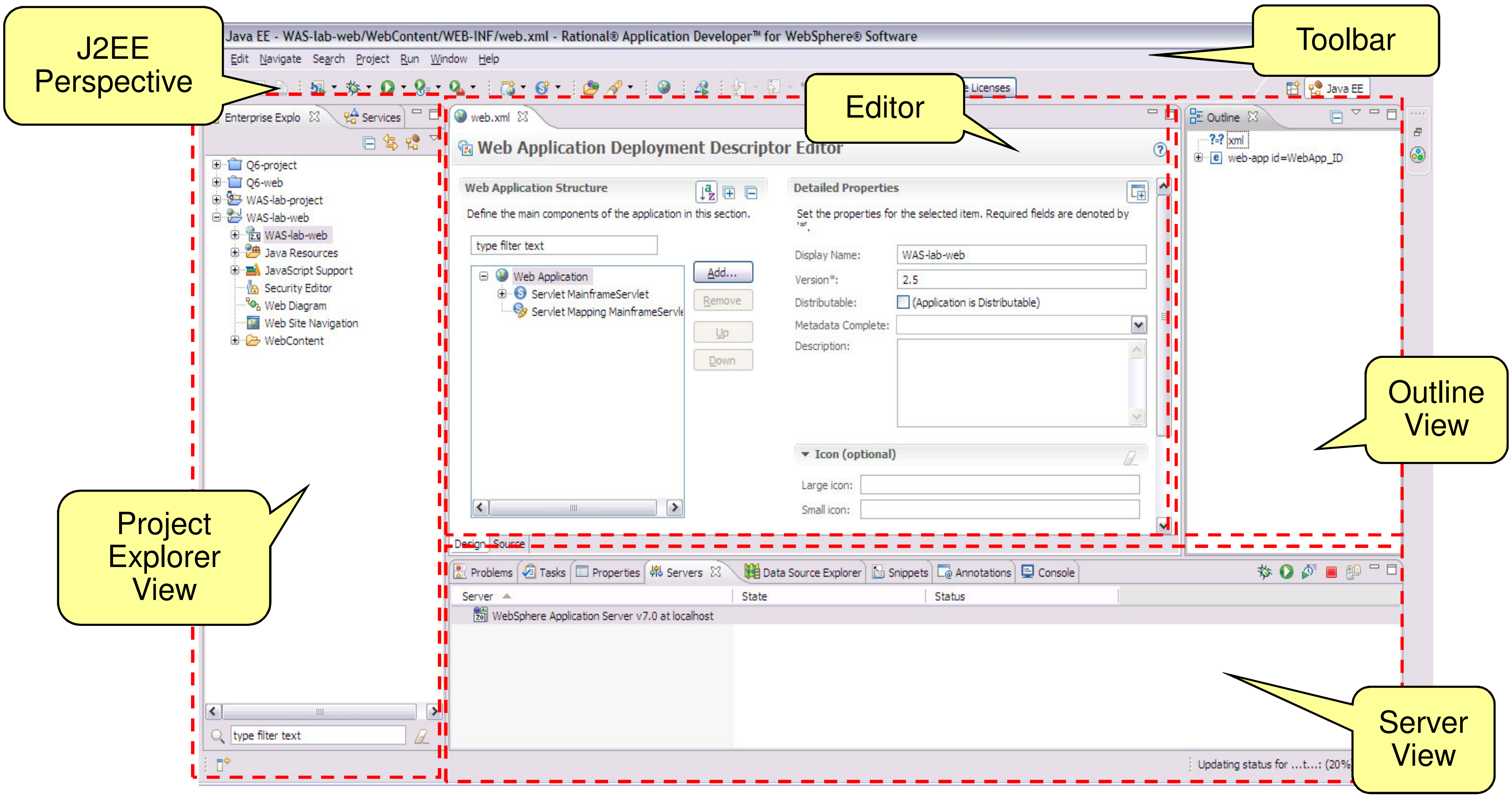


# WebSphere Application Packaging




DD = Deployment Descriptor

# The Application Server Toolkit Workbench



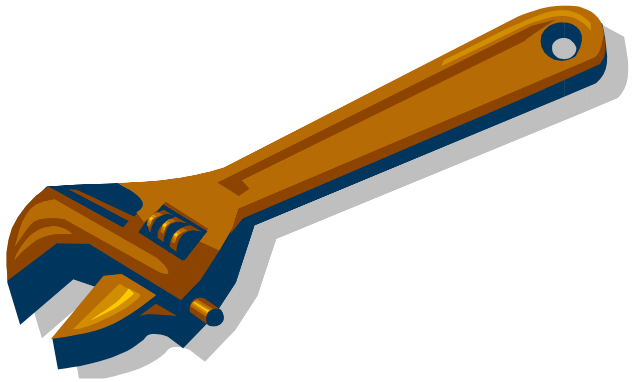

 

# Agenda

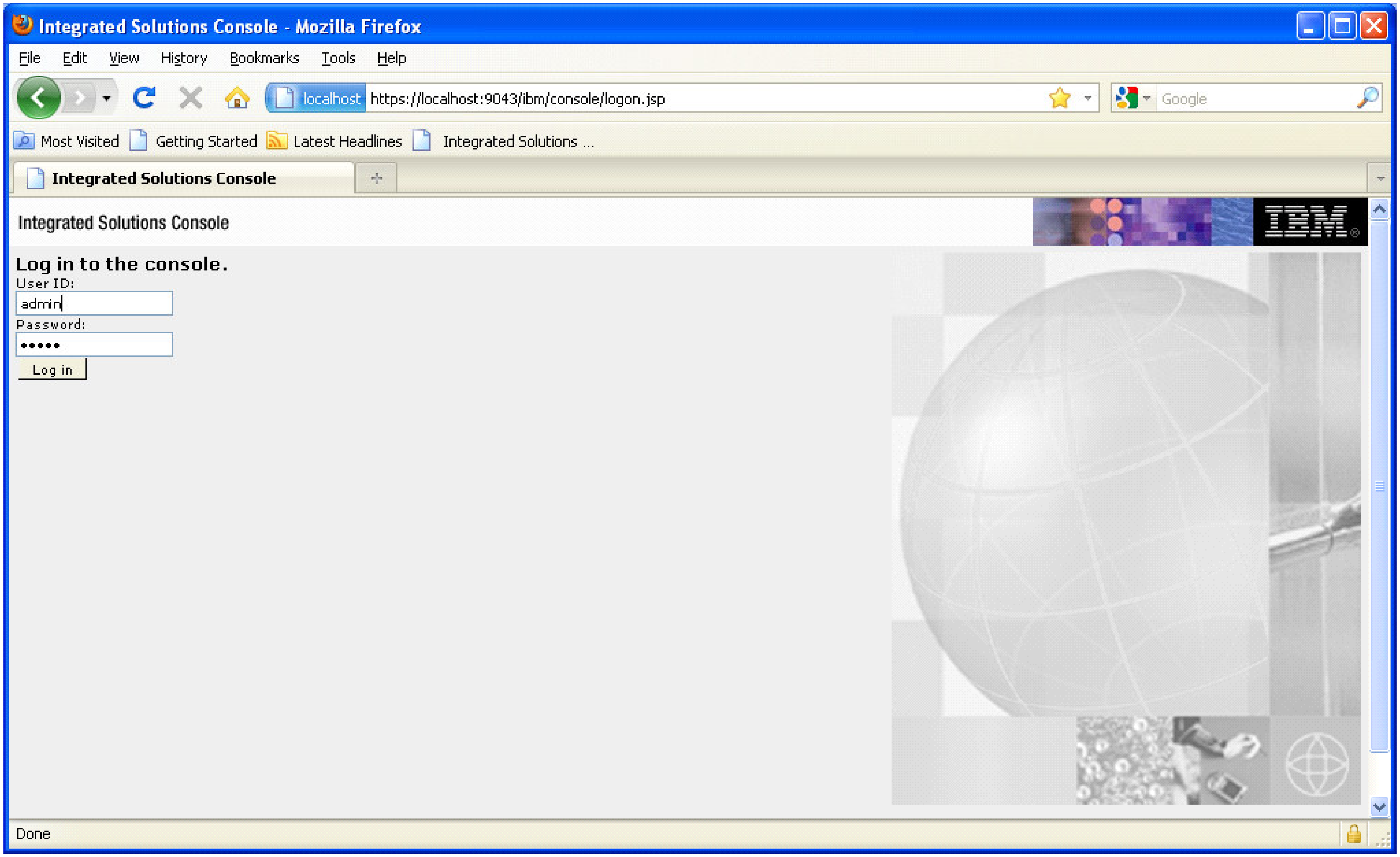
- What is an Application Server?
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- Application Server Architectural Overview
- Application Server in Cloud Computing
  - Topology Deployment
  - Infrastructure Virtualization
  - Health Management

# System Administration Tools

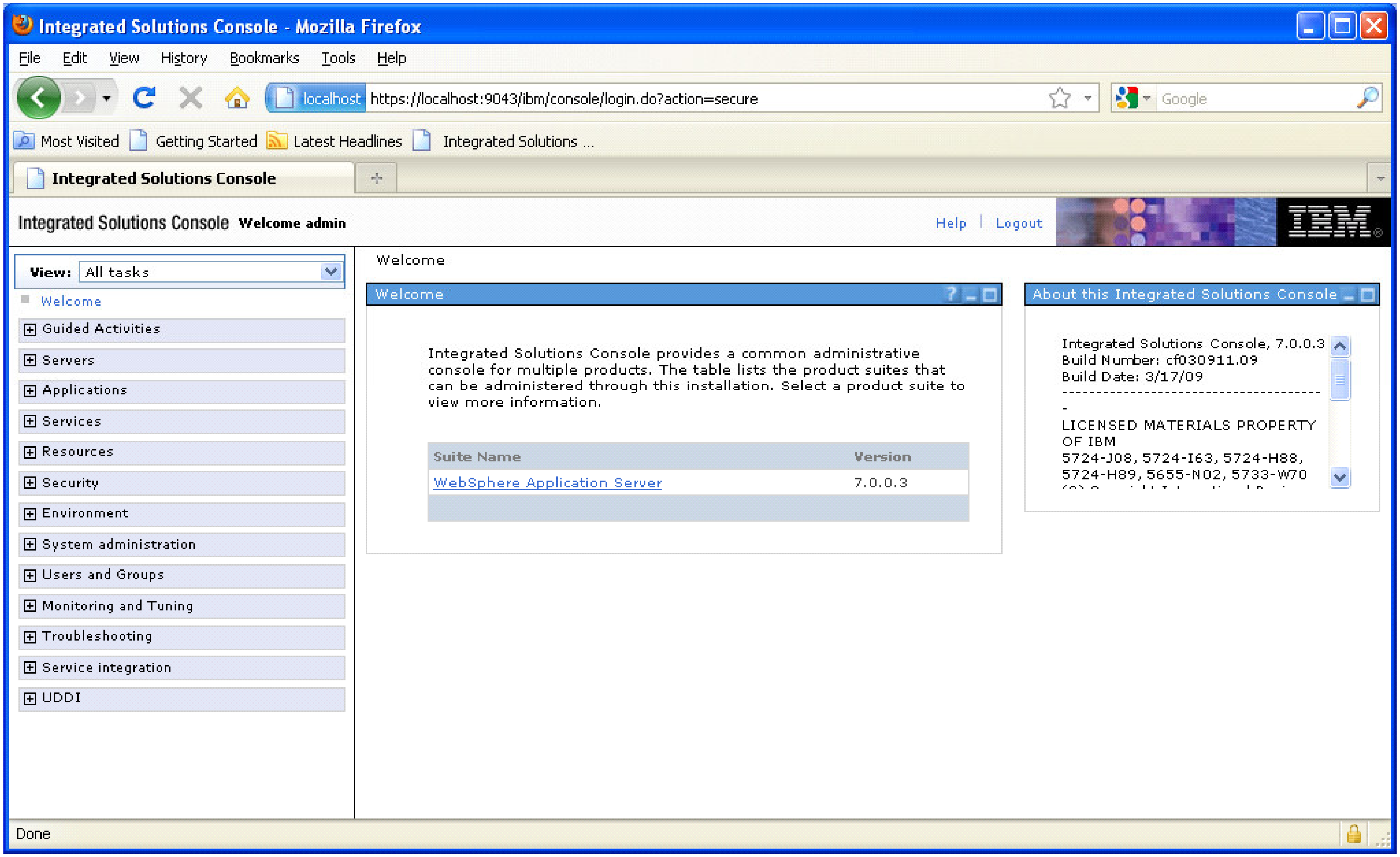
- WebSphere Integrated Solutions Console
  – Browser-based interface
- Command-line operation tools
  – Available in the bin directory
- wsadmin scripting
  – Interactive and batch modes
  – Supports JACL or Jython scripts
- Ant
  – Java-based build and automation tool
- Java-based JMX APIs
  – Programming interface for custom Java applications



