# JUDCon

## JBoss Users & Developers Conference

# 2012:India


Bruno Georges
Director, Software Engineering
JBoss by Red Hat

# Agenda


- JBoss Application Server 7
  - Goals
  - Key Features
  - Roadmap
- Domain Model Deep Dive
  - Key Operations, Administration & Management (OA&M)
  - Domain Mode and Standalone Mode
  - Key Management Model Concepts
  - Management APIs and Interfaces
  - Domain Configuration
- Q &A


# JBoss AS7 - Goals and Motivations

- JBoss Everywhere
  - Cloud
  - Mobile Devices
- Improve Performance
  - Footprint
  - Bootstrap
- Operational Effectiveness
  - Simple
  - Manageable
- Developer Productivity
  - EE6
  - Testability


# JBoss AS7 - What's new in EE6

- **JAX-RS** – RESTful Web Services (JSR 311)
  - Create web services without WSDL
- **CDI** – Context & Dependency Injection (JSR 299)
  - Reference Implementation is Weld
  - Based on JBoss Seam
  - Brings transactional support to the web tier
  - Defines a rich set of lifecycle contexts to more easily design your web apps
- **Bean Validation** (JSR 303)
  - Reference implementation is Hiberate Validator

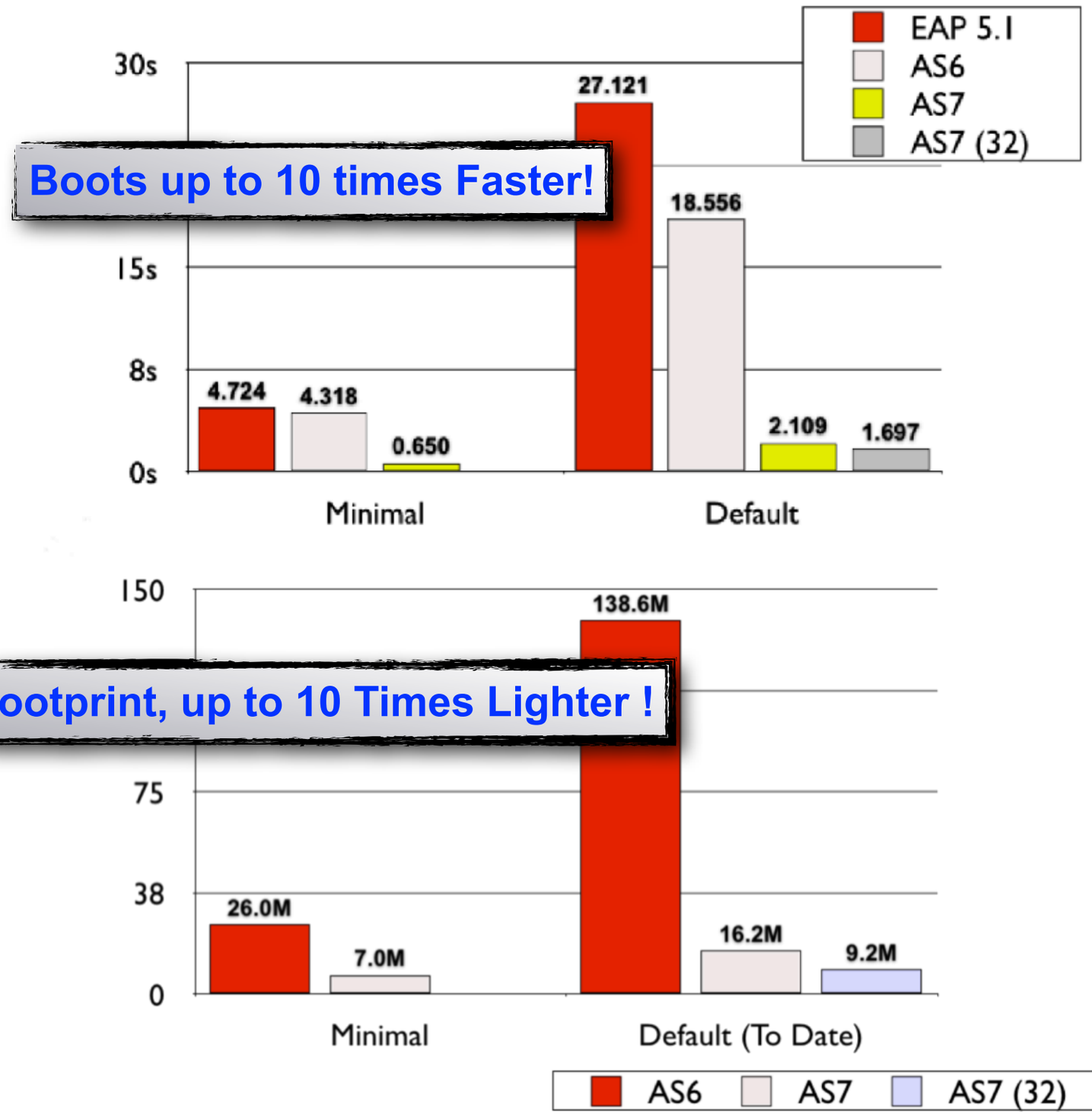
# JBoss AS7 – 7 reasons to love it!

- Fast
- Lightweight
- Modular
- New Domain Architecture
  - Standalone mode
  - Domain Mode
- Administration
  - Simple Unified User Configuration
  - Multiple Management Interface (CLI, API, HTTP, Console)
- Hot, Parallel Deployments
- First Class Components

