SOA Today with

envoisolutions

Agenda

- SOA defined
- Introduction to XFire
- A JSR 181 Service
- Other “stuff”
- Questions

Service Oriented

1. *to orient around services*
2. *buzzword*
3. *...*

Service oriented is NOT (but can be)

NEW

XML

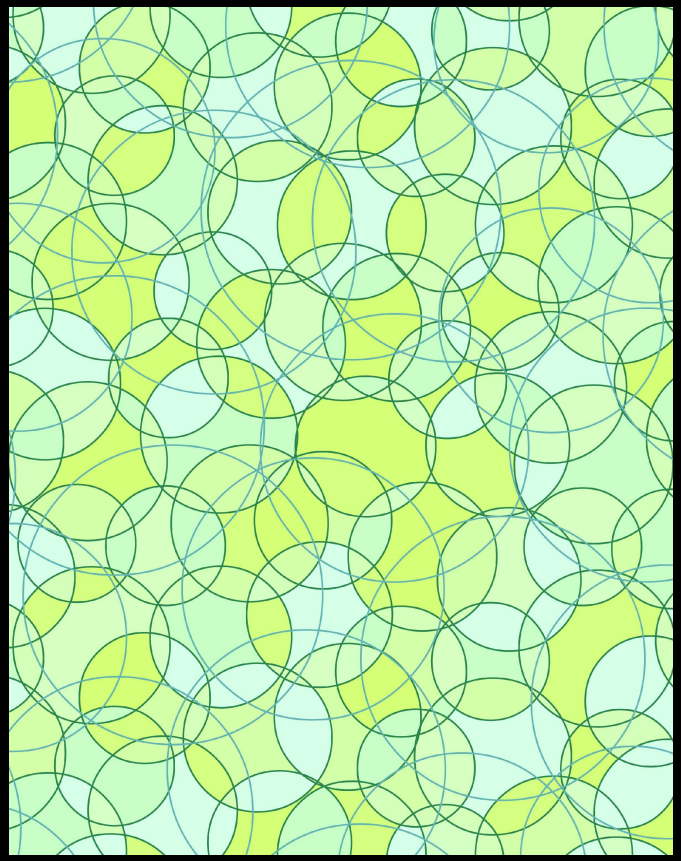
Web Services

ESBs

[insert cool product here]

Service oriented is an
architectual style

Loosely Coupled



NOT

Highly Interoperable

Built on standards Crosses boundaries

Service oriented is
business oriented

XFire

- Web Services for Java
- Standards based: XML, SOAP, WSDL, WS-*
- Open Source (MIT license)
- Easy to use:
  - JSR181
  - Spring support
  - Code->WSDL
  - WSDL->code

Architecture

- Streaming xml (StAX)
- Pluggable bindings – POJOs, JAXB, XMLBeans, Castor, Custom types
- Multiple transports – HTTP, XMPP, JMS, In-JVM transports
  - (and write your own easily via our Transport API!)
- Embeddable

Performance

- Traditional Web Services

- Requires building two object representations
  - Document Object Model
  - Business domain objects

XFire Performance

- Streaming Model

- Builds domain objects directly from the XML stream
- Low latency and high throughput model
- *2-5+x the throughput of Axis and 1/2-1/5 the latency*

XFire and JSR 181

- Easy code first deployment
- Model service via annotations
- Java 5 and Java 1.4 through commons-attributes or Retroweaver

The OrderService

```
@WebService
public interface OrderService
{
…
}
```

Our Operations

```
@WebMethod
public Collection<Order> getOrders();

@WebMethod
public long addOrder(Order o);

@WebMethod
public void removeOrder(long id);
```

Our Operations Take 2

```
@WebMethod
@WebParam(name="OrderId")
public long addOrder(
   @WebParam(name="Order") Order o);
```

Service Implementation

```
@WebService(
  endpointInterface="OrderService",
  serviceName="OrderService")
public class OrderServiceImpl
{
  // implement operations here
}
```

Configure the Service

- *A ServiceFactory* creates a *Service*
- A *Service* is registered in the *ServiceRegistry*
- *XFireHttpServer* creates a Jetty instance configured for XFire
- An XFireProxyFactory creates a proxied Client

Create the Service

```
XFire xfire = XFireFactory.newInstance().getXFire();

AnnotationServiceFactory factory =
   new AnnotationServiceFactory(
      new Jsr181WebAnnotations(),
      xfire.getTransportManager());

Service service = factory.create(OrderServiceImpl.class);
xfire.getServiceRegistry().register(service);

XFireHttpServer server = new XFireHttpServer();
server.start();
```