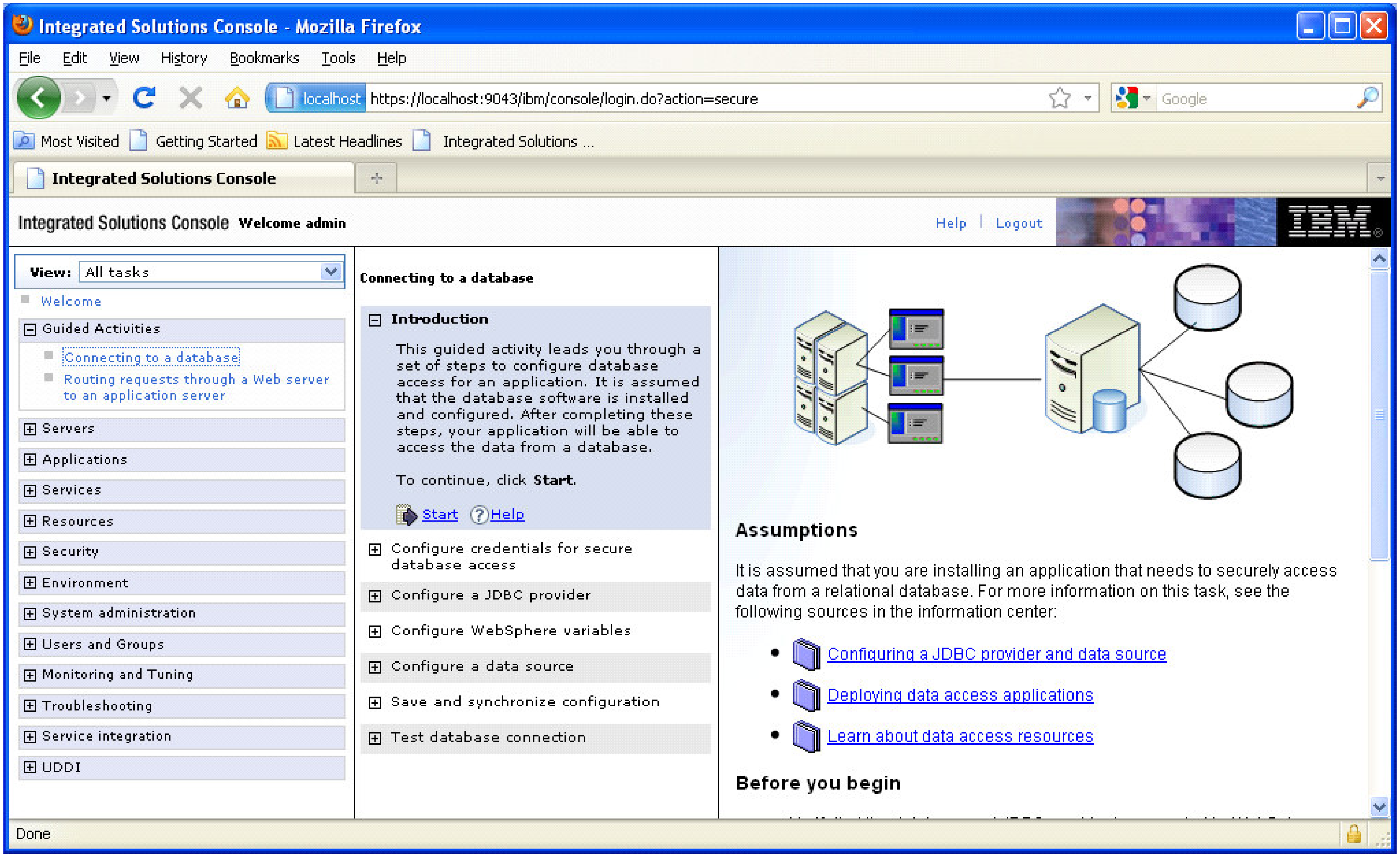
# Integrated Solutions Console



# Integrated Solutions Console – Welcome Page



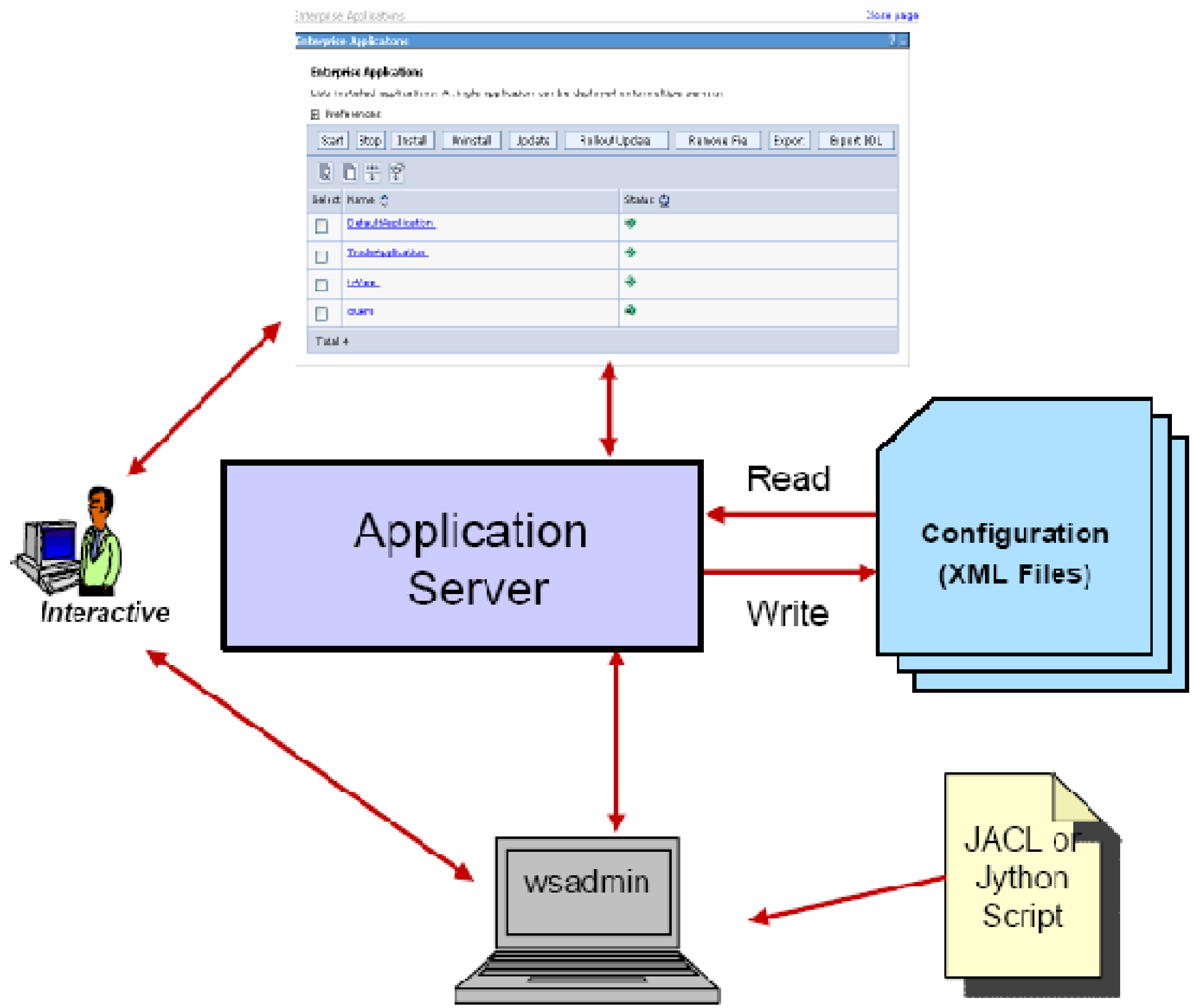
# Integrated Solutions Console – Cheat Sheet



# Agenda

- What is an Application Server?
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- Application Server Architectural Overview
- Application Server in Cloud Computing
  - Topology Deployment
  - Infrastructure Virtualization
  - Health Management

# Installing Enterprise Application



# Installation Tasks

- Configure the application environment as required
  – Variables, virtual hosts, classpath, security, and so forth
- Configure application resources
  – JDBC provider, DataSources
- Install application

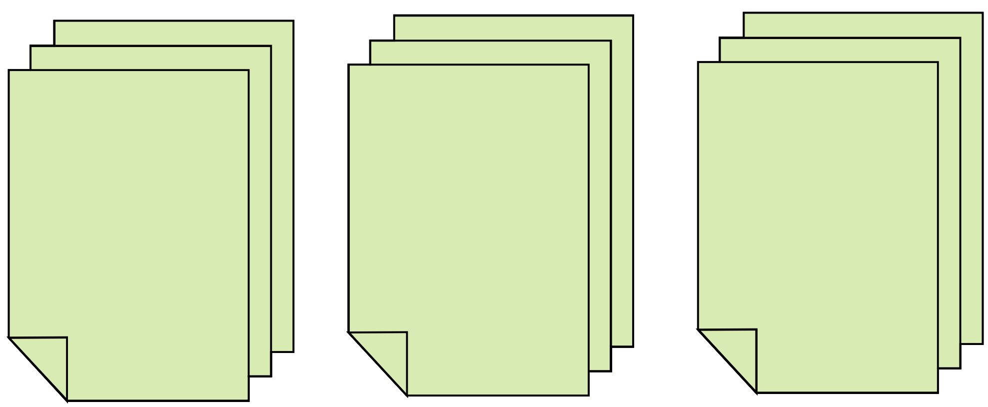
 IBM Confidential 

# Agenda

- What is an Application Server?
- JEE Packaging Overview
- Application Server Administration
  – Integrated Solutions Console
  – Application Install
  – Problem Determination
- Application Server Architectural Overview
- Application Server in Cloud Computing
  – Topology Deployment
  – Infrastructure Virtualization
  – Health Management

# Logging and Tracing

- Message logging (messages) and diagnostic trace (trace) are conceptually similar since they are using the same Java logging API
- They have important differences:
  – A **message entry** is an informational record that is intended for end users, systems administrators, and support personnel to view. The text of the message must be clear, concise, and interpretable by an end user.
  – A **trace entry** is an information record that is intended for service engineers or developers to use. As such, a trace record might be considerably more complex, verbose and detailed than a message entry.



