SKILLS MATTER

Making SOA Groovy

Paul Fremantle, pzf@apache.org

2007

GRAILS eXCHANGE

http://www.grails-exchange.com | http://www.grails.org | http://skillsmatter.com

Who am I?

- Paul Fremantle
  - Co-founder of WSO2 - open source SOA middleware company
  - Member of the Apache Software Foundation
  - Committer and Release Manager on Apache Synapse
  - Co-chair of the WSRX Technical Committee at OASIS
  - Previously a Senior Technical Staff Member at IBM

Making SOA Groovy

- Explain my simple views on SOA
- Introduce Apache Synapse
- Show how Synapse and Groovy work together
- Demonstrate
- Using Groovy to create a dead simple DSL
- Futures

Service Oriented Architecture

Building connected systems

Using structured formats
  {XML, JSON}

Distributed re-use

SOA infrastructure

- Main components
  - XML parsers
  - Web Service toolkits
  - JMS / Messaging
  - Enterprise Service Bus (ESB)
  - Registry/Repository/Store

SOA and Dynamic Languages

- SOA is about adding more *flexibility* and *dynamism* to your IT systems
- A flexible IT model needs different aspects with different lifecycles
- Dynamic languages add a more flexible, fast approach to integration

Groovy and SOA

- Groovy already has some nice SOA enablers
  - GroovyWS – Apache CXF binding into Groovy
  - Also some Axis2 integration
  - Great XML support
    - Slurper, StreamingMarkupBuilder
  - Simple HTTP support
  - plus access to everything Java can do ☺

Apache Synapse

- A smart router for SOA
  - Listens for messages
    - TCP, HTTP, JMS, SOAP, WSRM, WSSec
  - Does stuff
  - Sends messages on (or back)
- Can be clustered, distributed

http://ws.apache.org/synapse

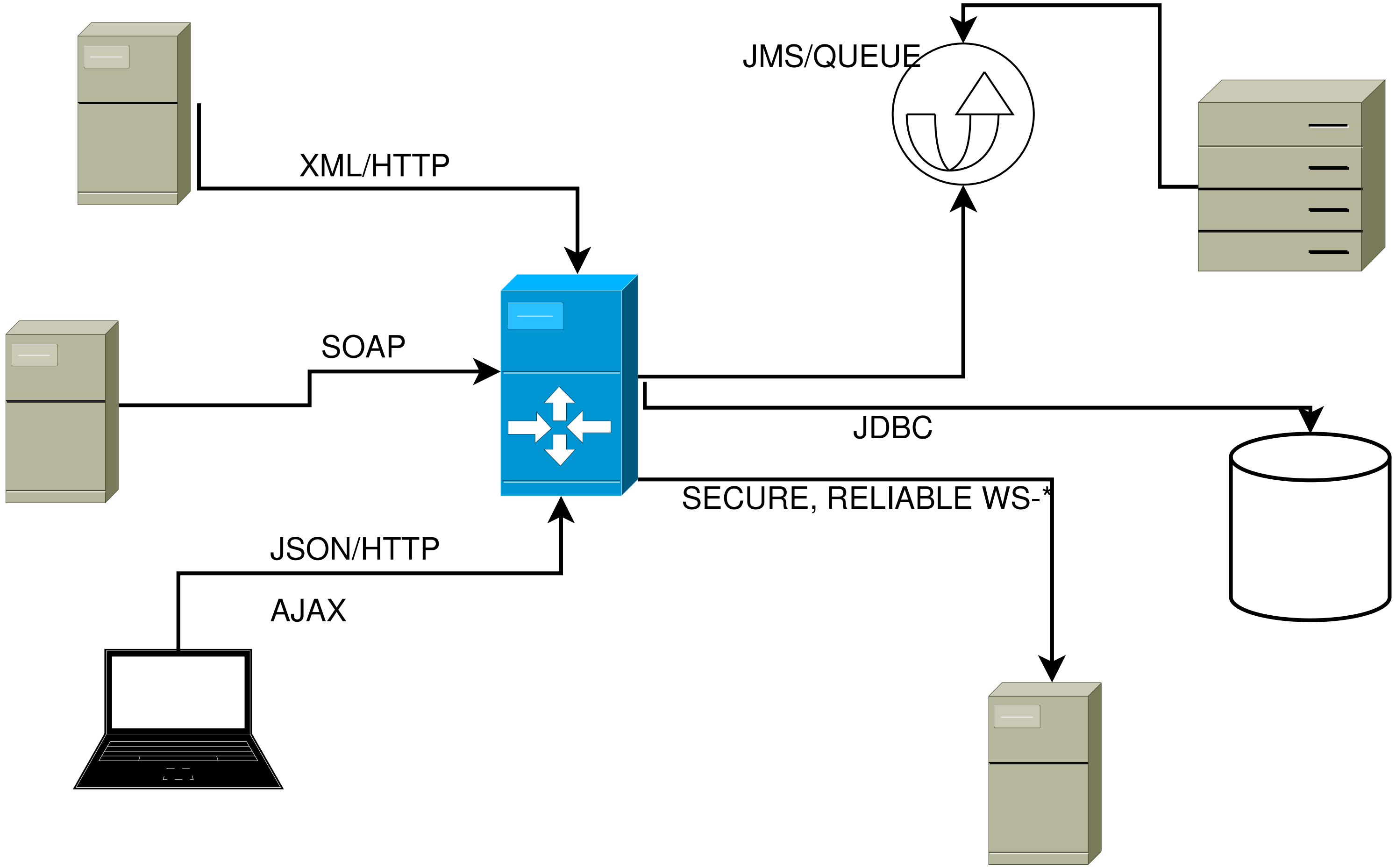
What is "Stuff"

