Enterprise Application Management with Spring

Why Manage Your Applications?

- Identify and eliminate performance bottlenecks
- Minimize application downtime
- Prevent problems before they occur
- Analyze trends in application usage and performance
- Control application at runtime
- Enforce SLAs

Agenda

- Introducing SpringSource Application Management Suite
- Instrumenting a Spring-powered Application
- Full Stack Monitoring with SpringSource AMS
- Demo: Managing the Spring Travel Application
- Summary
- Q&A

Introducing…..
SpringSource Application Management Suite (AMS)

- Manage and monitor Spring-powered applications and the platforms and servers they run on
- Granular monitoring and control of Spring components
- Monitor and control the entire cluster from one easy-to-use dashboard
- Bridges the gap between development and operations staff

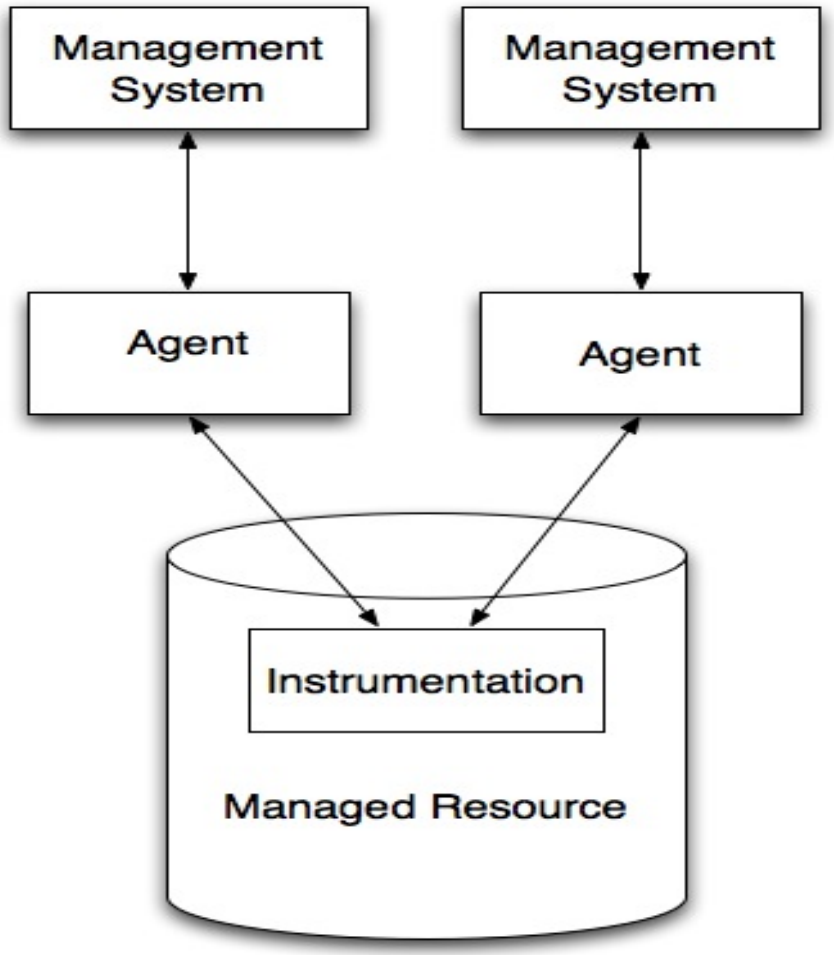
SpringSource AMS Features

- Auto-Discovery
- Monitoring
- Alerting and Corrective Control
- Enterprise Reporting
- Configuration and Log Event Tracking
- Metric Base-lining
- Integration with Existing Management Tools

