360|Flex Atlanta
Building Java Powered Flex Applications

Andrew Powell

Introductions

- Sr. Consultant, Universal Mind
- UM is a 2007 Inc. 500 Honoree
- 10+ Years Experience In Software Development
- ASP --> Java --> ColdFusion --> Flex & AJAX
- Multiple Publications on Flex, Java, & ColdFusion
- http://www.infoaccelerator.net

What Will We Learn Today?

- Using Java Objects as Remote Objects
- Inter-Application Messaging With JMS
- Leveraging Spring & Hibernate
- Binary Remoting Without AMF

The Technologies

- **Using Java Objects as Remote Objects**
- Inter-Application Messaging With JMS
- Leveraging Spring & Hibernate
- Binary Remoting Without AMF

What Is Remoting?

- The Ability to Call POJOs as ActionScript Local Objects
- Integration With Existing Java Security Policies
- Transparent Data Transfer Between Java and ActionScript
- No Need To Write Marshalling Code
- Improved Transfer Performance via AMF

Implementing Remote Objects

- Two Types
    - Remoting Proxy
    - Remoting Service
        - Java or ColdFusion
- Configured In: services-config.xml
    - Must Specify Location of services-config.xml To Compiler

SHOW ME THE CONFIG

Implementing Remote Objects

- <mx:RemoteObject>
  - concurrency
  - destination
  - id
  - endpoint
  - showBusyCursor
  - source
  - makeObjectsBindable
  - fault
  - result

SHOW ME THE CODE

Factories

- Factory - Your Own Component Creation and Maintenance System
- Allows Integration With:
  - EJB
  - Spring
- Configured In services-config.xml

Factory Configuration

```
<factories>
    <factory id="spring" class="flex.samples.factories.SpringFactory"/>
</factories>
```

Factory Implementation

```
<destination id="myDestination">
  <properties>
    <source>mypackage.MyRemoteClass</source>
    <factory>myFactoryId</factory>
    <myfactoryinstanceattribute>myfoobar2value</myfactoryinstanceattribute>
  </properties>
</destination>
```

Factory Implementation

```
<destination id="WeatherService">
   <properties>
      <factory>spring</factory>
      <source>weatherBean</source>
   </properties>
</destination>
```

SHOW ME THE CONFIG

- Using Java Objects as Remote Objects
- **Inter-Application Messaging With JMS**
- Leveraging Spring & Hibernate
- Binary Remoting Without AMF