# Log Files and Locations

- The destination and names for the log files are configurable. The default location is:
  <was_root>\profiles\<profile_name>\logs\<server_name>

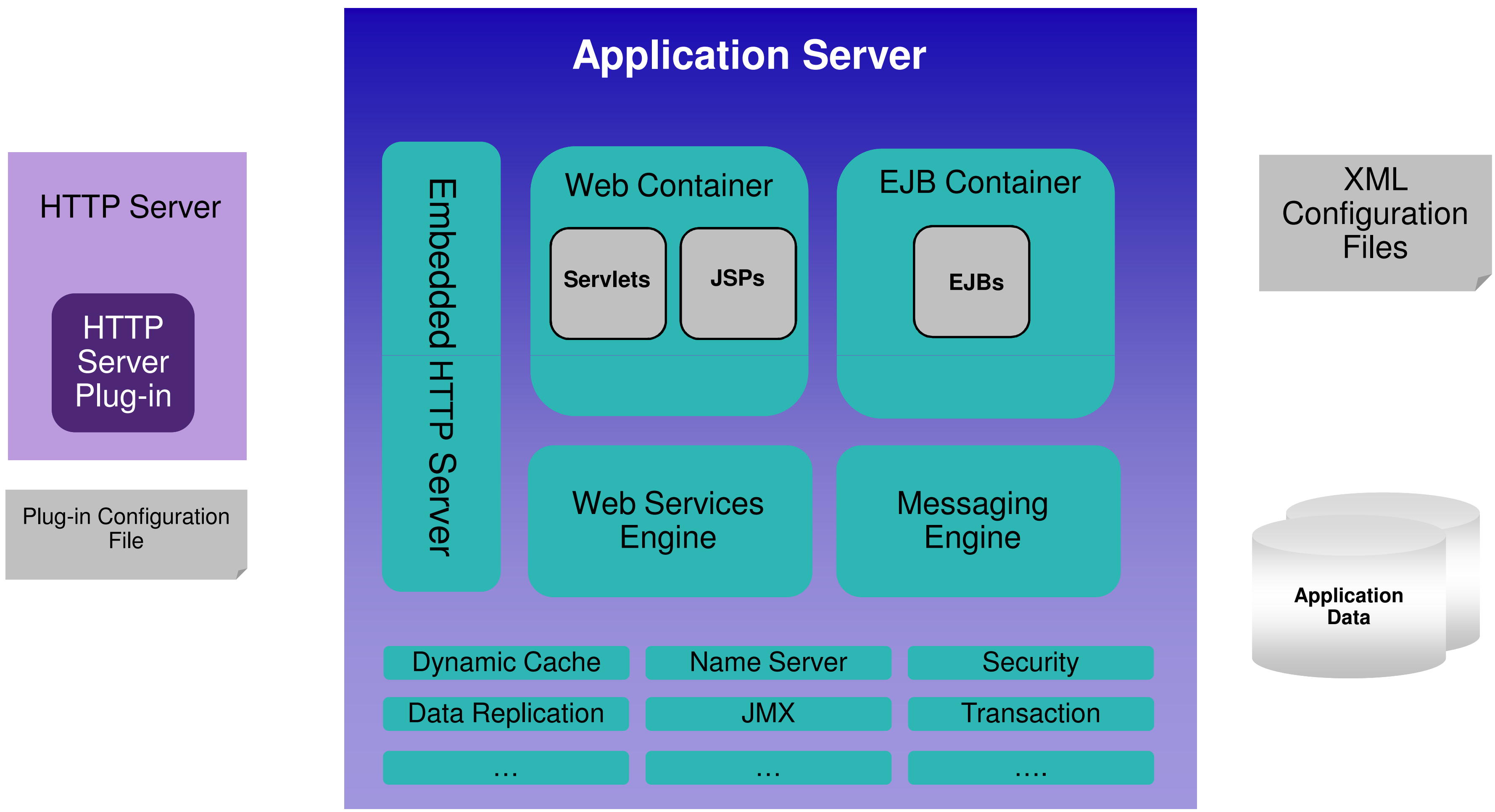
- Log Files:
  – **SystemOut.log** and **SystemErr.log** - Standard JVM output and error log
  – **startServer.log** and **stopServer.log**
    • Startup and shutdown of the Application Servers
  – **activity.log** - Events that show a history of installation activities
    • Use Log Analyzer to read output from this file
  – **trace.log** – Output from diagnostic trace
    • Destination and name are configurable
  – **http_plugin.log** – Not in <was_root>
    • Location: <plugin_root>\logs\<webserver_name>

 IBM Confidential 

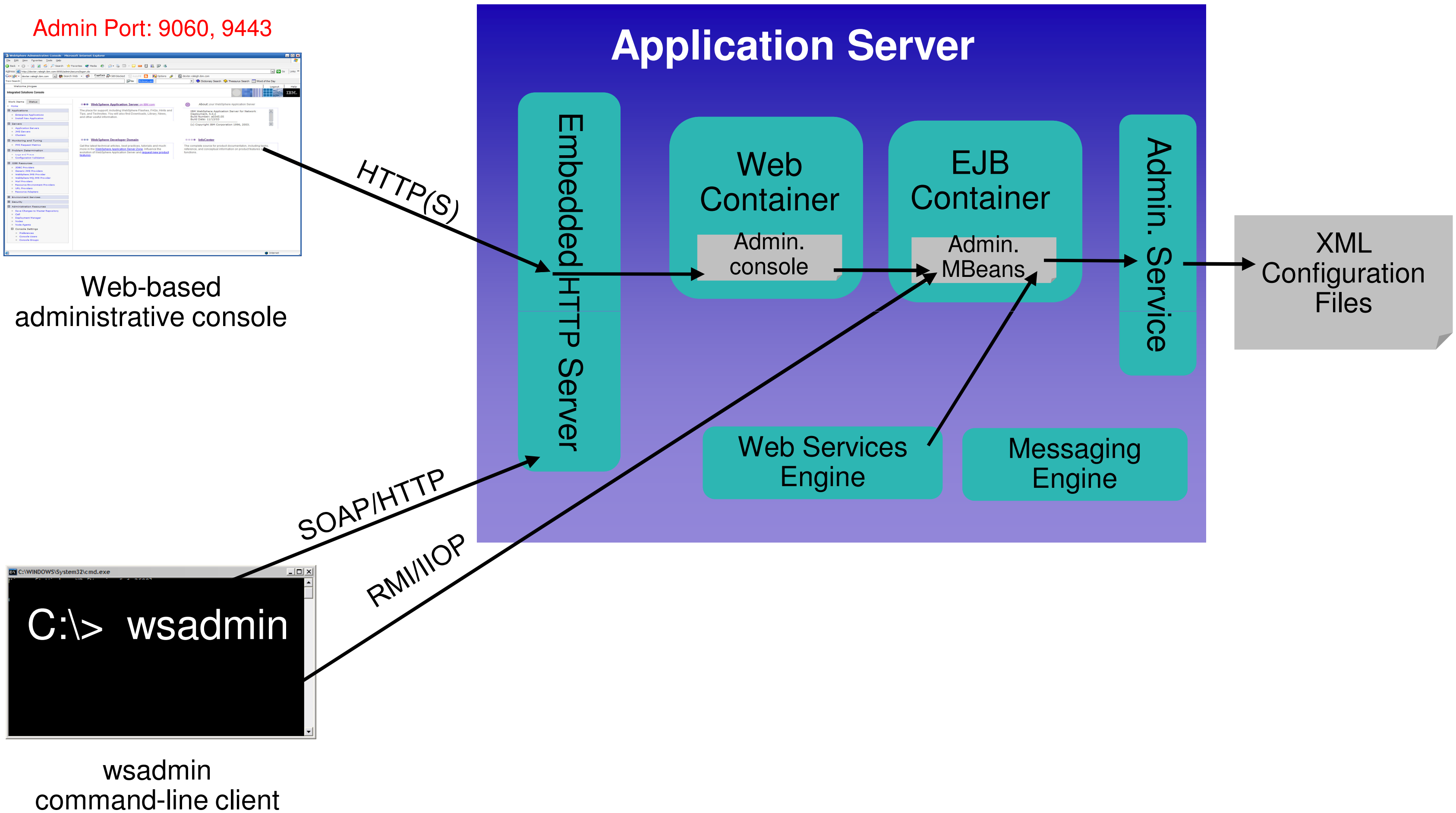
# Agenda

- What is an Application Server?
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- Application Server Architectural Overview
- Application Server in Cloud Computing
  - Topology Deployment
  - Infrastructure Virtualization
  - Health Management

