



# JavaOne<sup>SM</sup>

## Kick-Start Your SOA with Open-Source Tools

Aaron Mulder  
Chariot Solutions

## About the Speaker

- > Consultant
- > Committer on several of the projects we'll discuss
  - Camel, ServiceMix, ActiveMQ, etc.
- > Get to see both the product development side and the customer usage side of the picture

# Agenda

- > Why open source SOA/integration?
- > Products and Categories
- > Scenarios from consulting projects
- > Discussion (your scenarios here)

## Why Open Source SOA/Integration?

## Why Integration?

- > Umm, because you need to?
- > We'll look at different styles of integrating external applications, business partners, etc.
- > Also of use when you want different parts of the same app to talk in some standard way
  - e.g. Full-featured Flex front end and limited JavaScript front end for users without Flash plugin
  - Both talk to a REST back end

## Why SOA?

- > If you have to ask...
- > Generally comes down to reuse
  - lower cost, higher agility, etc.
- > Often limited by organizational issues more than technical issues
- > But if you're going there, open source is a good way to get off the ground

## Why Open Source?

- > It's the economy, stupid!
- > And, easier to use a decent development process
  - No vendor-specific tools
  - Easier for testing and CI
  - Less opaque generated code
- > Easier to debug (with the source)
- > Easier to get started
  - no licensing issues, community support, etc.

## Products and Categories

## XML Binding

- > Maps XML to POJOs and back
  - Used for dealing with XML content conveniently
- > Numerous options, all open source
  - JAXB, XmlBeans, JiBX, Castor, JaxMe...
- > Used to matter a lot -- JAXB 1 kind of stank and there were comparison sessions at JavaOne at all
- > Now JAXB 2 is often good enough
  - Only need to look further if it doesn't work out

## XML Binding, con't

### > Two main approaches

- Can start from a schema (XSD) and generate Java code
- Can start from POJOs and use JAXB annotations to control their XML form

### > Depends on your overall approach (XML or Java first)

- Probably depends on whether the endpoint is for outside consumption or not

## SOAP & REST

- > Two styles of XML integration
  - Won't really cover the pros and cons here
- > Several open source products available
  - Classic SOAP and now some REST: Axis2
  - JAX-WS and JAX-RS REST: CXF
  - JAX-RS REST: Jersey, RESTEasy, ...
- > Typically use JAXB 2 or friends to handle XML mapping

## SOAP & REST, con't

- > Multiple usage models
  - Generate code from WSDL/Schema
  - Driven by annotated POJOs/EJBs
  - Generate clients
- > Multiple deployment models
  - Included in a Web app
  - Included in an ESB
  - Standalone (e.g. for testing)

# Messaging

- > Lots of commercial products
- > Open source options
  - ActiveMQ, JBoss Messaging
- > Every messaging scenario is different
  - Topics vs queues, persistence or no, can miss or duplicate messages or no, number of publishers vs. subscribers vs. destinations, size of messages, ...
  - Makes it hard to give general advice

## Messaging, con't

- > Certainly worth starting with open source options
  - Still easy to configure, test, embed, etc.
  - Easy to use with other products discussed here
- > Test out the limiting factors in your case
  - Message load, 10,000's of destinations, fail-over, etc.
  - Plenty of tuning to be done
  - But if it doesn't work out, look at commercial products

## “Reusable Integration Logic”

- > First Apache Camel, now Spring Integration too
- > Perhaps the easiest way to get started on an integration project
- > Can set up in & out endpoints
  - SOAP/REST, file polling, messaging, Bean calls, Velocity templates, etc.
- > Then add data transformation, routing, filtering, ...
- > All driven by simple XML, DSL, and/or annotations

## Camel & Spring Integration, con't

- > Great Spring integration, very easy to mock/test
- > Not as formal/robust a “bus” or component model as an ESB
- > Will have to think about crash recovery, deployment, monitoring
- > But lightweight and dead simple to get going
  - I would certainly consider starting here before going to an ESB

