IBM Technical Summit

Stay ahead.

Leveraging Rational and Cloud Technologies To Integrate Development, Test and Production Environments

Tom Kamimura (kamimura@us.ibm.com)
Distinguished Engineer and Rational CTO for Asia Pacifc

Please note the following

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Agenda

- DevOps and Continuous Delivery
- IBM cloud solutions with Virtual System Patterns
- DevOps Tools and UrbanCode

Business innovation is increasingly being delivered via software

Rapid pace of change and the digitization of business drives the need for agility

Software drives almost **80%** of the automotive industry's functional innovation

An F-35 contains **5.6 million lines of code**

Technology is the **#1** force impacting business today

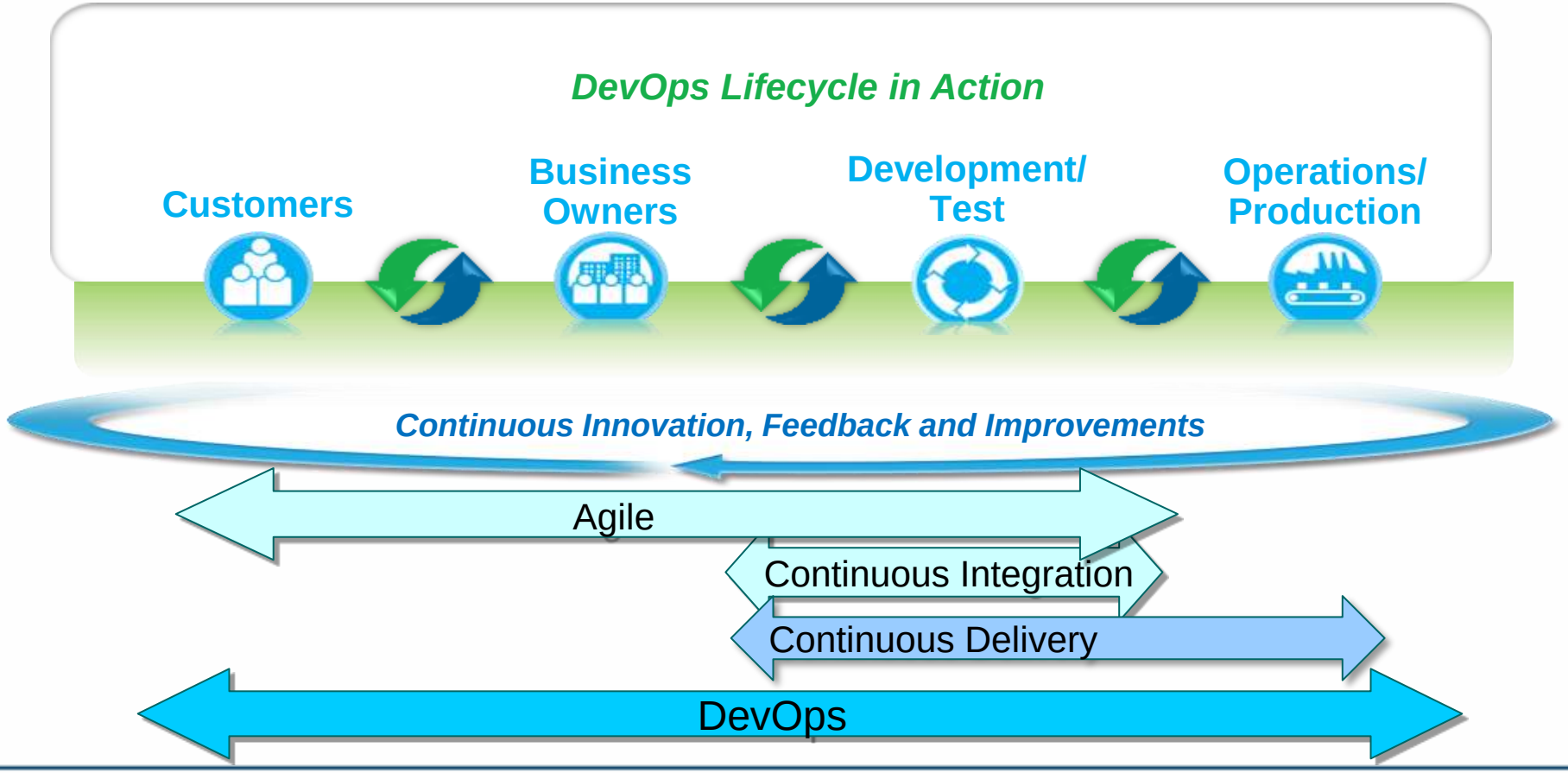
Robotic surgical systems have the **computational capacity of 7 laptops**

By the year 2020 in the U.S., **90% of financial transactions** will be **cashless**

Customer experience increasingly drives **business needs**

DevOps: A blueprint for continuous delivery of software innovation

dev·ops *noun* \'dev-äps\
Enterprise capability for continuous software delivery that enables clients to seize market opportunities and reduce time to customer feedback.



Challenge in Continues Delivery

- Differences in environments can lead to problems
  - Hand-off is time consuming.
  - Development components don't run in the test environment
  - Quality assured solutions don't run in the production environment
- Keeping all three environments identical is very difficult
- Agile development rapid releases compound the difficulties
- Result: Errors, Delays, Extra Cost

Cloud technologies can help address these challenges

Continuous Delivery for Cloud

- Virtualization
  - Flexible and scalable
  - Delivery instructions defined as code.
- Automation
  - Fast and reliable delivery without human errors
- Standardization
  - Repeatable and reliable process
  - Reuse of artifacts for delivery

