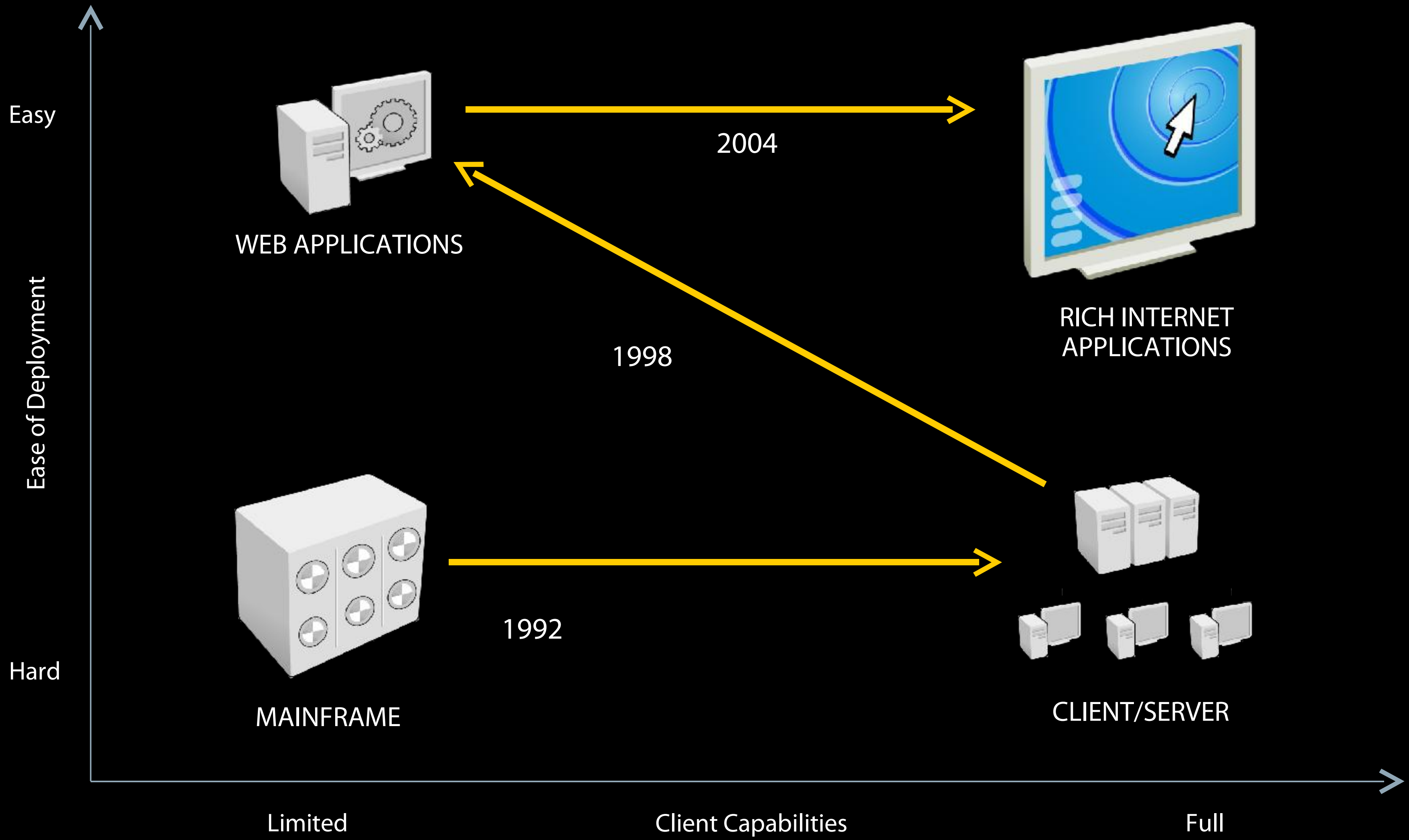
Flex and Java

James Ward

http://www.jamesward.com

twitter://jlward4th

Applications have evolved



Adobe's Software Development Platform

Applications

Adobe Media Player

Open Bug Database: http://bugs.adobe.com

Designer/Developer Tools

60 Day **Free** Trial