JBoss by Red Hat

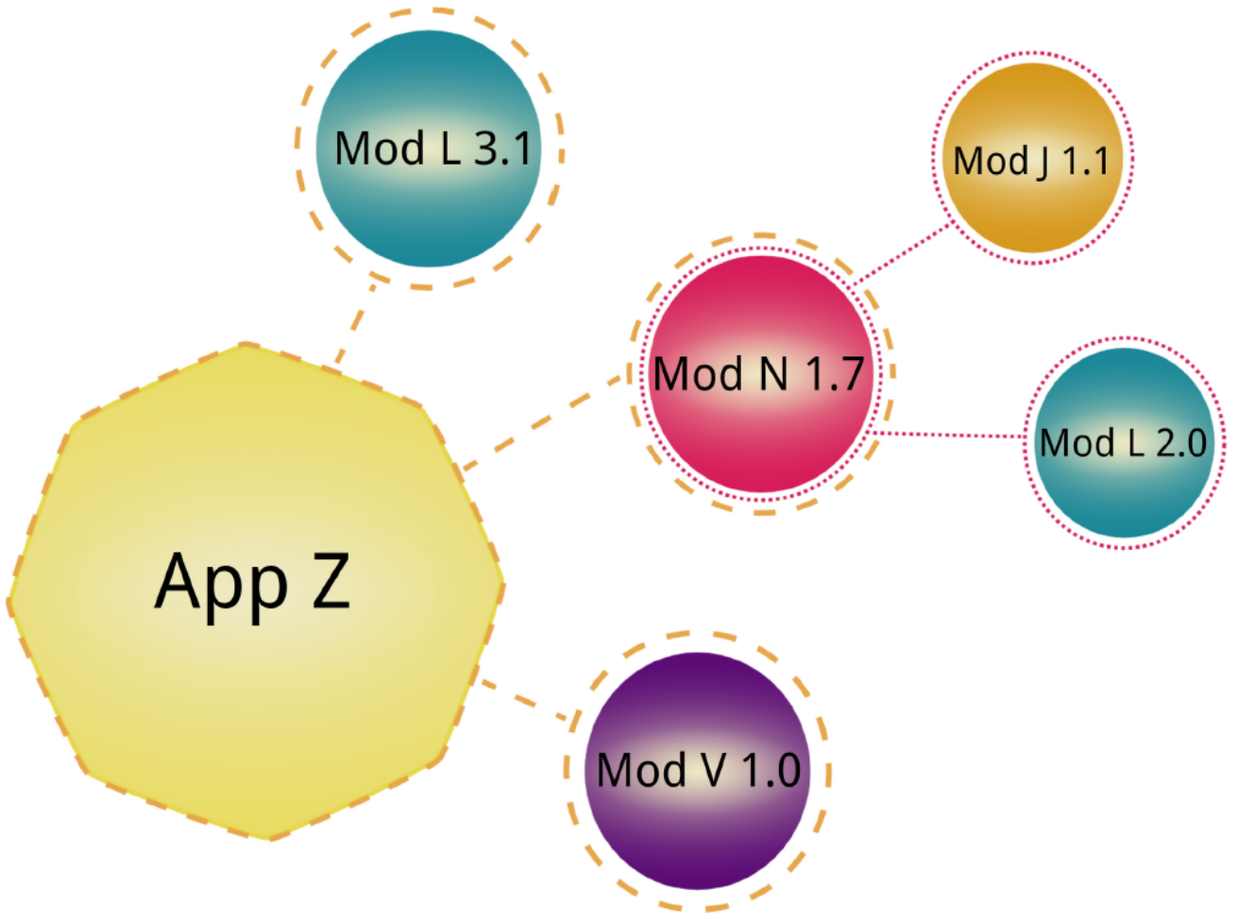
# JBoss AS7 - Fast and Light by Design

- New Lightweight Kernel
- Highly Concurrent
- Scalable and Dynamic
- Multiple Start Modes
  - Active, Passive, On-demand,...
  - Async Start & Stop
- more



# JBoss AS7 - Modularity - JBoss Modules

- Modular Class Loading (non-hierarchical)
- ClassLoader and Module isolation
- Lightweight - small memory footprint
- Highly efficient dependency resolution mechanism
- High level of Concurrency
- Addresses well-know issues
  - JAR duplication
  - PermGen space
  - GC and cache related leaks
  - CCE, CNFE,...
- On the Java 8 roadmap



# JBoss AS7 - Domain Architecture

- Domain Management Capabilities
- Standalone
  - Traditional JBoss single JVM server
  - Management facilities IN-VM
  - No lifecycle management (only shutdown)
- Domain
  - Multi-JVM, multi-server model
  - Management coordinated by Domain Controller Process
  - Multiple server instances (JVMs) per Host
  - Full lifecycle managed by Process Controller

JBoss by Red Hat

# JBoss AS7 - Management

- AS7 offers a set of features to make management easier
- **One** configuration file (standalone.xml / domain.xml)
- **Management API** that allows for **persistent changes** to the configuration
- Management API can manage all servers in the **domain**
- Management **console** to provide user friendly management in a web browser
- **Command line tool** for use in scripts

JBoss by Red Hat

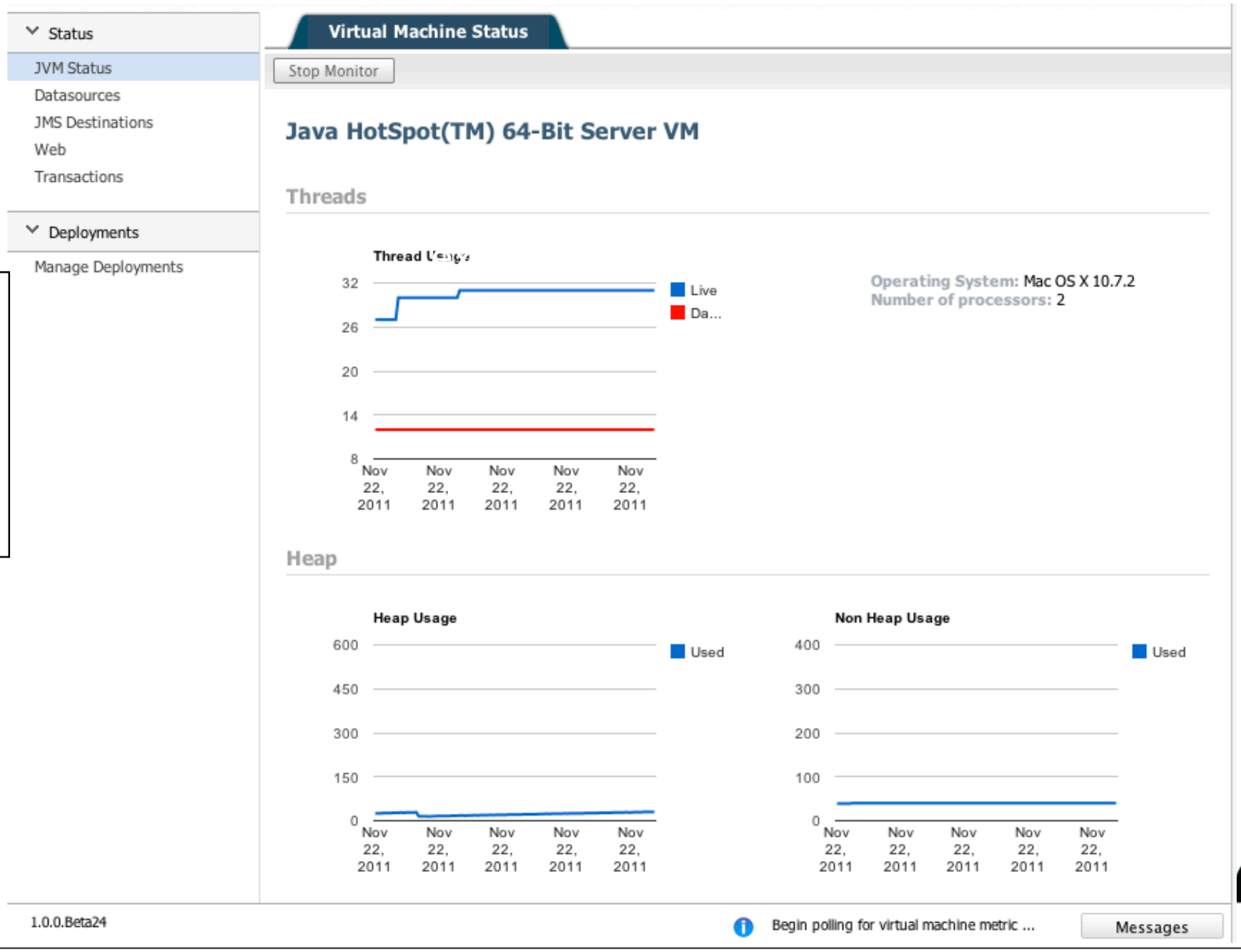
# JBoss AS7 - Management - API

- Can control a standalone server or an entire domain
- De-typed API uses a small set of Java classes
- All management interfaces are based on this API