Easy client

```
XFireProxyFactory xpf = new
   XFireProxyFactory();
OrderService service = (OrderService)
   xpf.create(model, OrderService.class);


service.getCustomers();
```

The services.xml version

```
<beans>
 <service>
   <serviceFactory>jsr181</serviceFactory>
   <serviceClass>OrderServiceImpl</serviceClass>
 </service>
</beans>
```

A “pure” Spring version

```
<import resource=“/org/codehaus/xfire/spring/xfire.xml”/>

<bean id=“orderService” class=“org.codehaus.xfire.spring.ServiceBean”>
  <property name=“serviceClass” value=“com.acme.OrderServiceImpl”/>
  <property name=“serviceFactory” ref=“jsr181”/>
</bean>

<bean id="webAnnotations"
    class="org.codehaus.xfire.annotations.jsr181.Jsr181WebAnnotations"/>

<bean id=“jsr181 class=“org.codehaus.xfire.anotations.AnnotationServiceFactory”>
  <constructor-arg ref=“webAnnotations”>
  <constructor-arg ref=“xfire.transportManager”/>
</bean>
```

A “pure” Spring version

```
<import resource=“/org/codehaus/xfire/spring/xfire.xml”/>

<bean id=“orderService” class=“org.codehaus.xfire.spring.ServiceBean”>
  <property name=“serviceClass” value=“com.acme.OrderServiceImpl”/>
  <property name=“serviceFactory” ref=“jsr181”/>
</bean>

<bean id="webAnnotations"
    class="org.codehaus.xfire.annotations.jsr181.Jsr181WebAnnotations"/>

<bean id=“jsr181 class=“org.codehaus.xfire.anotations.AnnotationServiceFactory”>
  <constructor-arg ref=“webAnnotations”>
  <constructor-arg ref=“xfire.transportManager”/>
</bean>
```

A “pure” Spring version

```
<import resource=“/org/codehaus/xfire/spring/xfire.xml”/>

<bean id=“orderService” class=“org.codehaus.xfire.spring.ServiceBean”>
  <property name=“serviceClass” value=“com.acme.OrderServiceImpl”/>
  <property name=“serviceFactory” ref=“jsr181”/>
</bean>

<bean id="webAnnotations"
    class="org.codehaus.xfire.annotations.jsr181.Jsr181WebAnnotations"/>

<bean id=“jsr181 class=“org.codehaus.xfire.anotations.AnnotationServiceFactory”>
  <constructor-arg ref=“webAnnotations”>
  <constructor-arg ref=“xfire.transportManager”/>
</bean>
```

A “pure” Spring version

```
<import resource=“/org/codehaus/xfire/spring/xfire.xml”/>

<bean id=“orderService” class=“org.codehaus.xfire.spring.ServiceBean”>
  <property name=“serviceClass” value=“com.acme.OrderServiceImpl”/>
  <property name=“serviceFactory” ref=“jsr181”/>
</bean>

<bean id="webAnnotations"
    class="org.codehaus.xfire.annotations.jsr181.Jsr181WebAnnotations"/>

<bean id=“jsr181 class=“org.codehaus.xfire.anotations.AnnotationServiceFactory”>
  <constructor-arg ref=“webAnnotations”>
  <constructor-arg ref=“xfire.transportManager”/>
</bean>
```

Testing

- Test against your target platforms
  - .NET, PHP, Ruby, Perl, [fancy lang here]
- Record messages and play back
- Use AbstractXFireAegisTest
  - invokeService(serviceName, document)

Security

- Authentication – HTTP, WS-Security, Custom headers
- Authorization – Acegi, Custom code
- Integrity – WS-Security
- Privacy – HTTPS, JMS, WS-Security
- Avoid WS-Security whenever possible. Its dead slow.

Custom authentication

```
public void addOrder(
  @WebParam(header=true) AuthToken token,
  @WebParam Order order);

public class AuthToken {
  String username;
  String password;
  ….
}
```

SOA with XFire perspective

- Loosely Coupled – OrderService is independent of others
- Highly interopable – All you need is the WSDL
- Business oriented
  - Shares data across business boundaries

Where to now

- Use XFire Mapping files to control WSDL/XSD
- Do Schema/WSDL first with JAXB, XMLBeans or Castor for more control
- Hook up services to JMS or ServiceMix or Mule
- Write more services!
- Investigate WS-* needs

Questions? More Information:

- Web site: http://xfire.codehaus.org
- Mailing Lists: http://xfire.codehaus.org/Support
- My Blog: http://netzooid.com/blog
- Commercial Support: http://envoisolutions.com
- Send email to dan@netzooid.com for performance stats

envoisolutions