System Requirements

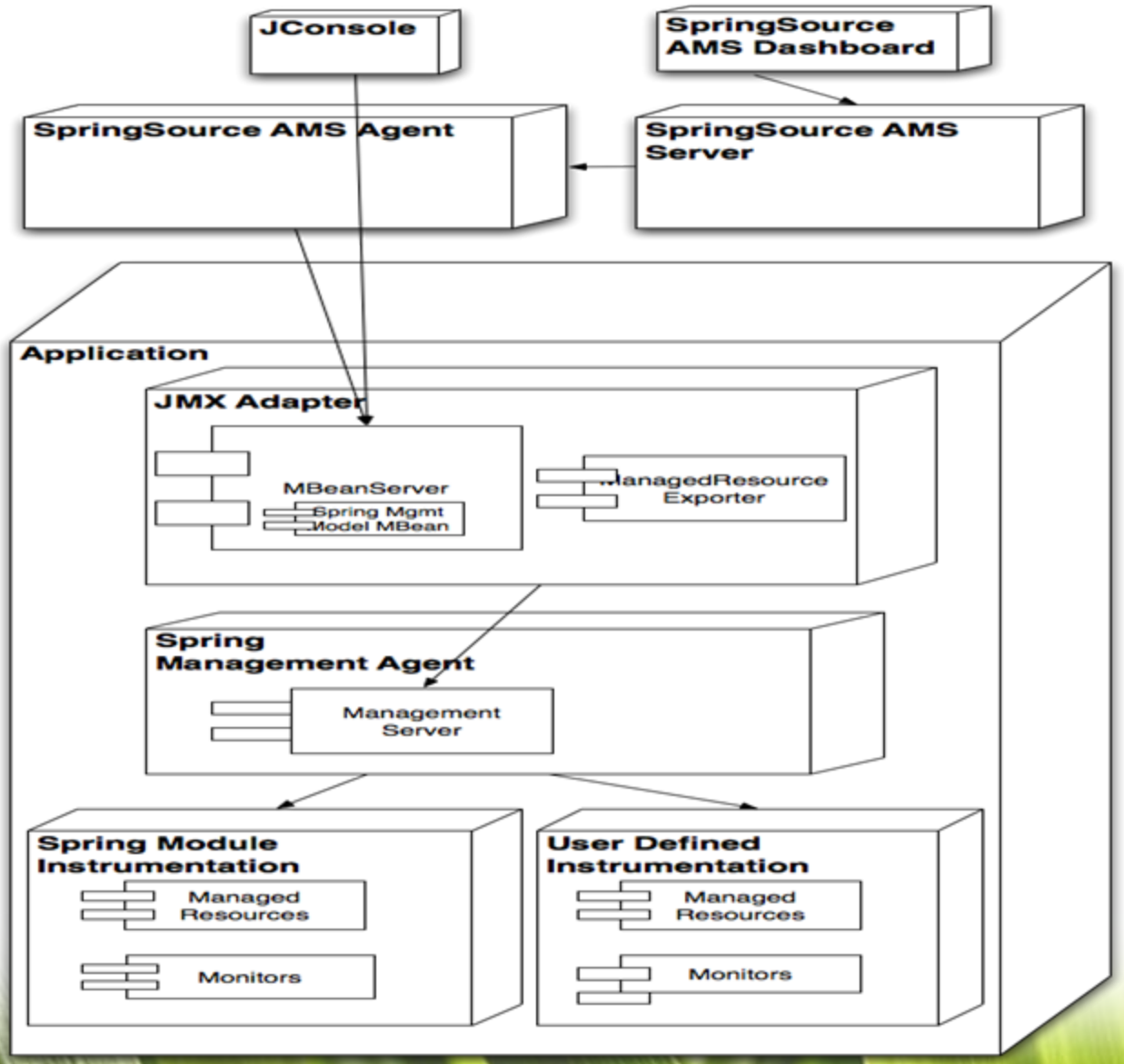
- Monitored Spring-Powered Applications must be run using Java 5 (and instrumented using latest Spring Frameworkcomponents)
- AMS specifically recognizes apps running in SpringSource Application Platform, WebSphere, WebLogic, Tomcat , and JBoss
- Standalone applications and applications in other web/application servers can be recognized by using a System Property

Traditional Architecture/Terminology



- SpringSource AMS consists of:
  - AMS Server (Management System)
  - AMS Agent (Agent)
  - Instrumentation Agent (Instrumentation)
  - Instrumented Spring Portfolio Libraries (Instrumentation)