**Web Console**

**Various transports:**

- Java Remoting
- JSON over HTTP



JBoss Application Server 7.1

(0) Messages

Profile | Runtime

- Status
- Server
  - Configuration
  - JVM
- Subsystem Metrics
  - Datasources
  - Transactions
  - Web
- Deployments
  - Manage Deployments
- Runtime Operations
  - OSGi

Data Sources | XA Data Sources

Refresh

## Data Source Metrics

Metrics for data sources.

### Datasource

| Name | JNDI |
|---|---|
| ExampleDS | java:jboss/datasources/ExampleDS |

1-1 of 1

Pool Usage | Prepared Statement Cache

### Pool Usage

| Metric | Actual | |
|---|---|---|
| Available: | 20 | |
| Active Count: | 0 | 0% |
| Max Used: | 0 | 0% |

1.0.0.RC1

JBoss by Red Hat

# JBoss AS7 - 1st Class Components

- Based on **JBoss AS7**
- JMS - **HornetQ**


- Replication, Caching- **Infinispan**


- CDI - **Weld**


- JPA - **Hibernate**


- REST - **RestEasy**


- Testability - **Arquillian**



# JBoss AS7 - Developer productivity

- **Arquillian**
  - Easy Testing with the App Server
- **Forge**
  - A Core Framework, standard based for Rapid application Development
- **OpenShift**
  - Red Hat's PaaS. Deploy your apps in the Cloud

OPENSHIFT™

PaaS by Red Hat® Cloud

JBoss Forge

JBoss by Red Hat

# AS7 Roadmap

- JBoss AS7.0
  - New Container Architecture
  - EE6 Web Profile focus
  - July 2011 - EE6 Web Profile Certified
- JBoss AS7.1
  - Goal Full EE6 implementation
  - Sub-systems management operations
  - 23 December - JBoss AS7.1.0.CR1
  - February 2012 - JBoss AS7.1.0.Final - EE6 Certified
  - Base for EAP6

# Domain Model

# Agenda

- Key Operations, Administration & Management (OA&M) Goals for AS7
- Domain Mode and Standalone Mode
- Key Management Model Concepts
- Management APIs and Interfaces
- Domain Configuration
- Q&A

# Key Goal – Centralized Configuration

- End user configuration centralized in one or two files
  - No longer scattered all over the distribution
- Config changes made via management tools are always persisted back to the config file
  - The config files provide a complete picture


# Key Goal - User-focused Configuration

```
<bean name="TransactionManager"
 class="com.arjuna.ats.jbossatx.jta.TransactionManagerService">
        <annotation>@org.jboss.aop.microcontainer.aspects.jmx.JMX(name="jboss:service=TransactionManager",
exposedInterface=com.arjuna.ats.jbossatx.jta.TransactionManagerServiceMBean.class,
registerDirectly=true)</annotation>

<annotation>@org.jboss.managed.api.annotation.ManagementObject(name="TransactionManager",componentType=@or
g.jboss.managed.api.annotation.ManagementComponent(type = "MCBean", subtype =
"JTA"),targetInterface=com.arjuna.ats.jbossatx.jta.TransactionManagerServiceMBean.class)
        </annotation>

        <property name="transactionTimeout">300</property>
        <property name="objectStoreDir">${jboss.server.data.dir}/tx-object-store</property>
```

```
<subsystem xmlns="urn:jboss:domain:transactions:1.0">
    <recovery-environment socket-binding="txn-recovery-environment"
                          status-socket-binding="txn-status-manager"/>
    <coordinator-environment default-timeout="300"/>
</subsystem>
```


# Key Goal – Robust Management API

- Complete: expose everything in the config schema
  - Plus metrics, runtime operations
- Stable: no incompatible changes across AS7.x series
- Secure remote access via:
  - Native Java interface
  - HTTP + JSON
  - CLI

