Developing Web Applications with Spring and Laszlo

Matt Raible
mraible@virtuas.com

VIRTUAS

Introductions

- Experience with Flash, Flex or OpenLaszlo?
- Experience with Web Services, Axis or XFire?
- Experience with Acegi Security, Spring or Ajax?
- What do you hope to learn today?

VIRTUAS

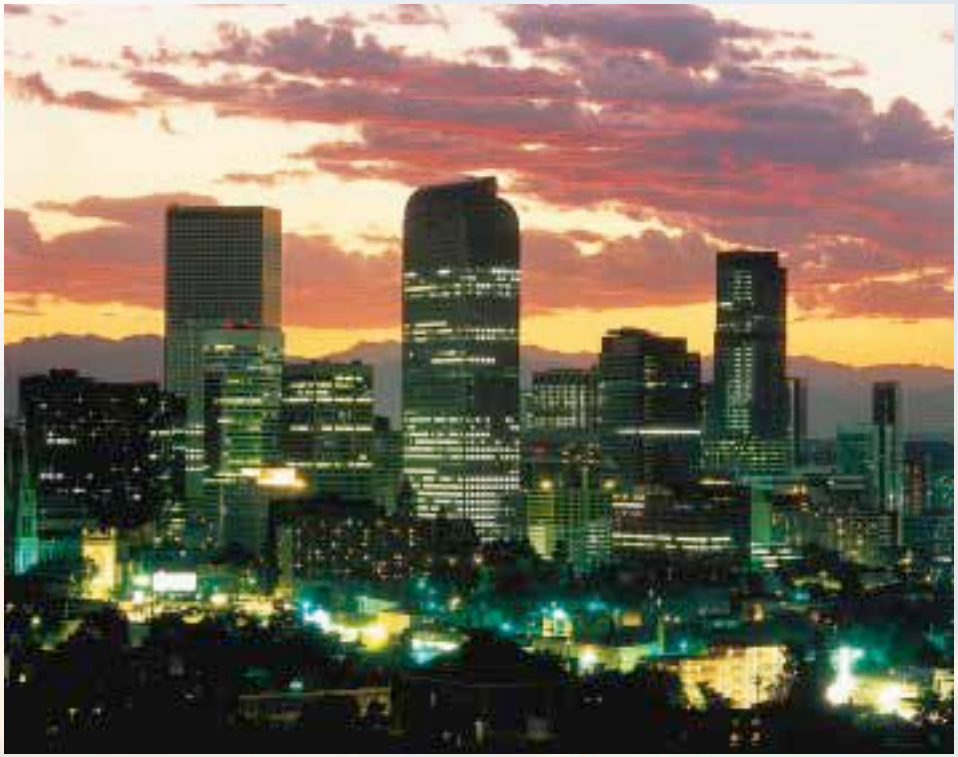
Matt

VIRTUAS

Raible

VIRTUAS

VIRTUAS



VIRTUAS

VIRTUAS

VIRTUAS

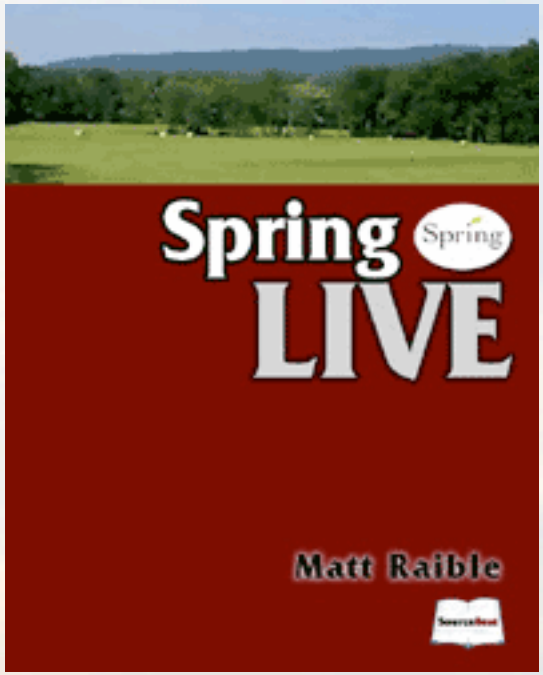
VIRTUAS

VIRTUAS

VIRTUAS

VIRTUAS

VIRTUAS



VIRTUAS

VIRTUAS

AppFuse

Roller

Struts Menu

VIRTUAS

VIRTUAS

Spring Remoting

- Supports many different types of remoting:
  - EJB
  - RMI (traditional and invoker)
  - Hessian and Burlap
  - HTTP Invoker
  - WSDL/SOAP via JAX-RPC

