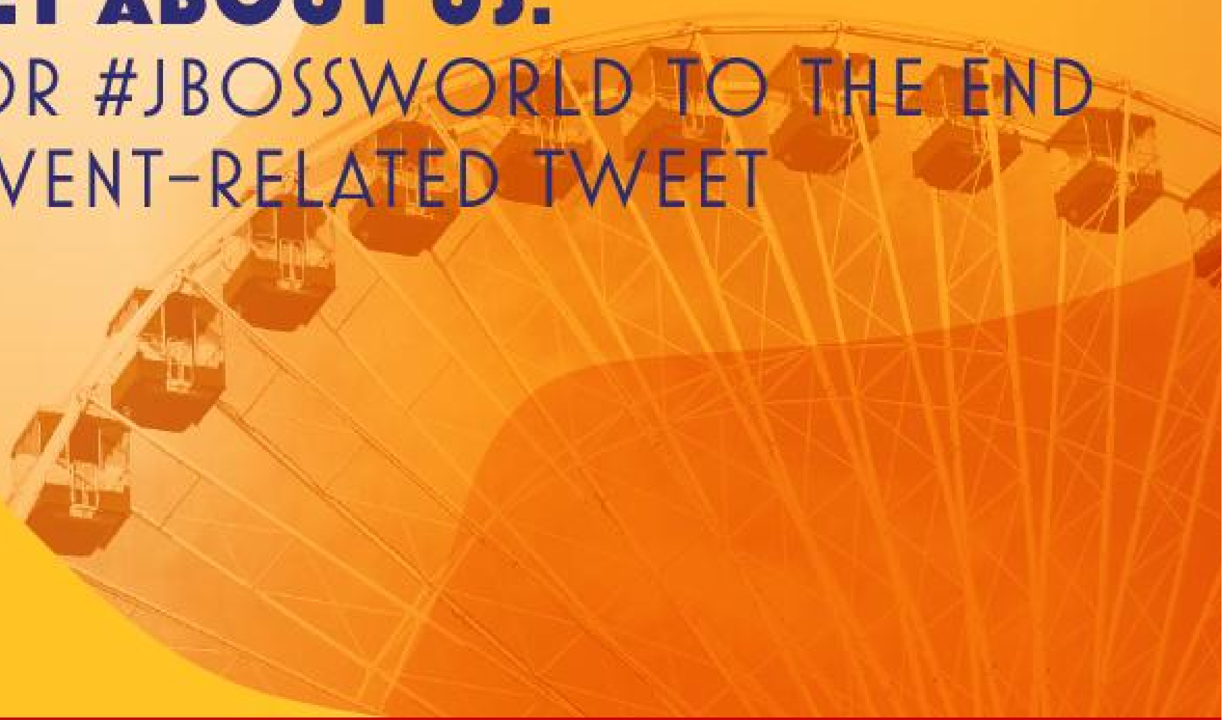
JBoss
WORLD
CHICAGO 2009

## FOLLOW US:

TWITTER.COM/REDHATSUMMIT

## TWEET ABOUT US:

ADD #SUMMIT AND/OR #JBOSSWORLD TO THE END OF YOUR EVENT-RELATED TWEET



presented by

JBoss WORLD
CHICAGO 2009

Patrick Black
JEE Technology Engineer,
Travelers Insurance
9/3/2009

presented by

# Travelers Insurance

- Second-largest writer of commercial U.S. property casualty insurance
- Second-largest writer of U.S. personal insurance through independent agents
- No. 99 on the Fortune 500 list of largest U.S. Companies
- Represented by approximately 12,000 independent agencies and brokerages countrywide
- Representatives in every U.S. state, Canada, Ireland and the U.K.
- Approximately 33,000 employees