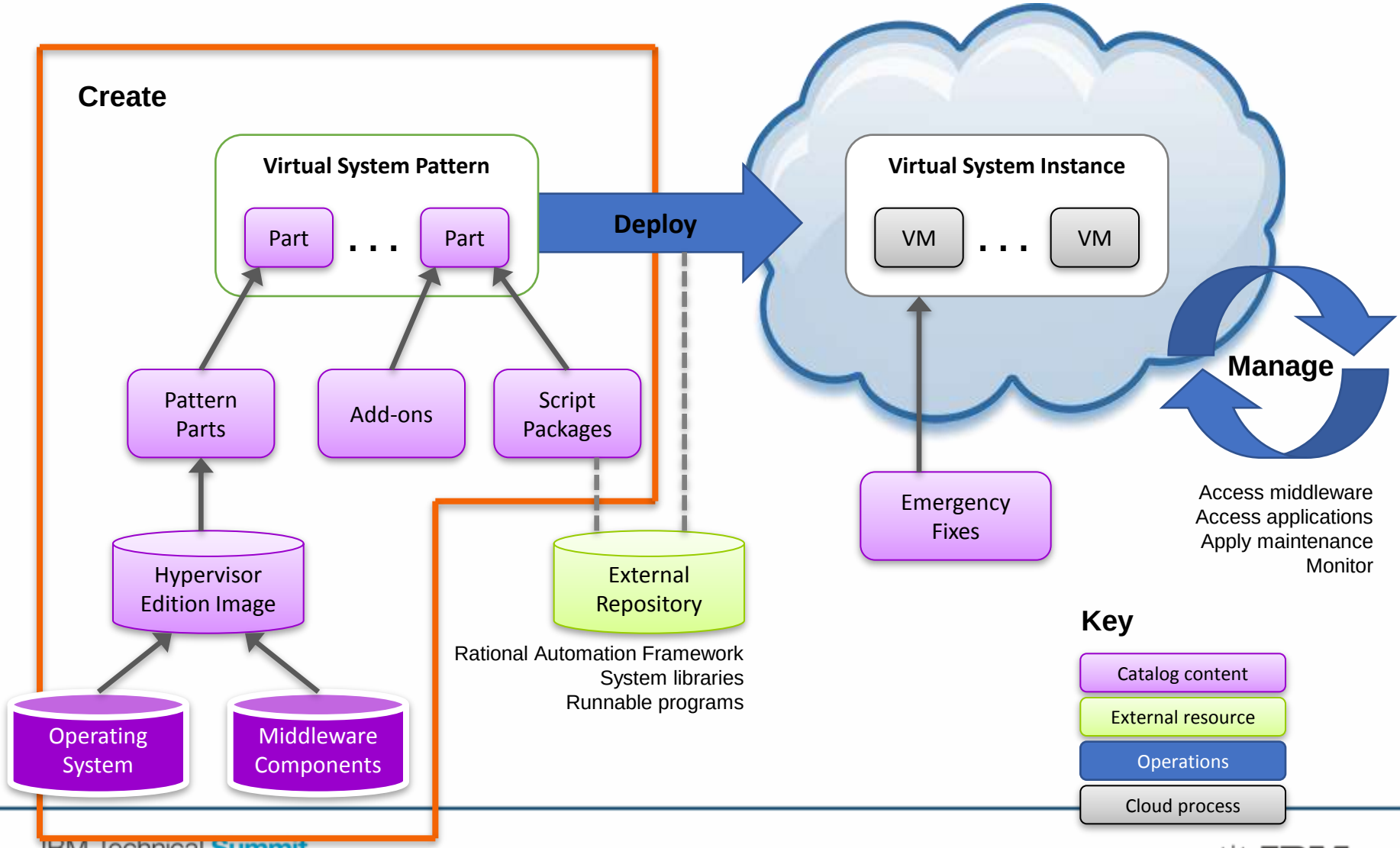
Agenda

- DevOps and Continuous Delivery
- IBM cloud solutions with Virtual System Patterns.
- DevOps tool chain and UrbanCode

Cloud hosts

- A Cloud Host is a public or private cloud capable of deploying virtual machines into a virtualized server environment
- Automatic deployment of virtual machines utilizes images, patterns, and scripts to create a pre-configured and reproducible running system of one or more virtual machines
- The virtual machines are a fully configured stack with virtualized processors, memory, and storage, operating system, middleware, and application code
- We have been exploiting technology Virtual Systems Patterns .