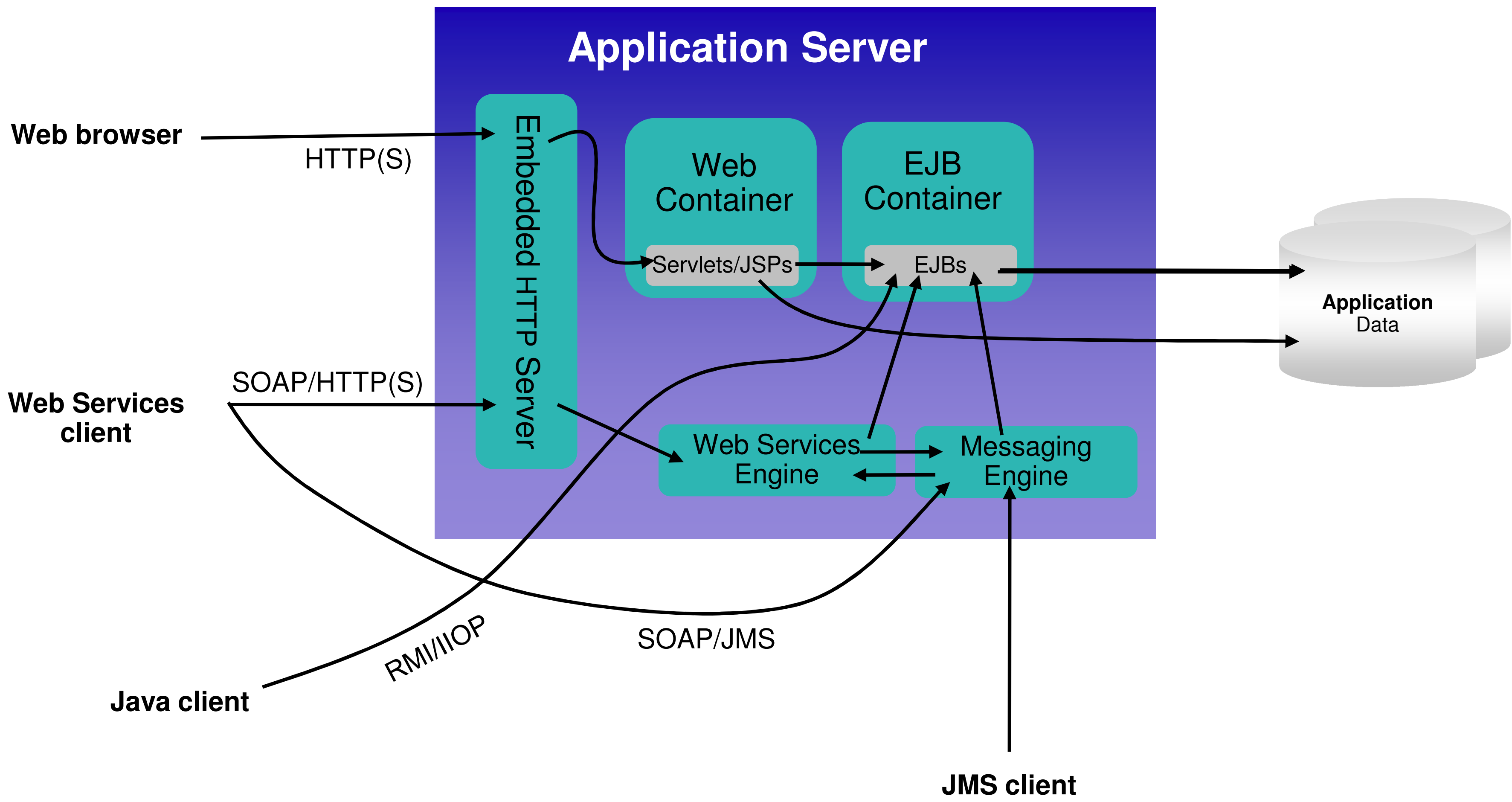
# Basic Architecture



# Administering the Server

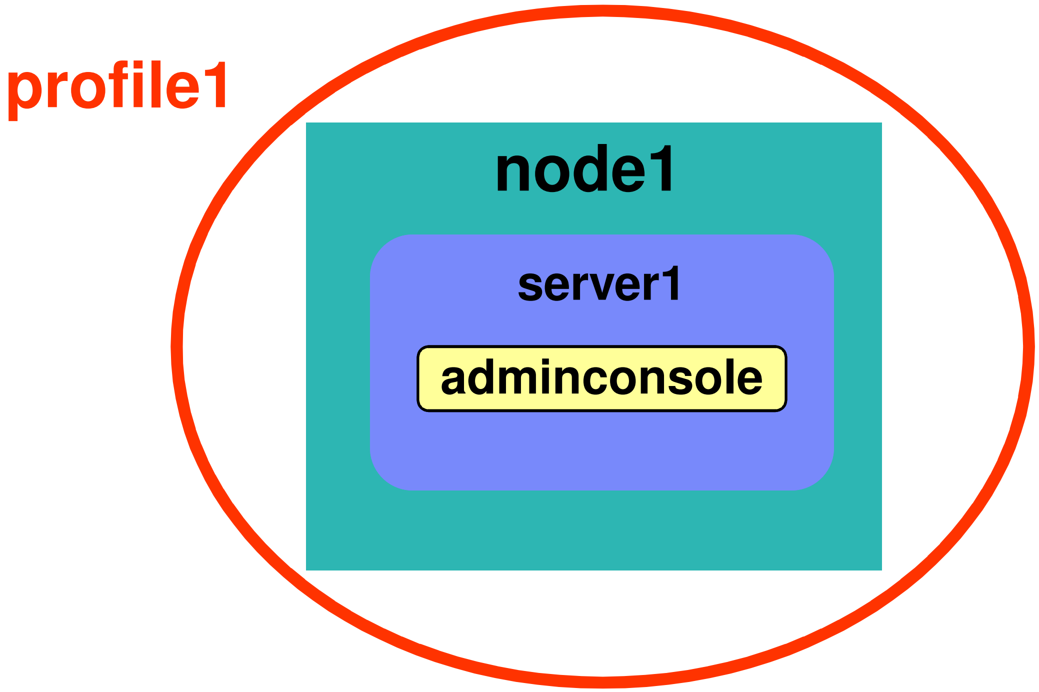


# Accessing Server Resources



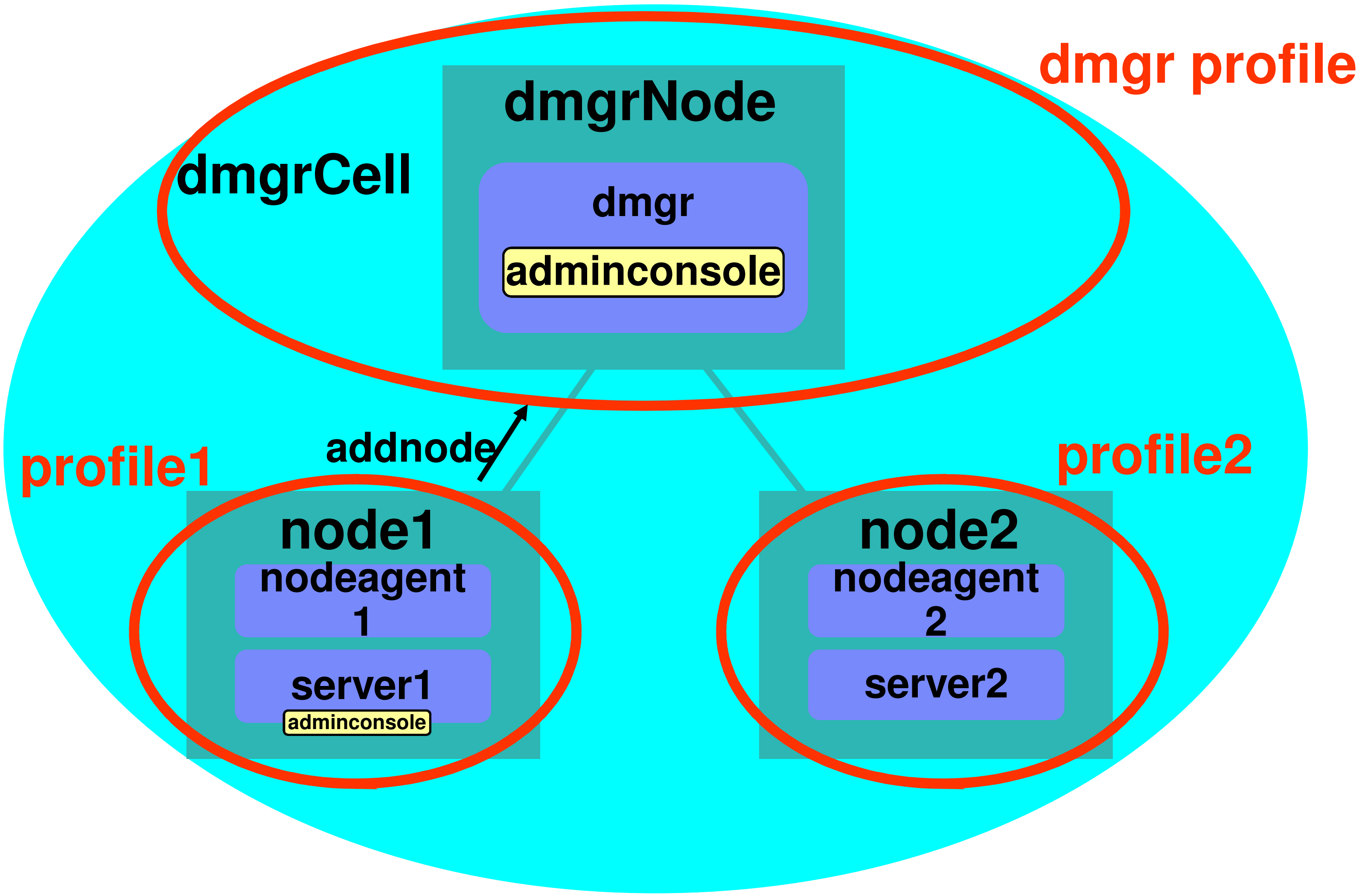

 IBM Confidential 

# Application Server – Standalone Solution

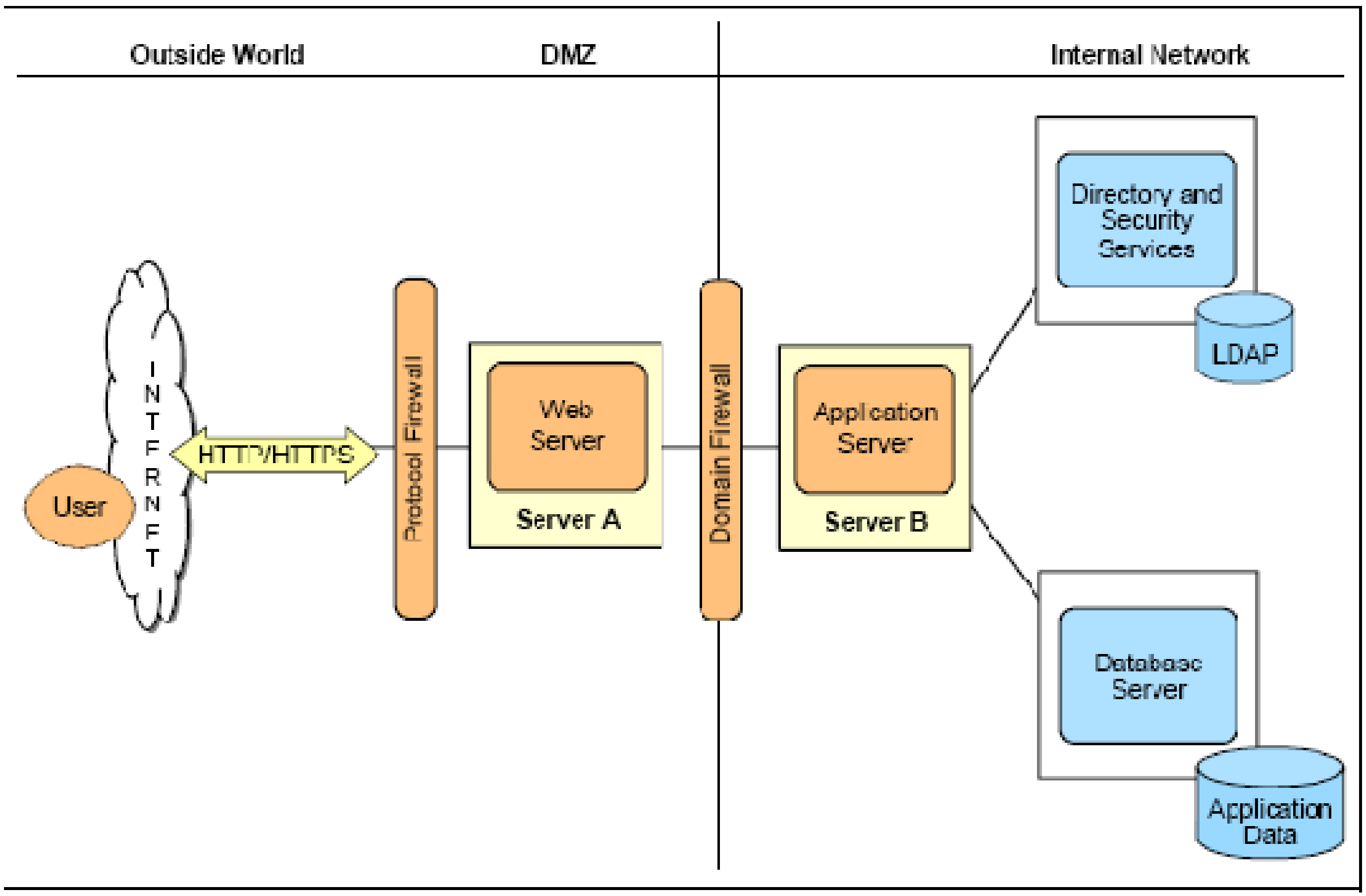



 IBM Confidential 

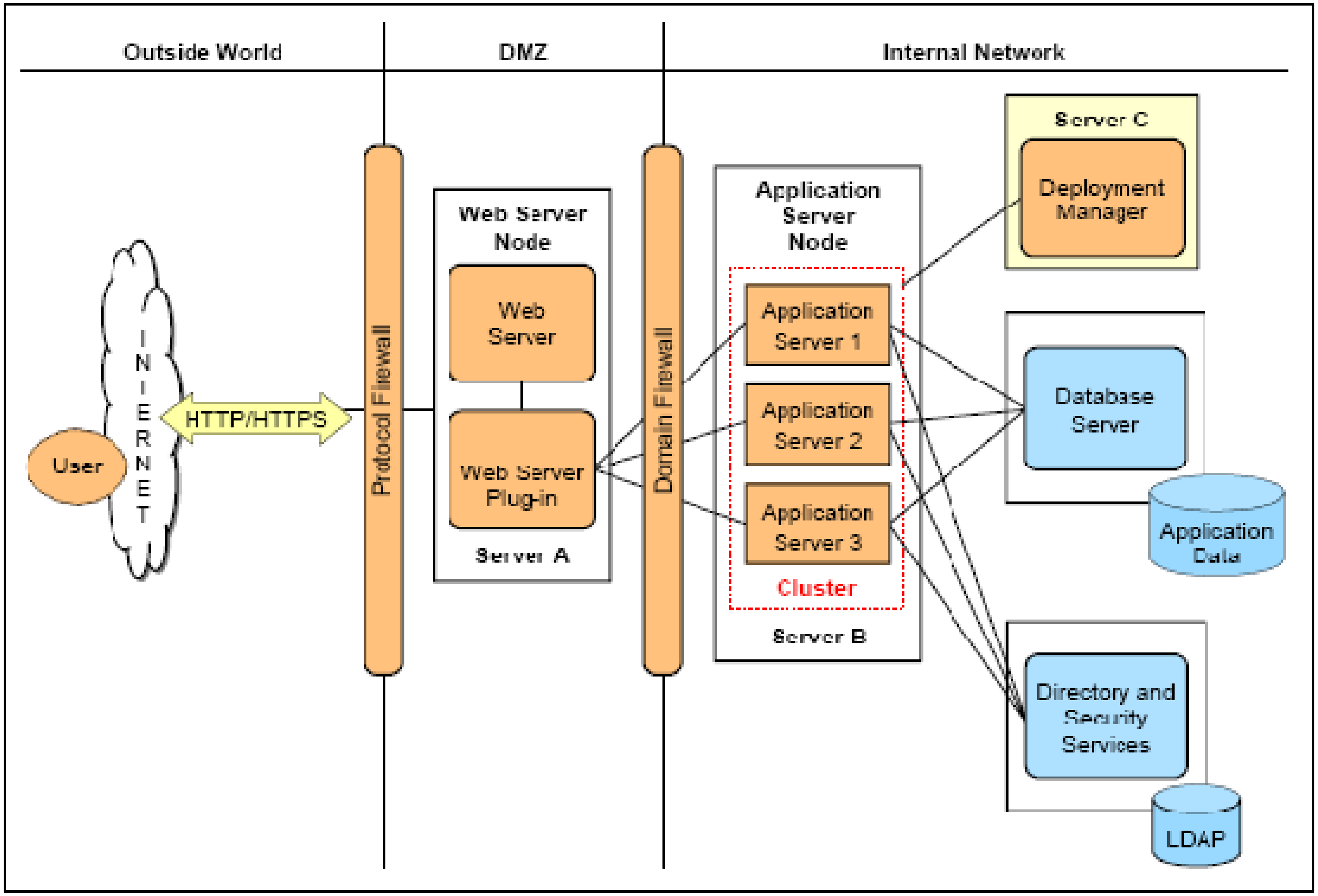
# Application Server – Cluster Solution



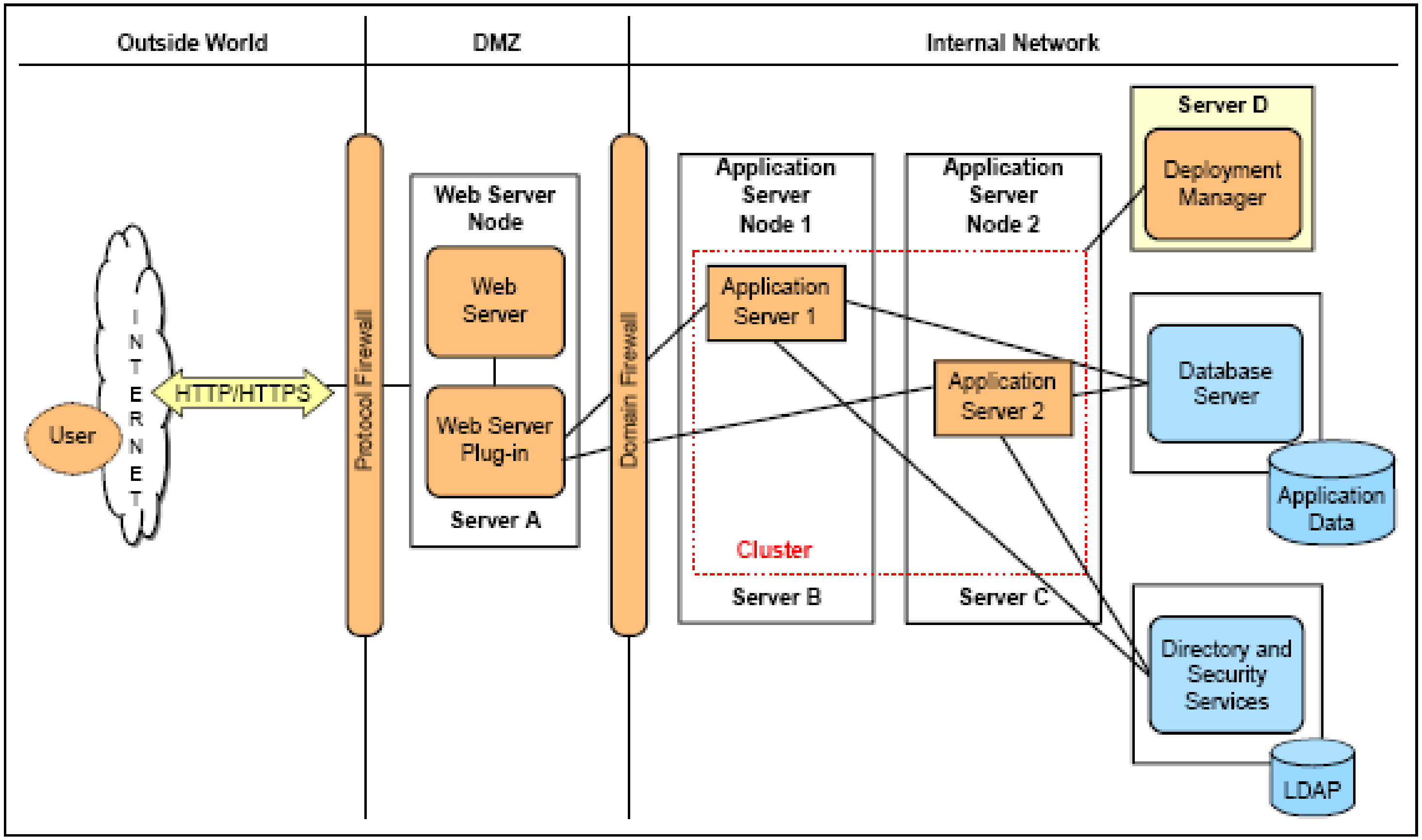
# Application Server Base Topology
## Single machine, single node, web server separated



# Application Server Vertical Scaling Topology



# Application Server Horizontal Scaling Topology



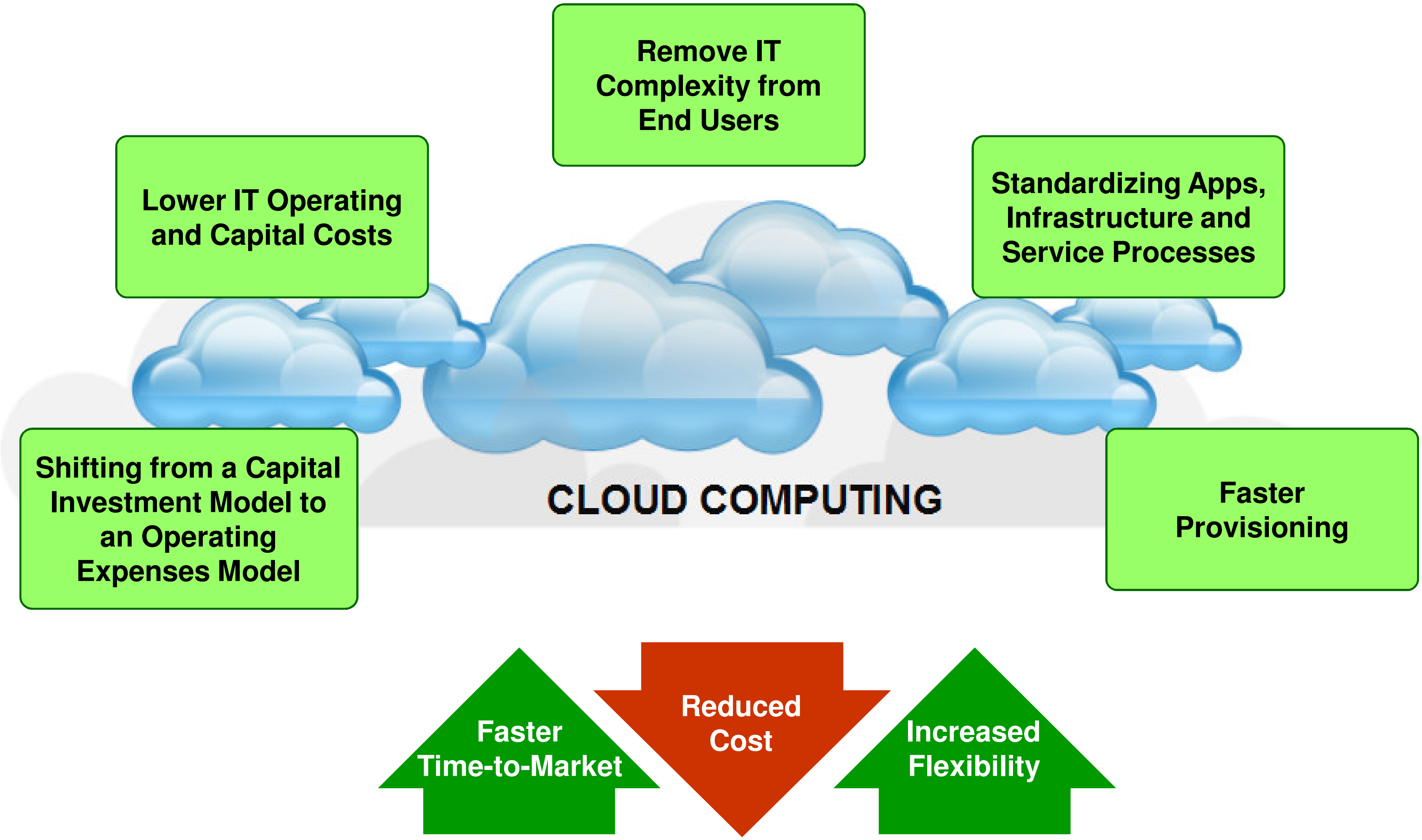
# Agenda

- What is an Application Server?
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- Application Server Architectural Overview
- Application Server in Cloud Computing
  - Topology Deployment
  - Infrastructure Virtualization
  - Health Management


lonelyplanetimages.com