- Logging
- Load-balance, throttle, failover
- XSLT, XQuery, XPath, CSV, JDBC, JSON
- Transport switch:
  - JMS<->SOAP, HTTP<->FTP
- Handle WS-* for you
  - WSRM, WS-A, WS-Sec
- Run regular tasks (Polling, File, FTP)

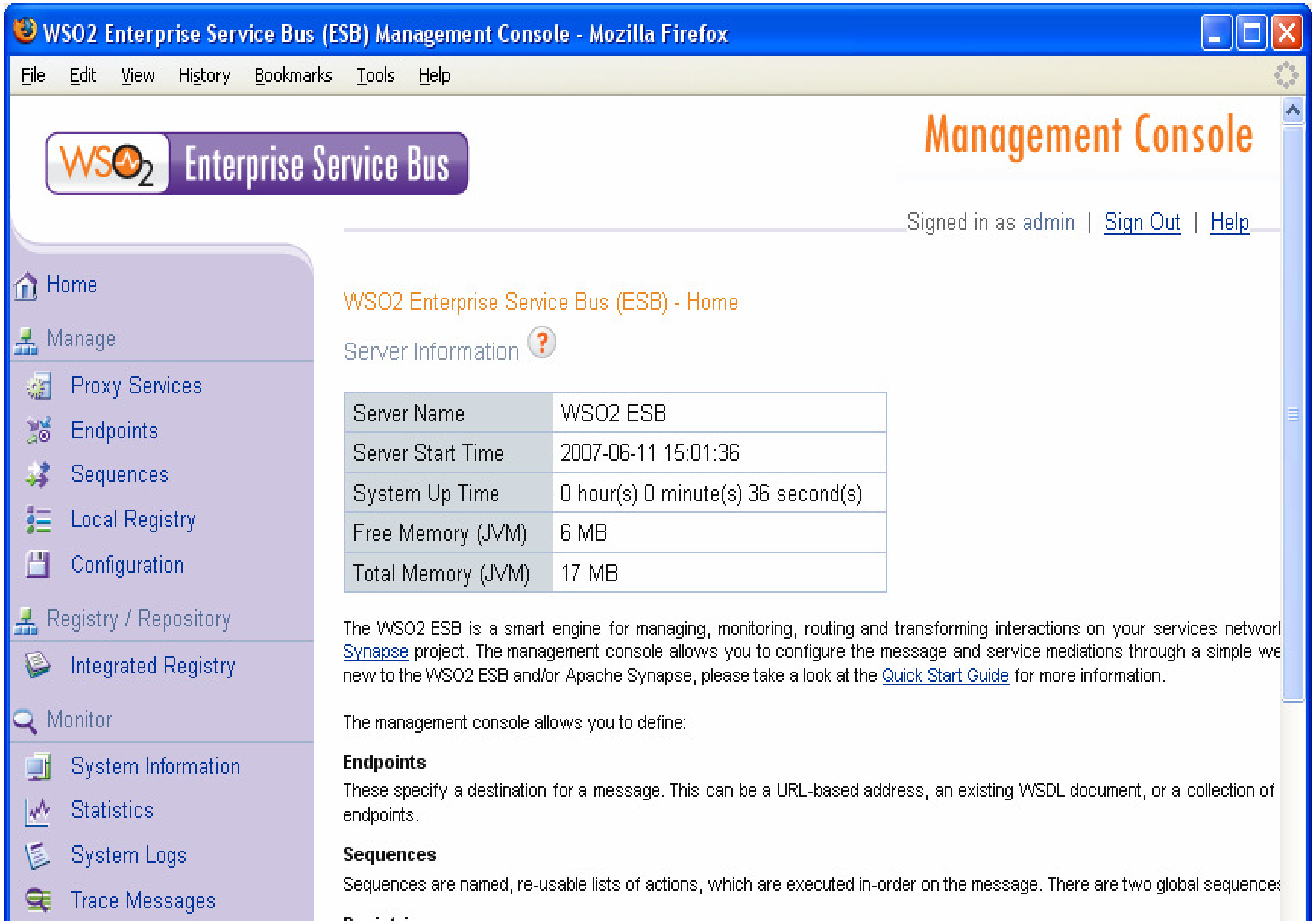
but there is more!