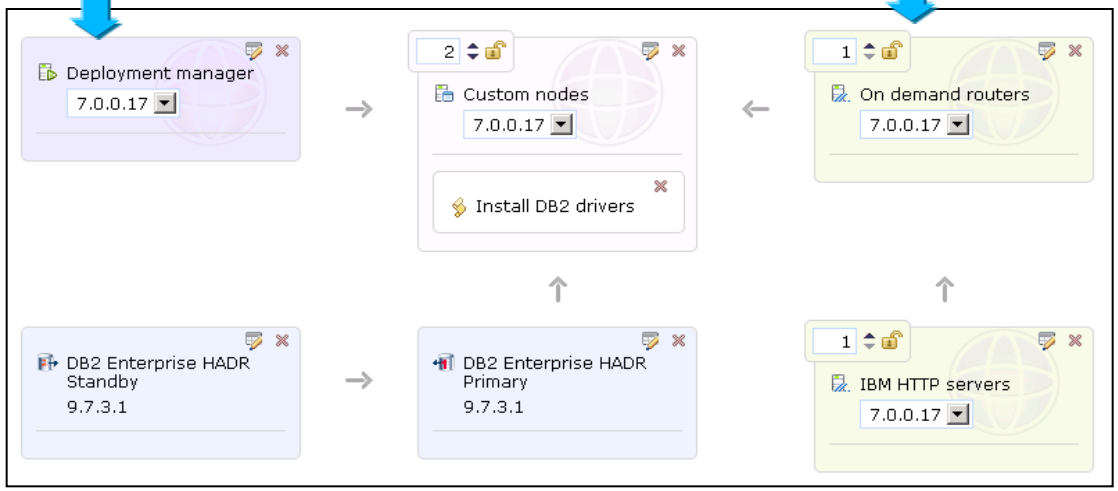
Anatomy of a virtual system



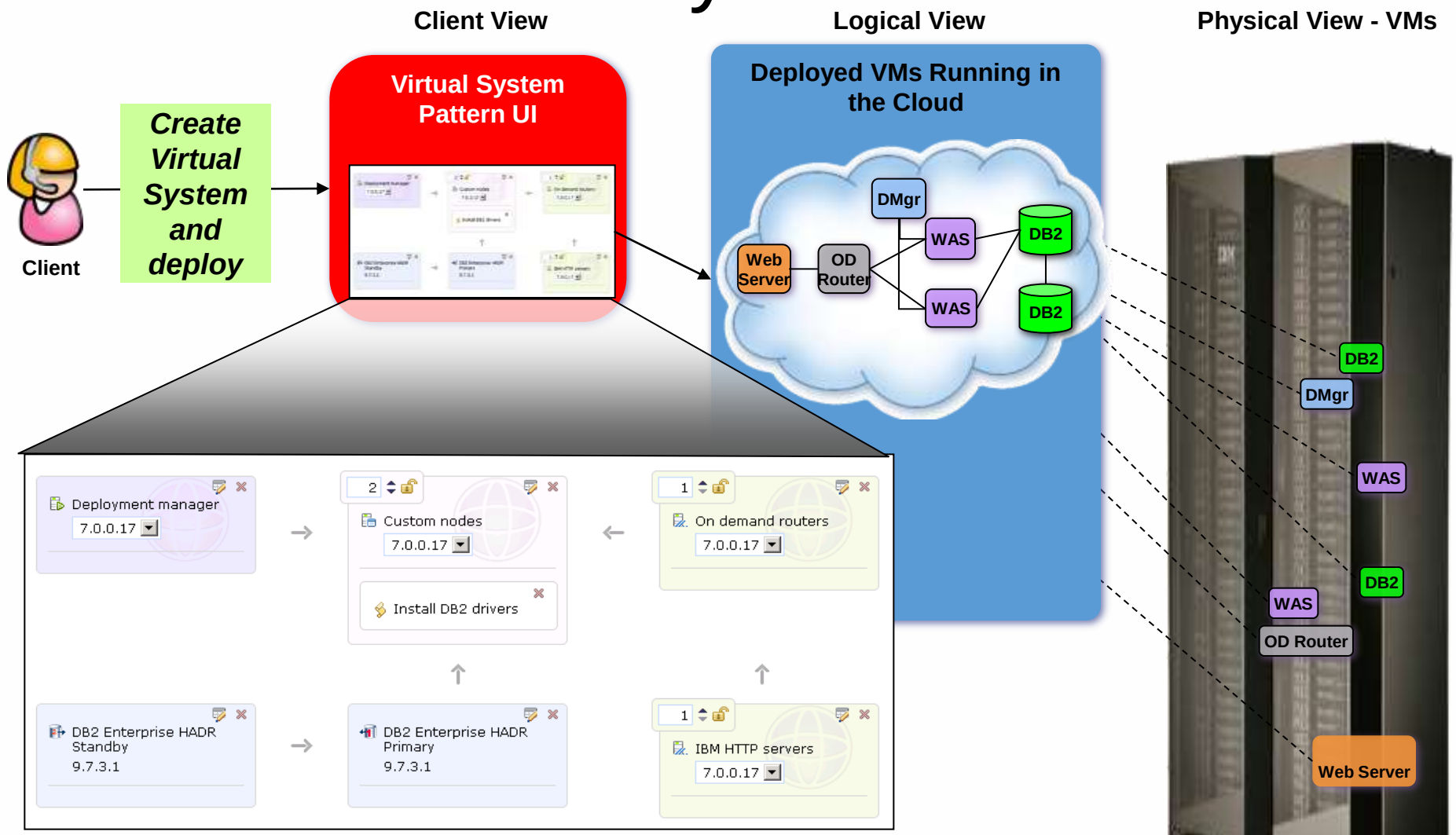
Virtual System Pattern

- A virtual system pattern represents system and middleware topology and automation required to build and wire that topology
  - For example: WebSphere Application Server Cluster pattern containing Deployment Manager, one or more Custom Nodes, IBM Http Server and configuration scripts for installing applications to the topology

What deployer defines **What system deploys**



Virtual System Views



- Focus is on the topology – client creates the topology pattern and deploys it
- Application and configuration scripts are added to the Virtual System pattern

IBM offers cloud host solutions that support Virtual System Patterns

- **SmartCloud Orchestrator**
  - Fully automates the deployment and lifecycle management of cloud services across resources, workloads and services
  - Built on a foundation of open standards – TOSCA, OpenStack, OSLC
  - Accelerated deployments with reusable workload patterns with Virtual System Patterns.
  - Includes capability of SmartCloud Provisioning.
- **PureApplication System**
  - Hardware, software, and a management appliance in a pre-built rack
  - PureApplication Manager can deploy Virtual System Patterns to it's own Intel and Power servers
- **SmartCloud Provisioning**
  - Can deploy Virtual Systems Patterns to Intel, Power, and zLinux hosting platforms
- **IBM Workload Deployer**
  - A hardware appliance that can deploy Virtual System Patterns to Intel, Power, and zLinux hosting platforms

Examples of pre-built IBM software patterns

Web and SOA Applications

- IBM **WebSphere Application Server** Hypervisor Edition for RHEL *
- IBM **Web Application Pattern** *
- IBM Web Experience Patterns for **WebSphere Portal Server** for Red Hat Enterprise Linux
- IBM Web Experience Patterns for **Web Content Manager** for Red Hat Enterprise Linux
- IBM **SOA Policy** Pattern
- IBM **SOA Policy Gateway** Pattern for Red Hat

Connecting Your World

- IBM **Connections** Hypervisor Edition
- IBM **WebSphere Message Broker** Hypervisor Edition for Red Hat Enterprise Linux Server
- **WebSphere MQ** Hypervisor Edition V7.5 for Red Hat Enterprise Linux
- IBM **Messaging Extension** for Web Application Pattern
- IBM **WebSphere Transformation Extender** with Launcher Hypervisor Edition
- IBM PureApplication System virtual application for **SAP CRM**

Data and Transactions

- IBM **DB2** Enterprise Server Edition *
- IBM **Transactional Database** Pattern *
- IBM **Data Mart** Pattern *
- IBM **Informix** Hypervisor Edition

Business Intelligence and Analytics

- IBM Business Intelligence Pattern (**Cognos**)
- IBM **InfoSphere Information Server** for production
- IBM **InfoSphere Information Server** for non production environments

Processes and Decisions

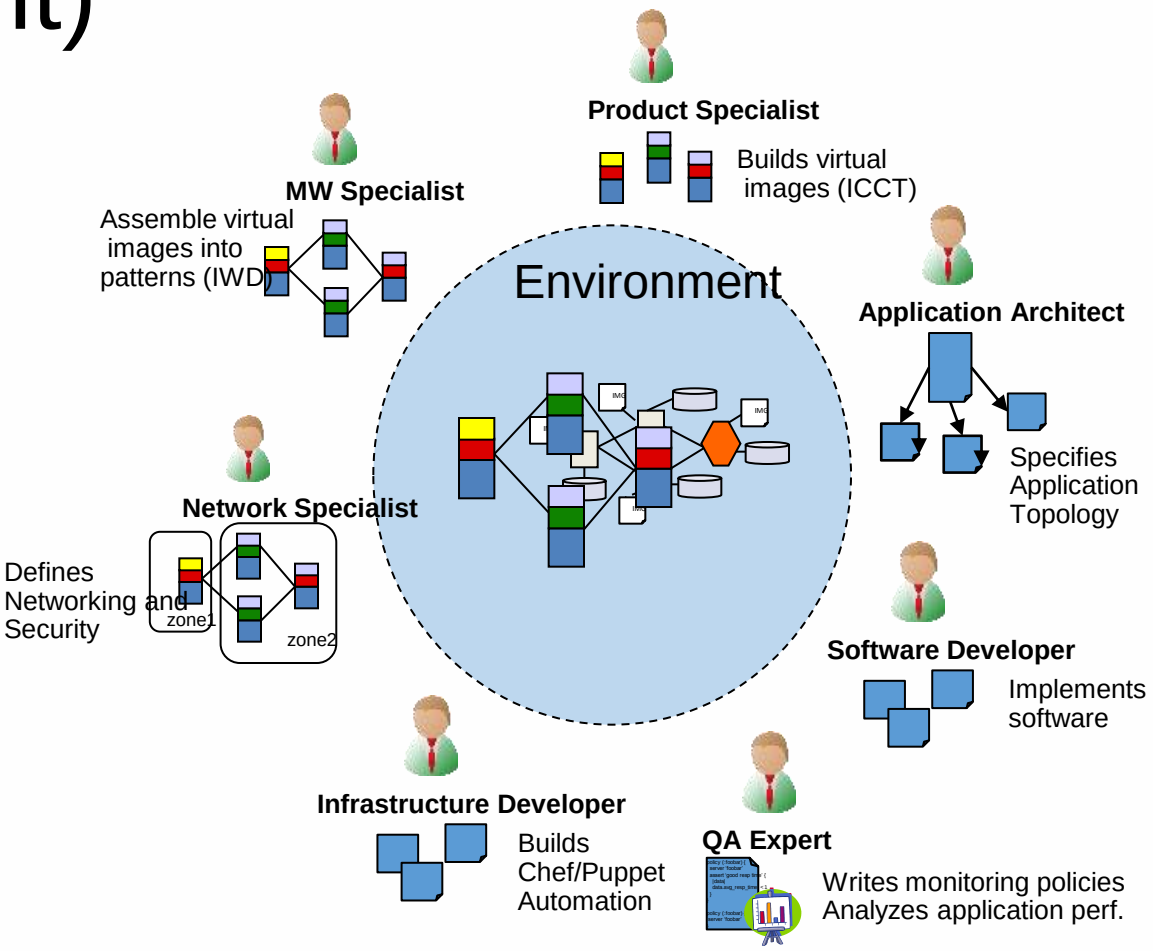
- IBM **BPM** Pattern (Process Center Hypervisor Edition on RHEL)
- IBM **BPM** Pattern (Process Server Hypervisor Edition on RHEL)
- IBM **Operational Decision Manager** Pattern (Decision Center Hypervisor Edition on RHEL)
- IBM **Operational Decision Manager** Pattern (Decision Server Hypervisor Edition on RHEL)

