December 4-7, 2006, Santa Clara Marriott, Santa Clara, CA

WS-Man and Wiseman

An Open Source Implementation of the Ws-Management Specification for Java

Presenters:

William Reichardt - ORACLE

Simeon Pinder - hp invent

John Tollefsrud - Sun microsystems
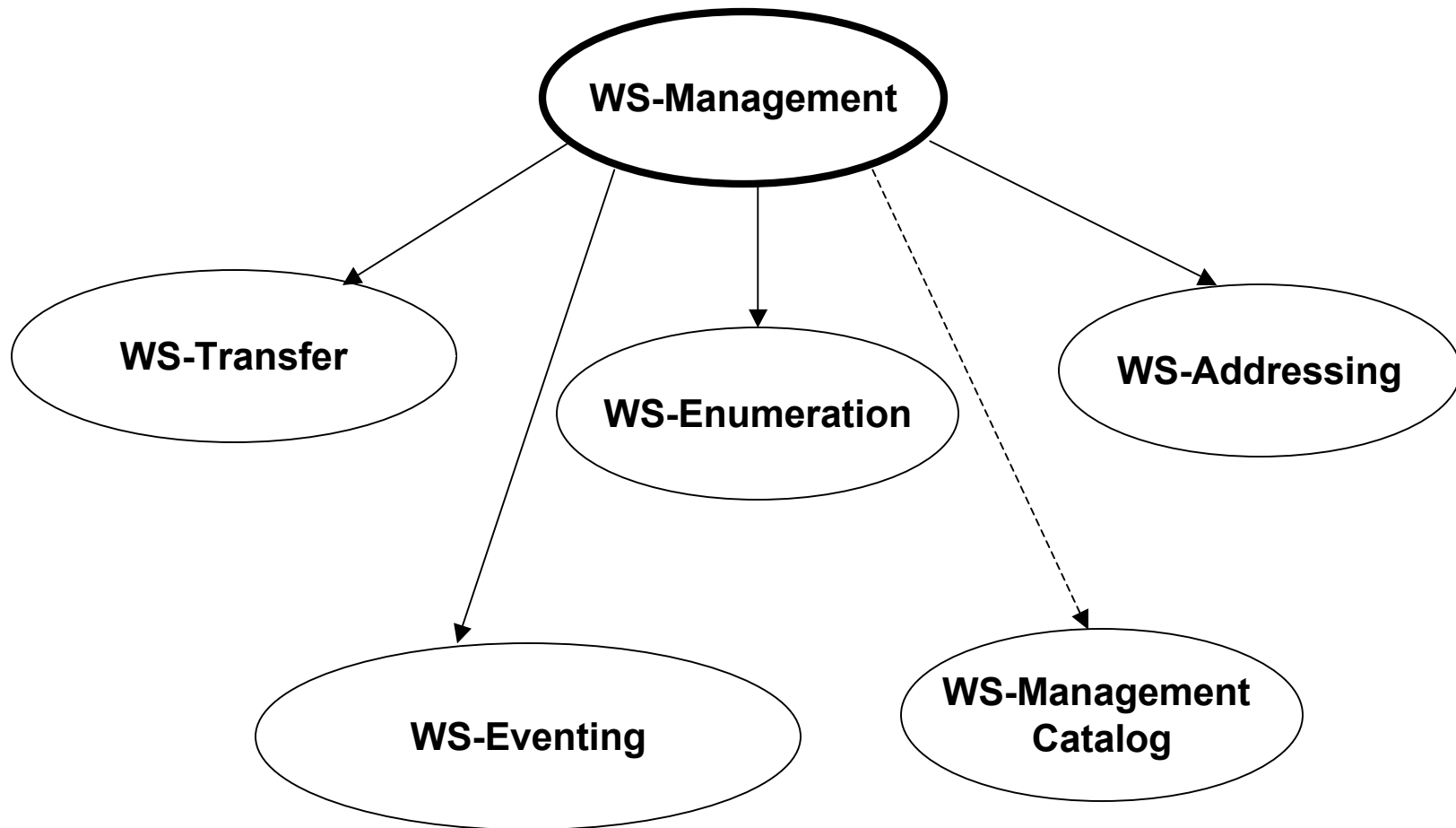
What is WS-Management?

- Made up of a collection of specs

What is WS-Man used for?