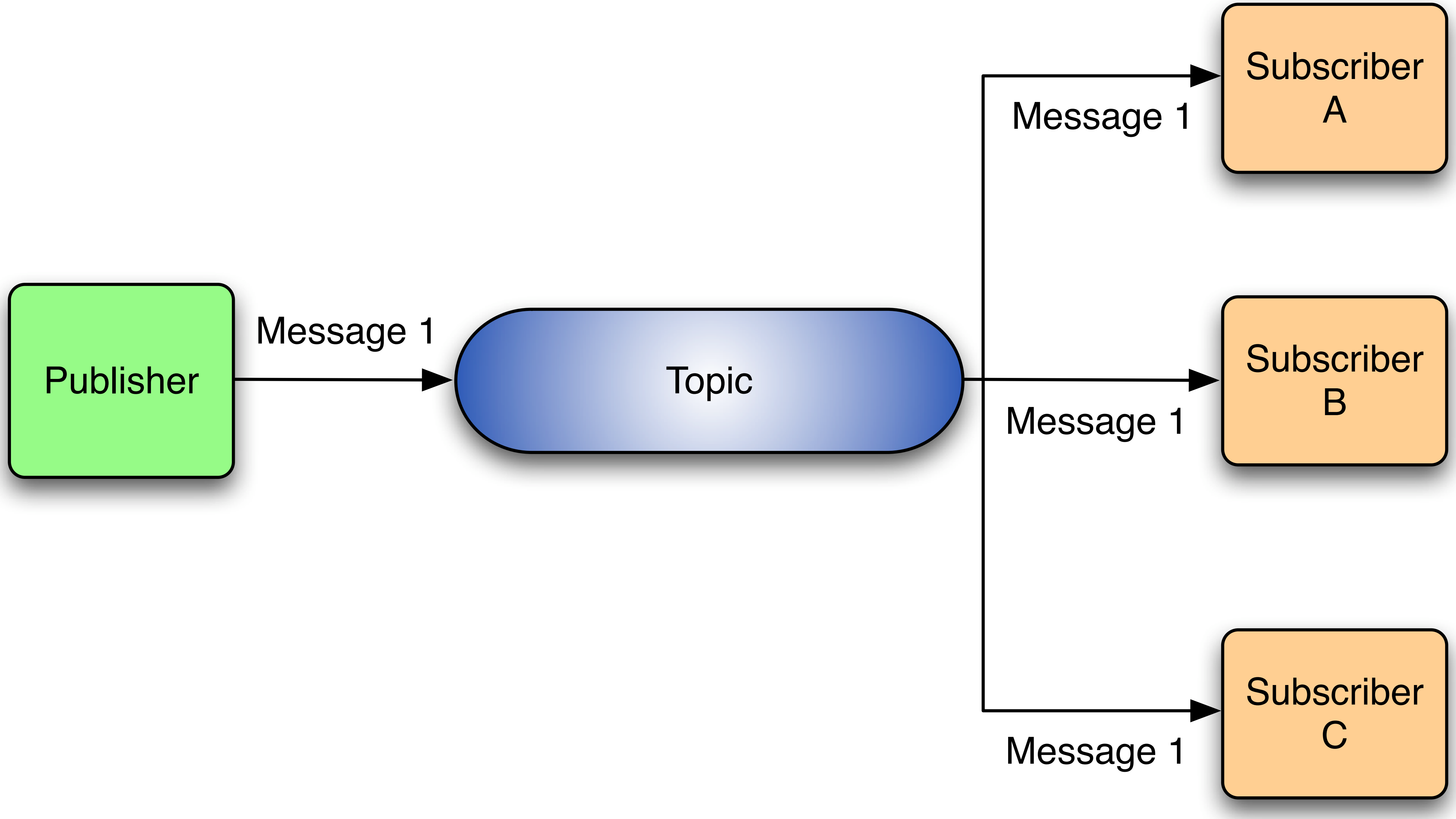
Messaging With JMS

- What Is JMS?
  - Java Message Service API
  - Message Oriented Middleware API
- API For Sending Messages Between Two or More Clients
- Allows For Loosely Coupled Distributed Communication