Concepts in SmartCloud Continues Delivery (SCD) of 2012

- First Generation of Continues Delivery offering for cloud.
- Leverages Virtual System Patterns
- Introduced SDE with Weaver technology for complex and flexible deployment
- Leverages Rational tools and IBM Cloud Host tools
  - RTC and RAM
  - SCP
- Also leverages open source tools
  - Chef, cucumber

SDE (Software Defined Environment)

- Cloud enables complete description and use of
  - application
  - middleware
  - infrastructure
- Provisioning and using them is complex task
  - Various roles of experts need to collaborate
- Opportunities for standardization, modularization and reuse.



Weaver

- SDE framework developed by IBM Research.
- DSL (Domain Specific Language) based on Ruby
  - Application
  - Infrastructure (system and middleware)
  - Environment
- Promote reuse with modules
- Reduces errors with validation.

Environment, Application and Infrastructure

- *Environment* = {*applications*} + *infrastructure*
- Each application runs in Environment on Infrastructure.
- Each section is organized as a module for reusability.

Example: **Topology** for JKE Banking *Application*

```
•topology (:jke) {
•  name “JKE”
•  version ”2.0.0”
•  description ”JKE Banking Web/DB Application”
•
•  node (:app_server) {
•    name ”JKE Banking Web application server”
•    description ”Describes all components hosted on your web application server”
•
•    component (:web_app) {
•      name "JKE Web Application”
•      description ”Web frontend in Dojo plus a bunch of REST resource"
•
•      requires :product => "WebSphere Application Server Community Edition", :vers
•      requires :product => /Java/i, :version => /(6|7).*/
•
•      # Configurable properties for the application component
•      property :context_root => ‘/jke’, :validators => [ not_empty ]
•      property :deploy_directory => “/opt/#{asset_id}-#{asset_version}”
•      property :asset_id => “JKE_app”, :validators => [ not_empty ]
•      property :asset_version => “2.0.3.I20120729-1116”, :validators => [ not_empty ]
•      property :war_file => ‘http://ram.ibm.com:13080/ram/artifact/#{asset_id}/#{asset_version},
•              :validators => [ valid_url ]
•  }

•  node (:db_server) {
•    name ”JKE Banking Database server”
•    description ”Describes all components in your DB server”
•    # component details omitted for brevity
•    . . .
•  }
•}
```

- Components of application
  - Application node
    - Uses AP server
  - DB node
- Validation logic

Example: **Topology** for *Infrastructure*

```
•import “virtual_images.weaver”

•topology (:qa_infrastructure) {
•  name “QA Infrastructure for JKE Banking”
•
•  # Represents the Tomcat node
•  node (:tomcat_vm) {
•    name “Tomcat Server”
•    # Declare the usage of role “tomcat_mysql_image” from the imported topology
•    use pure_app_images.tomcat_mysql_image

•    property :cpu => 4
•    property :memory => 4096, :description => “Memory in MB”
•  }
•
•  # Represents the MySQL node
•  node (:mysql_vm) {
•    name “MySQL Server”
•    # Declare the usage of role “tomcat_mysql_image” and alias it to ‘img’
•    use pure_app_images.tomcat_mysql_image => img
•
•    property :cpu => 4
•    property :memory => 8192, :description => “Memory in MB”
•  }
•}
```

- Components of infrastructure
  - AP server
    - Tomcat
    - # of CPU
    - Memory size
  - DB node
    - MySQL
    - CPU,
    - memory,…
- Validation logic

Example: **Topology** for *Environment*

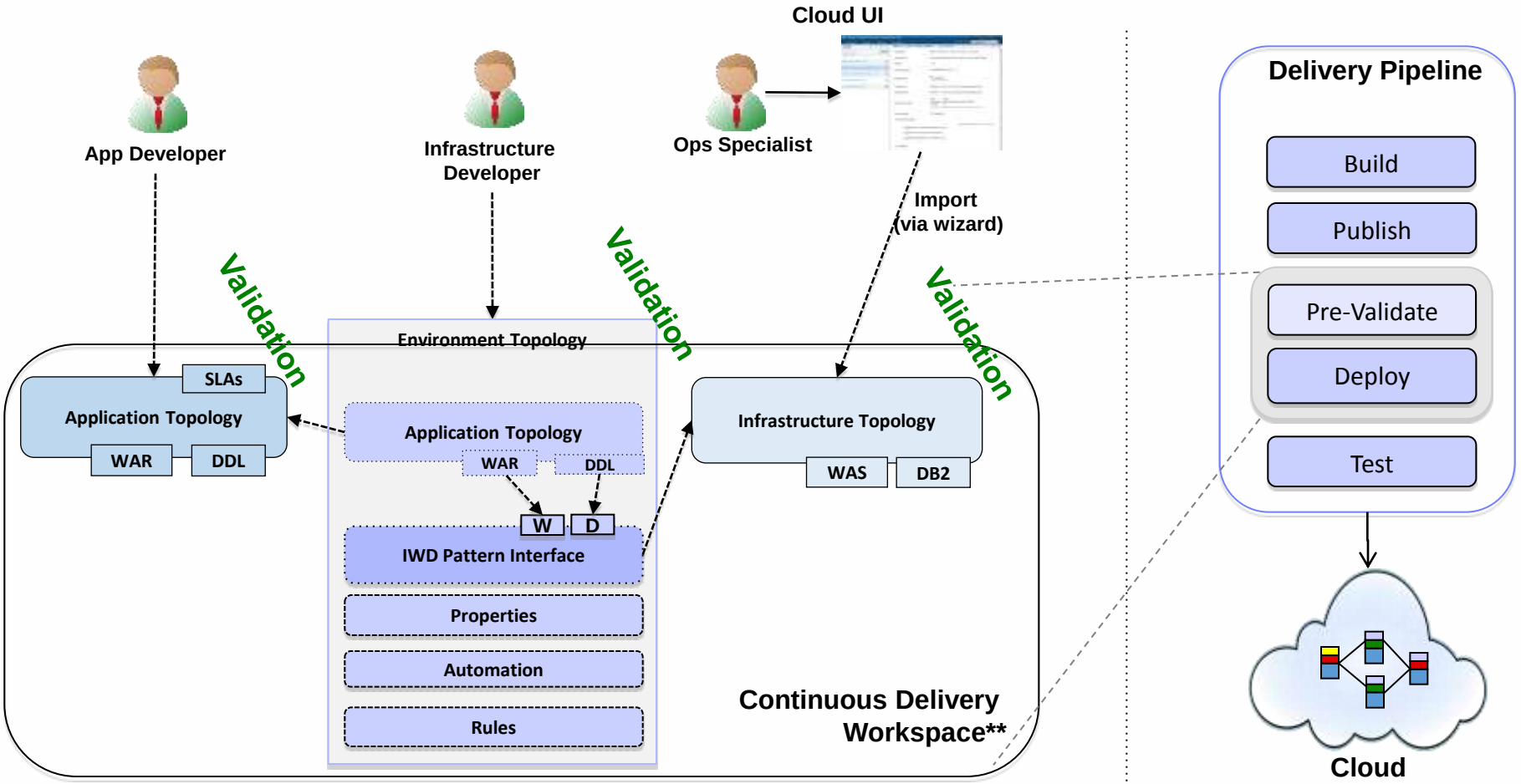
```
•import “qa_infrastructure.weaver”     # Topology for Infrastructure
•import ”jke_app.weaver”               # Topology for Application
•
•topology (:qa_environment) {
•  name ”Integration Testing Environment"
•  description "JKE App on two node IWD pattern”

•  use :qa_infrastructure
•  use :jke
•
•  # Realize application nodes onto pattern nodes
•  realizes jke.app_server => qa_infrastructure.tomcat_server
•  realizes jke.db_server => qa_infrastructure.mysql_server

•  # Environment specific properties
•  property :build_tag => ”TODO", :description => "Build id from build system”

•  # Set properties for the infrastructure node (excerpt)
•  qa_infrastructure.name = “JKEBanking_#{build_tag}”

•  qa_infrastructure.was_server.cpu = 4
•  qa_infrastructure.was_server.memory = 4096

•  qa_infrastructure.db_server.cpu = 4
•  qa_infrastructure.db_server.memory = 4096

•  # Override other values in any of the application or pattern topology (if needed)
•  jke.app_server.web_app.context_root = “/”
•}
```