- Exposing Management Models/Resources as Web Services
  - CIM, JMX or Custom Models
- Vendor Independent, Architecture Independent
- XML based
- Built into Windows Vista
- Being adopted by HP OpenView

What is Wiseman?

- A pure Java 5.0 Implementation of WS-Man
- Provides both Server and Client APIs
- Server requires Tomcat or J2EE Servlet Support
  – Deploys as a WAR
- Uses parts of JWSDP
  – JAXB - XML to Java Binding
  – SAAJ - SOAP API

Resource Features In A Nutshell

- **Transfer**

  - Resource 'States' are represented as XML documents defined with a Schema

  - Request and Response documents appear as Java Beans via JAXB

  - Glue Code to your back end simply copies state data into these Beans

  - WS-Transfer defines Java Bean like actions (get,set, create and delete)

  - Your glue code copies data to or from from you model into these beans

- **Enumeration**

  - WS-Enumeration defines collections of resources

  - Wiseman's tooling generates an enumeration for every resource you define automatically.

  - You can delegate filtering queries to your model or use the default XPath on the state documents in the collection

```xml
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope>
  <env:Header>
    <wsa:Action env:mustUnderstand="true">
        http://schemas.xmlsoap.org/ws/2004/09/transfer/Get
    </wsa:Action>
    <wsa:ReplyTo>
      <wsa:Address env:mustUnderstand="true">
        http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
    </wsa:Address>
    </wsa:ReplyTo>
    <wsa:MessageID env:mustUnderstand="true">
        uuid:3eb09bda-e3be-4450-8edb-1f74d77bc0c3
    </wsa:MessageID>
    <wsa:To env:mustUnderstand="true">http://localhost:8080/wsman/</wsa:To>
    <wsman:ResourceURI>wsman:system/2005/02/this</wsman:ResourceURI>
    <wsman:OperationTimeout>PT30.000S</wsman:OperationTimeout>
    <wsman:SelectorSet>
      <wsman:Selector Name="SystemName">
        sun-v20z-1
      </wsman:Selector>
    </wsman:SelectorSet>
  </env:Header>
  <env:Body/>
</env:Envelope>
```

Maps to a Action/Method

Maps to a handler class

Maps to a Particular Resource Instance

Non WSI Compliant

- Why don't you use AXIS or JAX-WS?
- Action Dispatching and WSI compliance
- The WSManServlet
- Document "Handlers"

```java
public interface Handler {

    void handle(final String action, final String resource,
            final HandlerContext context,
            final Management request, final Management response) throws Exception;
}
```

- Handlers are Mapped by ResourceURI/ Introspection
- Handlers map Actions to Methods to create responses