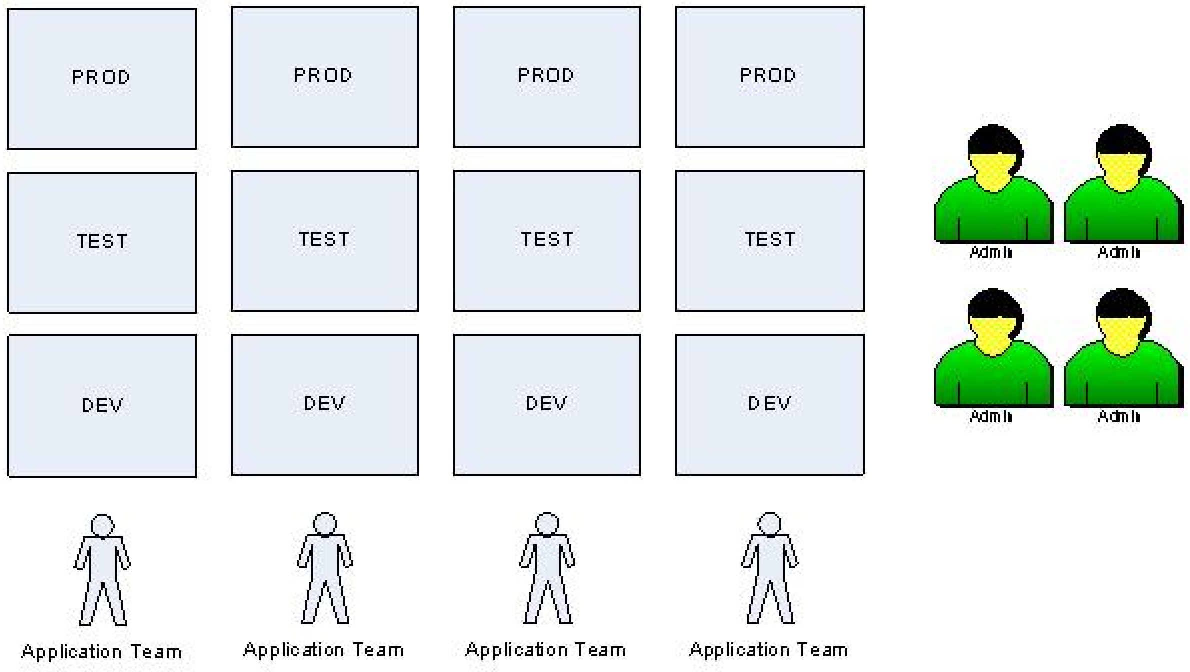
# Travelers JEE Infrastructure Overview

- Hundreds of applications running on thousands of Java Virtual Machines.
- Support staff of approximately 12.
- Application owners have near complete control of their configuration.
- Applications share hardware to save cost, but are isolated to limit adverse affects.
- Fast Provisioning of resources for applications.
- Touch less – in most cases we do not have server access.

# The Old Model

- Application area has total control.
- Application area builds its own infrastructure. Dev, Test and Production.
- Application area maintains a support staff for the application and the JBoss configuration - enterprise staff provides support.
- Server hardware is under-utilized.
- Pockets of poorly configured Applications – security and audit exposures.
- Persistent errors are rarely corrected.
- Problem determination is difficult as each server is unique.

# The Old Model



| | | | |
|---|---|---|---|
| PROD | PROD | PROD | PROD |
| TEST | TEST | TEST | TEST |
| DEV | DEV | DEV | DEV |
| Application Team | Application Team | Application Team | Application Team |

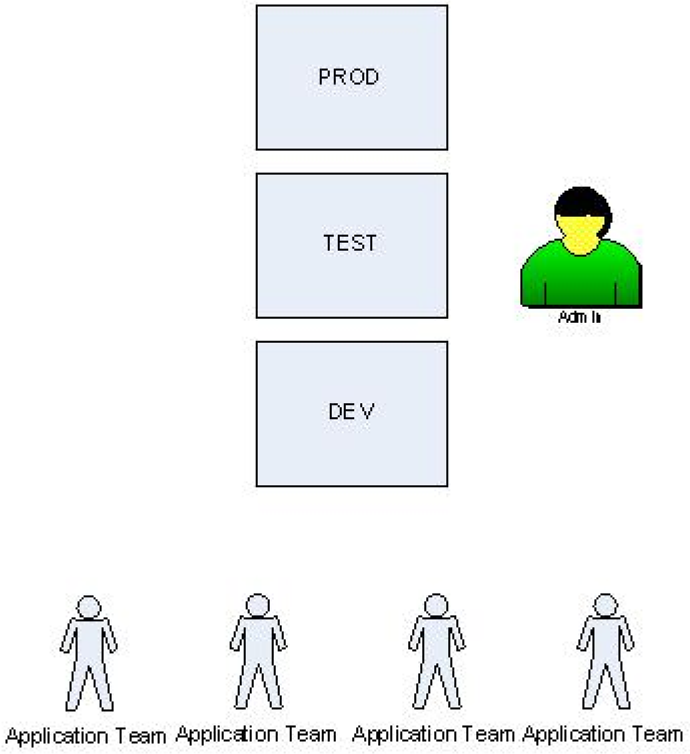
# New Model

- Each JBoss installation is identical to every other installation
- JBoss JVM configurations are standardized.
- All applications profit from the discovery of errors by one application.
- Application areas do not need a JBoss person.
- Problem determination is simplified with centralized monitoring and reporting tools.
- Standardized deployment methods allow quick access to the configurations and logs.
- Consistent security and audit compliance.

