



Flex Integration with Java EE Scenarios

Shashank Tiwari

 stiwari@saventech.com

Saven Technologies

(<http://www.saventech.com>)

877-SAVEN-21



Take Off

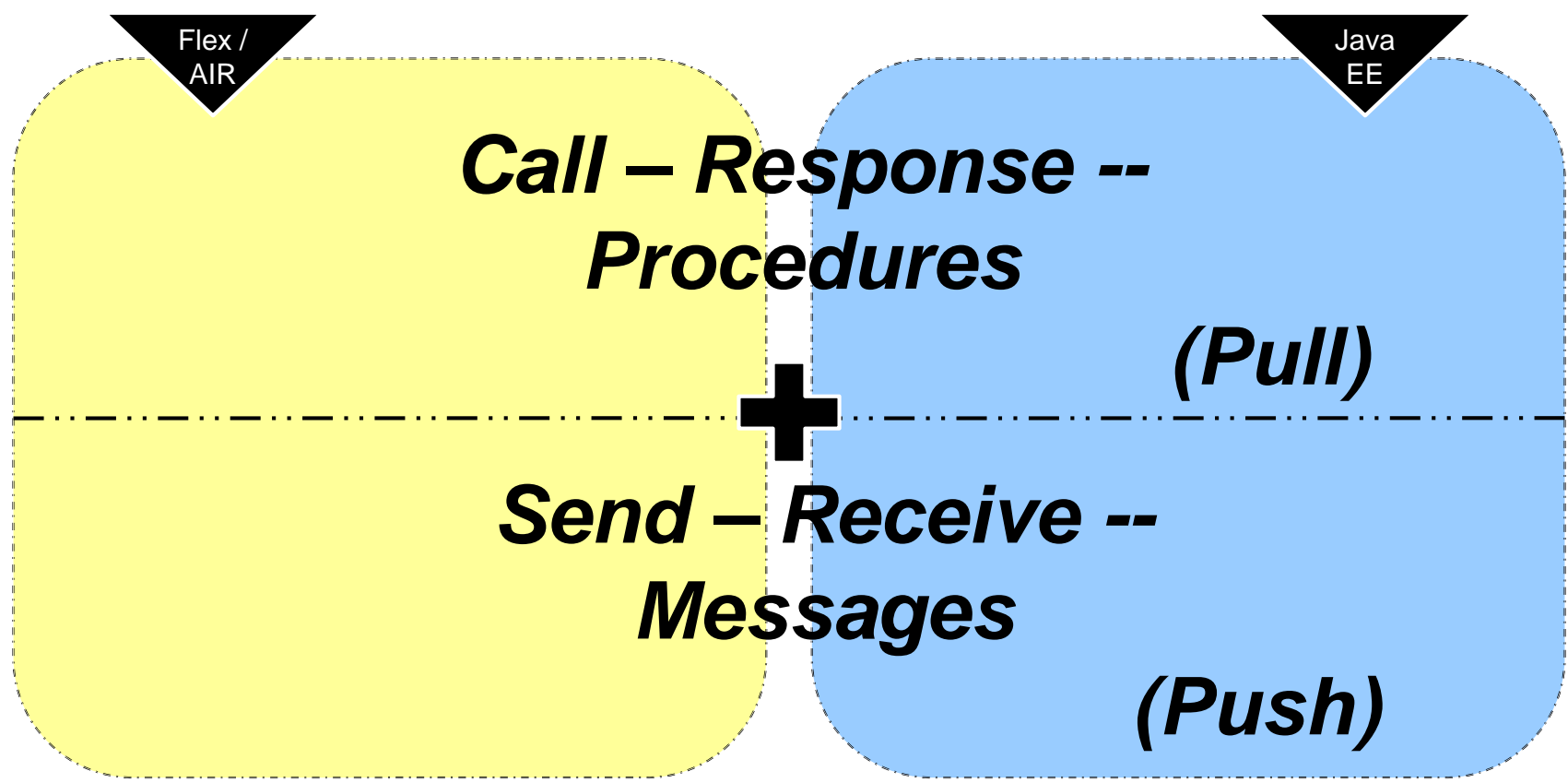


What will we try and achieve in this session?