VIRTUAS

Configuration

- For all remoting solutions:
  - **Proxy factory**: service interface, URL of service, used to import remote server into client
  - **Proxy behavior**: proxy for remote service, delegates method invocations to remote service
  - **Exceptions**: propagated from remote service to proxy, generic ***RemoteAccessException*** for all remoting strategies

VIRTUAS

Exporting Services via HTTP

```
<servlet>
    <servlet-name>remoting</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
    <servlet-name>remoting</servlet-name>
    <url-pattern>/remoting/*</url-pattern>
</servlet-mapping>
```

Server: HttpInvoker

```
<bean id="userManager" parent="txProxyTemplate">
    <property name="target">
        <bean class="org.appfuse.service.impl.UserManagerImpl">
            <property name="userDAO" ref="userDAO"/>
        </bean>
    </property>
</bean>

...

<bean name="/UserService"
    class="org.springframework.remoting.httpinvoker.HttpInvokerServiceExporter">
    <property name="service" ref="userManager"/>
    <property name="serviceInterface" value="org.appfuse.service.UserManager"/>
</bean>
```

Client: HttpInvoker

```
<bean id="httpInvokerProxy"
    class="org.springframework.remoting.httpinvoker.HttpInvokerProxyFactoryBean">
    <property name="serviceUrl" value="http://remotehost/remoting/UserService"/>
    <property name="serviceInterface" value="org.appfuse.service.UserManager"/>
</bean>
```

```
<bean id="httpInvokerProxy"
    class="org.springframework.remoting.httpinvoker.HttpInvokerProxyFactoryBean">
    <property name="serviceUrl" value="http://remotehost/remoting/UserService"/>
    <property name="serviceInterface" value="org.appfuse.service.UserManager"/>
    <property name="httpInvokerRequestExecutor">
        <bean class="org.springframework.remoting.httpinvoker.\
                     CommonsHttpInvokerRequestExecutor"/>
    </property>
</bean>
```

Exposing Services: JAX-RPC

```
public class UserServiceJaxRpc extends ServletEndpointSupport implements UserService {
    private UserManager userManager;

    protected void onInit() {
        this.userManager = (UserManager) getWebApplicationContext().getBean("userManager");
    }

    public User getUser(String username) {
        return userManager.getUser(username);
    }

    public User[] getUsers() {
        List list = userManager.getUsers();
        User[] users = new User[list.size()];
        for (int i=0; i < list.size(); i++) {
            users[i] = (User) list.get(i);
        }
        return users;
    }

    public void removeUser(String username) {
        userManager.removeUser(username);
    }
```

server-config.wsdd

```
<?xml version="1.0" encoding="UTF-8"?>
<deployment xmlns="http://xml.apache.org/axis/wsdd/"
    xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
    ...
    <service name="UserService" provider="java:RPC">
        <parameter name="allowedMethods" value="*"/>
        <parameter name="className" value="org.appfuse.service.UserService"/>
        <beanMapping qname="appfuse:User" xmlns:appfuse="urn:AppFuse"
            languageSpecificType="java:org.appfuse.model.User"/>
    </service>
    ...
</deployment>
```