# Do you feel like this?

# Cloud computing benefits



# Characteristics of a Cloud

- ***More Responsive***: Dynamically allocates resources to meet demands
- ***More Optimized***: better utilizes system resources and lowers TCO
- ***More Agile***: better aligns IT capabilities with business needs
- ***More Resilient***: prevents, isolates, and recovers from failures

***Pre-Cloud middleware was…***

- ***Scalable***: Add additional resources to meet demands
- ***Available***: Redundancy to avoid outages
- ***Consolidated***: Shared hardware resources

# Agenda

- What is an Application Server?
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- Application Server Architectural Overview
- Application Server in Cloud Computing
  - Topology Deployment
  - Infrastructure Virtualization
  - Health Management

# Topology Deployment Pains

- Customer Pains
  - *Takes too long to create middleware infrastructures*
  - *Manual and error-prone process*

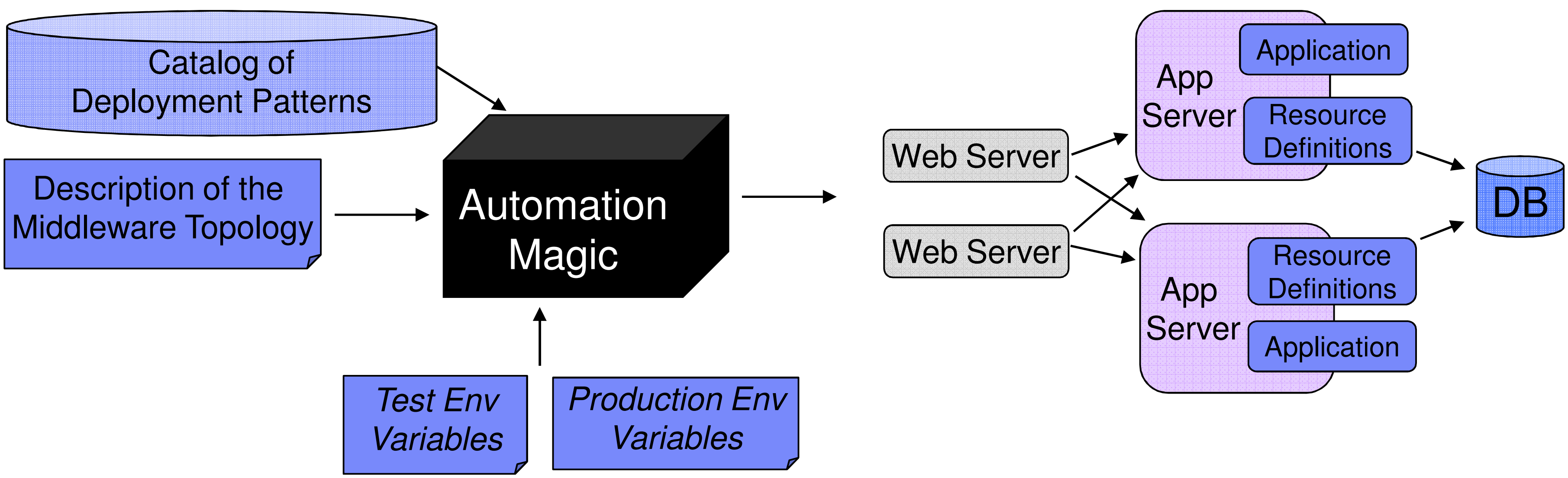
## What admin & management efficiencies can be made?