Free for students and educators

Flex Builder

Client Runtimes

Adobe AIR

Flash Player

PDF

High Performance JIT'ing VM: Mozilla Tamarin

Adobe Flex 3

Servers/Services

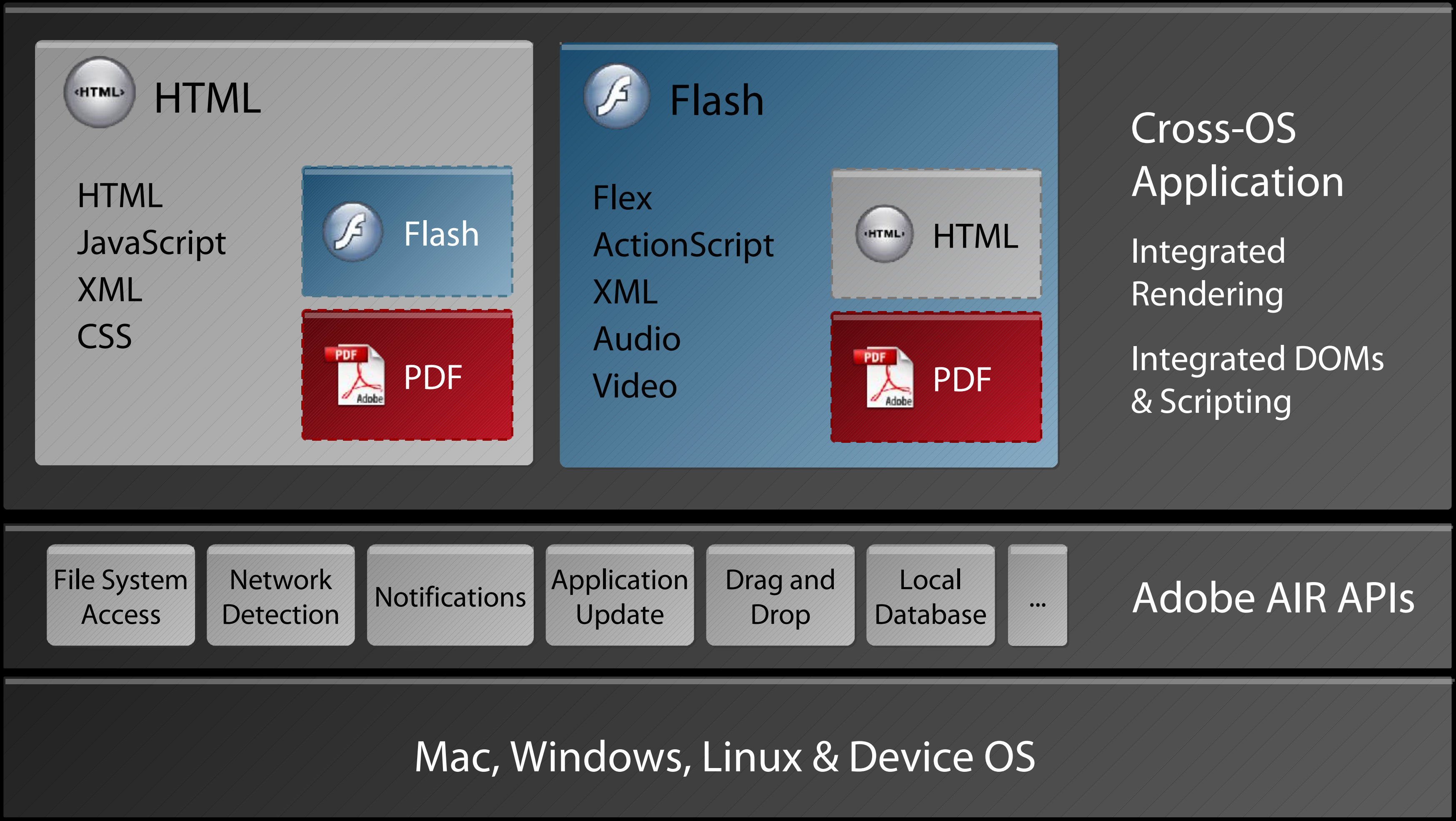
LiveCycle

ColdFusion

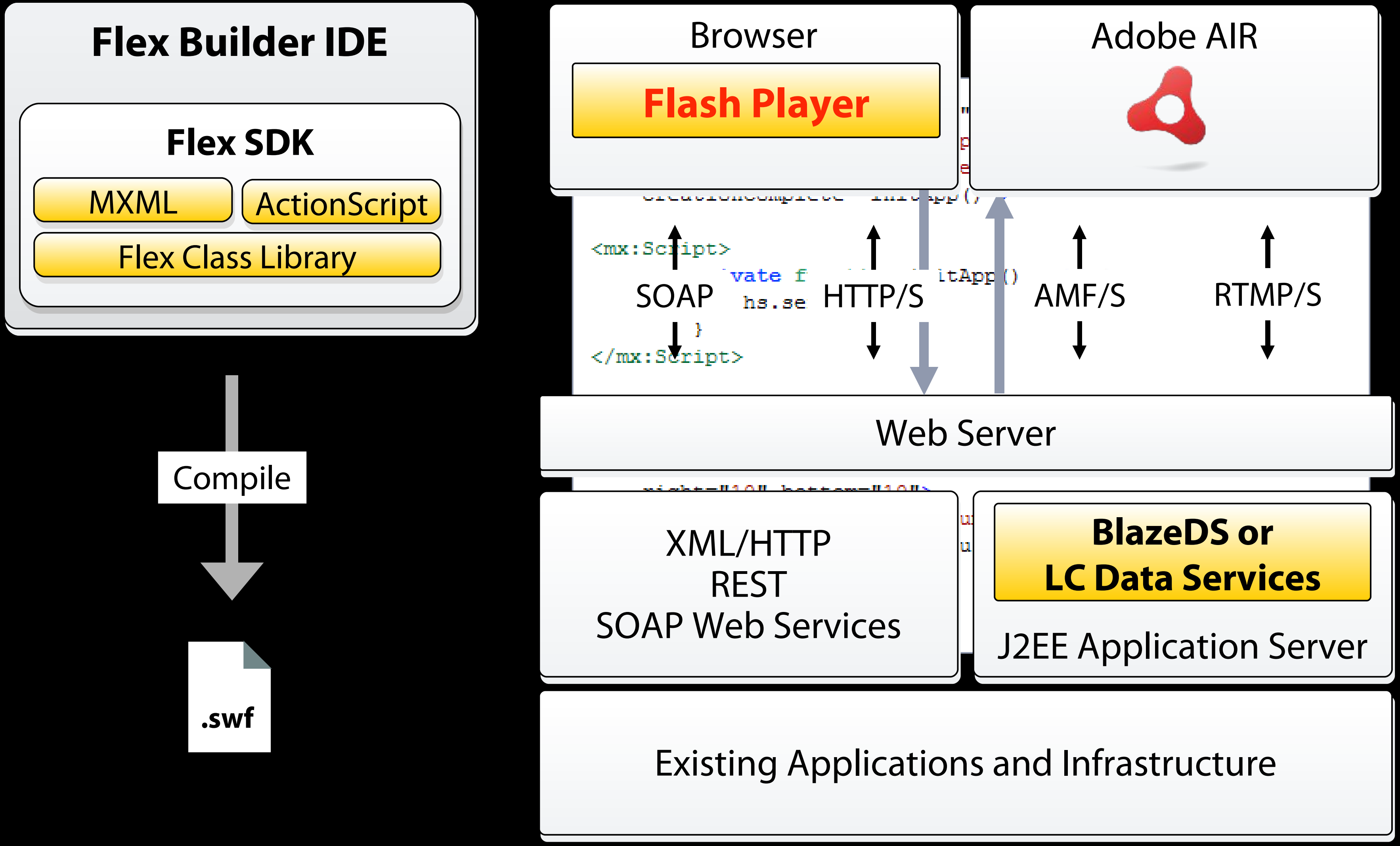
Open Source: MPL

Use any text editor / IDE