VIRTUAS

Client: JAX-RPC

```
<bean id="userWebService" class="org.springframework.remoting.jaxrpc.JaxRpcPortProxyFactoryBean">
    <property name="serviceFactoryClass" value="org.apache.axis.client.ServiceFactory"/>
    <property name="serviceInterface" value="org.appfuse.webservice.UserService"/>
    <property name="wsdlDocumentUrl"
              value="http://localhost:8080/equinox-laszlo/services/UserService?wsdl"/>
    <property name="namespaceUri"
              value="http://localhost:8080/equinox-laszlo/services/UserService"/>
    <property name="serviceName" value="UserServiceJaxRpcService"/>
    <property name="portName" value="UserService"/>
    <property name="servicePostProcessors">
        <list>
            <bean class="org.appfuse.webservice.BeanMappingServicePostProcessor"/>
        </list>
    </property>
</bean>
```

```
private UserService userService;

public void setUserWebService(UserService userService) {
    this.userService = userService;
}
```

VIRTUAS

Register Bean Mappings

```
public class BeanMappingServicePostProcessor implements JaxRpcServicePostProcessor {

    public void postProcessJaxRpcService(Service service) {
        TypeMappingRegistry registry = service.getTypeMappingRegistry();
        TypeMapping mapping = registry.createTypeMapping();
        registerBeanMapping(mapping, User.class, "User");
        registry.register("http://schemas.xmlsoap.org/soap/encoding/", mapping);
    }

    public void registerBeanMapping(TypeMapping mapping, Class type, String name) {
        QName qName = new QName("urn:AppFuse", name);
        mapping.register(type, qName,
            new BeanSerializerFactory(type, qName),
            new BeanDeserializerFactory(type, qName));
    }
}
```

XFire

- Support for important Web Service standards - SOAP, WSDL, WS-I Basic Profile, WS-Addressing, WS-Security, etc.
- High performance SOAP Stack
- Pluggable bindings POJOs, XMLBeans, JAXB 1.1, and JAXB 2.0
- Embeddable and Intuitive API
- **Spring**, Pico, Plexus, and Loom support

VIRTUAS

XFire Service

```
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:org/codehaus/xfire/spring/xfire.xml</param-value>
</context-param>

...

<bean id="echo" class="org.codehaus.xfire.spring.remoting.XFireExporter">
    <property name="serviceFactory" ref="xfire.serviceFactory"/>
    <property name="xfire" ref="xfire"/>
    <property name="service" ref="userManager"/>
    <property name="serviceInterface" value="org.appfuse.service.UserManager"/>
</bean>
```

Which remoting solution works best?

VIRTUAS

OpenLaszlo

- An open-source platform for development and delivery of rich internet applications
- Common Public License (CPL 1.0)
- 2000-2001: Prototyping and Development
- 2002-2003: LPS Preview Release, LPS 1.0/1.1
- October 2004: LPS 2.0, LPS open sourced
- 2005: LPS -> OpenLaszlo, 3.0/3.1 Released

VIRTUAS

OpenLazslo Basics

- Authored in XML, JavaScript and XPath
- Basic Elements:
  - <canvas>
  - <view>
  - <text>
  - <button>
- Demo:
  - http://openlaszlo.org/img/laszlo-basics.mov