- The average lead time to get a new application environment up and running is 4-6 weeks
  - Approvals, procurement, shipment, HW installation, license procurement, OS installation, application installation, configuration

- 30% of bugs are introduced by inconsistent configurations
  - These bugs are often of the most difficult variety to detect
  - They often emerge when moving between dev/test, QA, production

- Because it's so expensive to set up an environment, there is an incentive to hold onto them even when no longer needed "just in case."
  - Future environments = new hardware, instead of recycling returned hardware, and this takes time and money

1. ***Speed*** - build out environments quickly
   - A new application cell should take you minutes, not weeks
2. ***Repeatability*** - automation to avoid redundancy
   - Your 2nd, 3rd, and Nth environment should be faster to build than your 1st environment
3. ***Consistency*** - limit the variables across deployments
   - Your configurations should get seamlessly promoted from test -> pre-production -> production



# Agenda

- What is an Application Server?
- JEE Packaging Overview
- Application Server Administration
  – Integrated Solutions Console
  – Application Install
  – Problem Determination
- Application Server Architectural Overview
- Application Server in Cloud Computing
  – Topology Deployment
  – Infrastructure Virtualization
  – Health Management

# Middleware Infrastructure Pain

1. Static Middleware Infrastructure
   - Doesn’t react well to spikes in demand
   - Resources are under-utilized
   - Not well-aligned with the business

   *The infrastructure should manage provisioning application and middleware resources to achieve some stated business level objectives.*

2. Fragile Middleware Infrastructure
   - System can’t detect that a failure will probably occur
   - Failures aren’t isolated, and impact more than it should

   *The infrastructure should monitor and react to conditions that effect the “health” of the cluster-member JVM’s*

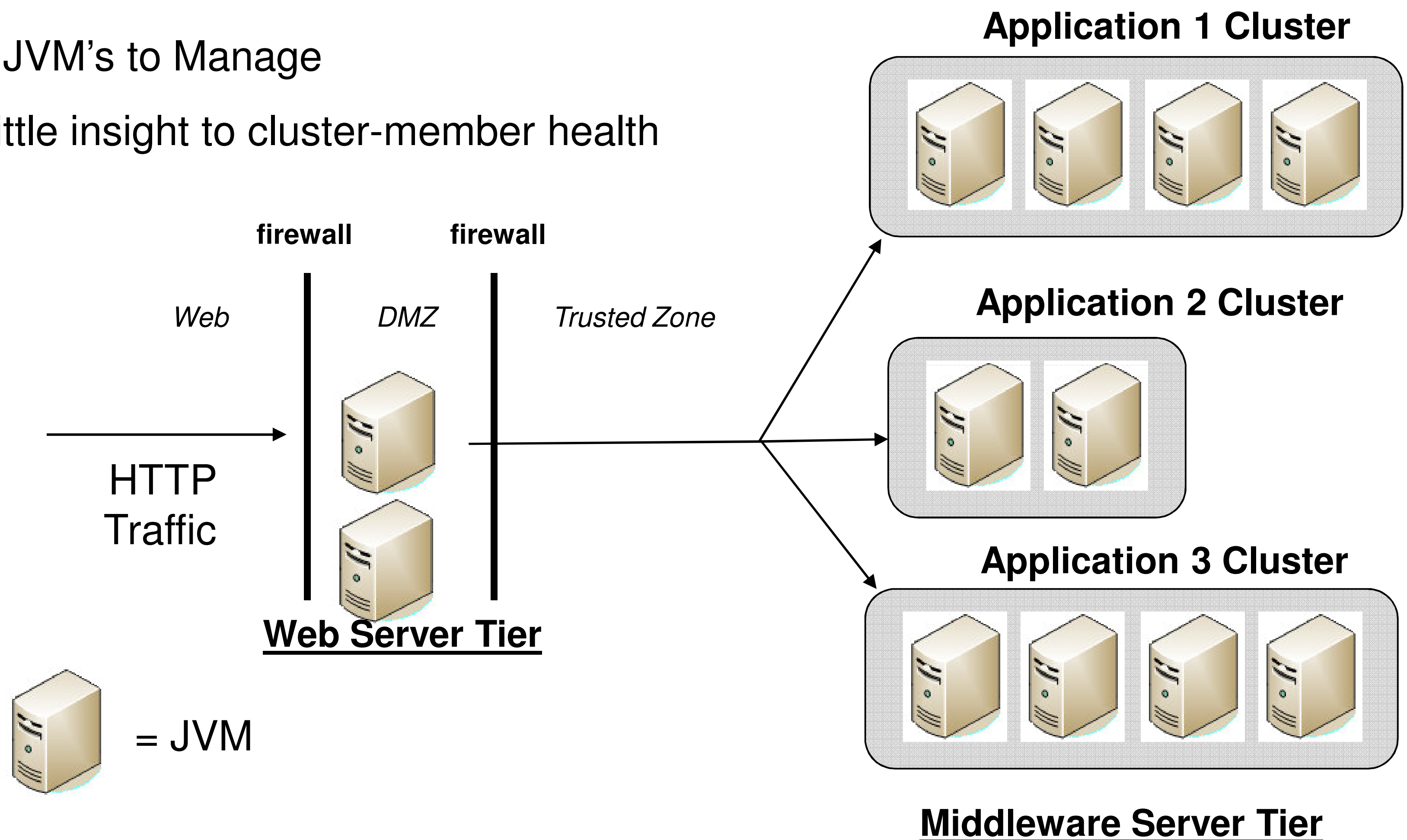
# Traditional Middleware Architecture

-Application clusters are statically defined

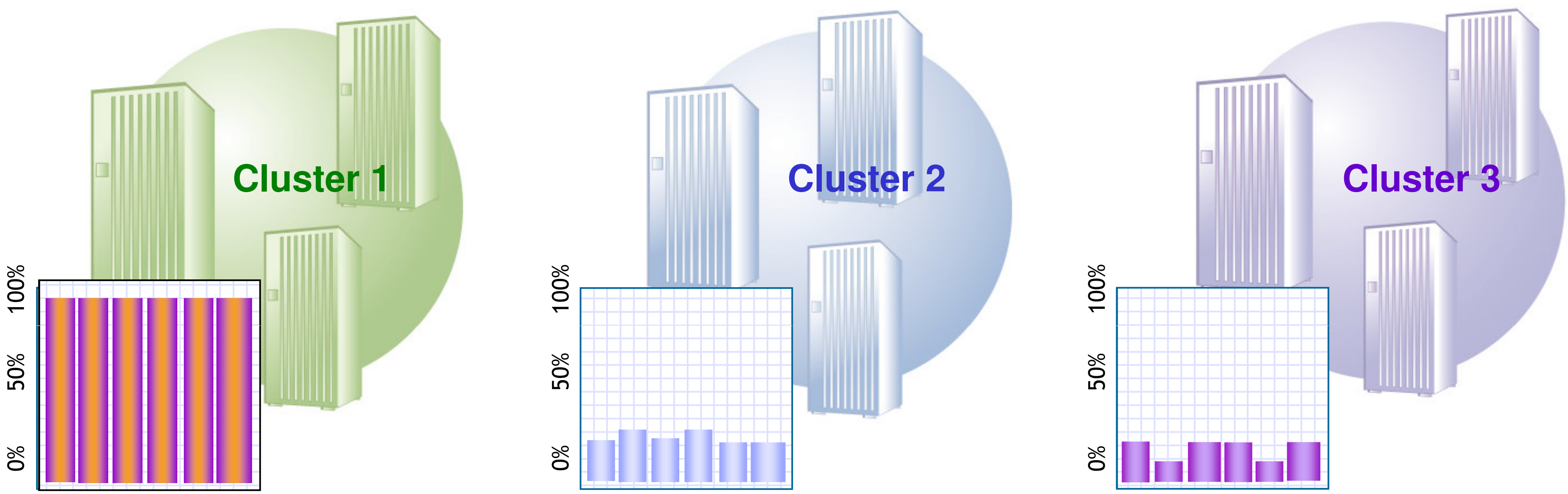
-Low CPU utilization

-Many JVM's to Manage

-Very little insight to cluster-member health



# Static clustered environment without IBM Smart Cloud




**Sudden change in market:**
**100% Utilized Servers**

15% Utilized Servers

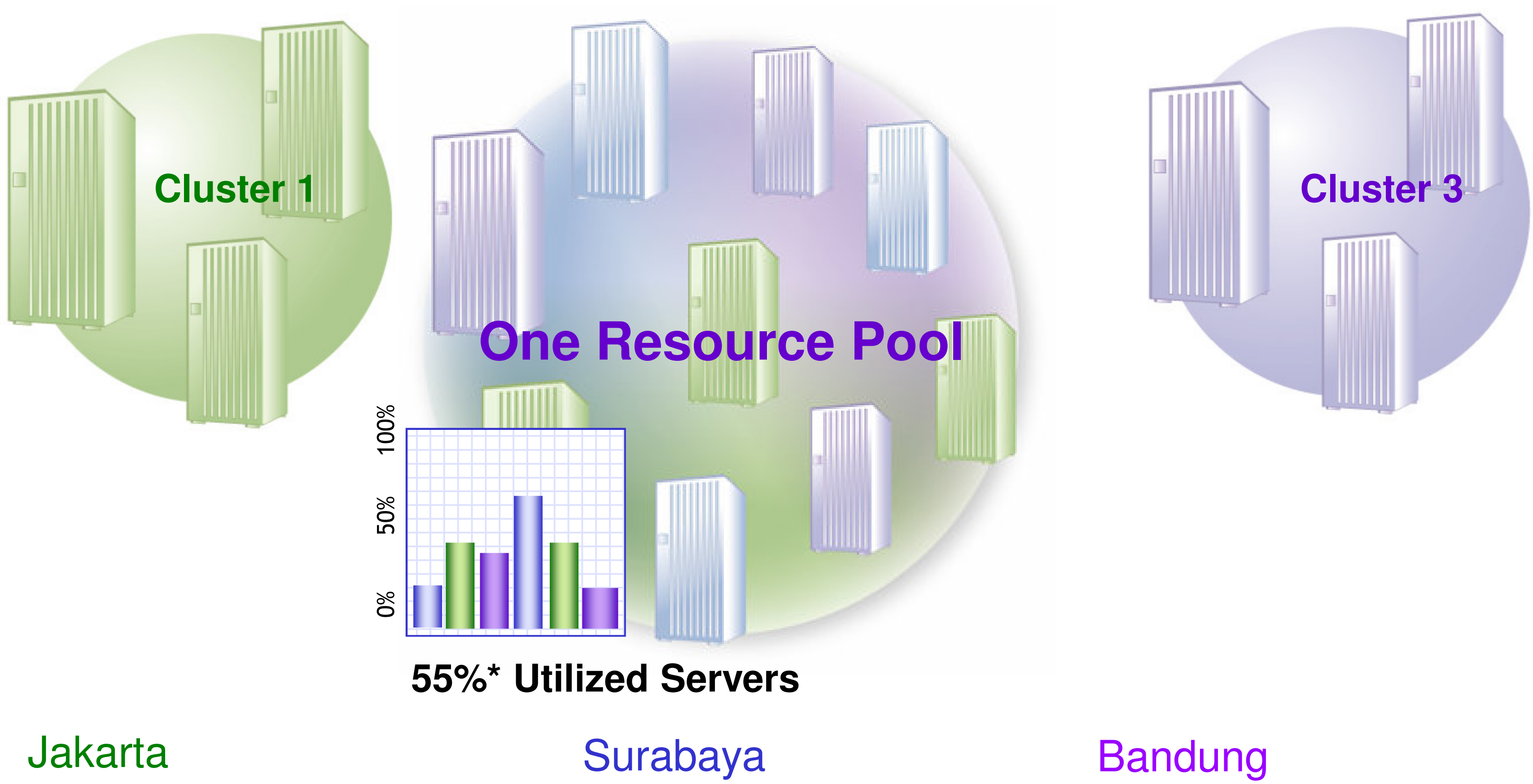
10% Utilized Servers

Jakarta

Surabaya

Bandung

# *IBM Smart Cloud provides application infrastructure virtualization capabilities to maximize application server utilization*