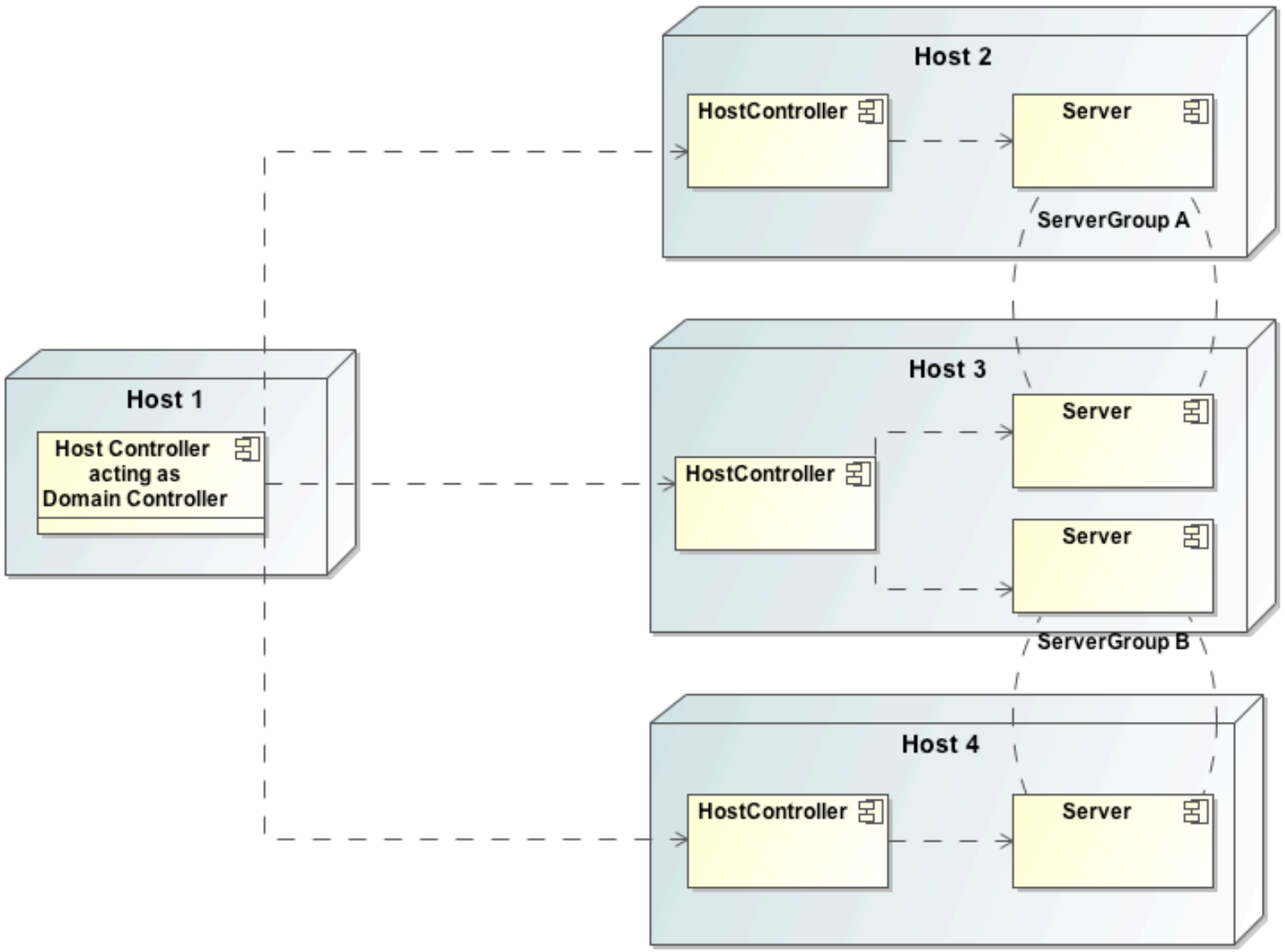
# Key Goal – Multi-Server Management

- Multi-server management as a core part of AS 7 / EAP 6 itself
- Manage multiple servers from a single control point
  - Start/quiesce/stop servers
  - Rolling deployment to a set of servers
  - Roll a config change out to a set of servers
  - Roll back changes

# Agenda

- Key Operations, Administration & Management (OA&M) Goals for AS7
- Domain Mode and Standalone Mode
- Key Management Model Concepts
- Management APIs and Interfaces
- Domain Configuration
- Q&A

# JBoss AS7 - Domain Topology




- **Domain Controller.** Management point for the entire domain.
- **Server Group.** A set of server instances that will be managed and configured as one.
- **Server.** Actual application server instance, runs in a separate JVM process from the Host Controller.
- **Host Controller.** Start/stop app server processes, interacts with the Domain Controller to help manage them.

# Choices for How to Manage Your AS Instances

- Two different operational modes
- Basically, do you want to take advantage of our multi-server management features?
  - Yes: run a *Managed Domain*
    - `[bin]$ ./domain.sh`
  - No: run a *Standalone Server*
    - `[bin]$ ./standalone.sh`
- Either way, you still get simplified configuration and a robust management API

# Standalone Server Mode

- Each server is independently managed, a la AS 3/4/5/6 and EAP 4/5
- User is responsible for coordinating changes across servers

- Single configuration file:
  - `standalone/configuration/standalone.xml`


# File Layout – Standalone Server

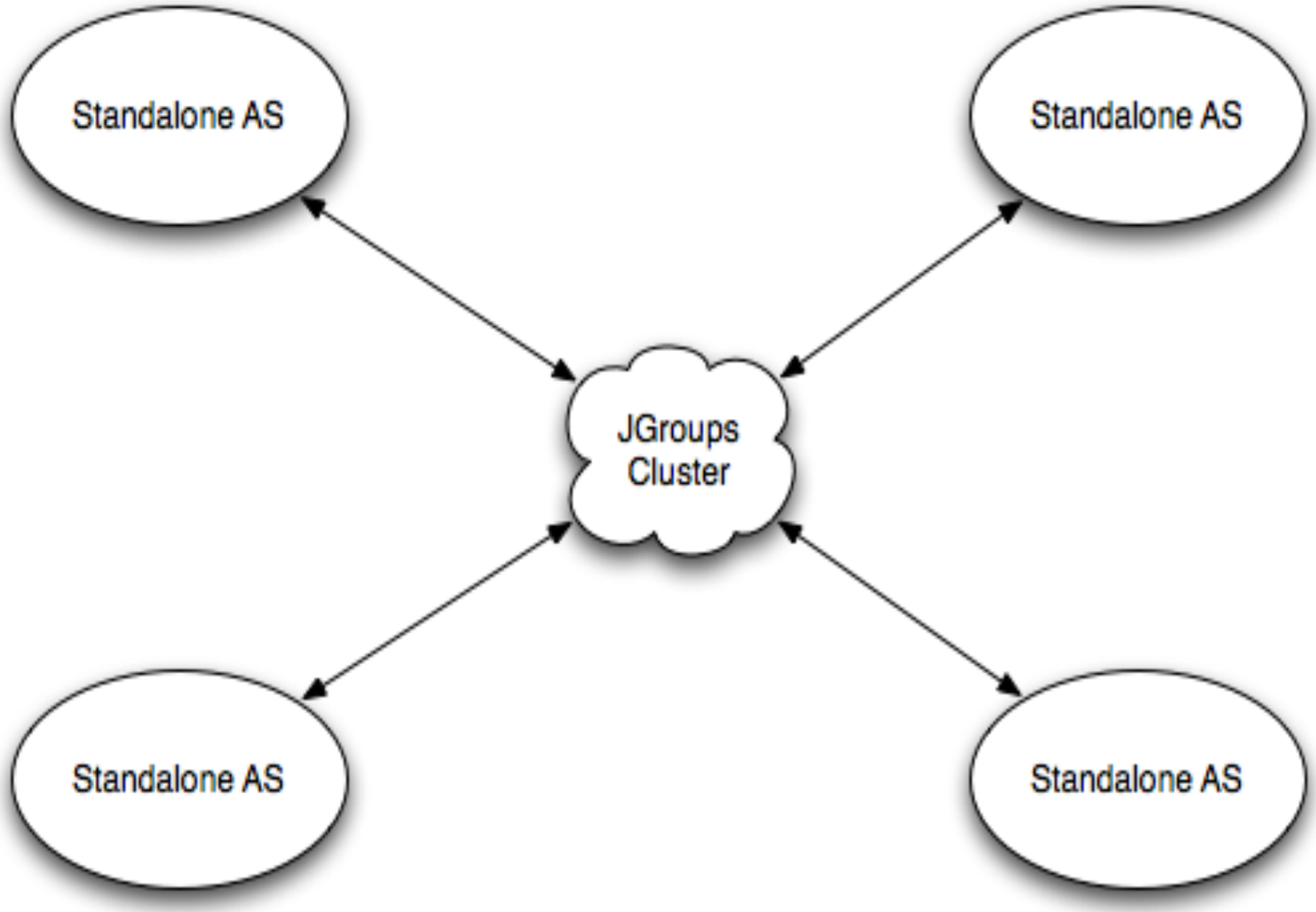
- jboss-7.1.0.CR1b
  - bin
    - standalone.conf
    - standalone.sh
  - modules
  - standalone
    - configuration
      - standalone.xml
    - deployments
    - logs
    - data