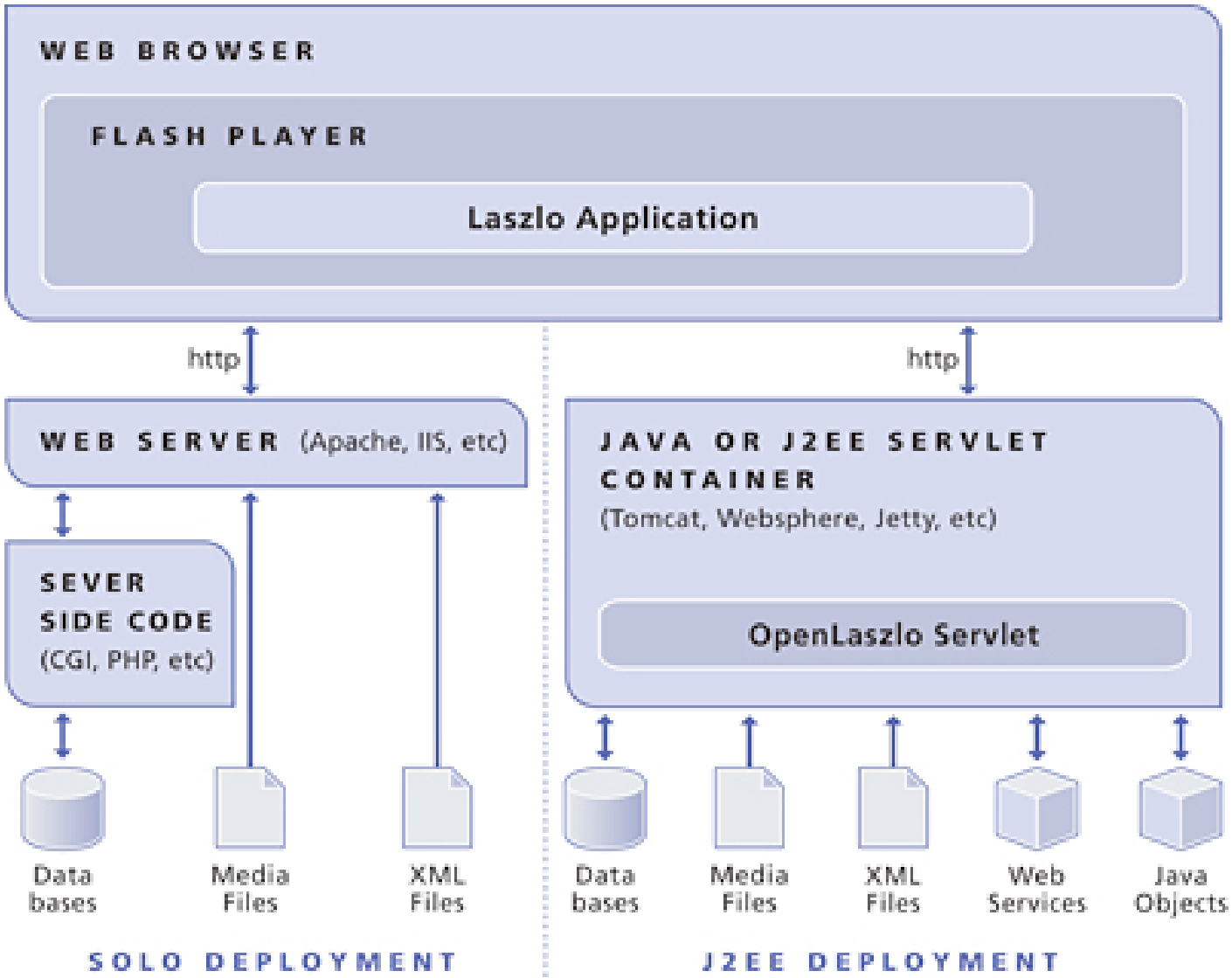
VIRTUAS

How do you code it?

- LZX is the language - XML similar to XHTML
- JavaScript for event handling
- XPath for data binding
- Authored using a text/XML editor or Eclipse plugin