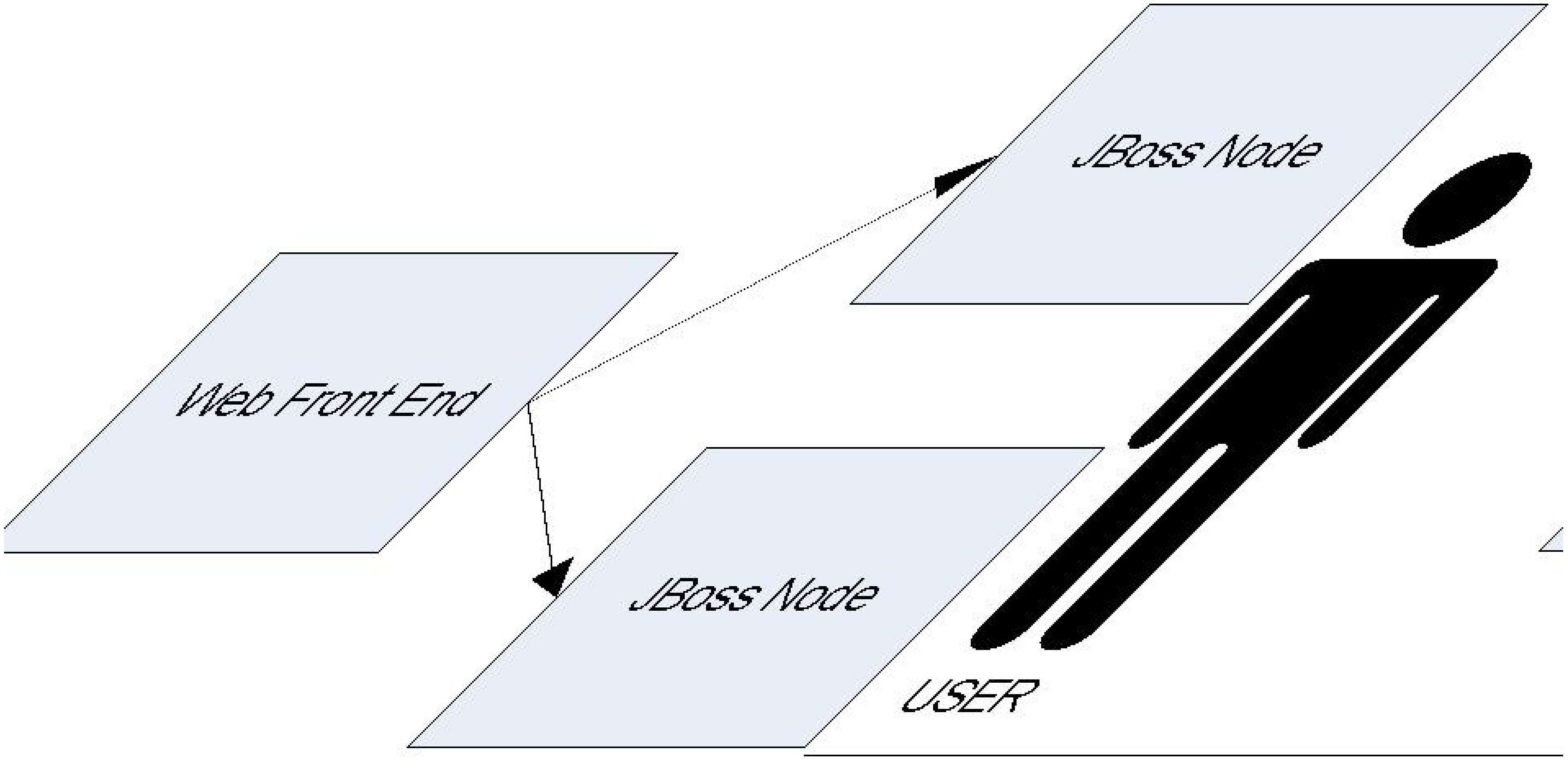
# Shared Server Model

- Applications assigned hardware to share with other applications.
- Fully utilize hardware.
- Minimized hardware, OS, and software costs.
- Lower administrative overhead.
- Decreased provisioning time and costs.
- Application area changes are controlled – increasing supportability.