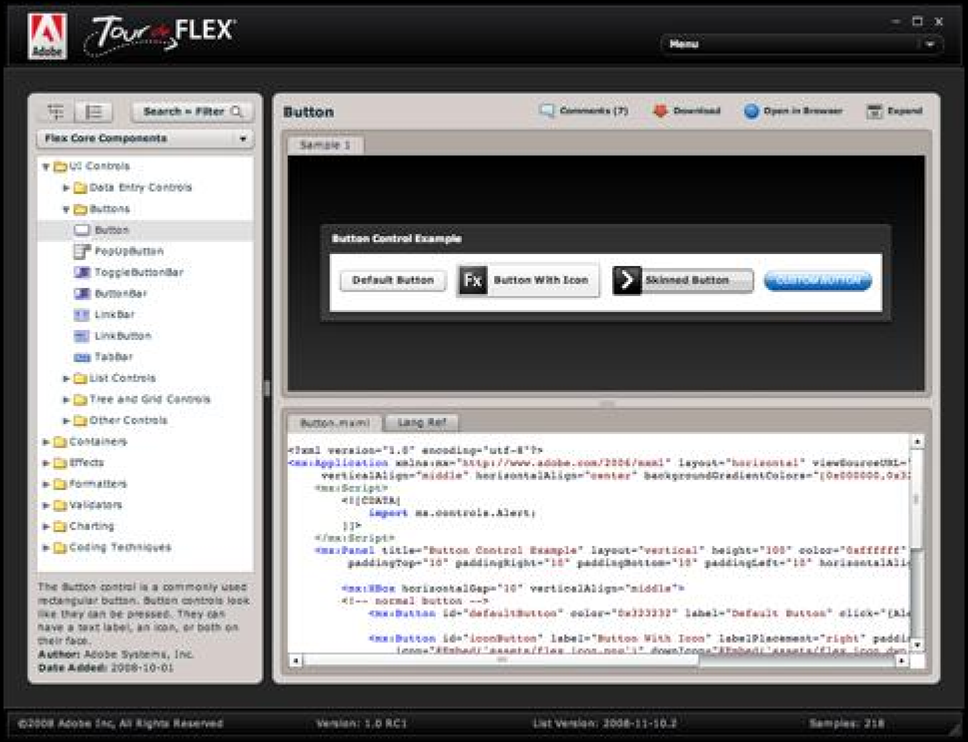
Adobe AIR Application Stack



How Flex Works



Tour de Flex - flex.org/tour



The Button control is a commonly used rectangular button. Button controls look like they can be pressed. They can have a text label, an icon, or both on their face.
Author: Adobe Systems, Inc.
Date Added: 2008-10-01