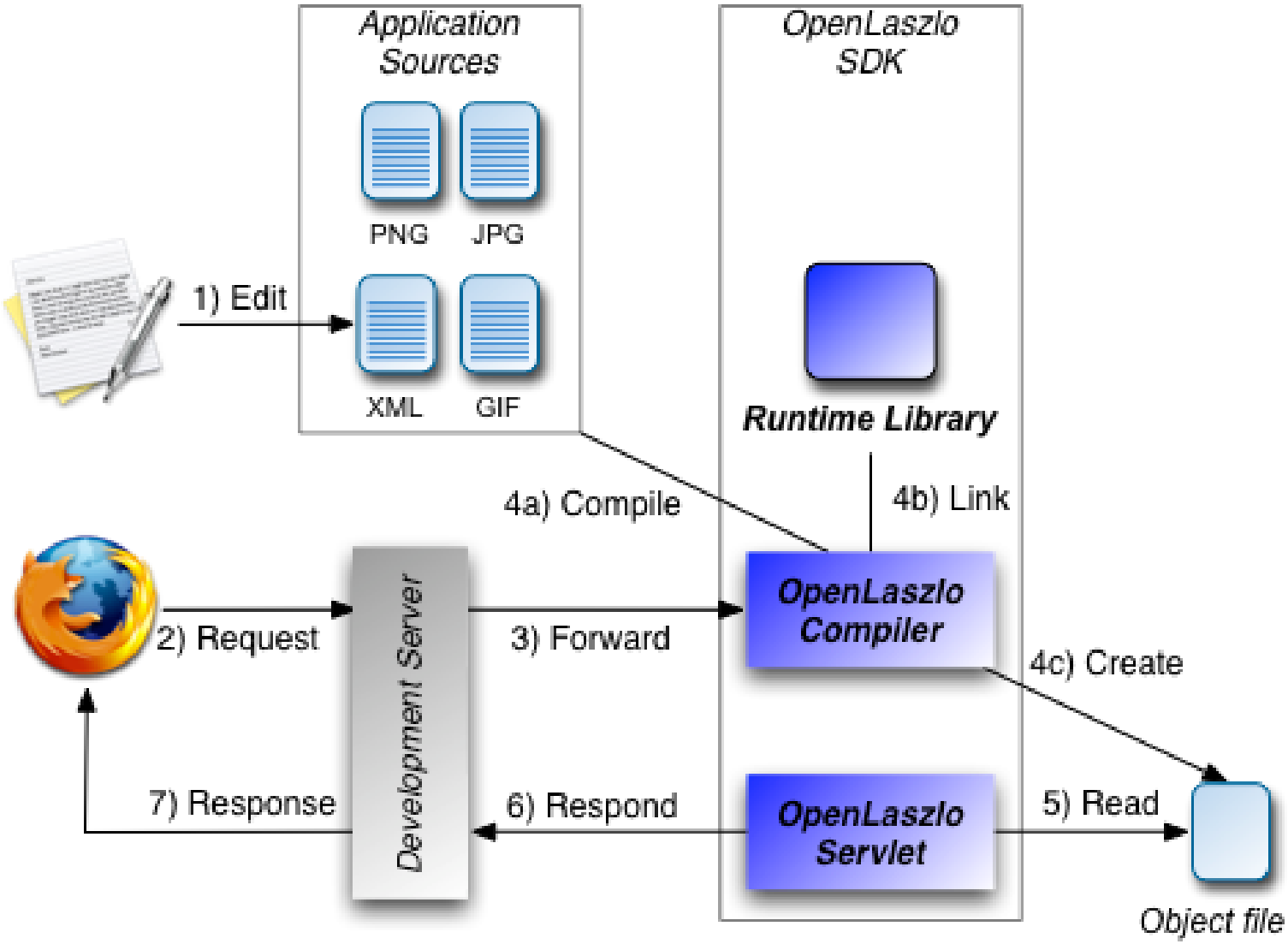
VIRTUAS

OpenLaszlo Architecture



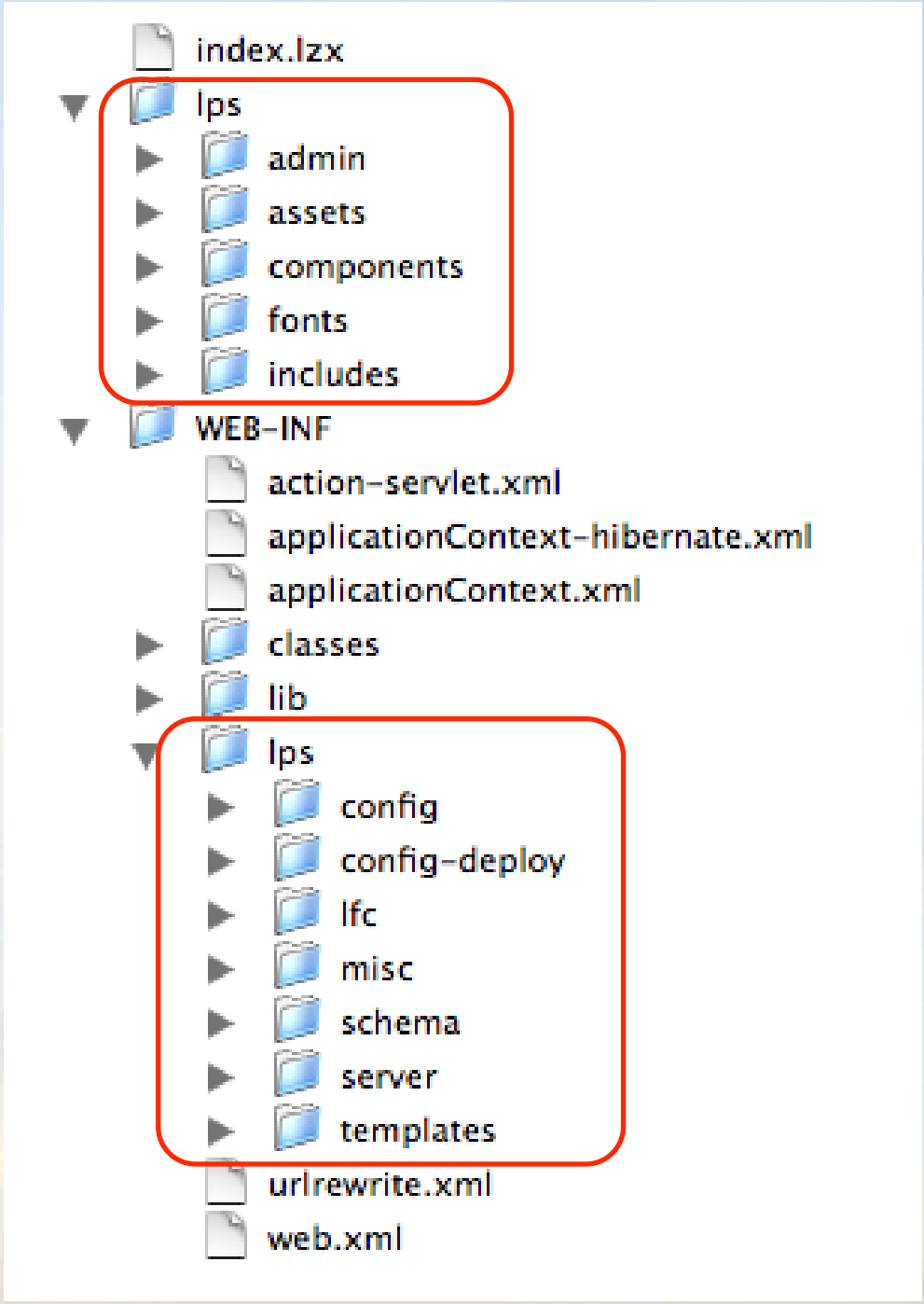
VIRTUAS

Development Cycle



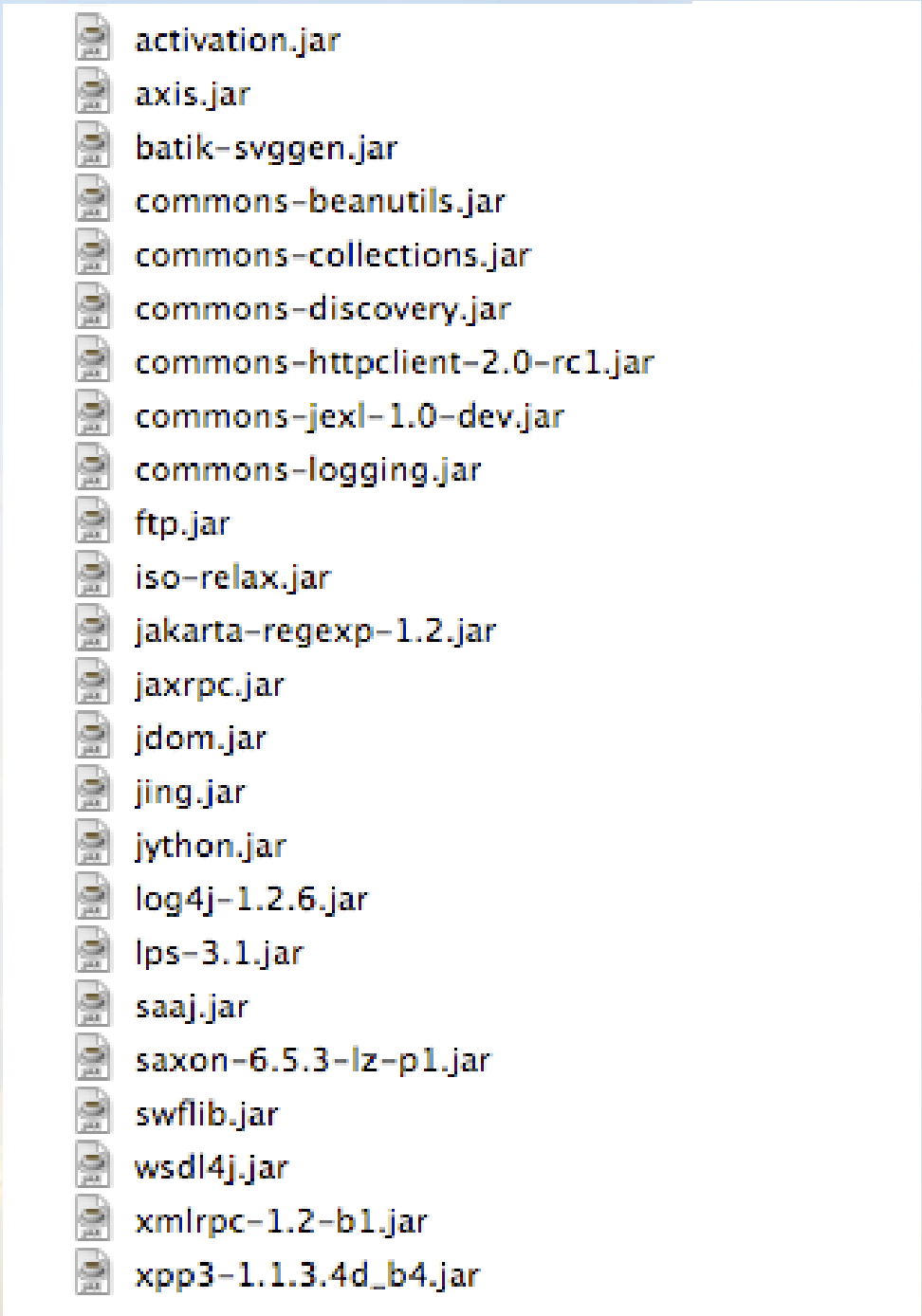
VIRTUAS

Laszlo Presentation Server



VIRTUAS

Dependencies



VIRTUAS

Retrieving Data

How is it included?	When is it loaded?	Code

Embedded	Compile-time	`<dataset name="users">`
		`  <user>`
		`  <!-- .. other XML tags .. -->`
		`  </user>`
		`</dataset>`

Included	Compile-time	`<dataset name="users"` `  src="users.xml"/>`

HTTP Data	Runtime	`<dataset name="users"`
		`  autorequest="true"`
		`  type="http"`
		`  src="users.html"/>`

Displaying Data

```
<canvas height="80" width="500">
  <dataset name="myData">
    <myXML>
      <person show="simpsons">
        <firstName>Homer</firstName>
        <lastName>Simpson</lastName>
      </person>
      <person show="simpsons">
        <firstName>Marge</firstName>
        <lastName>Simpson</lastName>
      </person>
    </myXML>
  </dataset>

  <view name="rowOfData" datapath="myData:/myXML[1]/person[1]">
    <simplelayout axis="x" />
    <text datapath="firstName/text()" />
    <text datapath="lastName/text()" />
    <text datapath="@show" />
  </view>
</canvas>
```


VIRTUAS

Displaying Data

```
<canvas height="80" width="500">
  <dataset name="myData">
    <myXML>
      <person show="simpsons">
        <firstName>Homer</firstName>
        <lastName>Simpson</lastName>
      </person>
      <person show="simpsons">
        <firstName>Marge</firstName>
        <lastName>Simpson</lastName>
      </person>
    </myXML>
  </dataset>
```