# The New Model

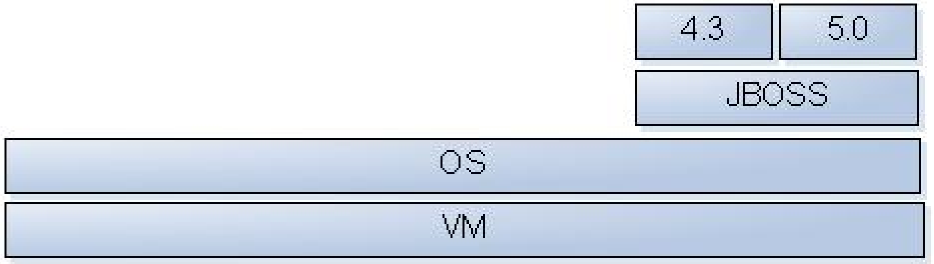


# A Standard Environment



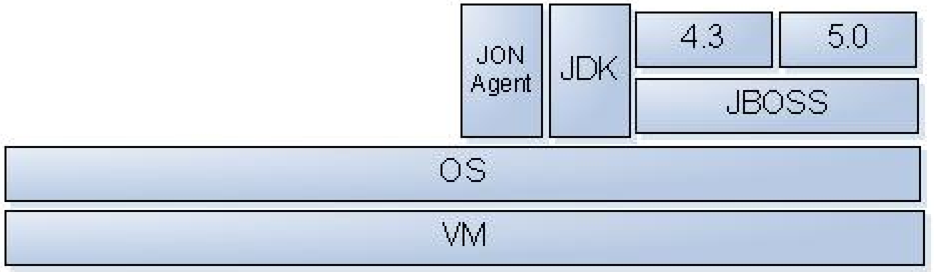
# Building a Shared Server – JBoss Installation

- VM
  - Better utilization of Hardware
  - Automated DR capabilities
  - Template Build Capabilities
- OS Support
  - Unix and Windows
- JBoss Installation
  - Standardized JBoss installation.
  - Multiple JBoss versions on the same server
  - N-1 version support



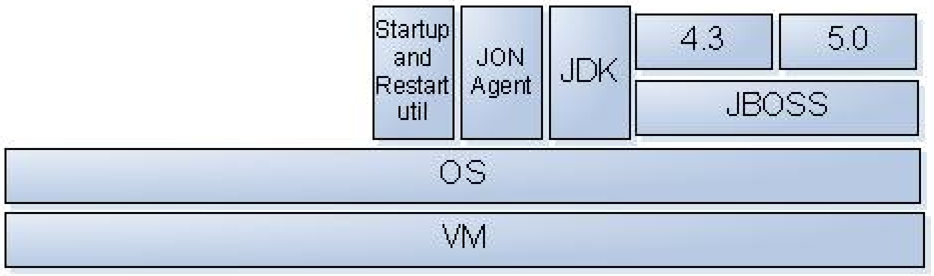
# Building a Shared Server – JDK and JON Agent

- JDKs
  - Standardized JDKs supported (1.5, 1.6, 32/64 bit, etc)
  - Custom JDKs allowed after exception review
  - N-1 version support
- Jon Agent
  - Pre-installed on server
  - Pre-configured to the JonServer for the environment



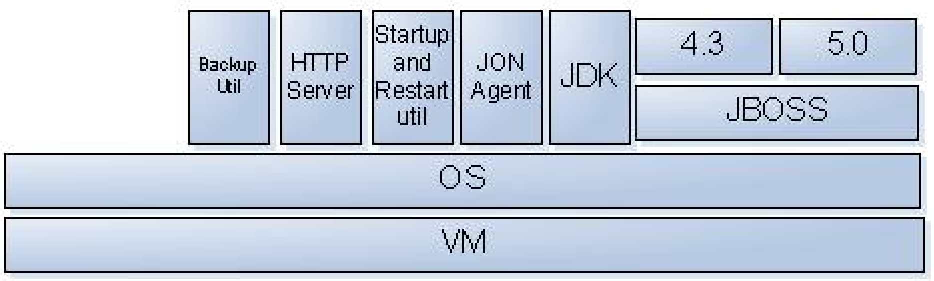
# Building a Shared Server – Startup and restart