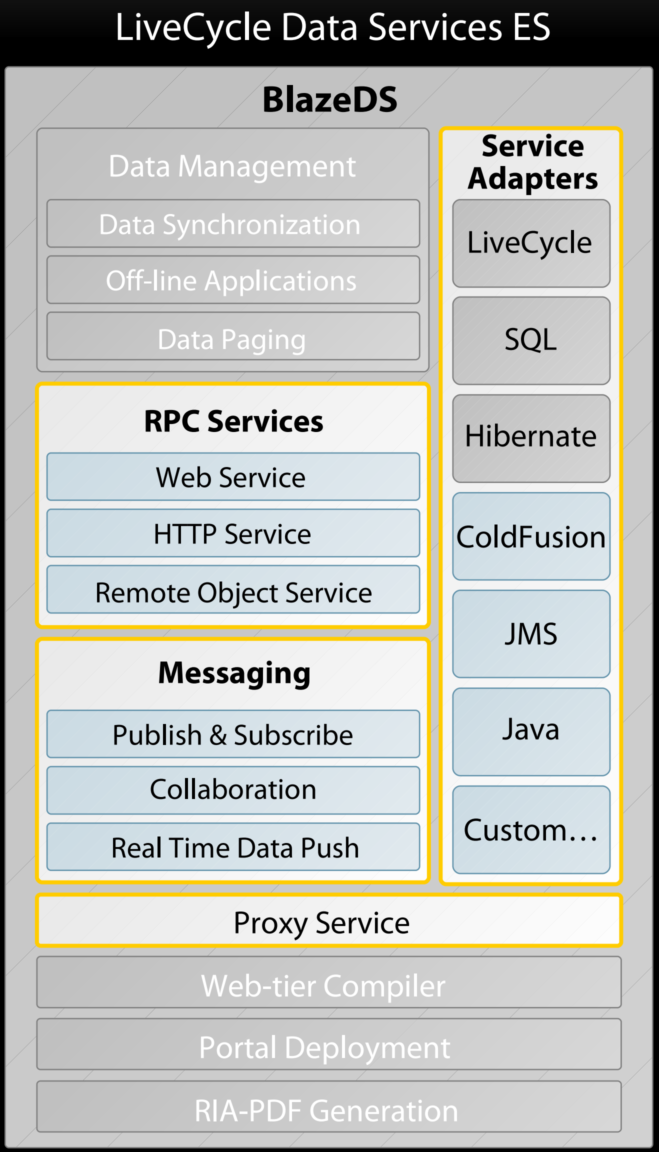
```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="horizontal" viewSourceURL="
    verticalAlign="middle" horizontalAlign="center" backgroundGradientColors="[0x000000,0x3
    <mx:Script>
        <![CDATA[
            import mx.controls.Alert;
        ]]>
    </mx:Script>
    <mx:Panel title="Button Control Example" layout="vertical" height="100" color="0xffffff"
         paddingTop="10" paddingRight="10" paddingBottom="10" paddingLeft="10" horizontalAli

        <mx:HBox horizontalGap="10" verticalAlign="middle">
        <!-- normal button -->
            <mx:Button id="defaultButton" color="0x323232" label="Default Button" click="{Al

            <mx:Button id="iconButton" label="Button With Icon" labelPlacement="right" paddi
```

©2008 Adobe Inc, All Rights Reserved   Version: 1.0 RC1   List Version: 2008-11-10.2   Samples: 218