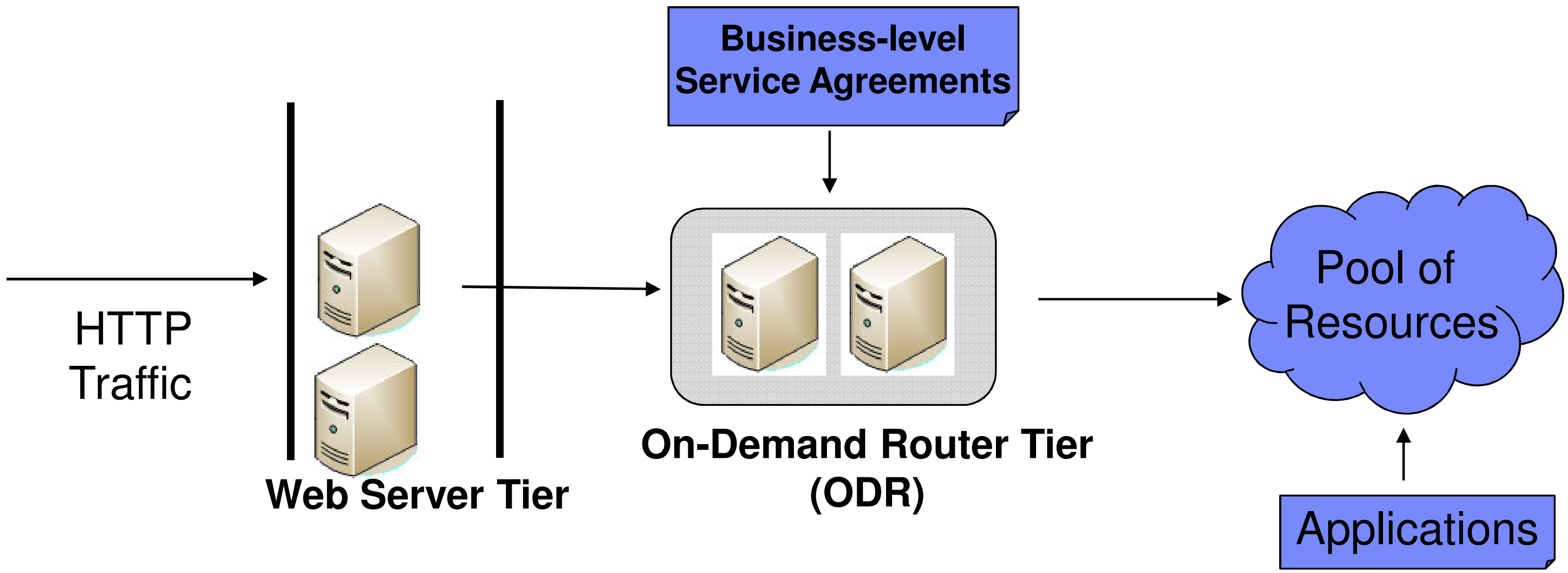



* Hypothetical, for illustrative purposes

# Policy-based Management

-ODR is responsible for:

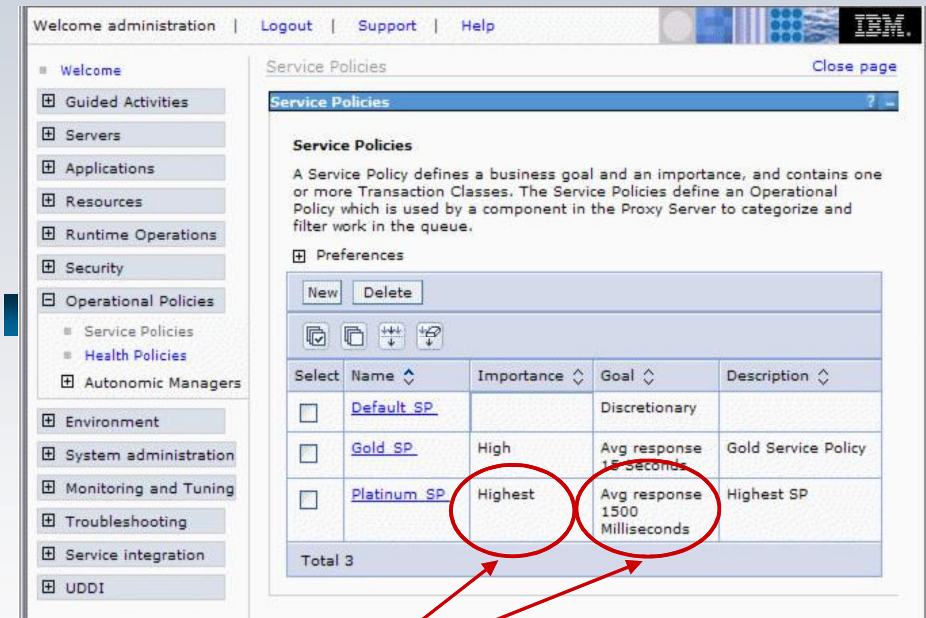
- -starting/stopping JVM's to meet demand
- -Ensuring higher priority applications/users/etc are serviced first (via flow control)
- -Administrators describe requirements (min/max application instances, etc) through policies
- -Resource-driven load-balancing within clusters
- -Keeping track of application/user/etc resource usage (for chargeback)



# Application Prioritization: Doing What's Important to You

IBM Smart Cloud easily allows an administrator to specify the relative importance of applications; IBM Smart Cloud then manages to it

- **Service policies** are used to define application service level goals
- Allow workloads to be classified, prioritized and intelligently routed
- Enables application performance monitoring
- Resource adjustments are made if needed to consistently achieve service policies



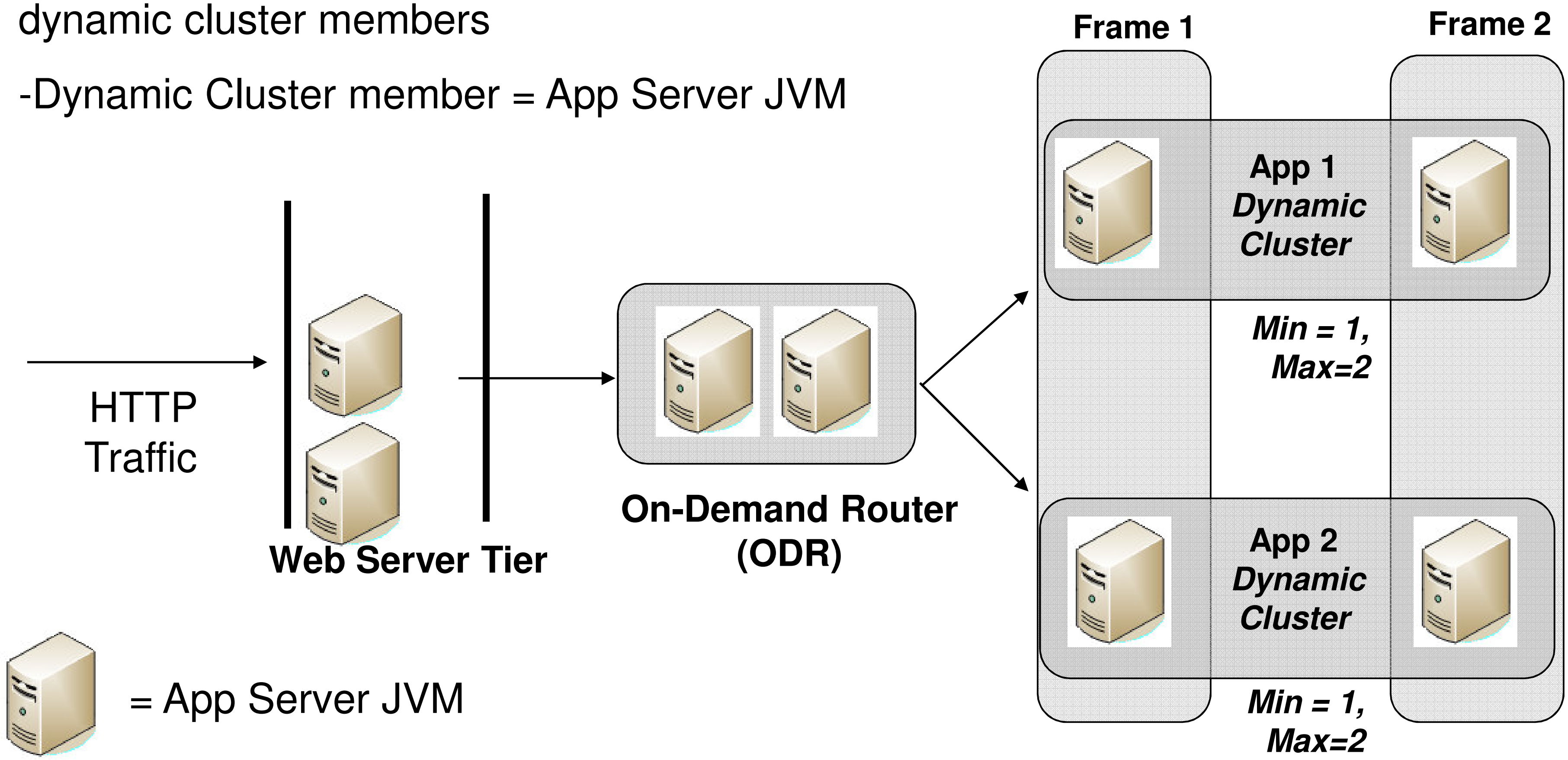

***Service Policies define the relative importance and response time goals of application services; defined in terms the end user result the customer wishes to achieve***

# Optimizing the Pool of Resources with Dynamic Clusters

-Applications are installed to a ***Dynamic Cluster***

-ODR uses policies to determine when to start/stop dynamic cluster members

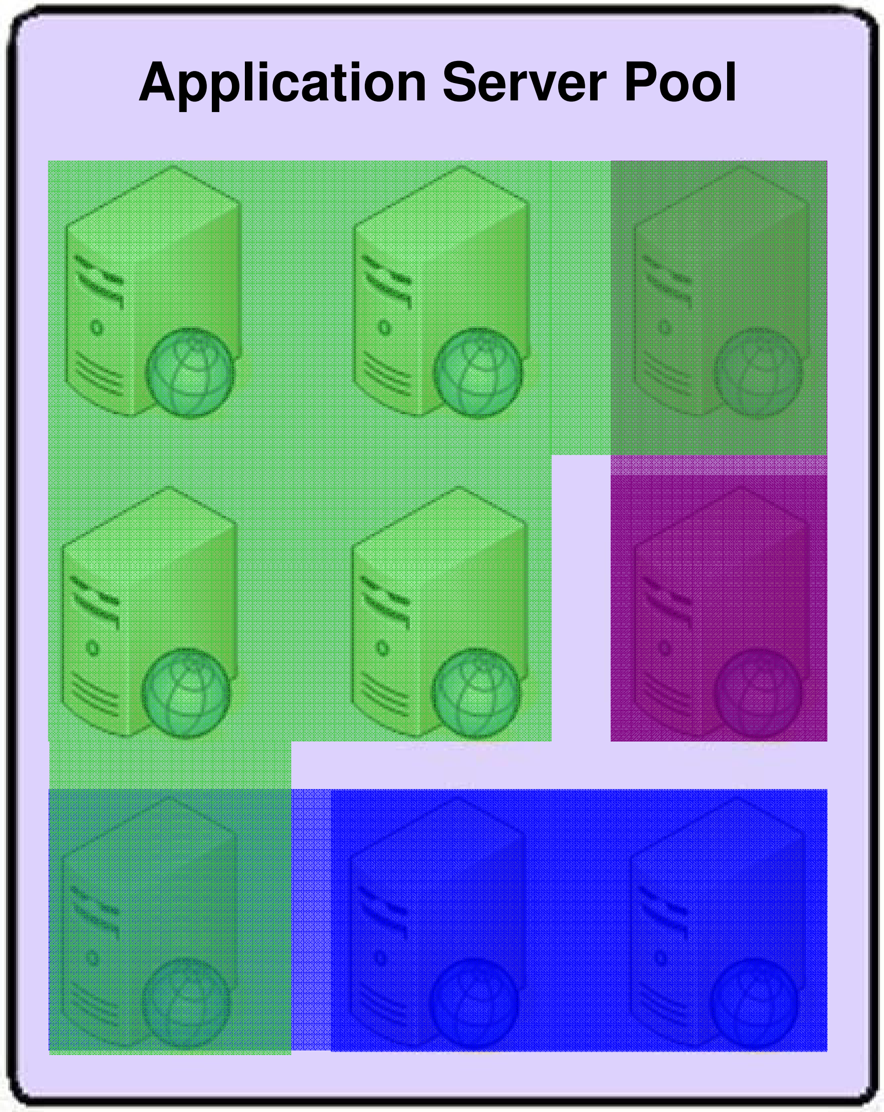
-Dynamic Cluster member = App Server JVM