- Startup Scripts
  - Scripts to Start JVMs when the server reboots
- Selective shutdown / startup scripts
  - Stop / Start JVMs based on JDK used.
  - Stop / Start JVMs based on JBoss version.
- Scripts to restart JVMs at specified intervals



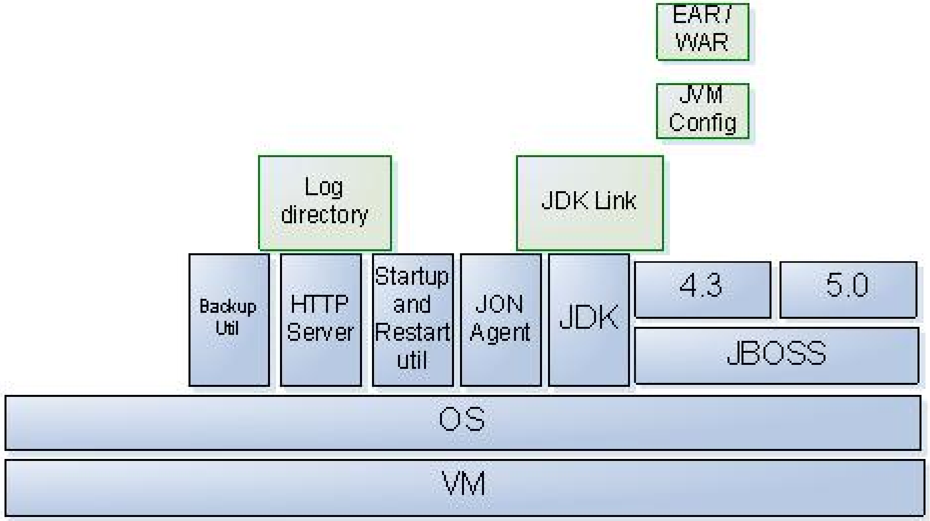
# Building a Shared Server – Shared Utilities

- HTTP Server to allow Application access to JVM and application logs
- Backup utility to run nightly backups of the JVM and system configurations.



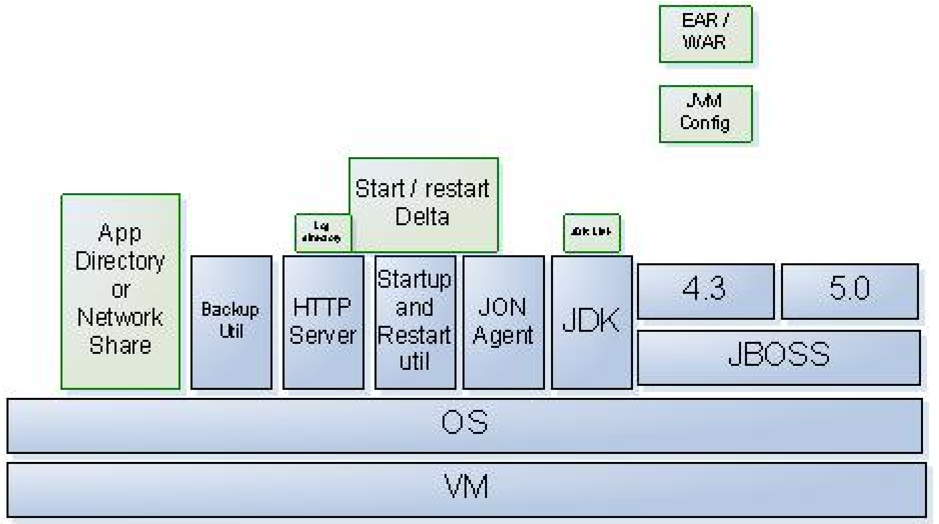
# Application Installation – JVM configuration

- Application requests specific resources. These resources are translated into a JVM configuration.
- The Application provided EAR/WAR is deployed to the JVM. (separate deploy directory)
- The JVM runs on one of the provided JDKs, switching the JDK is an application configurations.
- Log location is configured.