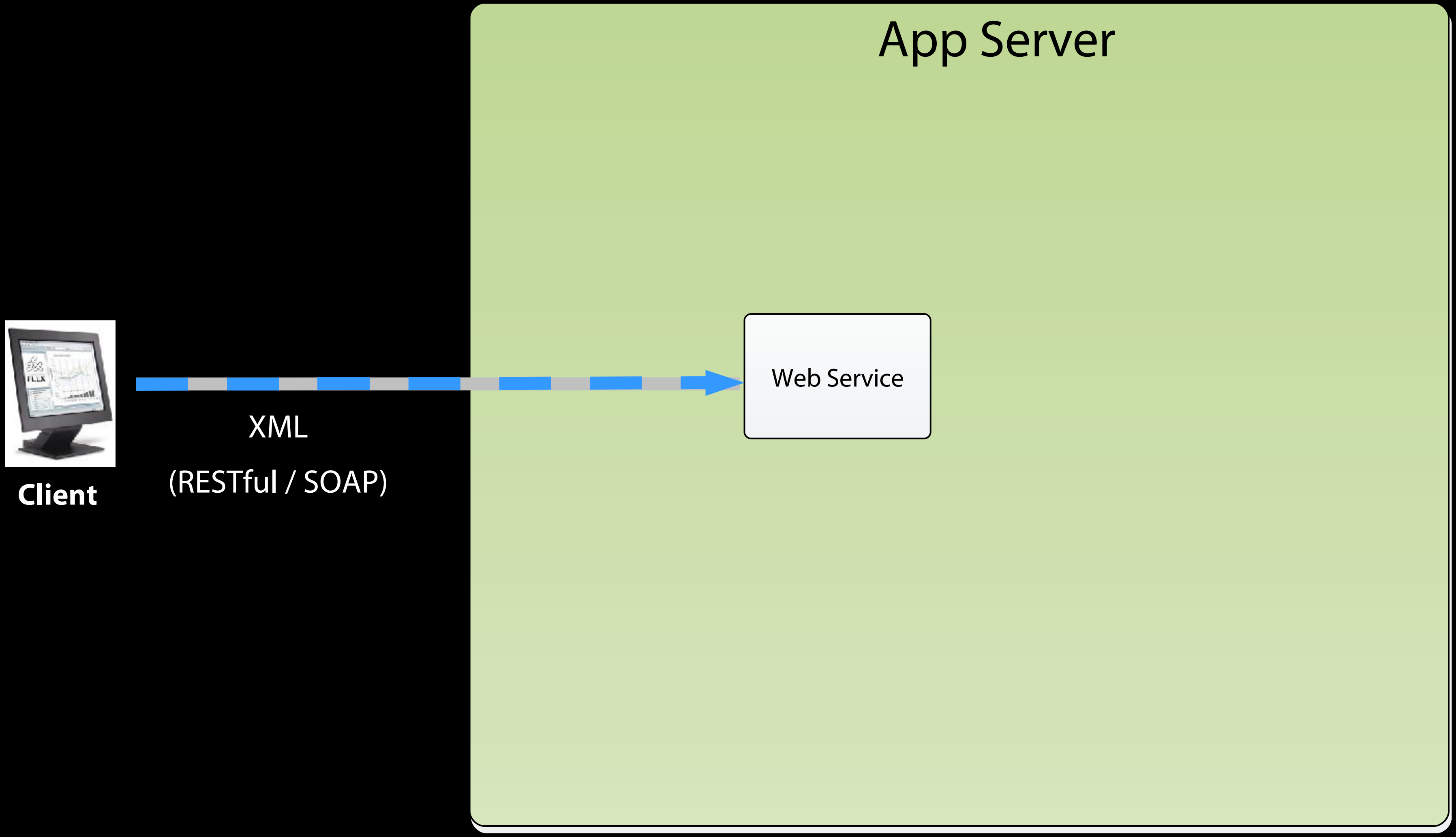
Introducing Open Source BlazeDS

BlazeDS is the remoting and HTTP-based messaging technology which Adobe is contributing to the community under LGPL v3