SpringSource AMS Architecture

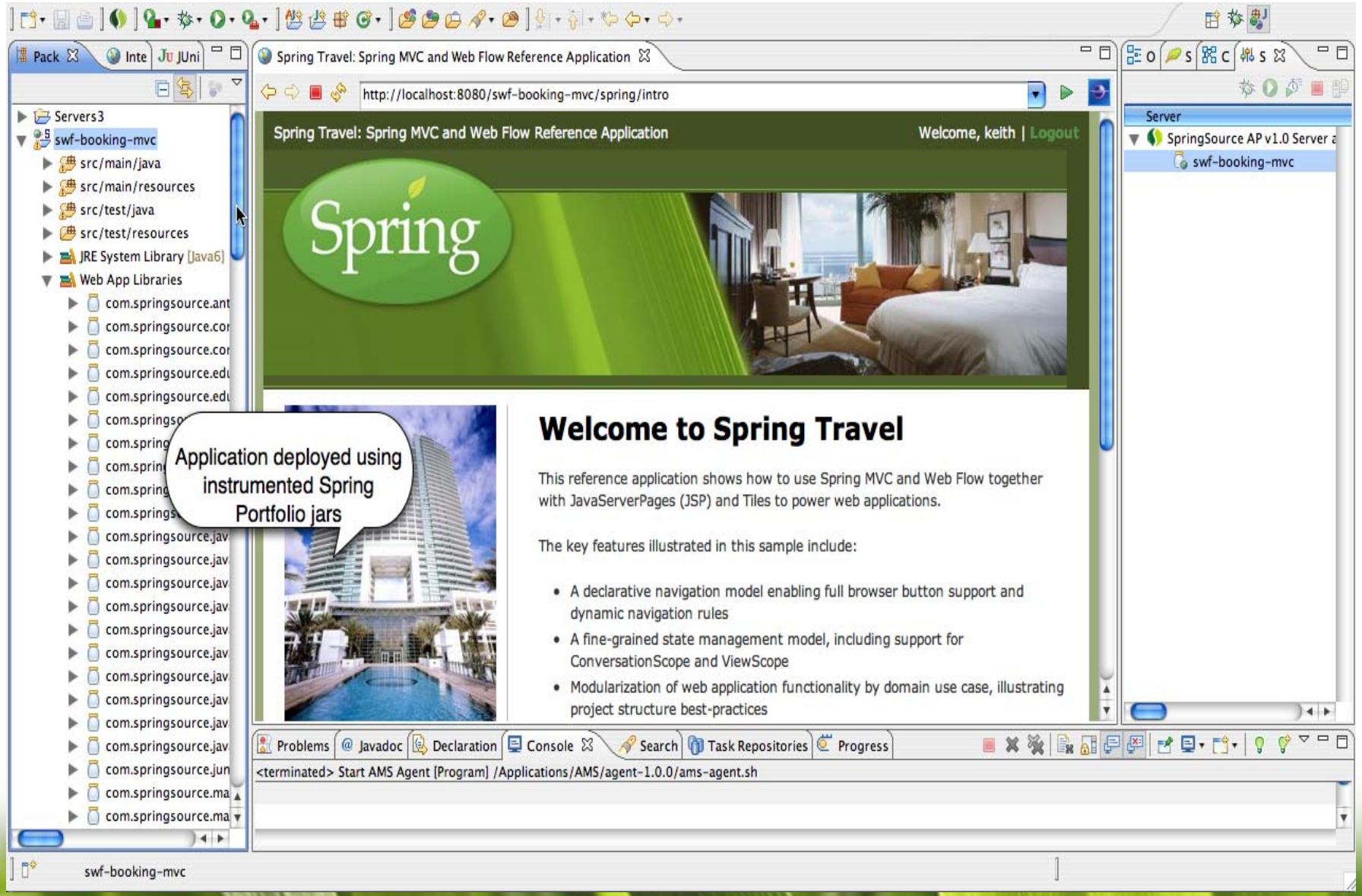


Deliverables

- Platform-Specific Installer Archive that will install SpringSource AMS server and optionally the AMS agent
- Platform-Specific AMS Agent Archive
- “Instrumentation Agent” Archive
- Latest Spring Releases Instrumented:
  - Spring Framework
  - Spring Web Flow
  - Spring Security
  - Spring Web Services

Instrumenting a Spring-Powered Application

Instrumentation



Application Discovery

Application Discovery

- Applications automatically discovered by advising refresh method of ApplicationContexts
- Aspects are used to add a BeanPostProcessor to any BeanFactory created in the application
- The BeanPostProcessor considers all newly created beans for automatic export to the management model

Inventory

Inventory - ManagementServer

- Main entry point into instrumentation agent
- ManagementServer Supports:
  - Retrieval of Domains