A Sample Request
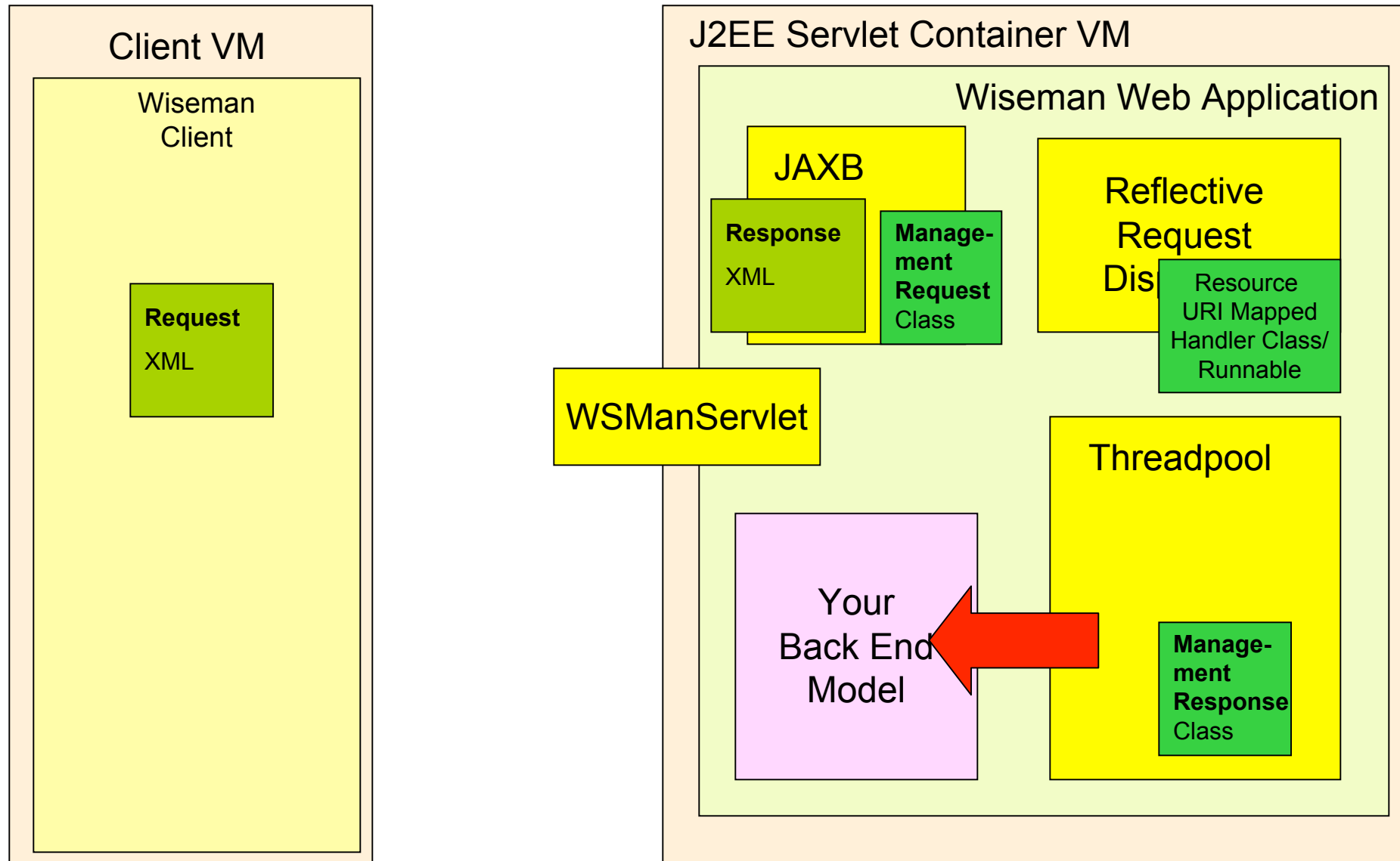
How do you expose a resource?

- Implement Handler Yourself
- Generate your application from Schema with tools.
- Genwsdl ant task
  - Converts Schemas to WS-Man compliant WSDLs (uses soapaction attribute)
- wsdl2wsman Ant Tasks
  - Generates JAXB classes to Wrap Schema Types
  - Generates a handler class from your WSDL
  - Verifies WS-Man WSDL

- Supports Push and Pull Delivery Modes
- Eventing Extentions are still under development
- Provides EventSource and SubscriptionManager interfaces.
- EventingSupport utility classes for simple Eventing implementations
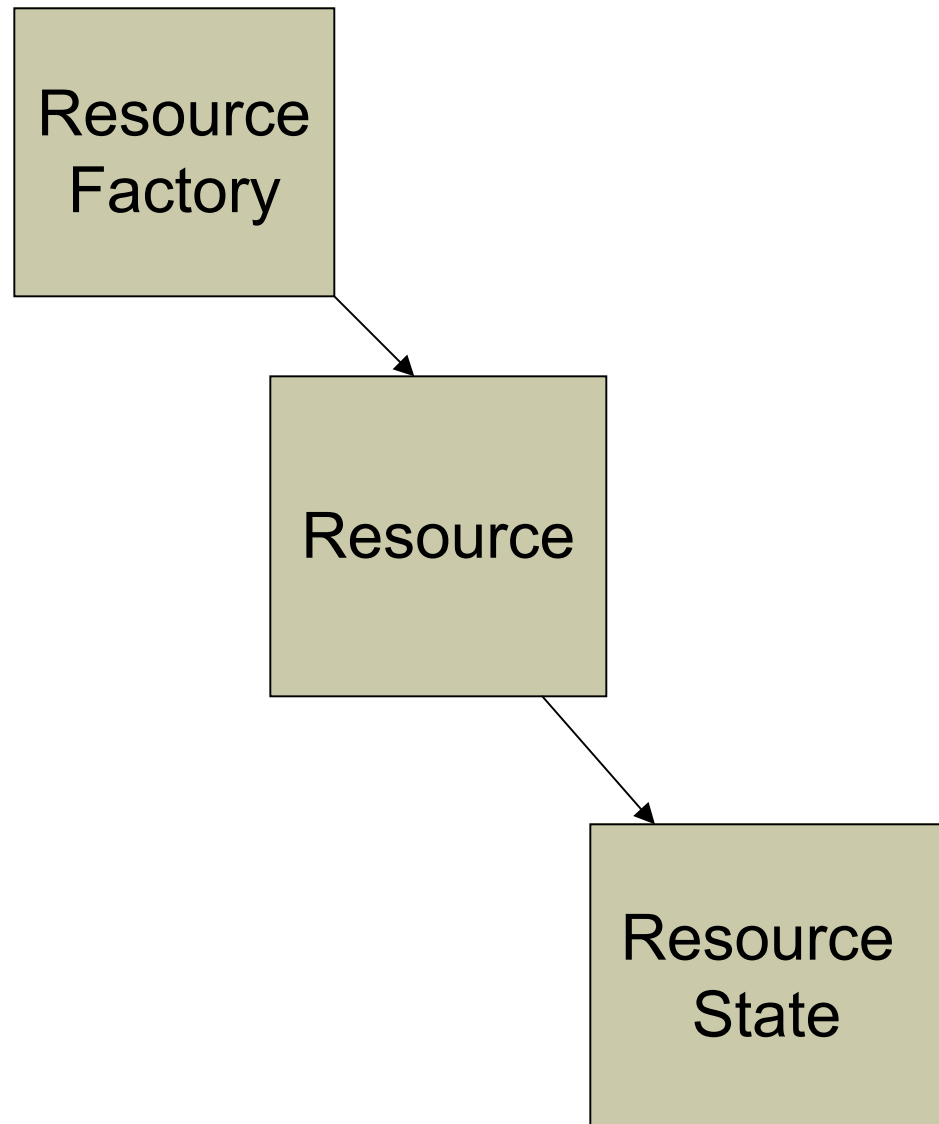- Provides an Enumeration of Available Event Sources