- Extensible using Dynamic Languages
  - JSR223/BSF
- Built-in support for
  - Groovy (of course)
  - JavaScript
  - JRuby

A picture


JMS/QUEUE
XML/HTTP
SOAP
JDBC
SECURE, RELIABLE WS-*
JSON/HTTP
AJAX

WSO2 ESB



WSO2 Enterprise Service Bus (ESB) - Home

Server Information

Server Name	WSO2 ESB
Server Start Time	2007-06-11 15:01:36
System Up Time	0 hour(s) 0 minute(s) 36 second(s)
Free Memory (JVM)	6 MB
Total Memory (JVM)	17 MB

The WSO2 ESB is a smart engine for managing, monitoring, routing and transforming interactions on your services networl Synapse project. The management console allows you to configure the message and service mediations through a simple we new to the WSO2 ESB and/or Apache Synapse, please take a look at the Quick Start Guide for more information.

The management console allows you to define:

Endpoints
These specify a destination for a message. This can be a URL-based address, an existing WSDL document, or a collection of endpoints.

Sequences
Sequences are named, re-usable lists of actions, which are executed in-order on the message. There are two global sequences

Intro to the Synapse Configuration Model

- A simple XML configuration that defines the processing behaviour of Synapse
  - Task
    - repeating events (interval or cron based)
  - Proxy
    - Listeners / virtual service endpoints
  - Property
    - Re-usable configuration items
  - Registry
    - Remote sources of configuration (e.g. SVN, UDDI)
  - Sequence
    - Ordered sets of processing behaviour
      - e.g. Log, then Cache, then Send

Example

```
<definitions
   xmlns="http://ws.apache.org/ns/synapse">
    <!-- log messages passing through -->
    <log level="full"/>

    <!-- Send the msg to implicit destination -->
    <send/>
</definitions>
```

Another example

```
<definitions xmlns="http://ws.apache.org/ns/synapse">
  <proxy name="StockQuoteProxy">
    <target>
      <endpoint>
        <address uri="http://localhost:9000/soap/SSQ"/>
      </endpoint>
      <outSequence>
       <send/>
      </outSequence>
    </target>
 </proxy>
</definitions>
```

Using Groovy in Synapse

<script language=“groovy”>

- Can either inline the script, load it from a file, or a registry
- Currently Groovy 1.0 but hope to update to 1.1-RC1 for the 1.1 release of Synapse