Inventory - Domains

- Domain – A logical grouping of managed resources
  - ex. Applications, Subsystems
  - Subsystems
- Domain Supports:
  - Retrieval of ManagedResources
  - Retrieval of Configuration Attributes and Operations

Inventory – Managed Resources

- Managed resources contain:
  - Attributes
  - Metrics
  - Operations
  - Configuration Attributes
  - Configuration Operations

Inventory - Attributes

- Attributes contain:
  - Name - unique wrt containing ManagedResource or Domain
  - Class type of the attribute value
  - Description
  - ReadOnly flag
- Attributes support:
  - getValue/setValue

Inventory - Metrics

- Metrics contain:
  - Name - unique wrt containing ManagedResource
  - MetricCategory
  - ChangeType
  - DefaultCollectionInterval
  - Class type of metric value
  - Unit of metric value
    - Units contain Dimension and Symbol
    - Support: Conversion of values between units, division, power, root
  - Indicator Flag

Inventory – Metrics (cont)

- Metrics support:
  - getValue
  - Values returned as Measures
    - Measure is the result of a measurement
    - Measures contain a numeric value and unit
    - Measures support conversion between units, representation of numeric values in long, int, double, and float

Inventory - Operations

- Operations contain:
  - Name – unique wrt containing ManagedResource
  - Description
  - OperationParameter Metadata
  - Class type of return value
- Operations support:
  - Invoke

Inventory - Configuration

- Configuration Attributes
  - Domains and Managed Resources have them
  - Configures property of management system (as opposed to the resource being managed)
  - ex. “monitoringEnabled”
- Configuration Operations
  - Domains and Managed Resources have them
  - Control operation that configures the management system (as opposed to the resource being managed)
  - ex. “resetMetrics”

Monitoring

Aggregate Monitoring

- Only aggregrate performance and utilization data maintained in memory
- Monitors stored in MonitorRepository by Monitor Key
  - MonitorInstanceKey
  - MonitorClassKey
- Individual operation information discarded after used to update monitors

Performance Monitors

- Updated via recordExecution and recordFailure methods
- Collect and expose metrics (as Measures):
  - Accumulated Time
  - Average Elapsed Time
  - Count
  - Failure Count
  - Max Time
  - Min Time
  - Throughput

Utilization Monitors

- Updated via incrementCount and decrementCount methods
- Collect and expose metrics (as Measures):
  - Count
  - Max Count
  - Min Count

Operation Monitoring

- Operations monitored through startOperationMonitor, endOperationMonitor, and endOperationMonitorFailure methods
- MonitoredOperations contain the following information about a single execution of an operation:
  - Count (ex. 10 messages)
  - Start Time
  - End Time
  - Failure Count
  - MonitoredOperationError
  - Parent MonitoredOperation

Spring AOP Monitoring Proxies

- Spring AOP used to monitor stereotyped components or third parties that cannot be compile-time woven
- Automatically exposes execution time metrics for each operation
- Uses same Operation Monitoring approach as aspects woven into Spring components at compile-time

JMX Adapter

MBean Server Discovery

- When JMX Adapter jar is on classpath, MbeanServer discovery happens on agent bootstrap
- Same discovery algorithm as Spring Framework
- Discovers WebLogic server by JNDI lookup
- Discovers WebSphere server using AdminClient interface
- Discovers local MBeanServer

MBean Export

- Constructs ModelMBeans representing Domains and ManagedResources
- Builds ModelMBeanInfos from Attribute, Operation, and Metric metadata
- Converts Attribute/Operation types to JMX OpenTypes