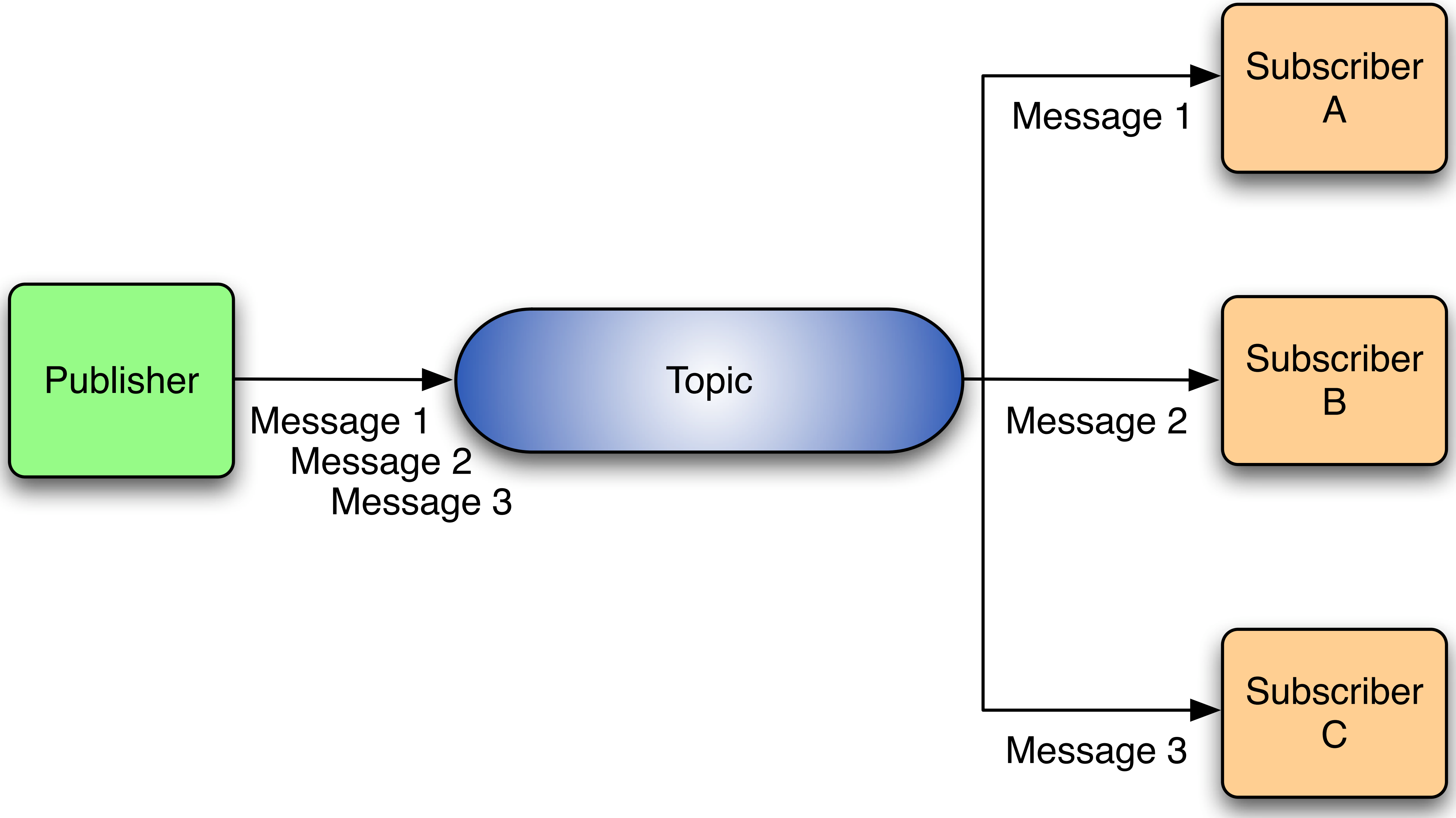
Messaging With JMS

- JMS Models
  - Topic
    - Publish / Subscribe
  - Queue
    - Point-To-Point

Messaging With JMS



Messaging With JMS



JMS Configuration

- Configured In services-config.xml
- Local Server Or Remote
- ANY JMS Implementation Will Work
    - JBoss Messaging
    - ActiveMQ
    - JRun - CoreMQ

SHOW ME THE CONFIG

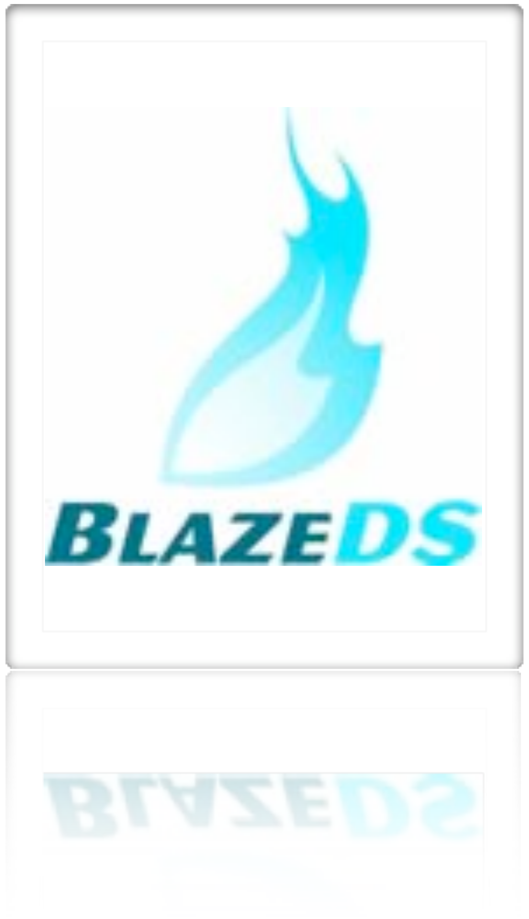
Why Use JMS?

- Ability To Send Either:
  - Plain Text
  - Objects
- Messaging Between Diverse Clients
  - All Clients Use The Same Queue / Topic
  - AIR / Flex / Swing / JavaFX

SHOW ME THE CODE

Functionality So Far...

- All The Previous Functionality Exists In Blaze DS



- Using Java Objects as Remote Objects
- Inter-Application Messaging With JMS
- **Leveraging Spring & Hibernate**
- Binary Remoting Without AMF

Introducing Spring

- What Is Spring?
- Spring Provides:
  - Inversion of Control
  - Aspect Orient Programming Model
- Many Sub-Projects:
  - Spring Security (Acegi Security)
  - Spring Web Services
- Leveraged In Flex via Factories

Spring Basics

- Configured Via XML
- Can Use Multiple XML Files
- Application Context Loaded in web.xml
- Does Not Provide Data Access, Logging, Transactions, etc.
    - Provides An Interface to Existing Libraries

SHOW ME THE CODE

Introducing Hibernate

- What Is Hibernate?
- Hibernate Provides:
  - Relational Persistence
  - Automatic Object and Query Caching
- Sub Projects:
  - Shards - Horizontal Data Partitioning
  - Search - Integration with Lucene
  - Validator - Data Integrity Validation API
- Assembler For LCDS Included With LCDS

Hibernate Basics

- Configured Via XML or Properties File
  - hibernate.cfg.xml - config file
  - myPOJO.hbm.xml - mapping file
- Inline, Modular Caching Configuration
- Inline, Modular Transaction Management
- Can Also Be Configured in Spring Configuration