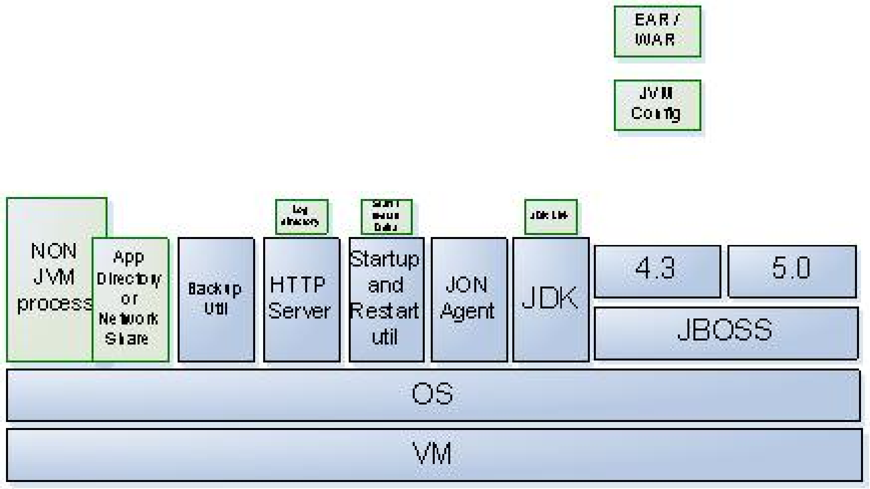
# Application Installation – Startup/Restart and App Directory

- A “Delta” File for the startup and restart utilities are added to make sure the JVM starts on reboot and restarts when requested.
- Each application is given an Application directory to place non EAR/WAR content, temp directories and private logs.
- Applications requiring large directories will utilize a network share.



# Application Installation – Other

- Applications always want more!
- A location is carved out for non-JBoss scripts, tools, or other application processes.