# Standalone Server Mode

- Good for many development use cases
  - where only a single server is needed
- A good option for enterprises with their own preferred tooling for multi-server management
  - *We expect many large customers to run large numbers of standalone servers rather than managed domains*


# Standalone Mode Allows HA Clusters

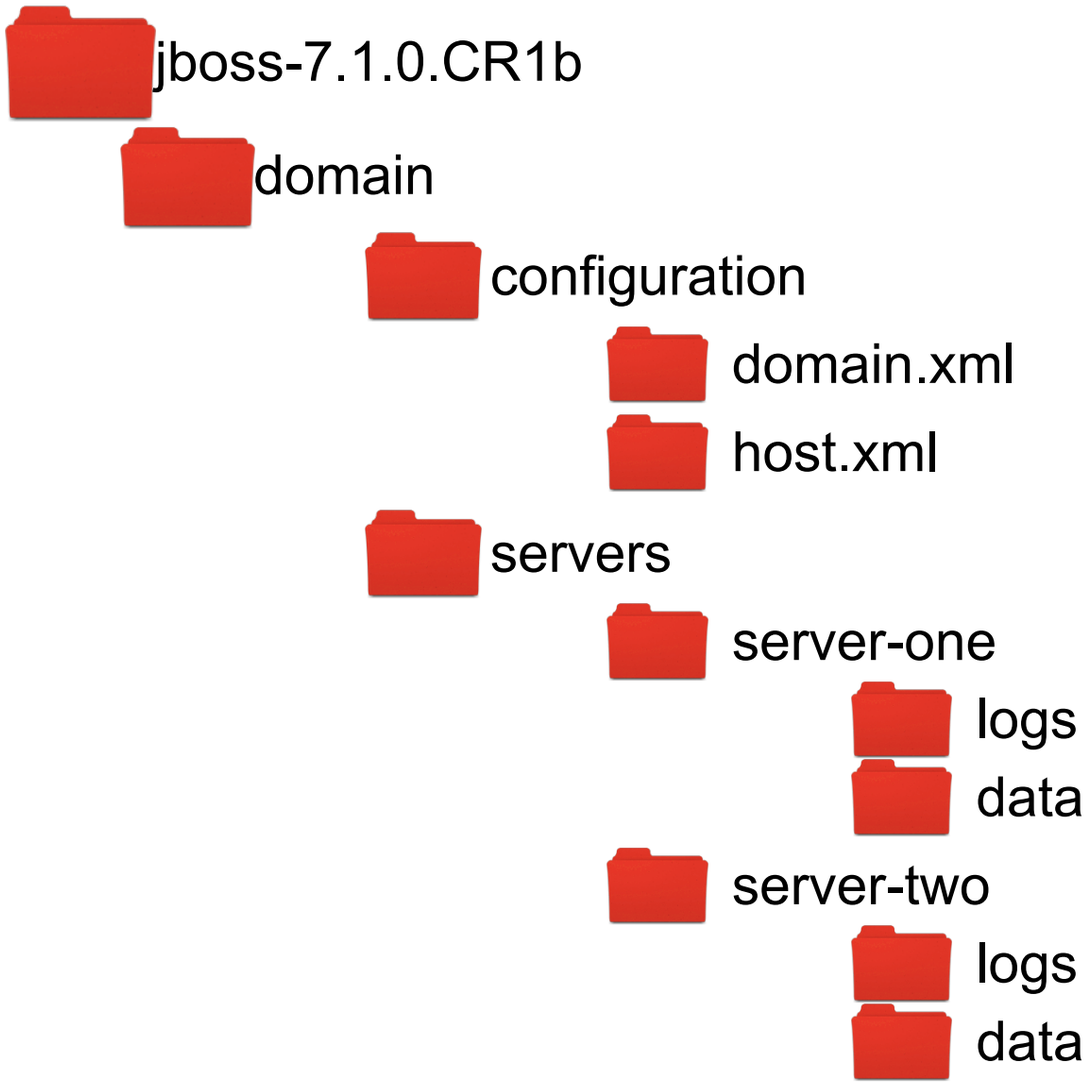
- Standalone mode is about *management*, not how managed services operate



# Managed Domain Mode

- Provides AS7's multi-server management features
- Domain: a set of servers with a common management policy
  - Policy is defined in `domain.xml` config file
  - Servers can be heterogeneous in a domain
- We ensure that all servers in the domain run in accordance with that policy