- Understand Flex and Java EE integration
 - from different levels of granularity
- Walkthrough a few example scenarios

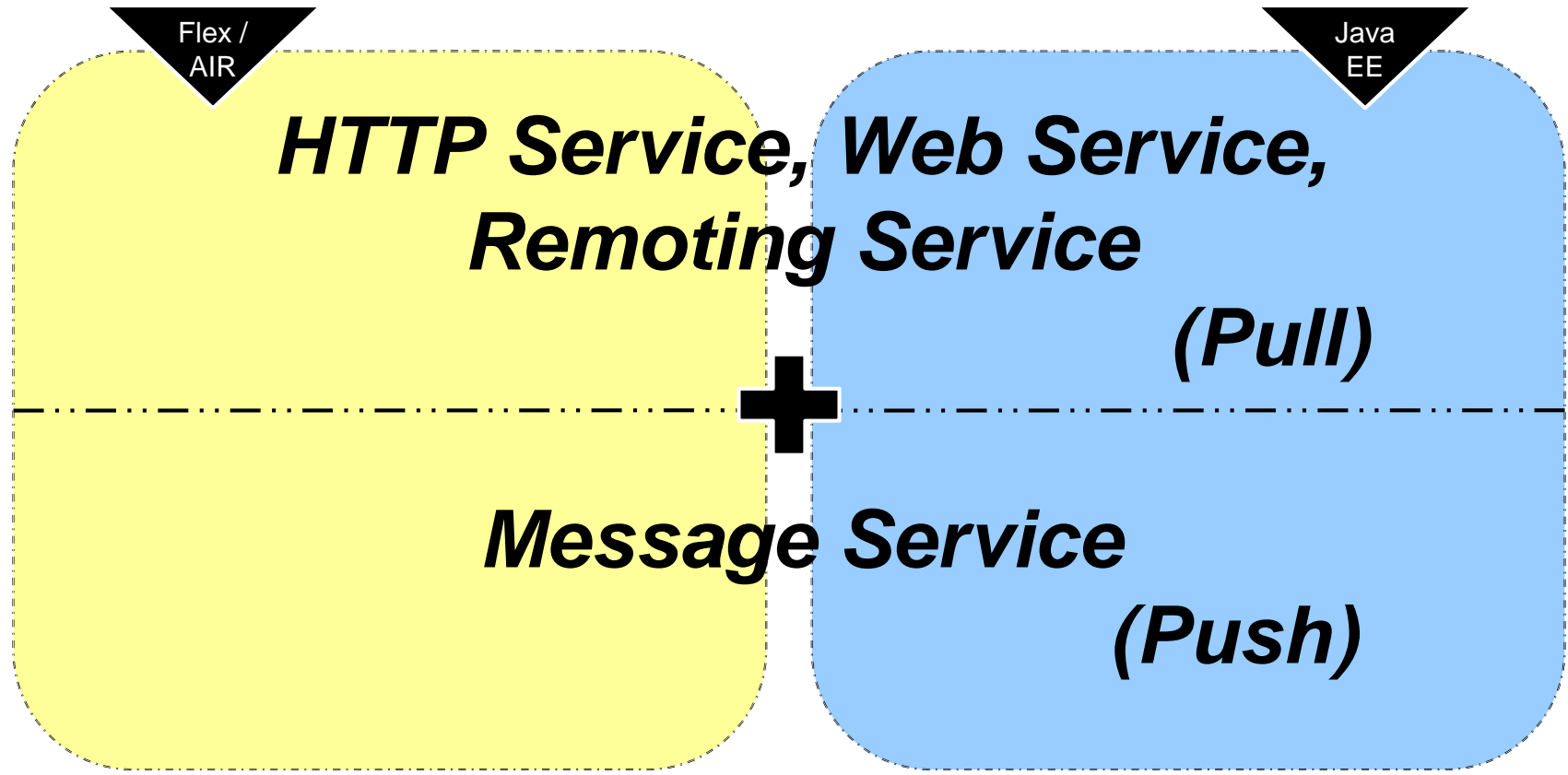


30,000 ft. View



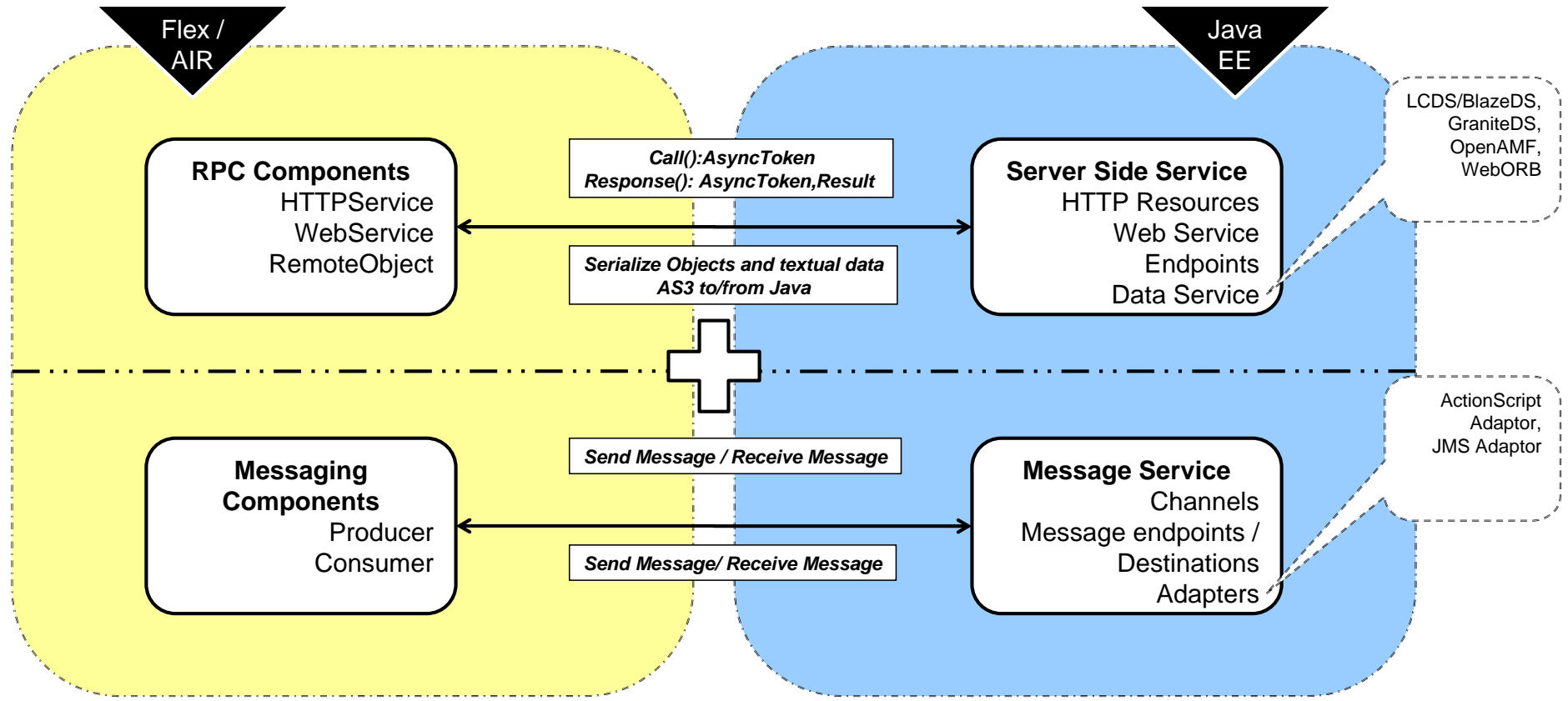


10,000 ft. View



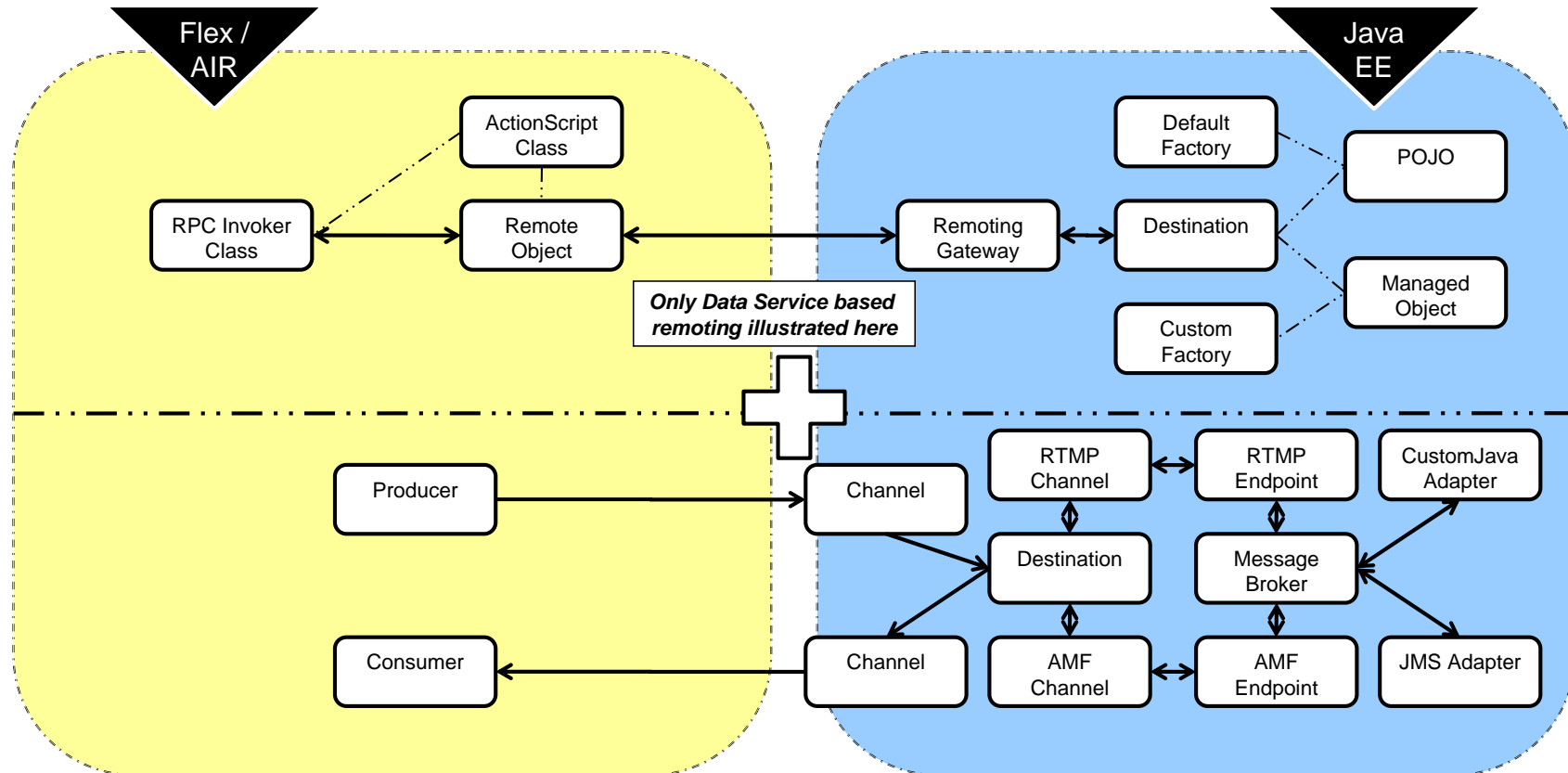


1,000 ft. View



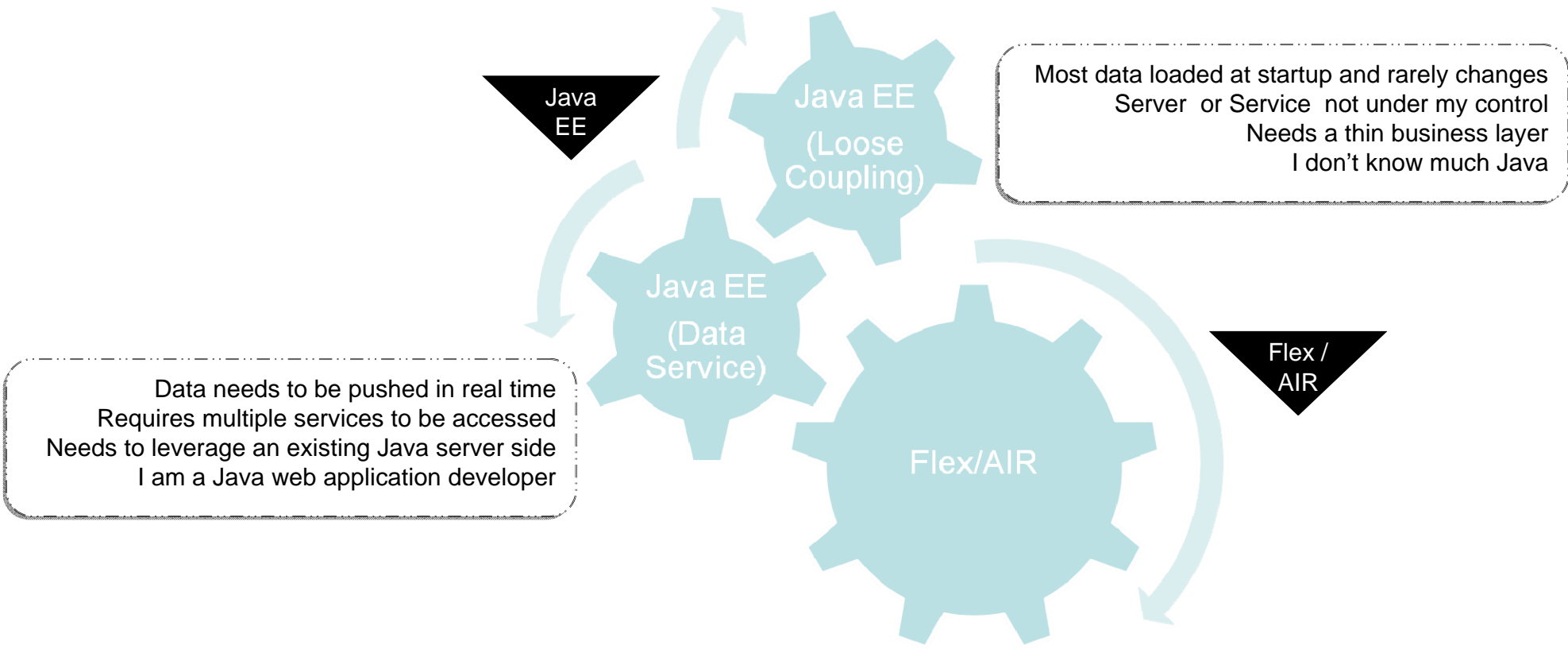


100 ft. View





Yet Another View





Scenario #1 -- POJO (1/4)

Steps to be followed:

- On the Java Server
 - Create your Java class / POJO