- Component of environment
  - Link to application Topology
  - Link to infrastructure Topology
- Validation logic
- Link to Chef script for application deployment.
- Deployment parameters

SDE

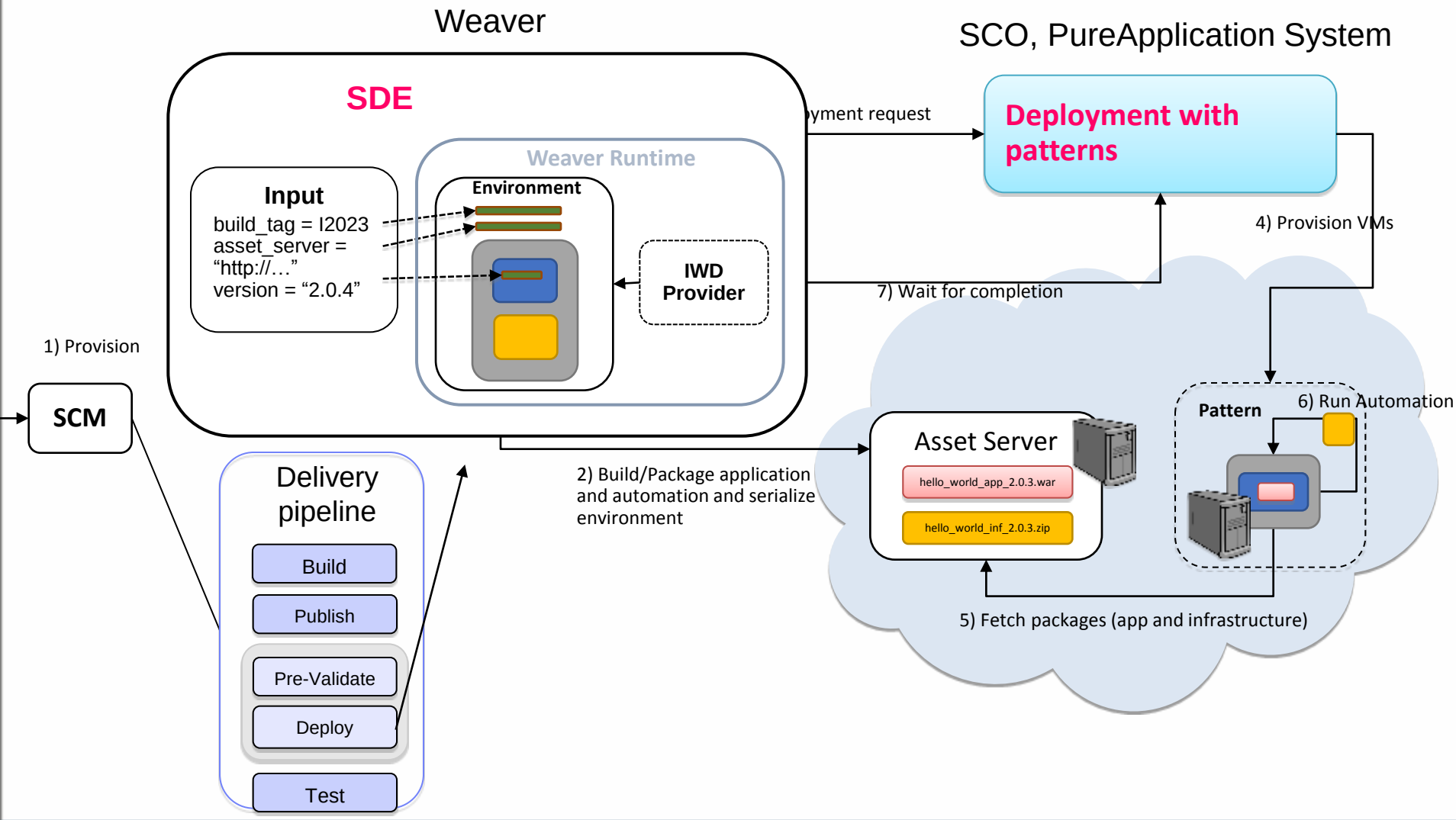
Development View

Runtime View

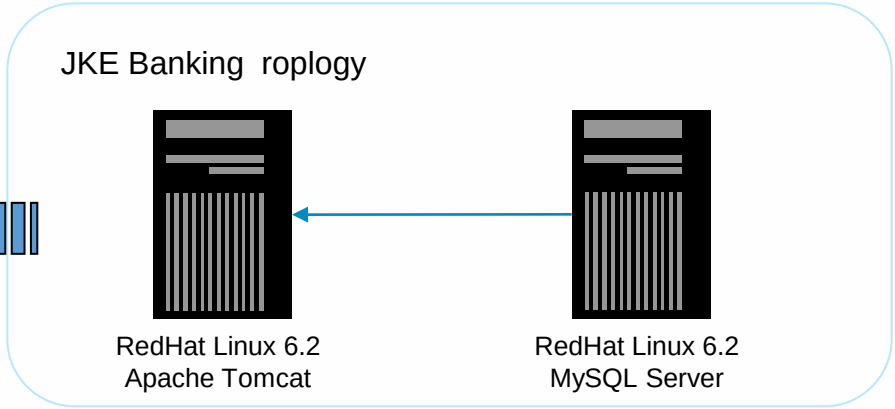


** Versioned in SCM

Deployment flow with SCD



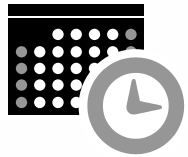
Comparison of sample deployment



Manual Deployment

Activities	time
Build validation and library registration	30 min.
Provisioning two Linux machines	2 hours
Installing Apache Tomcat and application war files	1.5 hour
Installing MySQL server and setting up DB.	1.5 hour
Setting up a connection between the application and DB.	30 min.
Smoke test and validation	30 min.
total	6.5 hours