## BPEL

- > The commercial standard for drag & drop business processes
- > Now with open source options including ODE, ActiveBPEL
- > Don't see this too much in practice
  - Easier to write and test POJOs

## ESB

- > The backbone of many an SOA
  - Allows hub & spoke integration instead of proliferation of point to point integration solutions
- > Both commercial and open source options
  - Mule, ServiceMix, OpenESB, JBoss ESB
  - Technology varies (JBI, Spring, etc.)
- > We typically start with open source products and normally don't need to go any further

## ESB, con't

- > Often use the other stuff under the covers
  - Camel endpoints
  - Axis/CXF endpoints
  - ActiveMQ messaging, in some cases even for messages on the bus
  - BPEL if you like
- > Ideally with good management, security, deployment models for routes/services/processes

## Governance

- > For if/when you graduate to managing a complex SOA
  - Not typically so useful for integration needs
- > Mule Galaxy the main open source option
- > Work out the policies and methodologies first, then look for a tool that's a good fit

## Scenarios from Consulting Projects

## SOA/e-Commerce

- > Company had multiple “store” platforms, in Java, Perl, etc.
- > Needed to implement new privacy/security regulations
  - Didn't want to do it 3 times
- > Used separate vendors (payment processing, etc.)
- > Went with Mule ESB, JAXB 2 for inbound XML requests, Axis for outgoing SOAP requests

## Telco

- > Had multiple conceptual “services”
  - Conference calls, e-mail/pager alerts, org charts, ...
- > Wanted to build business processes on them
  - When notified of a big problem, set up a conference call, notify participants and/or their assistants, ...
- > Wanted to go with the JBI standard
- > Prototyped with ServiceMix ESB, ODE BPEL, and a variety of endpoints to talk to different services

## Business Partners

- > Company supplied widgets
- > Their customers entered widget needs in their own (ERP) systems
- > Wanted to avoid duplicate entry on both ends
- > Every customer used a different electronic format/protocol (Excel/CSV/XML on e-mail, FTP, HTTP)
- > Used Camel to set up endpoints, transform data to canonical format, route to/from internal systems, notify customers

## e-Commerce 2

- > Complex home-grown internal systems
  - Web Store, inventory, etc.
- > Web stores cache all product data in memory
- > Needed to add real-time availability updates (for those hot Xmas items)
- > Needed to poke all Web Stores when a product's status changed
- > Tried JMX and some other not-so-good solutions
- > Went with clustered ActiveMQ messaging

## SOA/e-Commerce 2.5

- > Later rebuilding some of the internal services using Spring and other more “standard” technologies
- > Building as REST services, schema-first
  - Using Jersey and JAX-RS to build services
- > Using Mule to manage/expose services
- > Right now just a handful of services... No real business processes, governance, etc.
  - May never go that way

## SOA 2

- > Magazine publisher managing Web sites for many publications
- > Historically run separately, but they all want updates (community content, etc.)
- > Strategic decision to adopt SOA
  - Reflexively bought a commercial ESB, built services there (not so easy to develop, test, etc.)
- > Tried Axis and CXF clients for Web apps invoking the services (driven by what worked in app server)

## Content Management

- > REST back end for Flex and other front ends, plus other systems as clients
- > CXF and JAX-RS for developing and testing services
- > Somewhat loose XML (driven by Java not XML)
  - But all consumers are fairly tightly tied so not such an issue
- > Actually getting a lot of mileage out of Groovy for XML processing on some clients

## Q&A / Discussion

## Your Scenarios

- > I'd be interested to hear about your SOA/integration scenarios
- > We can talk about which of these products might be most appropriate



# JavaOne<sup>SM</sup>

# Thank You

Aaron Mulder

[ammulder@chariotsolutions.com](mailto:ammulder@chariotsolutions.com)

<http://chariotsolutions.com/presentations>

