



**December 5-8, 2005, Santa Clara Marriott, Santa Clara, CA**

# **The Wiseman project: An open-source Java<sup>TM</sup> implementation of WS-Management**

**Akhil Arora**  
**Sun Microsystems, Inc.**



# Objectives

- To familiarize Java developers with the Wiseman project
- Be able to build, develop, deploy and test manager and agent applications using the Wiseman codebase
- Be able to customize and extend the codebase for your needs

# Agenda

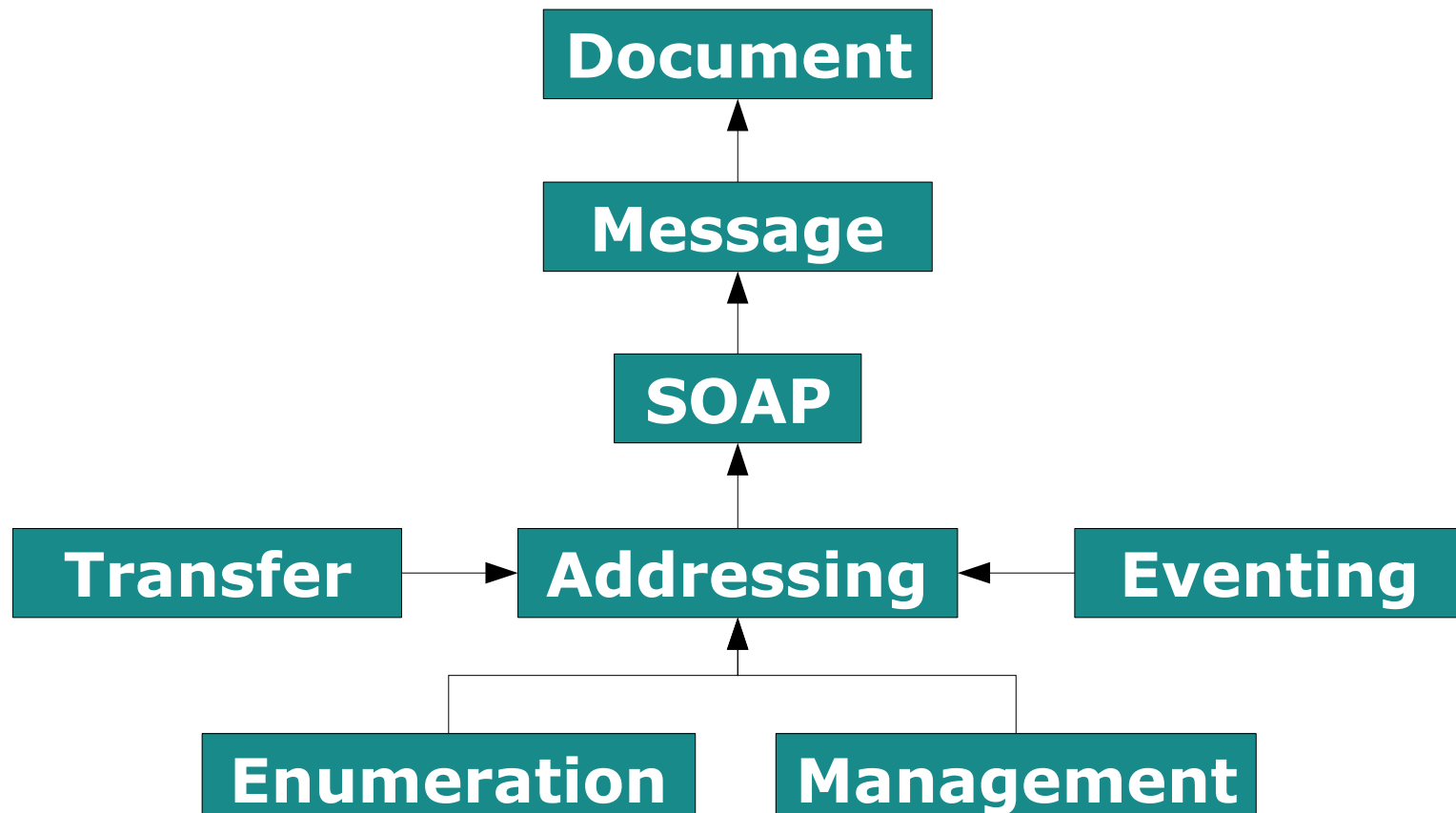
- Overview
- Architecture
- Sample code
  - Manager
  - Agent
- Futures

# The Wiseman project

- <https://wiseman.dev.java.net>
- Java SE 5+ based implementation of the WS-Management protocol, 2005/06 version and dependent specs
- Apache License, Version 2.0
- JUnit-based unit tests
- Prototype code