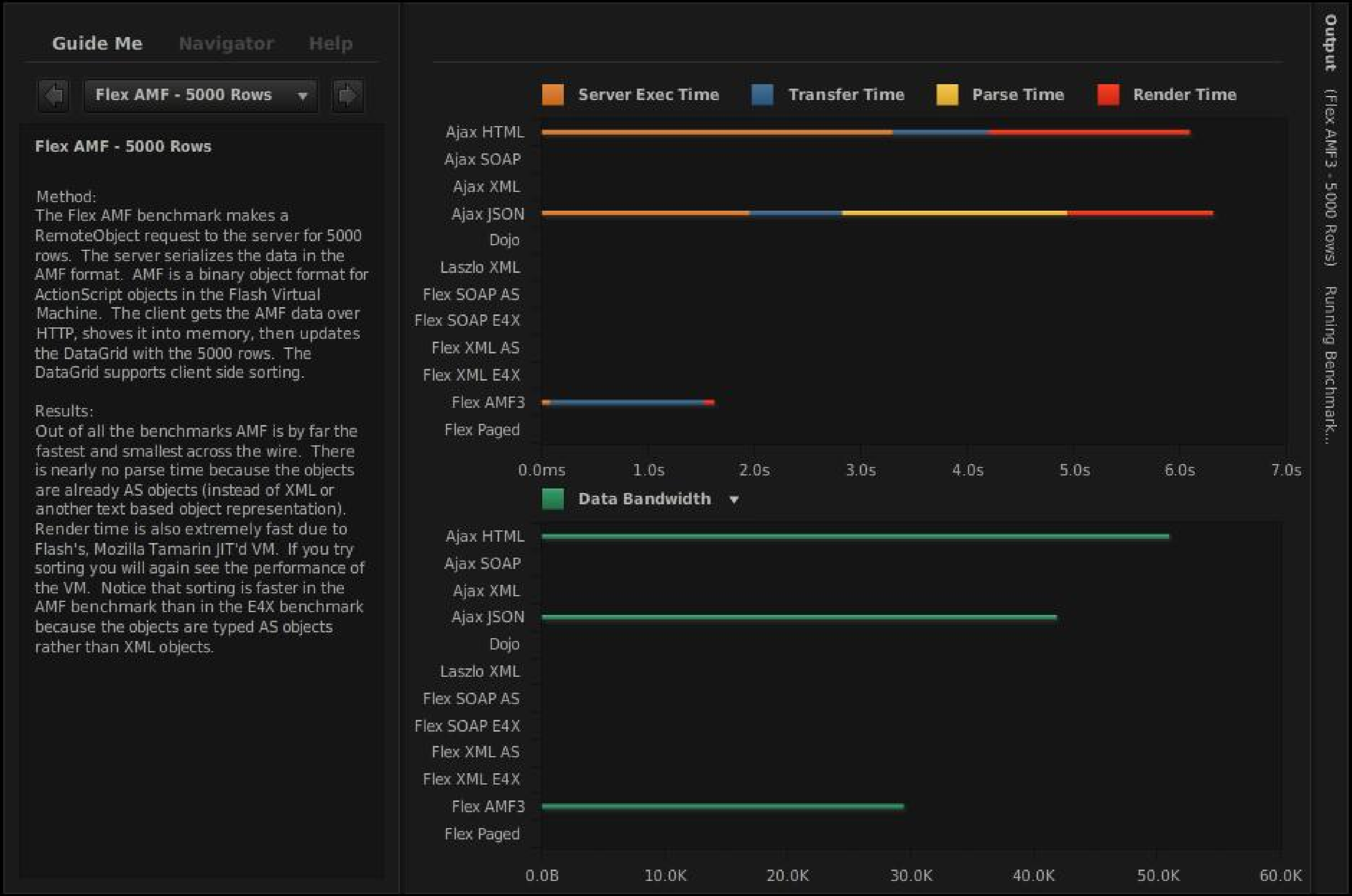
- Capabilities
  - Easily connects Flex & AIR applications to existing server logic
  - High performance data transfer for more responsive applications
  - Real-time data push over standard HTTP
  - Full pub/sub messaging that extends existing messaging infrastructure
- Publication of the Action Message Format (AMF3) binary data protocol specification
- Certified builds, warranty protection and enterprise support subscriptions available



Flex with Java via XML (RESTful / SOAP)