Configuring Remote JMX Access

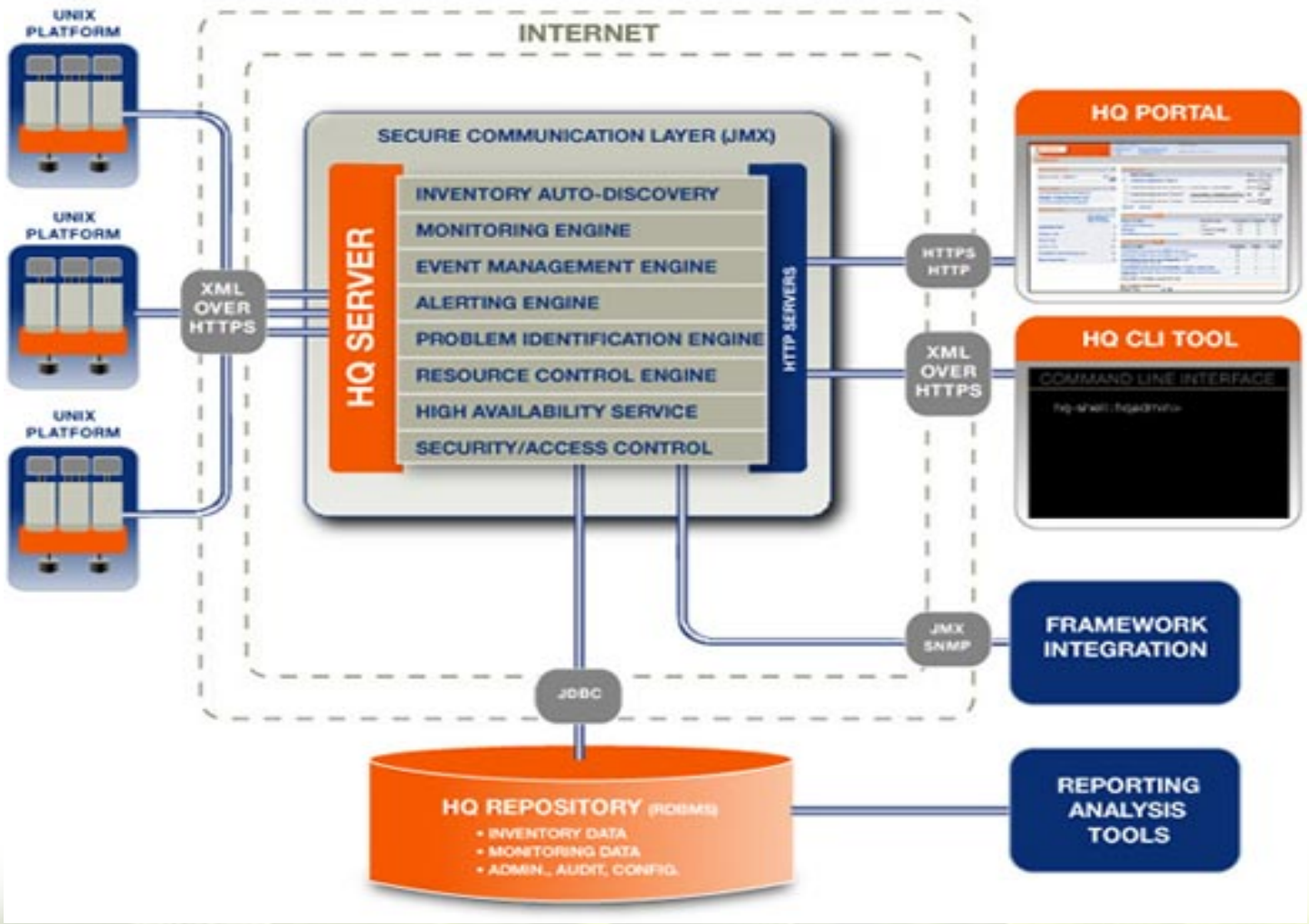
- Standalone applications and Tomcat must expose a JSR-160 Connector to view JMX MBeans
- Can be created programatically using ConnectorServerFactoryBean
- Can be created automatically by VM using system properties
  - ex. -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=6969 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false