- Java SE 5+
  - XML
  - org.w3c.dom
    - Document, Element, Node, Attribute
- SOAP with Attachments API for Java (SAAJ 1.3)
  - Message, Envelope, Body, Header, Fault
- Java Architecture for XML Binding (JAXB 2.0)
  - XML ↔ Java based on Schema

# Architecture



# Build and Run

- checkout source from cvs repository
  - instructions on website
- satisfy dependencies in lib
- ant war
- deploy war
- enable http basic auth
- start app server
- ant test to run the unit tests

# Client setup

- Client Initialization

- supply credentials for http basic auth

`HttpClient.setAuthenticator(new MyAuthenticator())`

- add vendor packages to XmlBinding, if needed

`SOAP.setXmlBinding(new XmlBinding())`



# Clients – part 1

- Create a blank Message

```
msg = new Management()
```

- Add Addressing headers

```
msg.setAction(Transfer.GET_ACTION_URI)
msg.setReplyTo(Addressing.ANONYMOUS_ENDPOINT_URI)
msg.setMessageId("uuid:" + UUID.randomUUID())
msg.setTo("http://localhost:8080/wsman/");
msg.setResourceURI("wsman:system/2005/02/this");
```

- Add Management headers

```
msg.setTimeout(datatypeFactory.newDuration(30000))
```

# Clients – part 2

- **Setting max envelope size**

```
maxEnvSize = new MaxEnvelopeSizeType()  
maxEnvSize.setValue(4096)  
msg.setMaxEnvelopeSize(maxEnvSize)
```

- **Setting Selectors**

```
selectors = new HashMap()  
selectors.put("SystemName", "v20z-1")  
msg.setSelectors(selectors)
```

- **Setting Locale**

```
locale = new LocaleType()  
locale.setValue("en-US")  
msg.setLocale(locale)
```

## Clients – part 3

- Add stuff to body using the saaj/dom apis

```
body = msg.getBody()  
ns = body.addBodyElement(NUMERIC_SENSOR_QNAME)  
lt = ns.addChildElement(LOWER_THRESHOLD_QNAME)  
lt.addTextNode("100")
```

- Send the request and receive a response

```
response = HttpClient.sendRequest(msg)
```

- Check for and handle fault

```
if (response.getBody().hasFault()) {  
    handleFault(new SOAP(response).getFault())  
    return  
}
```

# Clients – part 4

- Extract information from response body
  - if response's action is enumerate response

```
Enumeration er = new Enumeration(response);  
PullResponse pr = er.getPullResponse();  
if (pr.getEndOfSequence() != null) {  
    return;  
}
```

```
Object context =  
    pr.getEnumerationContext().getContent().get(0);
```

# Request Dispatchers

- An extensible mechanism to dispatch requests to providers
- Default implementation uses reflection to dynamically load providers
- Executed in a thread-pool, may be cached
- Providers are located by transforming the ResourceURI to a handler class name

`"wsman:system/2005/02/this" ->`

`"wsman.system._2005._02._this_Handler"`

# Handlers

- Implement the Handler interface

```
void handle(String action, String resource,  
            Management request, Management response)  
    throws Exception
```
- Some response headers are pre-filled
- Any exception is mapped to a SOAP Fault and returned
- May be interrupted if OperationTimeout is exceeded

# Writing Handlers

- Check for supported Action

```
if (Transfer.GET_ACTION_URI.equals(action)) {  
    response.setAction(Transfer.GET_RESPONSE_URI)  
    ...  
} else {  
    throw new ActionNotSupportedFault(action)  
}
```

- Fill-in response body

```
doc = response.newDocument()  
element = doc.createElementNS(...)  
doc.appendChild(element)  
...  
response.getBody().addDocument(doc)
```

# Handler Support Classes

- EnumerationSupport
  - provides default implementations of Enumerate, Pull, Release
  - calls back to EnumerationIterator for data
- EventingSupport
  - provides subscription management and event delivery
- EventingExtensionsSupport
  - coming soon



# Still to do (help wanted!)

- WS-Management
  - Fragment-level transfer
  - Eventing extensions
  - Productization
- CIM Mapping
- WS-management Catalog
- Documentation
  - Javadoc, FAQ, Developer guide
- Unit tests
- Compliance test-suite

# Resources

- Resources

<https://wiseman.dev.java.net>

[dev@wiseman.dev.java.net](mailto:dev@wiseman.dev.java.net)

[users@wiseman.dev.java.net](mailto:users@wiseman.dev.java.net)

- Standards

<http://dmtf.org/apps/org/workgroup/wsman>

<http://dmtf.org/apps/org/workgroup/interop/ws-cim>

# Questions

# Thank You