http://www.jamesward.com/census/

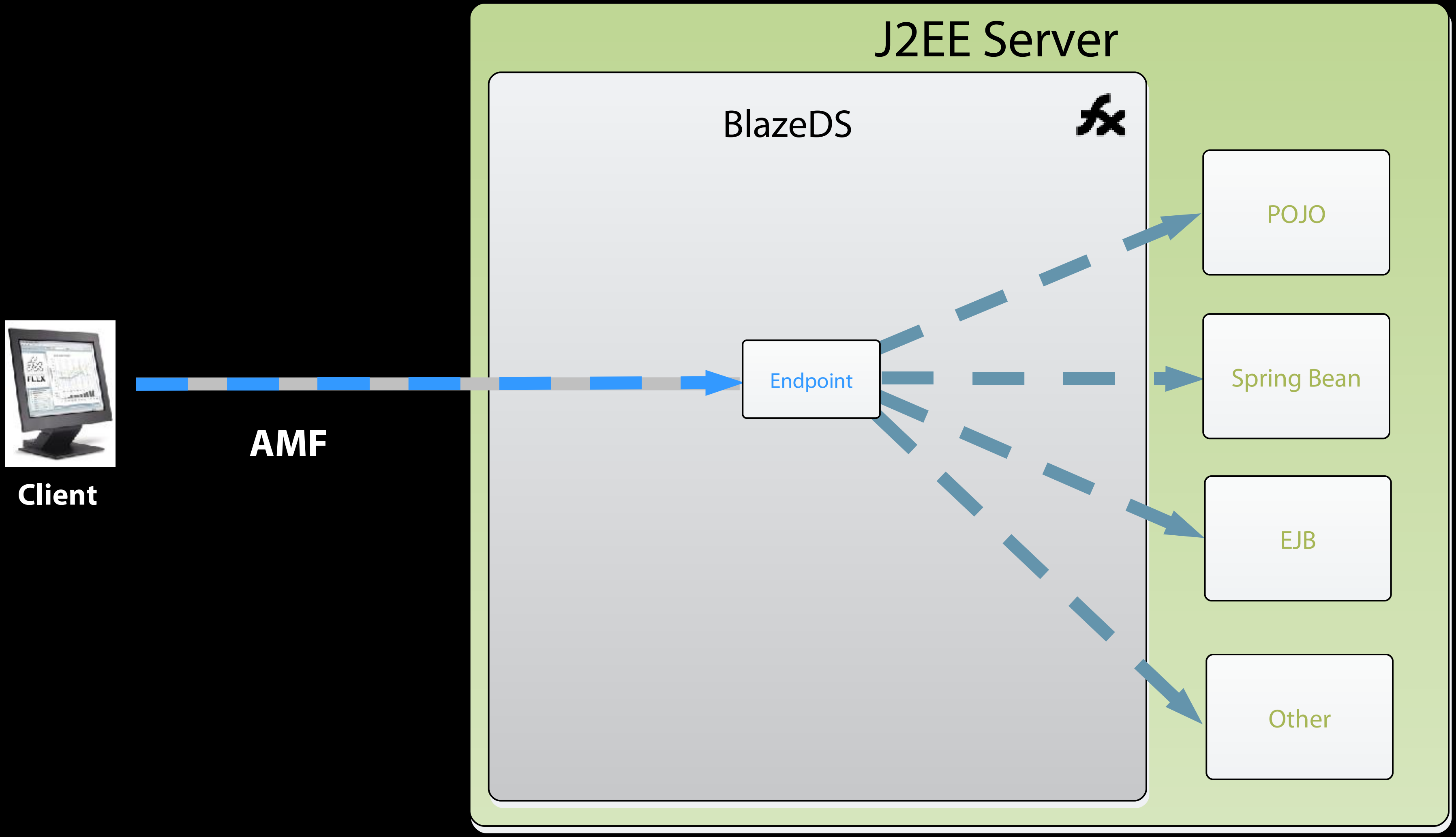


Flex AMF - 5000 Rows

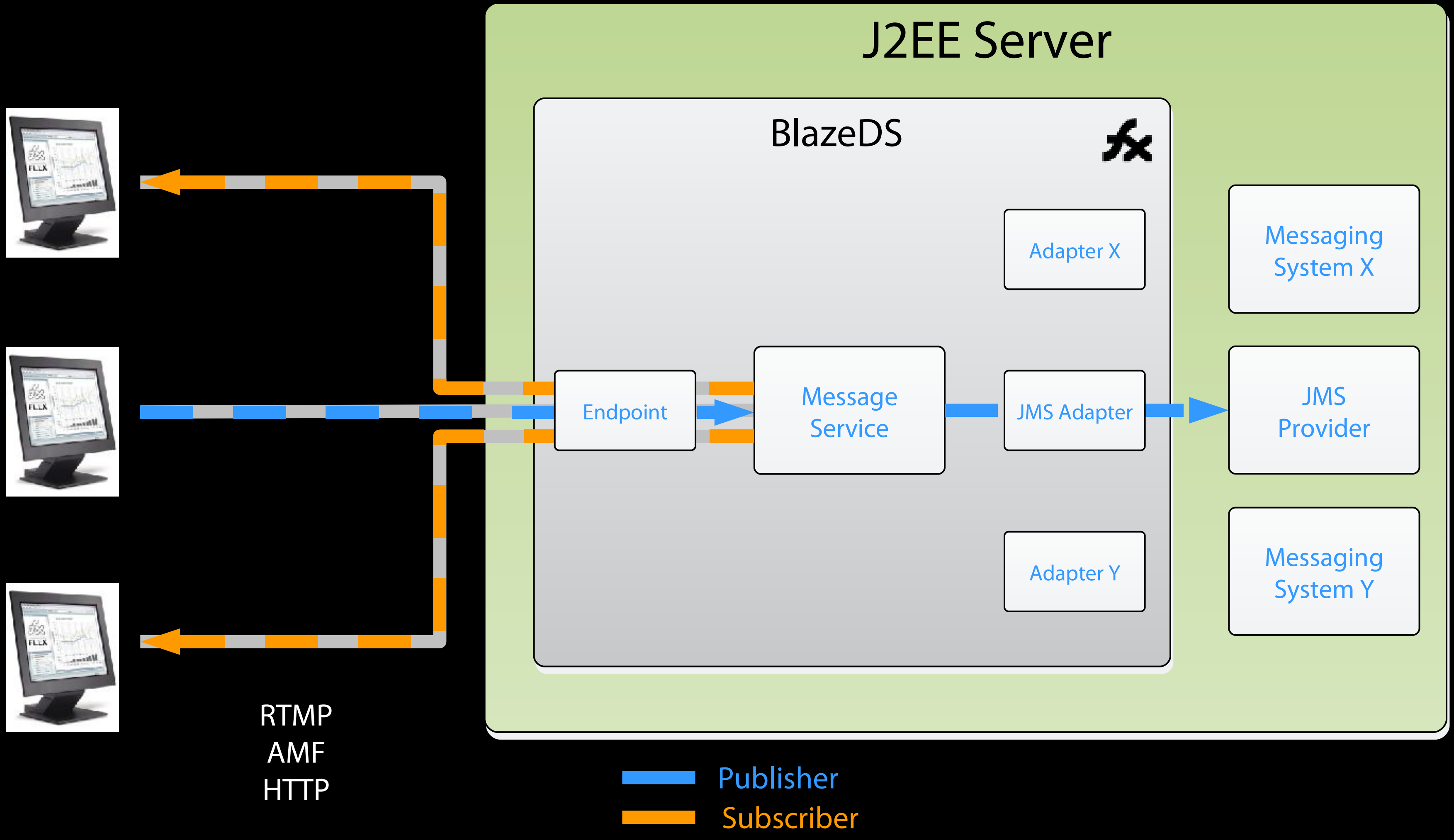
Method:
The Flex AMF benchmark makes a RemoteObject request to the server for 5000 rows. The server serializes the data in the AMF format. AMF is a binary object format for ActionScript objects in the Flash Virtual Machine. The client gets the AMF data over HTTP, shoves it into memory, then updates the DataGrid with the 5000 rows. The DataGrid supports client side sorting.