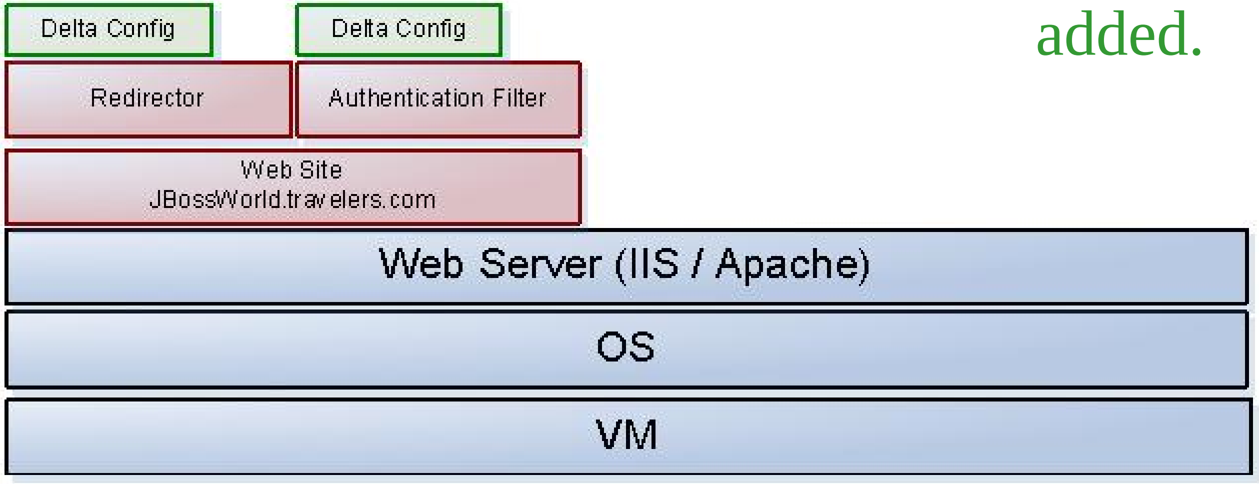


# Front End Server – Virtual Machine Template

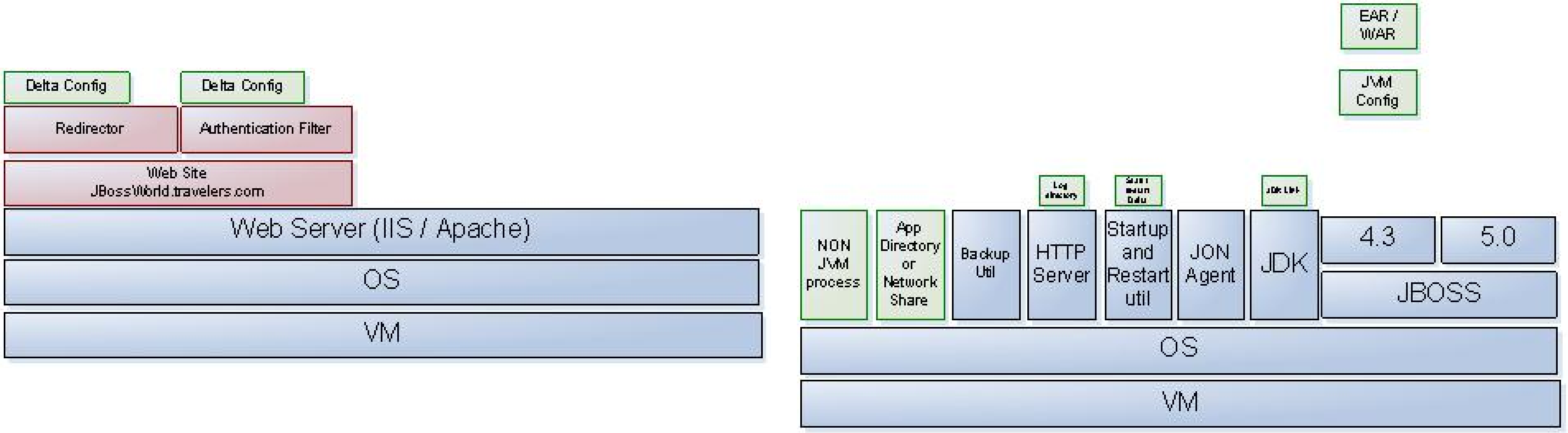
- Each Front End Server is built off a VM template.

- The Website is created.
- Jakarta Redirector is added.
- Authentication Filter is added.

- The Redirector specific entries are added.
- The Authentication filter entries are added.
- Static Content is added.



# Infrastructure Layout



# Application Provided Configurations

- Application Security Access Data
- Application Directory Contents
- JVM restart Timings
- JDK version
- JVM Configuration (Resources, Classloader, Role mappings)
- Context Root
- Binary (EAR or WAR)

# Infrastructure Provided Configurations

- Servers
- JVM names and ports.

## Derived Locations

- Application Directory
- Logs Directory
- Other Directory

# JVM Configuration Pattern

- Basic JVM configuration template pre-located with JBoss instance installation.
- Centralized configuration store generates configuration files to overlay the JVM template.
- Each JVM configuration becomes a list of changes from the standard JVM template.
- Variations from the normal configuration is easy to spot.