# File Layout – Managed Domain

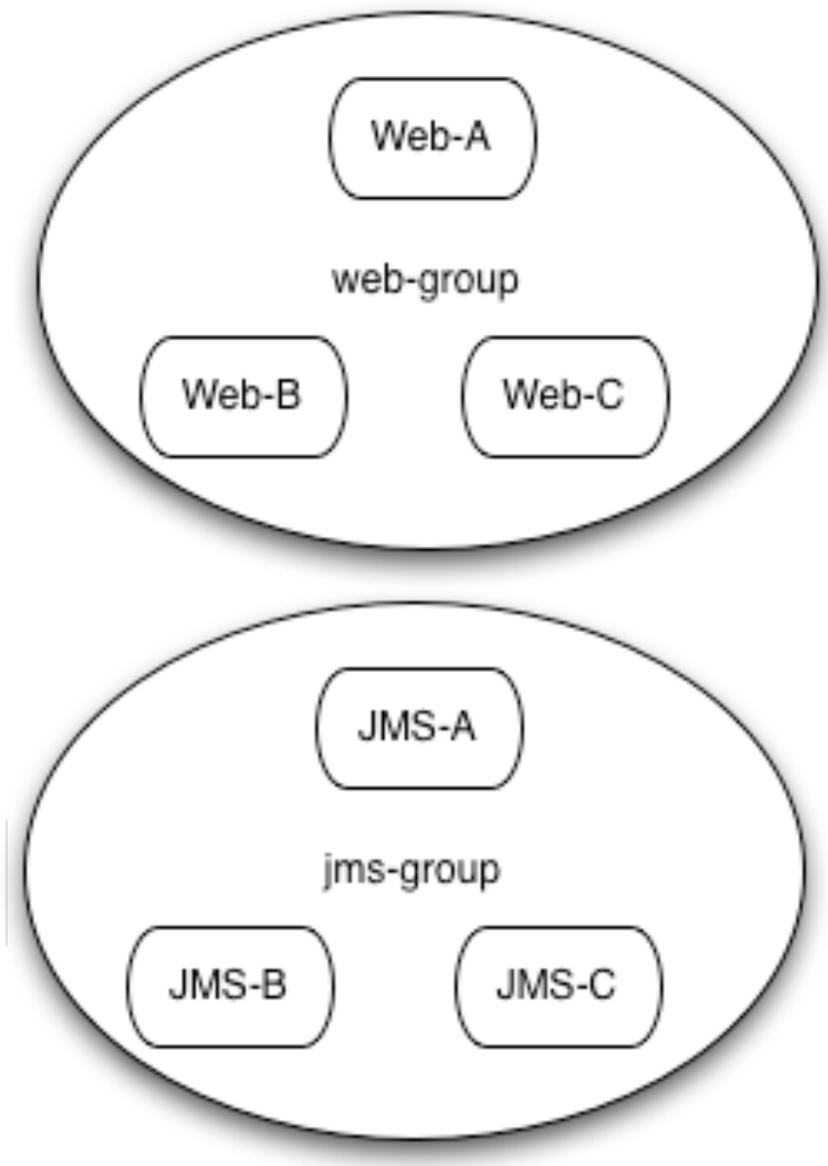


# Physical and Logical Domain Topologies

- Host Controllers can run multiple servers
- Different servers under a Host Controller can belong to different server groups
- Physical Topology:
  - Organization of servers by host
- Logical Topology:
  - Organization of servers by server group

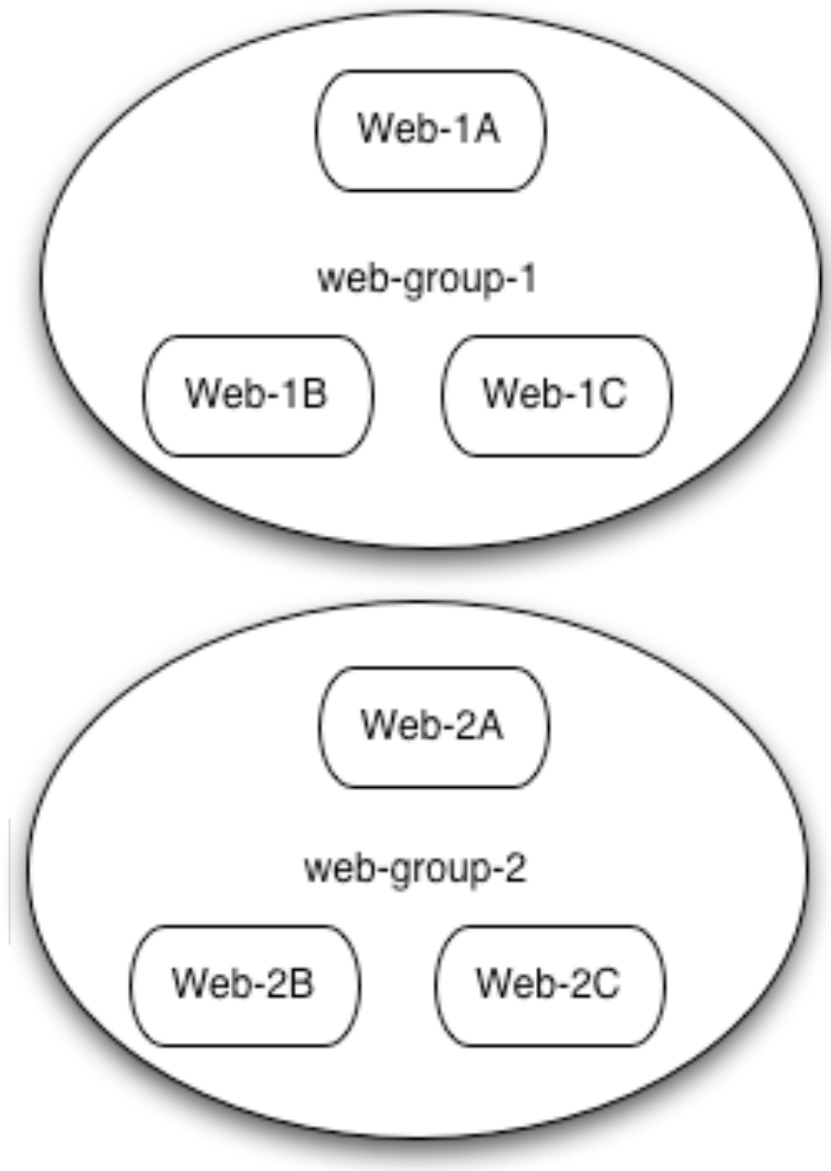

# Logical Topology Examples – Tiers

- Server Group per tier
  - e.g. web tier, JMS+MDB tier
- Similarly, could also be server group per application