Client Access

- Client Library does not require JAXB
  - You can get the DOM of responses
  - Provides XPath based extraction of fields
  - You can use your own XML binding tools
- Currently supports
  - Transfer and Fragment Transfer
  - Enumeration and Filtering, custom dialects
  - OptionsSet Support
- Coming soon, Eventing and HTTP Listener Support without J2EE

Client Model

- A Factory creates or Finds existing resources
- Resources contain a selector set and ResourceUri
- Resources produce resource states
- Resource States provide property and XPath tools

Resource Factory

Resource

Resource State

How complete is Wiseman?

Addressing	Transfer	Enumeration	Eventing	Catalog
(Complete)	(Complete)	(Complete)	(Partial)	(None)
Message ID Generation Action Dispatching Selector Access OptionSets Time-outs Size Limits Locals	Get Put Create Delete Rename Custom Actions Fragments	Enumerate Pull Release Filtering Fragments Object and EPR Enumeration	Listener Creation Message Dispatching Subscription Persistence & Timeouts Filtering and Selector Support Connection Retry No Pull Support	**To Be Done**

JSR262: WS connector for JMX

- Defines a Web Services Connector for JMX

- Will be based on WS-Management
  - Recent decision taken by JSR 262 Expert Group
  - Motivated by WS-Management/WSDM convergence
  - http://weblogs.java.net/blog/emcmanus/archive/2006/11/jsr262_and_wsma.html

- Reference Implementation protocol layer will use Wiseman

- JSR 262 is a Java SE technology
  - Will require JAX-WS which is in J2SE 6
  - No Servlet Engine Required

We Need Your Help

- **Areas Under Development**
  - Eventing Framework
  - CIM Integration
    - CIM Mapping - Tool Exists but needs updating
    - WS-Man CIM Binding - Not implemented
  - JMX Integration (JSR262)

Where Can I Get It?

- The Wiseman project is at Java.net
  - https://wiseman.dev.java.net/
- Our most recent release is 0.5
  - https://wiseman.dev.java.net/files/documents/3436/40769/wiseman_0.5.zip

- Scenario
  - WS-Man used to deploy and monitor traffic lights placed on a map
  - Traffic lights controlled from a back end model
  - Using Swing-UI to display and place lights on screen
  - Traffic light resource generated from schema
  - A swing UI controls light placement and state (Color)

Summary

- Wiseman is open source toolkit for creating Java based WS-Man Web services
- Has client and server support
- Has tooling for WSDL and service generation from schema
- Has a binary release ready to download and use
- Needs your involvement!

Questions?