A simple Groovy example

```
<script language="groovy"><![CDATA[
  mc.setPayloadXML('''
       <hello xmlns="http://grails-exchange.com">
         <to>Everyone</to>
        <from>Paul</from>
       </hello>
]]></script>
```

Demo

Reading the XML

- XMLSlurper
  - Really simple tool for reading XML
  - Converts XML into a recursive map
  - Now you can navigate XML as Groovy properties

```
def bodyXML =  new
   XmlSlurper().parseText(mc.getPayloadXML())
bodyXML.a.b.c.
   findAll
   { it.@key.text().contains('cpu'); }
```

Creating XML

- StreamingMarkupBuilder - internal DSL for XML

```
import groovy.xml.StreamingMarkupBuilder
def xml = new StreamingMarkupBuilder().bind{
   tagname(attribute:"value") {
        child("value")
        child2(att:"val",att2:"val2") {
                another()
        }
   }
}
println xml
```

Output

```
<tagname attribute='value'>
  <child>value</child>
  <child2 att='val' att2='val2'>
      <another/>
  </child2>
</tagname>
```

Demo2 scenario intro

Ganglia

- Excellent scalable distributed monitoring system
- Simple to use
- Simple (but UGLY) XML output

```
<GANGLIA_XML VERSION="3.0.0" SOURCE="gmond">
  <CLUSTER NAME="unspecified" LOCALTIME="1192724982"
    OWNER="unspecified" LATLONG="unspecified"
    URL="unspecified">
  <HOST NAME="localhost" IP="127.0.0.1"
    REPORTED="1192724963" TN="19" TMAX="20" DMAX="0"
    LOCATION="unspecified" GMOND_STARTED="1192723129">
  <METRIC NAME="disk_total" VAL="0.000" TYPE="double"
    UNITS="GB" TN="1820" TMAX="1200" DMAX="0" SLOPE="both"
    SOURCE="gmond" />
```

Demo2 Scenario

- Ganglia
- Synapse
  - GMondPoller
- Groovy
  - XMLSlurper
  - StreamingMarkupBuilder
- Simple Atom proxy

DEMO

More about Synapse

- Synapse can be configured to be completely dynamic and distributed
  - Pull config from a remote source
    - HTTP “registry”
  - Cache
  - Reload as needed

Creating a simple DSL for Synapse

- We always had in mind that XML was only one way to configure
- You can also use
  - Java
  - Spring
- Looking at other models

The easiest approach

- Just use StreamingMarkupBuilder

my first synapse dsl

```
definitions {
  task(class:"MessageInjector", name:"messageInjector1") {
    trigger(interval:"5000")
    property(name:"message") {
      xmlmessage { mydata() }
    }
    property(name:"to",value:"urn:paul")
  }
  "in" {
    log(level:"full")
    send()
  }
  out {
    send()
  }
}
```

Twangle.groovy

```
def prefix = '''
import groovy.xml.StreamingMarkupBuilder;
def xml = new StreamingMarkupBuilder().bind{
  mkp.declareNamespace(s:'http://ws.apache.org/ns/synapse');
'''
def suffix = '''
}
println xml.toString();

'''
StringWriter writer = new StringWriter();
writer.write(prefix as String);
new File(args[0]).eachLine { line -> writer.write(line+"\n") }
writer.write(suffix as String);
def nw = writer.toString();
Eval.me(nw);
```

Futures for Synapse DSL

- Actually design it!
  - <> design is not the same as {}
  - Better error reporting!
  - Built into Synapse not an external tool

Futures for Groovy & ApacheWS

aka whats on the TODO list

- More Groovy examples
- Upgrade to 1.1
- Better XML integration
  - We did this for JS with a 5x improvement
- Better Axis2 integration
- WSO2 Mashup Server

More information

- Me
  - pzf@apache.org
  - http://pzf.fremantle.org
- Synapse Website
  - http://ws.apache.org/synapse
  - See sample 500 onwards
- Apache Synapse Blog
  - http://apache-synapse.blogspot.com/
- Ganglia
  - http://ganglia.sourceforge.net/
- XMLSlurper/StreamingMarkupBuilder
  - http://groovy.codehaus.org/Processing+XML

Thanks for listening
Any questions?