Results:
Out of all the benchmarks AMF is by far the fastest and smallest across the wire. There is nearly no parse time because the objects are already AS objects (instead of XML or another text based object representation). Render time is also extremely fast due to Flash's, Mozilla Tamarin JIT'd VM. If you try sorting you will again see the performance of the VM. Notice that sorting is faster in the AMF benchmark than in the E4X benchmark because the objects are typed AS objects rather than XML objects.

Flex with Java via Remoting



Flex with Java via Messaging



Spring and Flex

- Flex moves the view and controller completely to the client
    - The return of client/server architecture
- Still want to utilize our rich Java model / business layer
    - Spring!

Spring and Flex!

- SpringSource and Adobe have formed a joint partnership to turn this idea into reality!
- The foundations of this new integration will be available as open source
    - A new Spring subproject in the web portfolio:

      Spring BlazeDS Integration
- Focus on integrating the open source BlazeDS with Spring

M1 - Dec 2008

- Bootstrap the BlazeDS MessageBroker as a Spring-managed bean (no more web.xml MessageBrokerServlet config needed)
- Route http-based Flex messages to the MessageBroker through the Spring DispatcherServlet
- Expose Spring beans for remoting using typical Spring remoting exporter configuration

M2 - March 2008

- Spring Security integration
    - Ensure that Spring security can secure any Springmanaged endpoints with credentials provided by the Flex app

post M2

- Spring JMS integration
  - Integration with the BlazeDS MessageService
  - Use Spring configuration to manage BlazeDS MessageDestinations
  - Let Spring manage the JMS details
  - Allows easy communication from Flex clients to Spring message-driven POJOs
- Spring 3.0 REST integration
  - Provides support for multiple client-types
  - Flex apps can already consume Spring 3.0 RESTful endpoints through HTTPService
  - Additional value could be realized by providing an AMFView implementation
    - Response for HTTP requests with a Content-Type=application/actionscript

web.xml

```
listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

/listener>

servlet>

<servlet-name>testdrive</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<load-on-startup>1</load-on-startup>
```

web.xml - Security Config

```
context-param>

<param-name>contextConfigLocation</param-name>

<param-value>

            /WEB-INF/config/web-application-config.xml

            /WEB-INF/config/web-application-security.xml

</param-value>

/context-param>


filter>

<filter-name>springSecurityFilterChain</filter-name>
```

web-application-config.xml

```
beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:flex="http://www.springframework.org/schema/flex"
  xmlns:security="http://www.springframework.org/schema/security"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="
          http://www.springframework.org/schema/beans
          http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
          http://www.springframework.org/schema/flex
          http://www.springframework.org/schema/flex/spring-flex-1.0.xsd
```

web-application-config.xml - Remoting Beans

```
 <!-- Expose the productDAO bean for BlazeDS remoting -->
 <flex:remote-service ref="productService" />


<!-- A secured version of productService -->
<bean id="securedProductService" class="flex.spring.samples.product.ProductDAO" >
   <flex:remote-service/>
   <constructor-arg ref="dataSource"/>
   <security:intercept-methods>
        <security:protect method="find*" access="ROLE_USER" />
 </security:intercept-methods>
</bean>
```

Frameworks

- Cairngorm
- Mate
- PureMVC

- Clear

- Swiz
- Spring ActionScript

fluint - Flex Unit and Integration Testing Framework

- Multiple simultaneous asynchronous operations
- Asynchronous setup and teardown
- Asynchronous returns before method body completion
- Support for UIComponent testing
- Support for test sequences
- Support for testing Cairngorm commands and controllers
- XML output of testing results
- Support for externalizing tests in modules
- Build automation integration with Apache Ant

Flex Monkey

- Records and plays back Flex UI interactions
- UI Interactions can be edited and replayed
- Generates FlexUnit TestCases, and can also be used with non-FlexUnit-based testing frameworks
- Tests can be run from build systems such as Ant
- Handles all Flex UI events
- Uses Flex Automation API to provide native control over your flex app. Requires no javascript or browser plug-ins to use.
- Unit tests are written entirely in ActionScript. No other programming or special purpose scripting languages are needed to develop comprehensive UI test suites.
- Non-invasive. Requires no modifications to your application source.