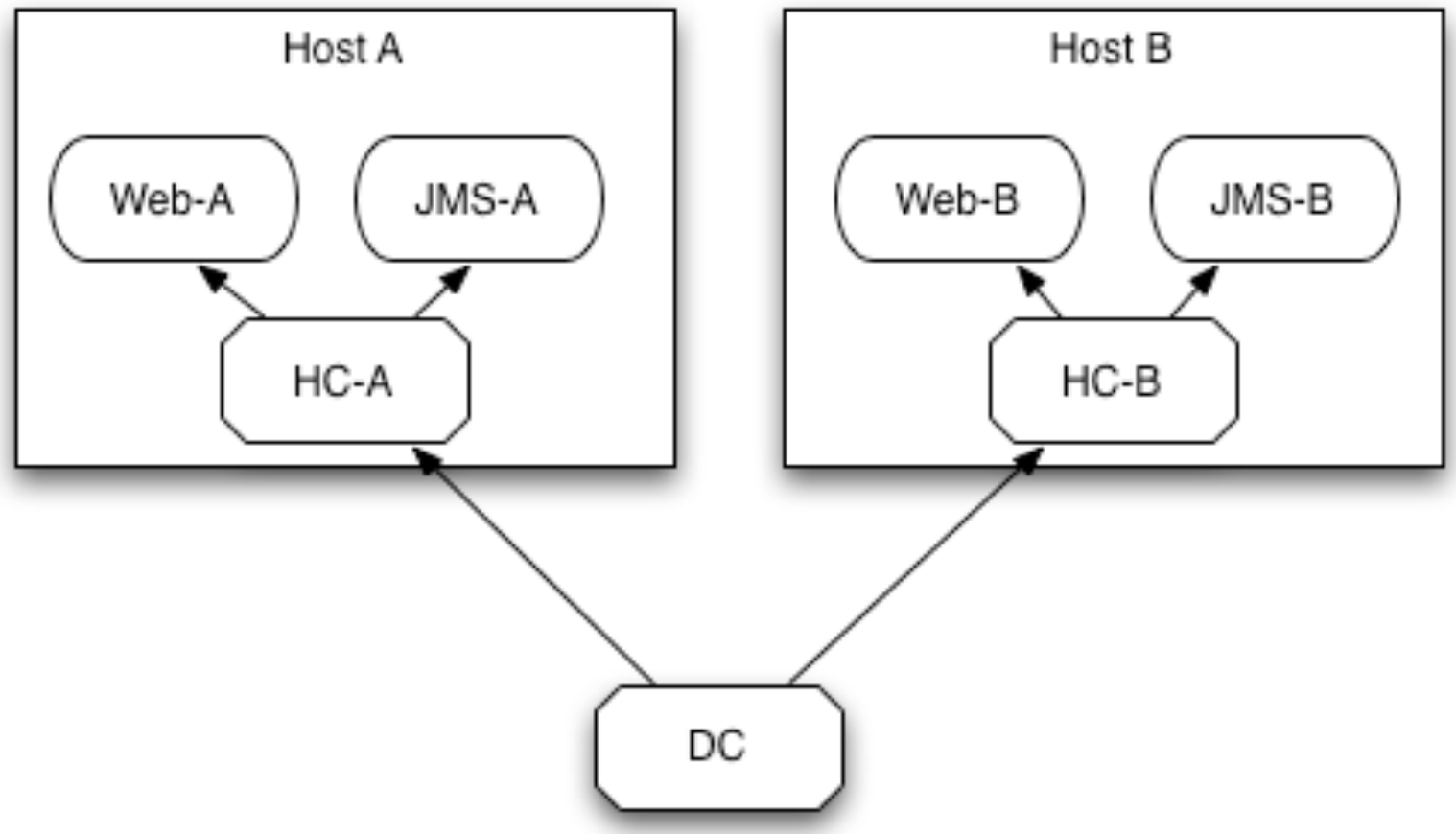
# Logical Topologies – Rollover Groups

- Split your servers into 2 identical server groups
- Roll out deployment upgrades one group at a time
- Avoid 100% outages



# Simple HA Physical Topology

- 2 hosts, 2 server groups, 4 servers



# Domain Management and HA

- The Domain Controller is not HA in AS7
- A running DC *is not needed*:
  - For servers to handle requests
  - To start a slave Host Controller (and servers)
  - To change host-specific configuration
  - To read server configurations, metrics, etc
- A running DC *is needed* to make changes to the domain.xml configuration


# Managed Domains and Deployment Scanning

- Deployment scanning: drop a deployment archive in a folder; AS notices and deploys it
- Not supported in a managed domain
  - Reason: it's unclear what Server Groups to deploy to
- Use the CLI or admin console to deploy content


# Agenda

- Key Operations, Administration & Management (OA&M) Goals for AS7
- Domain Mode and Standalone Mode
- Key Management Model Concepts
- Management APIs and Interfaces
- Domain Configuration
- Q&A

# Key Concept - Subsystems

- A particular set of capabilities that extend the application server core
  - Webserver, Transaction Manager, EJB3, CDI, HornetQ, OSGi, JCA, JGroups, Infinispan, etc, are all subsystems
- Each subsystem has its own section in the configuration file (standalone.xml or domain.xml)


# Key Concepts - Profiles

- The set of subsystems run by a standalone server or all servers in a server group
  - Add/remove subsystems in your profile to expand or narrow the capabilities of your servers
- A standalone server has a single profile
- The domain configuration (domain.xml) can include many profile configurations, for use by different server groups


# Key Concepts - Sockets

- Other configuration elements refer to sockets and interfaces by *logical names*, not specifics
  - `<connector socket-binding="http" .../>`
  - Not `<connector address="10.0.0.5" port="8080".../>`
- Allows centralized socket configs
- In a Managed Domain, each host can control how a logical interface name resolves to an actual IP address


# Sockets and -b

- ./standalone.sh -b 192.168.100.10
  - Sets system property `jboss.bind.address`
- The configs *we ship* use that property to define the interface used for most sockets

```
<interface name="public">
  <inet-address value="${jboss.bind.address:127.0.0.1}"/>
</interface>
```

- So, using -b *indirectly* controls what interface is used


# Sockets and -b

- Users are free to change our standard interface configuration xml in ways that mean -b will no longer have any effect!

```
<interface name="public">
  <nic name="eth0"/>
</interface>
```

- The configuration file controls the configuration; *-b is not an override*

# Agenda

- Key Operations, Administration & Management (OA&M) Goals for AS7
- Domain Mode and Standalone Mode
- Key Management Model Concepts
- Management APIs and Interfaces
- Domain Configuration
- Q&A

# Management Interfaces

- AS7 offers a number of interfaces for remote management
  - Native Java
  - CLI
  - HTTP + JSON
  - Web based admin console
  - For standalone servers, JMX
- All interact with the same underlying management model
- **All are secured by default !**


# Management Resources

- Everything manageable is exposed via a tree of addressable management resources
  - Address is an ordered list of key/value pairs
  - /profile=default/subsystem=web/connector=http
- Resources expose attributes and operations
- Quite similar to JMX Open MBeans
  - But, resources are organized in a tree
  - Atomic multi-step operations supported
  - Operations across servers supported


# Command Line Interface

- Scriptable command line management tool
- Uses the management API internally
- Allows access to high level user friendly commands
- Also allows direct access to the domain model, giving access to the full functionality of the management API

CLI Example:

```
[standalone@localhost:9999 /] batch
[standalone@localhost:9999 / #] jms-queue add --queue-address=tradeQueue --entries=queue/tradeQueue
#1 jms-queue add --queue-address=tradeQueue --entries=queue/tradeQueue
[standalone@localhost:9999 / #] deploy ~/jboss-as-login.war
#2 deploy ~/jboss-as-login.war
[standalone@localhost:9999 / #] undeploy jboss-as-login.war
#3 undeploy jboss-as-login.war
[standalone@localhost:9999 / #] jms-queue remove --queue-address=tradeQueue
#4 jms-queue remove --queue-address=tradeQueue
[standalone@localhost:9999 / #] run-batch
The batch executed successfully.
[standalone@localhost:9999 /]
```

# Raw HTTP Management Interface

- Accesses the same management resources as the native java interface or the CLI
- Resource addresses readily convert to URLs