# Ensures SLAs are met

| Application Name | Priority | Avg. CPU Utilization |
|---|---|---|
| Jakarta | High | [illegible] |
| Surabaya | Medium | [illegible] |
| Bandung | Low | [illegible] |

Hypothetical scenario #1:
Interest rates DROP in JKT

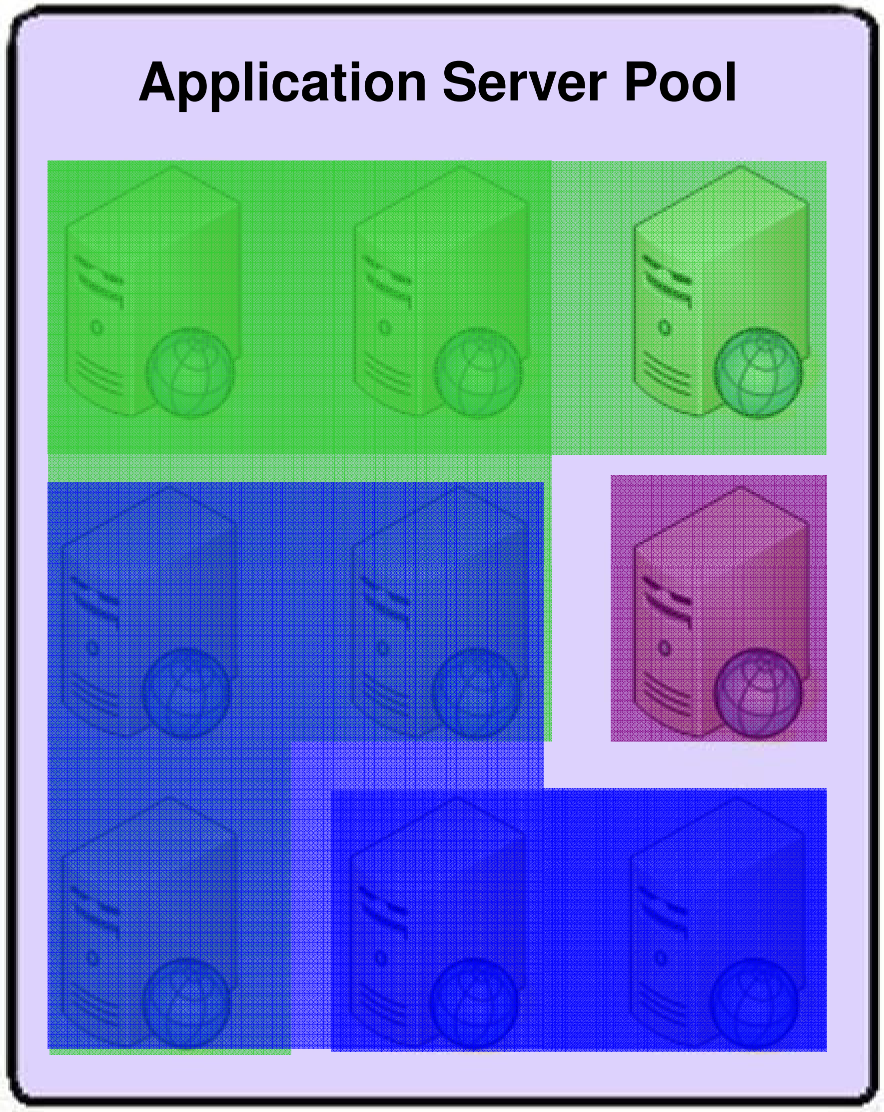



* Hypothetical, for illustrative purposes

## *Ensures SLAs are met*

| Application Name | Priority | Avg. CPU Utilization |
|---|---|---|
| Jakarta | High | 10% |
| Surabaya | Medium | [illegible] |
| Bandung | Low | 5% |

Hypothetical scenario #2:
Jobs increase 20% in SUB




* Hypothetical, for illustrative purposes

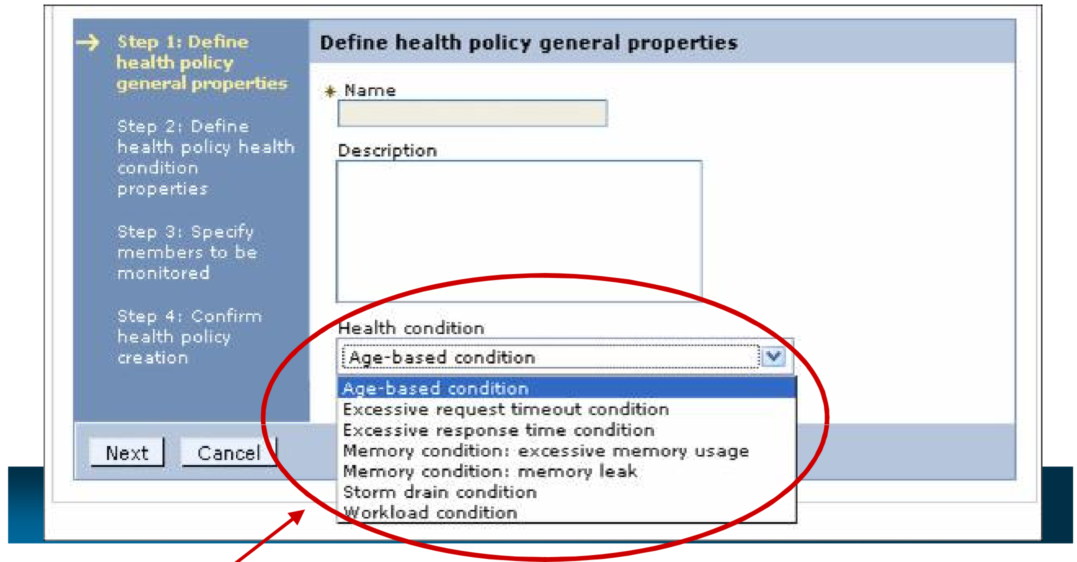
# Agenda

- What is an Application Server?
- JEE Packaging Overview
- Application Server Administration
  - Integrated Solutions Console
  - Application Install
  - Problem Determination
- Application Server Architectural Overview
- Application Server in Cloud Computing
  - Topology Deployment
  - Infrastructure Virtualization
  - Health Management

# Health Management – Health Policies

*Helps mitigate common health problems before production outages occur*

- Health policies can be defined for common server health conditions
- Health conditions are monitored and corrective actions taken automatically
  - Notify administrator
  - Capture diagnostics
  - Restart server
- Application server restarts are done in a way that prevent outages and service policy violations




***Health Conditions***

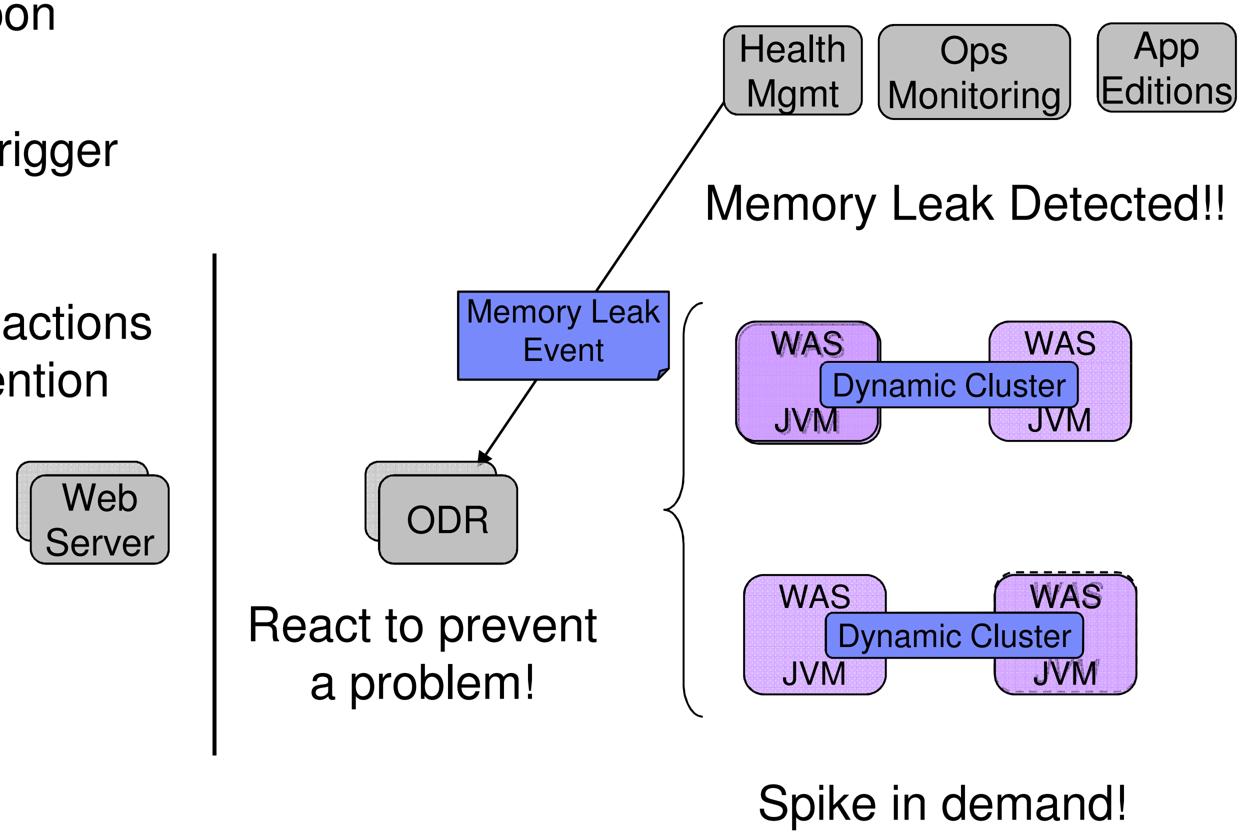
- **Age-based:** *amount of time server has been running*
- **Excessive requests:** *% of timed out requests*
- **Excessive response time:** *average response time*
- **Excessive memory:** *% of maximum JVM heap size*
- **Memory leak:** *JVM heap size after garbage collection*
- **Storm drain:** *significant drop in response time*
- **Workload:** *total number of requests*

# IBM Smart Cloud- Some examples

- A memory leak in the JVM could be detected and reacted upon

- A spike in demand could trigger additional JVM’s be started

- Decisions can made, and actions taken without Admin intervention



# IBM Smart Cloud Key Capabilities

Title Search

Assess risk

Issue Title

Tax Records

Manage account

**Applications**

## a. Server Consolidation

- Dynamic workload adjustment

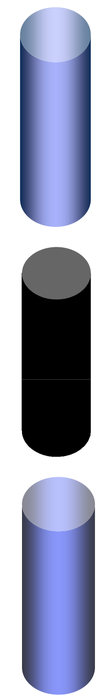
## b. Service Level Management

- Pooled resources
- Prioritized workloads using service policies

## c. Health Management

- Application performance monitoring
- Prevent outages by taking corrective action for common server health problems

**Application Resources**



**Information Resources**


Questions
&
Answers