SpringSource AMS Server and Agent
Full Stack Monitoring Solution

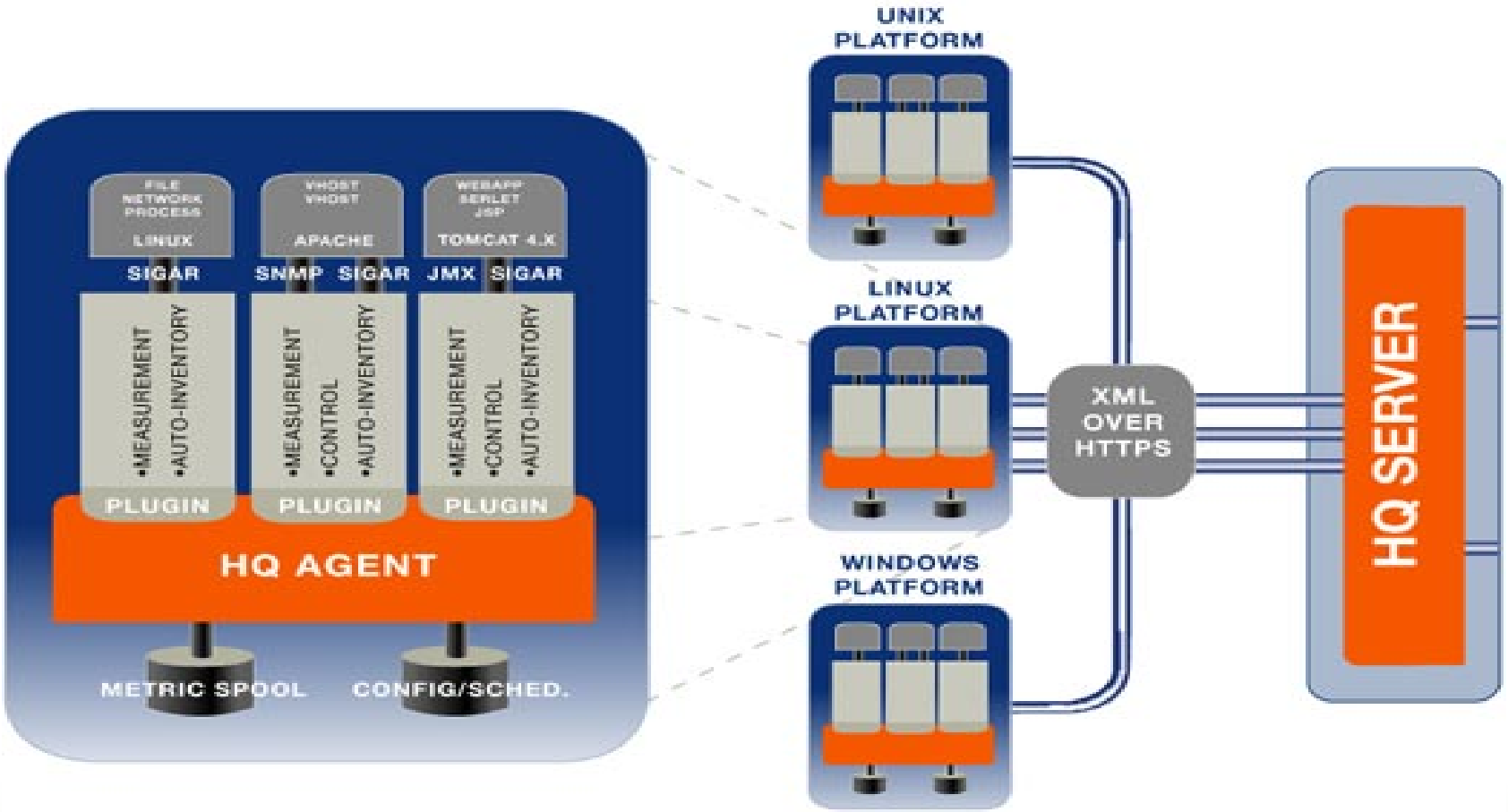
SpringSource AMS Server/Agent

- SpringSource AMS contains Hyperic HQ Enterprise Edition inside.
- The AMS server is SpringSource's version of the HQ Server. The AMS agent is SpringSource's version of the HQ Agent.

SpringSource AMS Server



SpringSource AMS Agent



Demo
Managing the Spring Travel Sample Application

Summary of Benefits

- Instrumentation – Consistent management model uniformly representing all components of your application
- AMS- Across-Cluster Views of Management Data (versus tools like Jconsole)
- Initial customers wil have large influence over future features/instrumentation

Benefits Continued

- Development time savings as well as production time savings
- Lower performance overhead on instrumentation due to compile-time weaving

Future Directions

- Add deeper and broader instrumentation
- Publish API/provide toolboxes for users to instrument their own components (includes integration with STS)
- Predefined alerts and corrective control
- Custom reports, diagnostic views (such as in-flight transactions)

SpringSource AMS Beta Program

- Participate in the AMS Beta Program
- http://www.springsource.com/products/suite/ams

Questions?

Jennifer Hickey

jennifer.hickey@springsource.com

Thank You for Attending

Find out more at:

http://www.springsource.com/products/suite/ams