Deployment with SCD



activities	time
Executing automatic deployment	50 min.
total	50 min

Agenda

- DevOps and Continuous Delivery
- IBM cloud solutions with Virtual System Patterns
- DevOps tool chain and UrbanCode

DevOps Tool Chain

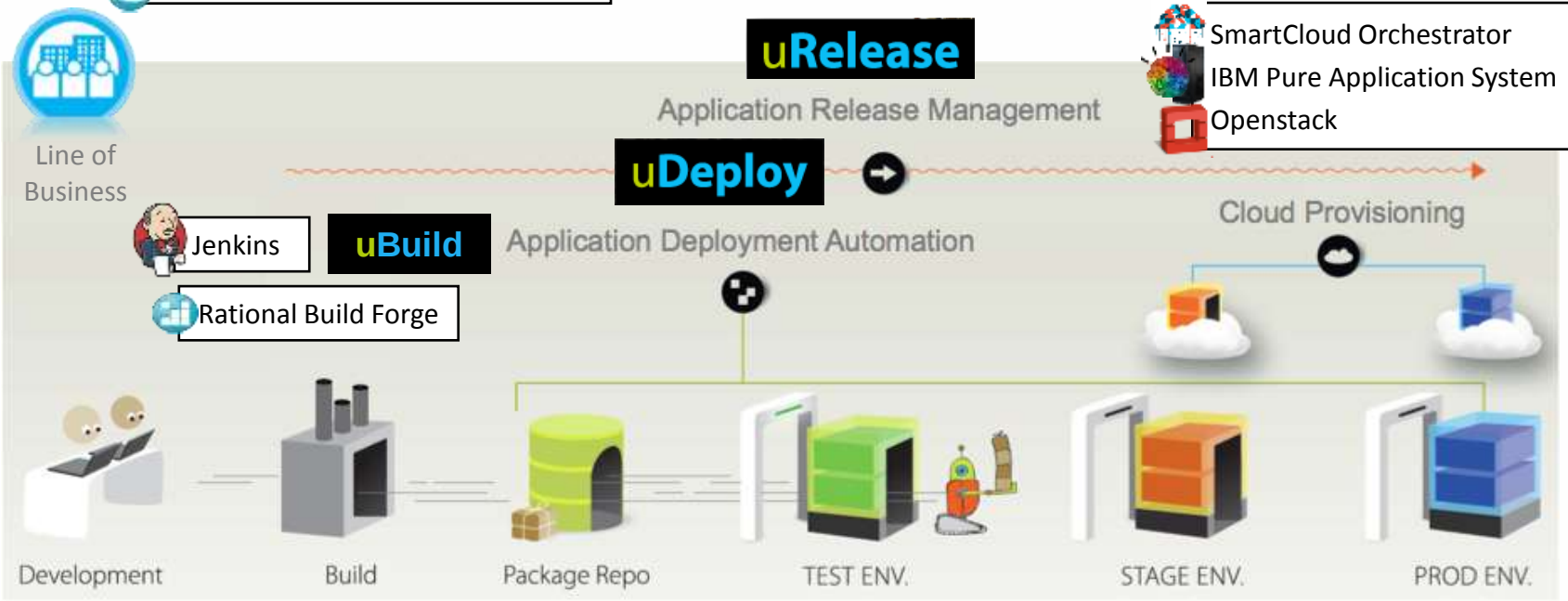
Plan and Measure

Rational Focal Point
Rational Requirements Composer

Release and Deploy

IBM SmartCloud

SmartCloud Orchestrator
IBM Pure Application System
Openstack



Rational Team Concert

Rational Quality Manager
Rational Test Workbench
Rational Test Virtualization Server

SmartCloud Control Desk

SmartCloud Application Performance Management

tealeaf

IBM Technical
Stay ahead.

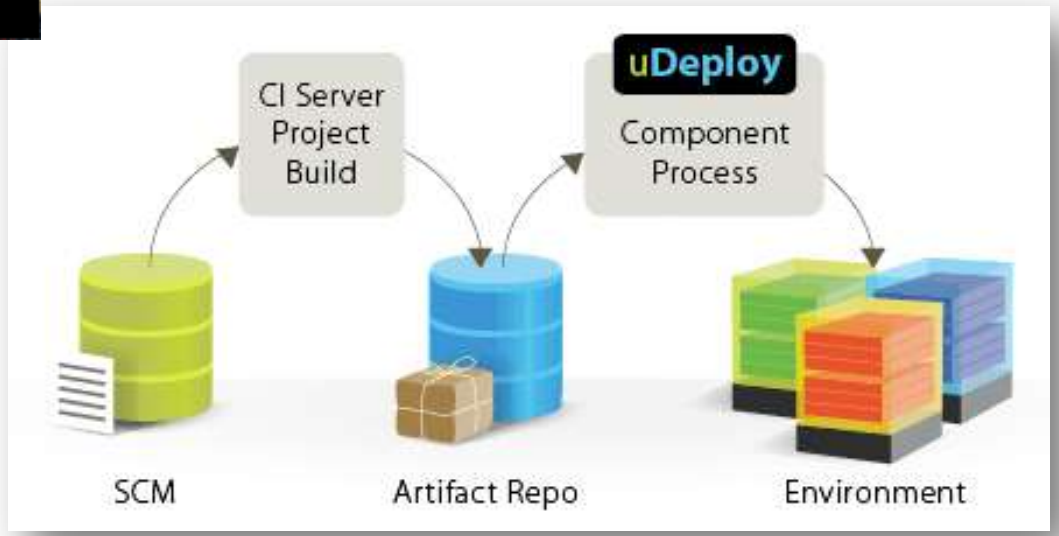
Develop and Test

Monitor and Optimize

uDeploy

Automating deployment of applications across dev, test, and production environments