Multiple Rows

```
  <view name="myTable">
    <simplelayout axis="y" />
    <view name="rowOfData" datapath="myData:/myXML[1]/person">
      <simplelayout axis="x" />
      <text datapath="firstName/text()" />
      <text datapath="lastName/text()" />
      <text datapath="@show" />
    </view>
  </view>
</canvas>
```

VIRTUAS

Sending Data

```
<text name="pk" visible="false" datapath="@email"/>
<text y="10">First Name:</text>
    <edittext name="firstName" datapath="@firstName" x="80" y="10"/>
<text y="35">Last Name:</text>
    <edittext name="lastname" datapath="@lastName" x="80" y="35"/>
<text y="60">Phone:</text>
    <edittext name="phone" datapath="@phone" x="80" y="60"/>
<text y="85">Email:</text>
    <edittext name="email" datapath="@email" x="80" y="85"/>

<method name="sendData" args="action">
  var d=canvas.datasets.dsSendData;
  var p=new LzParam();
  p.addValue("action", action, true);
  p.addValue("pk", pk.getText(), true);
  p.addValue("firstName", firstName.getText(), true);
  p.addValue("lastName", lastName.getText(), true);
  p.addValue("phone", phone.getText(), true);
  p.addValue("email", email.getText(), true);
  d.setQueryString(p);
  d.doRequest();
</method>
```

VIRTUAS

Sending Data

```
<text name="pk" visible="false" datapath="@email"/>
<text y="10">First Name:</text>
    <edittext name="firstName" datapath="@firstName" x="80" y="10"/>
<text y="35">Last Name:</text>
    <edittext name="lastname" datapath="@lastName" x="80" y="35"/>
<text y="60">Phone:</text>
    <edittext name="phone" datapath="@phone" x="80" y="60"/>
<text y="85">Email:</text>
    <edittext name="email" datapath="@email" x="80" y="85"/>

<method name="sendData" args="action">
  var d=canvas.datasets.dsSendData;
  d.setQueryString( { action: "action",
                      pk: pk.getText(),
                      firstName: firstName.getText(),
                      lastName: lastName.getText(),
                      phone: phone.getText(),
                      email: email.getText() } );
  d.doRequest();
</method>
```


VIRTUAS

OpenLaszlo RPC

- Invoke remote procedure calls or services over the network
- Current implementations:
  - **<javarpc>**
  - **<xmlrpc>**
  - **<soap>**
  - **<webapprpc>**
  - **<sessionrpc>**
  - **<XMLHttpRequest>**

VIRTUAS

Configuration

- ***lzt*** parameter can be used to customize output

  http://host[:port]/app/file.lzx?**lzt**=*type*
- /WEB-INF/lps/config/**lps.properties** - LPS config
- /WEB-INF/lps/config/**lps.xml** - logging configuration and filters for data requests and user agents
- /WEB-INF/lps/config/**lzusers.xml** - user database for default LPS security servlet
- /WEB-INF/lps/config/**crossdomain.xml** - sample cross domain policy file

VIRTUAS

Deployment

- Remove **lps/utils** - contains source code viewer
- Use dataoption="pooling" for <list> elements
- Time, tune and optimize JavaScript
- Tune cache settings
- Turn off development features, especially *allowRequestKRANK* in lps.properties
- Use **KRANK** to "compile" your application

VIRTUAS

Related Projects

- **Laszlo on Rails**

  http://wiki.openlaszlo.org/Laszlo_on_Rails

- **RIFE/Laszlo**

  http://rifers.org/blogs/gbevin/2005/5/28/announcing_rifelaszlo

- **Faces4Laszlo**

  http://www.alphaworks.ibm.com/tech/faces4laszlo

- **IDE for Laszlo**
  - http://eclipse.org/laszlo/

VIRTUAS

VIRTUAS

Laszlo + Spring Demo

VIRTUAS

Questions?

mraible@virtuas.com

VIRTUAS