 - Configure a remote object destination
 - Place Java class in the classpath
- In the Flex/AIR Application
 - Create a RemoteObject
 - Create ActionScript object (~Java class)
 - Call a remote method on the RemoteObject

Assumption from here on – DS (BlazeDS/LCDS) is installed and working properly



Scenario #1 -- POJO (2/4)

services-config.xml

☞ remoting-config.xml

```
.... <destination id="MyRO">  
<properties>  
  <source>package.subpackage.MyRO</source>  
  <scope>application</scope>  
</properties> <adapter ref="java-object"/> </destination>
```

....



Scenario #1 -- POJO (3/4)

```
<mx:RemoteObject id="myService"  
destination="MyRO"  
result="myResultHandler(event)"  
fault="myFaultHandler(event)" />
```

MXML 

ActionScript 

```
var myService:RemoteObject = new RemoteObject();  
myService.destination = "MyRO";  
myService.addEventListener("result", myResultHandler);  
myService.addEventListener("fault", myFaultHandler);
```



Scenario #1 -- POJO (4/4)

Complex Situations

Multiple methods – different result and fault handlers

```
<mx:RemoteObject id="myService" destination="MyRO"
  fault="faultHandler(event)">
  <mx:method name="methodOne" result="methodOneHandler(event)"/>
  <mx:method name="methodTwo" result="methodTwoHandler(event)"/>
</mx:RemoteObject>
```

Pass arguments

- Explicit parameter passing
- Parameter binding (within MXML)



Scenario #2 -- EJB (1/2)

What's different here?