uDeploy
Application
Deployment
Automation



Application Model of uDeploy

uDeploy

Components

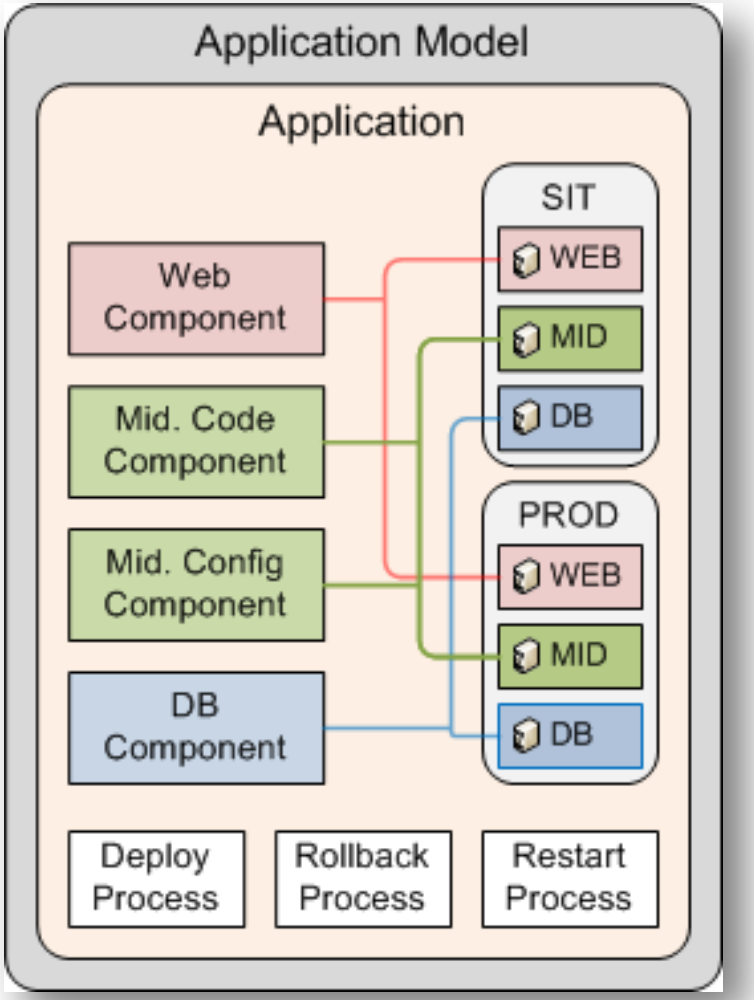
- Tiers or services

Environments

- Target servers
- Servers have roles
- Env. Specific Config

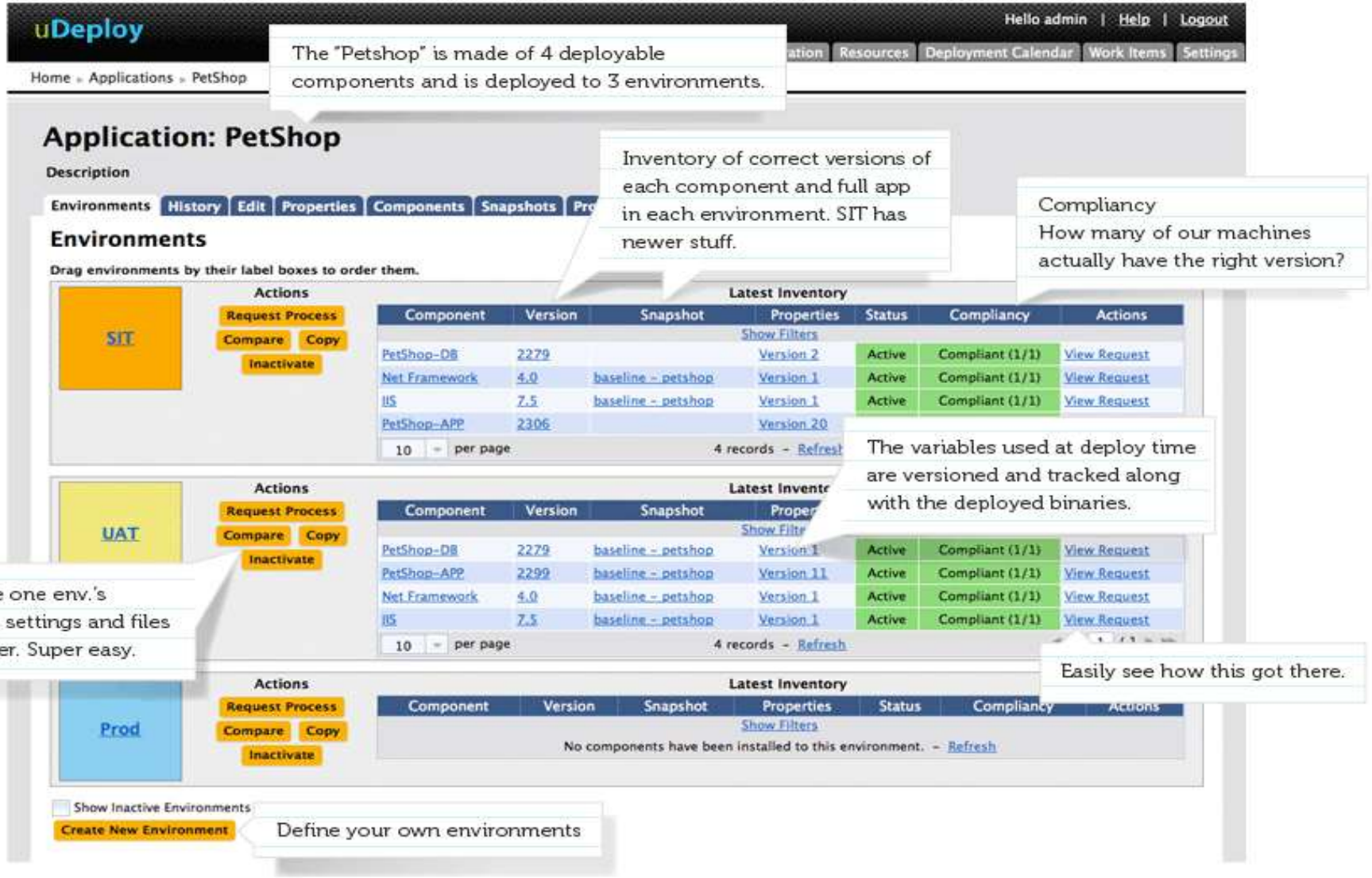
Processes

- Coordinates Component processes

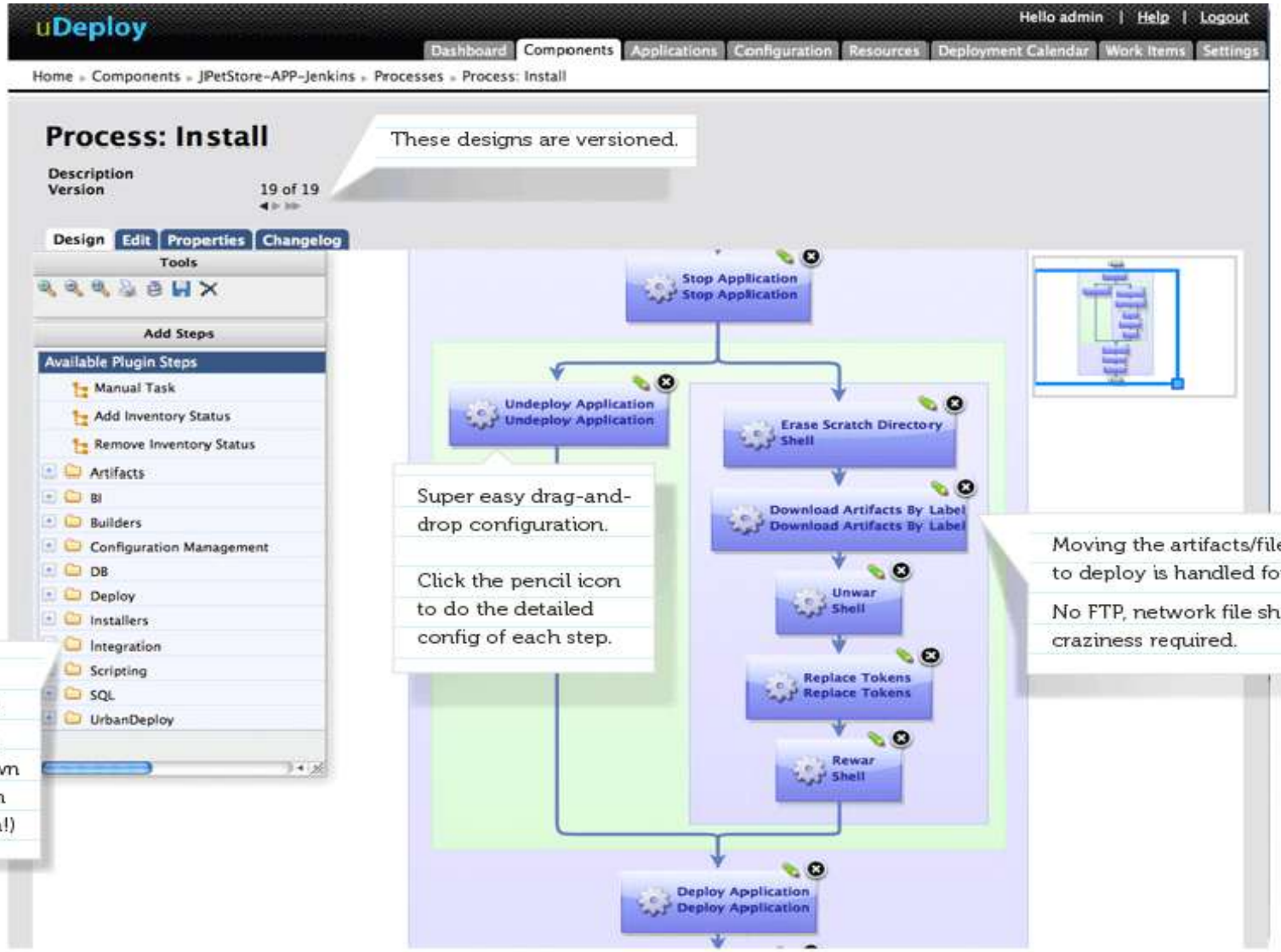


uDeploy configuration settings

Application environment inventory



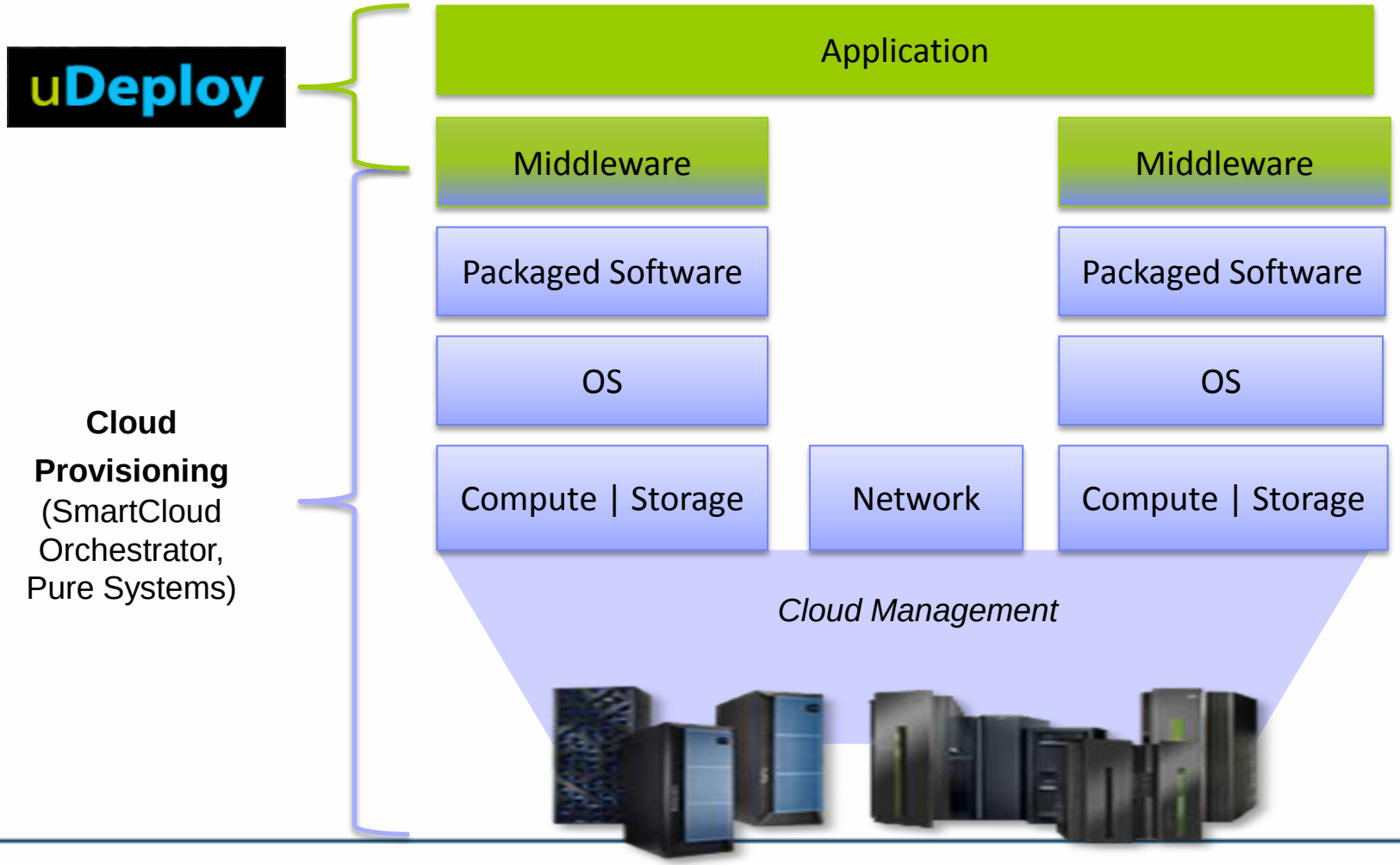
uDeploy process designer



What's New in UrbanCode uDeploy v6.0

- **Continuous Delivery with Cloud**
- Continuous Testing with Virtualized Services (Green Hat)
- DevOps for Mobile
- Discover and configure WebSphere middleware
- ALM Integration with
  - Rational Asset Manager
  - Rational Quality Manager
  - Rational Build Forge

uDeploy & Cloud Provisioning



Cloud Provisioning (SmartCloud Orchestrator, Pure Systems)

Demo

Summary

- To meet the challenge of latest trends (mobile, social, big data and cloud), IBM has defined DevOps as enterprise capability of end to end software delivery with accelerated speed, quality and fast feedback cycle.
- IBM UrbanCode is key solution for continued delivery in our DevOps solutions.
- Its uDeploy V6 supports continues delivery for Cloud with IBM Cloud Host products.
- Technical innovations include
  - Virtual system patterns for repeatable and reusable system deployment.
  - Software Defined Environment with Weaver.
- Rational has adopted continues delivery for our product delivery.
  - CLM has moved from a yearly release to quarterly release of new features.

Questions

Innovate2013
The IBM Technical Summit

Acknowledgements and disclaimers

Availability: References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Thank You