SHOW ME THE CODE

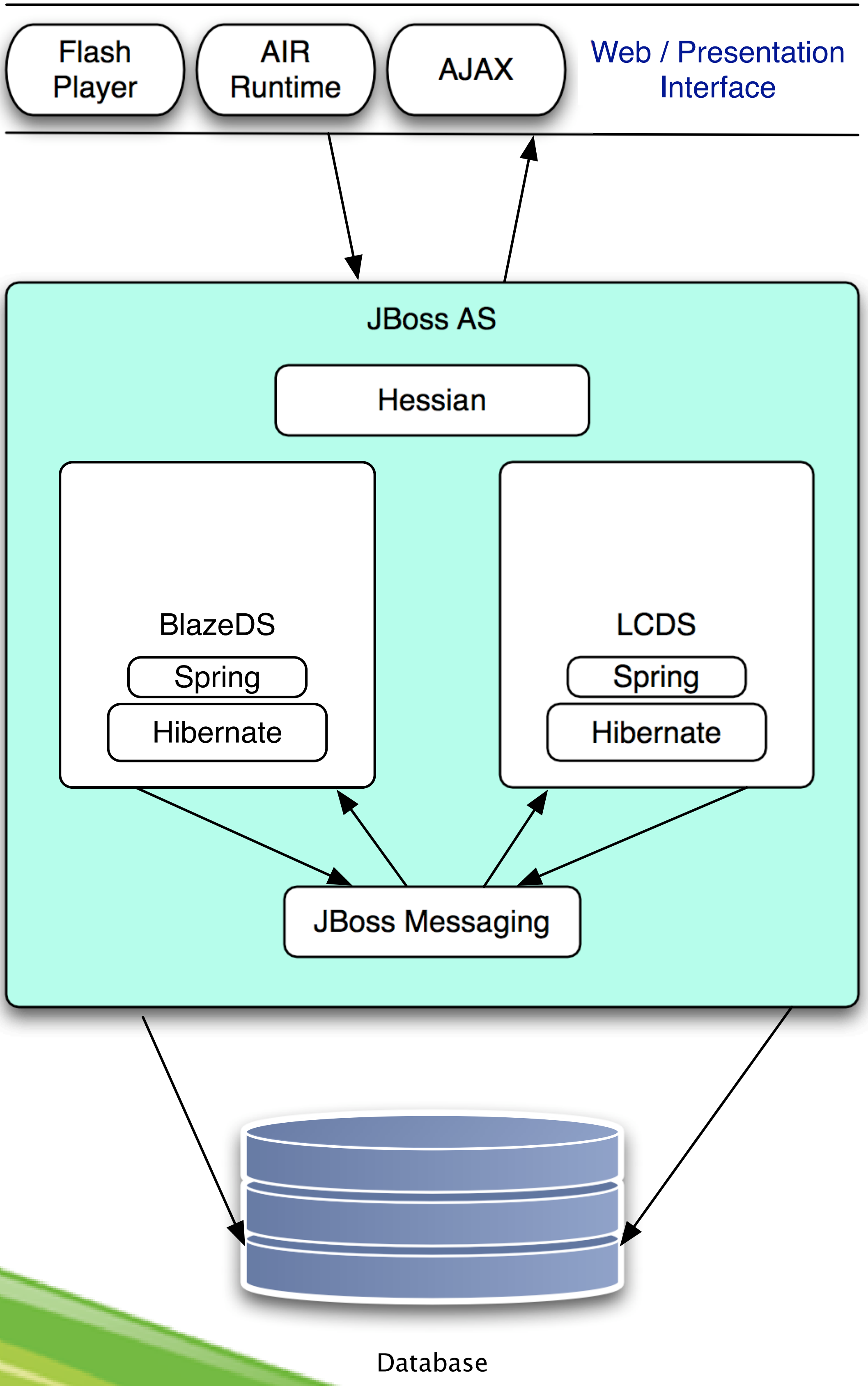
- Using Java Objects as Remote Objects
- Inter-Application Messaging With JMS
- Leveraging Spring & Hibernate
- **Binary Remoting Without AMF**

Hessian

- Binary Web Service Protocol
- Very Lightweight Compared To AMF
- Eliminates Need For External API Descriptors (WSDL, etc.)
- Configurable Stand-Alone or Via Spring

SHOW ME THE CODE

Architecture



Q&A

Resources

- http://labs.adobe.com/technologies/blazeds/
- http://www.springframework.org
- http://www.hibernate.org
- http://hessian.caucho.com
- http://blog.universalmind.com