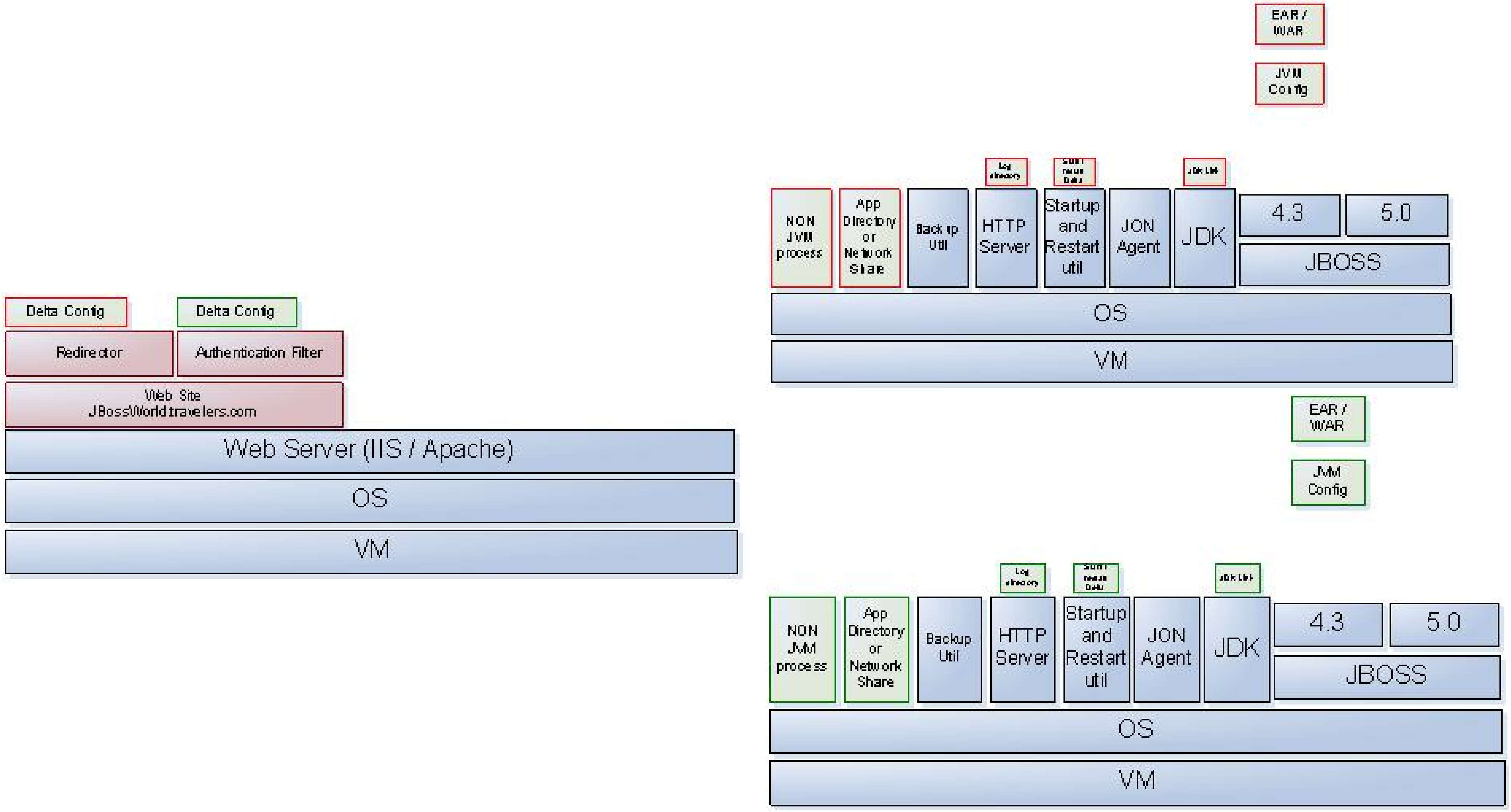
# Centralizing the Configuration Data

- Small amount of configuration data per application simplifies centralized storage of the configuration data
- Centralized configuration detail allows for authoritative reports of configuration trends. (Capacity planning, trend analysis, threat assessment, customer reporting)
- Repeatable application deployments
- “Best Practices” - Configuration checking scripts
- Implement and Inform capabilities

# Moving an Application

- Despite best plans, we eventually have to move a JVM
- Deployment scripts remove the JVM from the old server, and add the JVM to a new server.
- Web Server or load balancer updated to match the new JVM location.

# Moving an Application



Delta Config

Delta Config

Redirector

Authentication Filter

Web Site
JBossWorld.travelers.com

Web Server (IIS / Apache)

OS

VM

EAR / WAR

JVM Config

Log directory

[illegible]

JDK Link

NON JVM process

App Directory or Network Share

Backup Util

HTTP Server

Startup and Restart util

JON Agent

JDK

4.3

5.0

JBOSS

OS

VM

EAR / WAR

JVM Config

Log directory

[illegible]

JDK Link

NON JVM process

App Directory or Network Share

Backup Util

HTTP Server

Startup and Restart util

JON Agent

JDK

4.3

5.0

JBOSS

OS

VM

# JBoss Operations Network (JON)

- Agent is pre-configured on all servers.
- Provides notification when servers are built.
- Extendable
  - Pre-setting values specific to our infrastructure.
  - Gathering Travelers' Specific Data for centralized analysis.

# JON Server – Customer Usage

- Grouping of JON monitored assets allow application areas to view just their pieces
- Developers can configure availability alerts and application threshold alerts.
- Application areas have direct visibility into their applications without needing to work with administrator staff.
- JVM start / stop.

# Jon Server – Administrator usage

- Environment health check
- Capacity Planning
- Historical usage patterns
- After the fact problem determination
- Alerting on server errors (CPU, disk, availability)
- Identify trends so we can pro-actively address systemic issues.

# Customer Reports

- Automated Monthly reports are generated from the centralized configuration store detailing every relevant piece of data for an application (Bank Statement report)
  - VM server currently running on.
  - Deployments / JVM restarts
  - Server utilization
  - Error reports, resolutions
  - “Best Practices” compliant report
  - Identified configuration anomalies

# JBoss upgrades

- Scripts for each Jboss version to take xmls provided by the centralized configuration location to generate JBoss configurations.
- We create scripts for each JBoss version to translate the configuration XMLs into JBoss configuration files.
- During upgrade, the configuration scripts are updated to the new version and run against the configuration XML for each JVM.

# QUESTIONS?

TELL US WHAT YOU THINK:
REDHAT.COM/JBOSSWORLD-SURVEY