- Managed objects
- Default DS factory won't be able to instantiate an EJB object

What do we do?

- Create a custom factory
- Configure it



Scenario #2 -- EJB (2/2)

Create a Custom Factory

- Implement the `flex.messaging.FlexFactory` interface
- Create a `FactoryInstance`
- Use `FactoryInstance` to look up the object instance
- Once you have the object, call the method on it



Scenario #3 -- JMS (1/3)

Steps to be followed:

- On the Java Server
 - Create a JMS topic (or reuse an existing one)
 - Configure a messaging destination
 - Setup remote destination for object messages
 - Create a JMS message publisher / (consumer)
- In the Flex/AIR Application
 - Create a Message Consumer/Producer
 - Subscribe to the topic
 - Send and receive messages
 - Create ActionScript class(~Java class) if remoting



Scenario #3 -- JMS (2/3)

services-config.xml

☞ messaging-config.xml

```
<destination id="MessageDestination">  
<properties>
```

....

```
<jms>
```

```
<destination-type>Topic</destination-type>  
<message-type>javax.jms.TextMessage</message-type>  
<connection-factory>ConnectionFactory</connection-factory>  
<destination-jndi-name>topic/updateTopic</destination-jndi-name>  
<delivery-mode>NON_PERSISTENT</delivery-mode>
```

....

```
</jms>
```



Scenario #3 -- JMS (3/3)

Producer

```
<mx:Producer id="producer" destination=" MessageDestination"/>
```

```
import mx.messaging.Producer;  
var pro:Producer = new Producer();  
pro.destination = " MessageDestination";
```

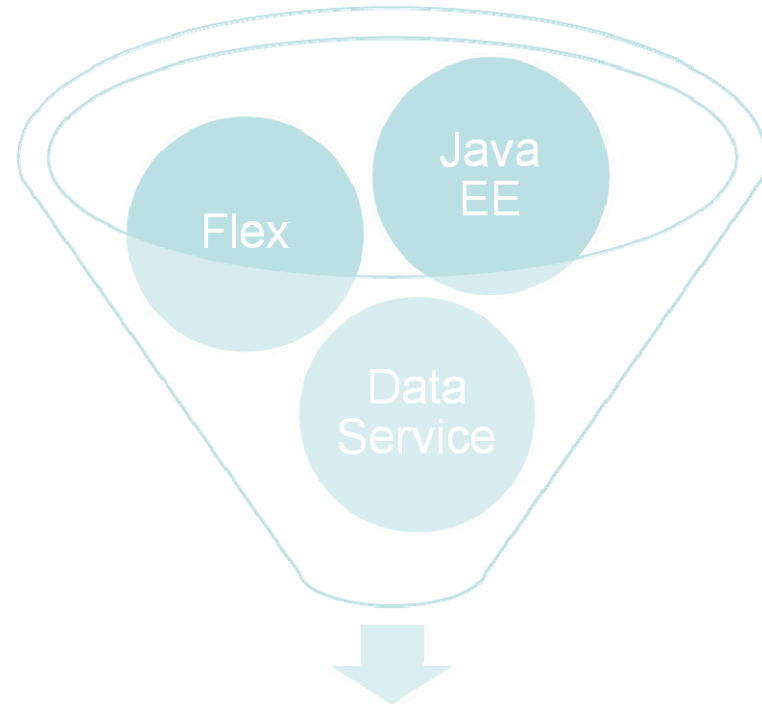
Consumer

```
<mx:Consumer id="consumer" destination=" MessageDestination"  
  message="myMessageHandler(event)"/>
```

```
import mx.messaging.Consumer;  
var consumer:Consumer = new Consumer();  
consumer.destination = " MessageDestination";
```



Back to the Hanger



Turbocharged RIA!



Contact Information



Shashank Tiwari

Email: stiwari@saventech.com

Web: www.saventech.com,
www.shanky.org

Blog: <http://www.oreillynet.com/pub/au/2799>

Phone: 877-SAVEN-21 / 847-9932001

Fax: 847-993-2016