`http://localhost:9990/management/subsystem/web`

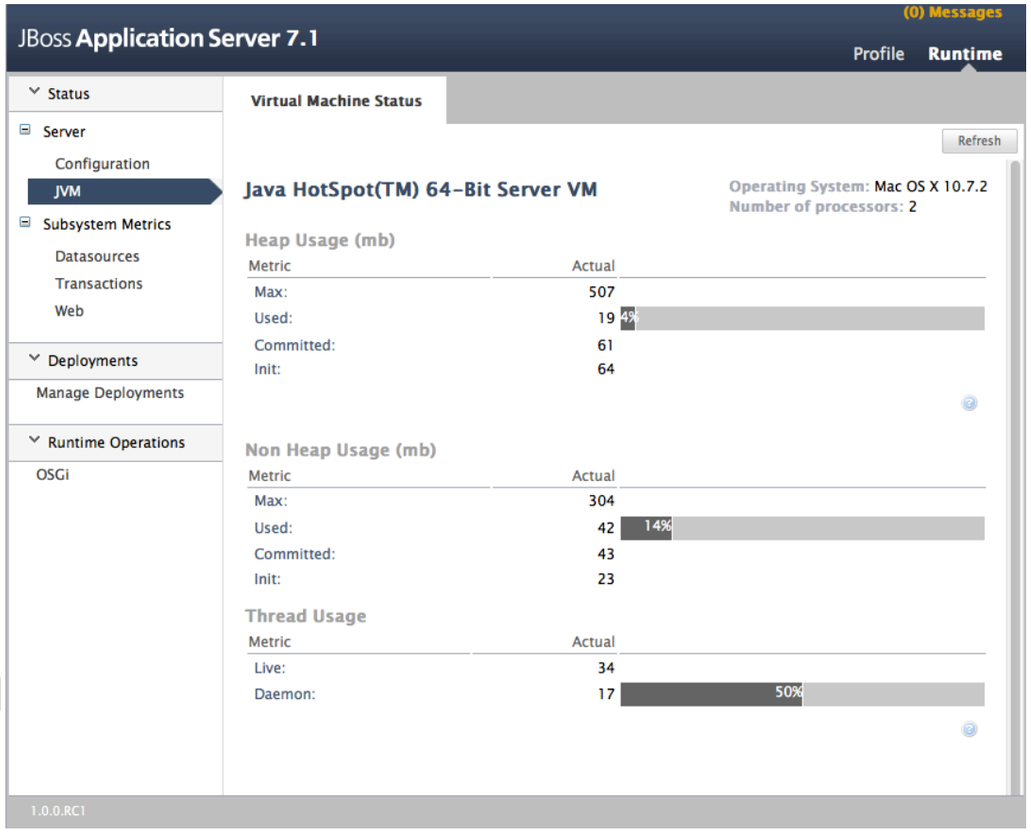
- HTTP GET for reads; POST for writes
- Response and POST content are JSON
  - encodings of the same simple data types used by the CLI and native interface


# Web Based Admin Console

- Available on same host:port as the raw http management interface:

http://localhost:9990/console

- Available for both a standalone server or for a  managed domain
  - consistent user interface for both



# JMX

- Standalone servers can also be administered via JMX
- Mbeans are under JMX domain jboss.as
- Addresses of management resources map naturally to object names:
  - jboss.as:subsystem=web,connector=http
- Resource attributes and operations map to Open MBean attributes and operations


# Agenda

- Key Operations, Administration & Management (OA&M) Goals for AS7
- Domain Mode and Standalone Mode
- Key Management Model Concepts
- Management APIs and Interfaces
- Domain Configuration
- Q&A

# Agenda

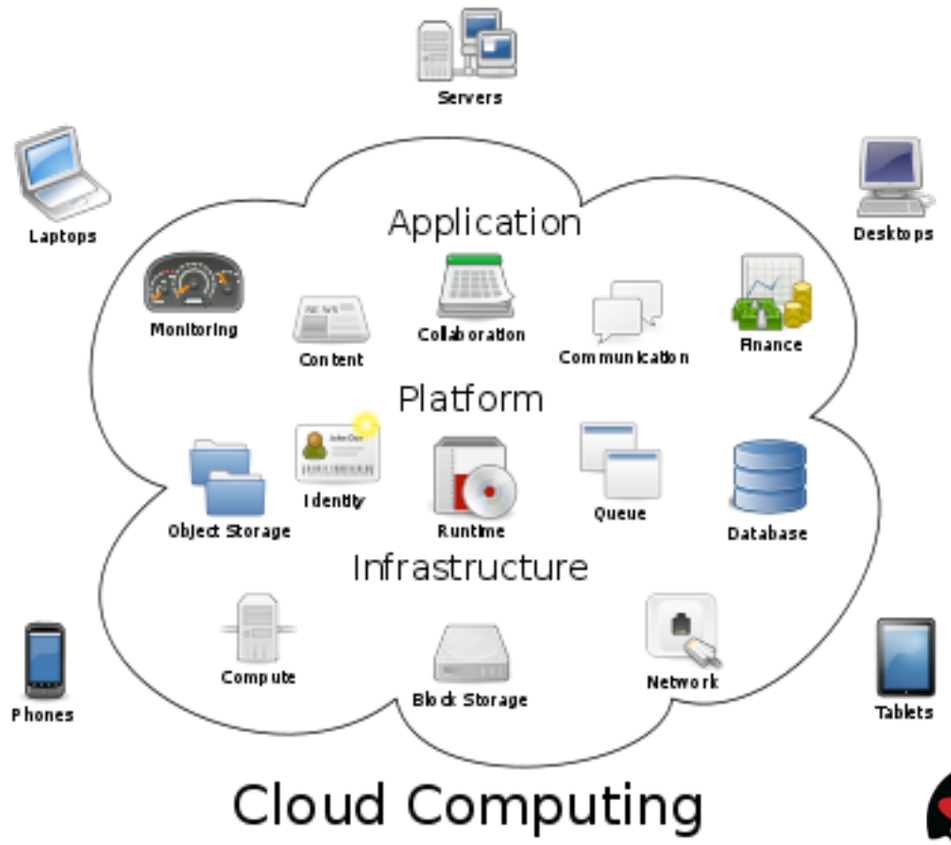
- Key Operations, Administration & Management (OA&M) Goals for AS7
- Domain Mode and Standalone Mode
- Key Management Model Concepts
- Management APIs and Interfaces
- Domain Configuration
- Q&A


# Configure an AS Instance in Domain Mode

- An individual server's config comes from 2 sources
  - `domain/configuration/domain.xml` on host with DC
    - Elements that are consistent across the domain
  - `domain/configuration/host.xml` on each host
    - Elements specific to the host the server runs on
- Host Controller process combines domain.xml data + host.xml data to derive server config(s)

# JBoss AS7 - Where to start ?

- **Home Page**
  - http://www.jboss.org/as7
- **Download**
  - http://www.jboss.org/jbossas/downloads/
- **Documentation**
  - https://docs.jboss.org/author/display/AS71/Documentation
- **Learn from our popular Webinars !**
  - http://vimeo.com/jbossdeveloper


JBoss by Red Hat