

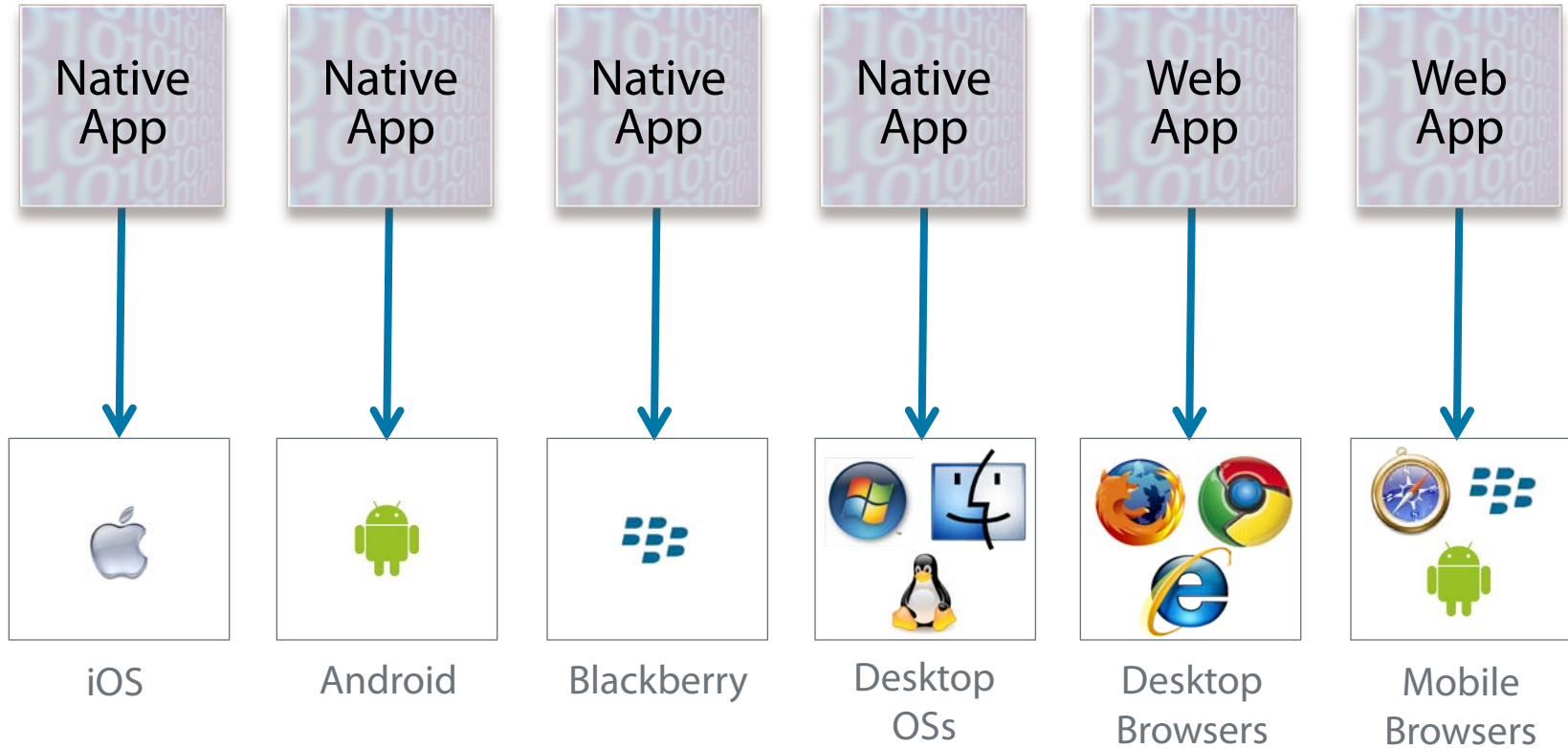


Rapid Mobile & Tablet App Development with Spring and Flex

@_JamesWard

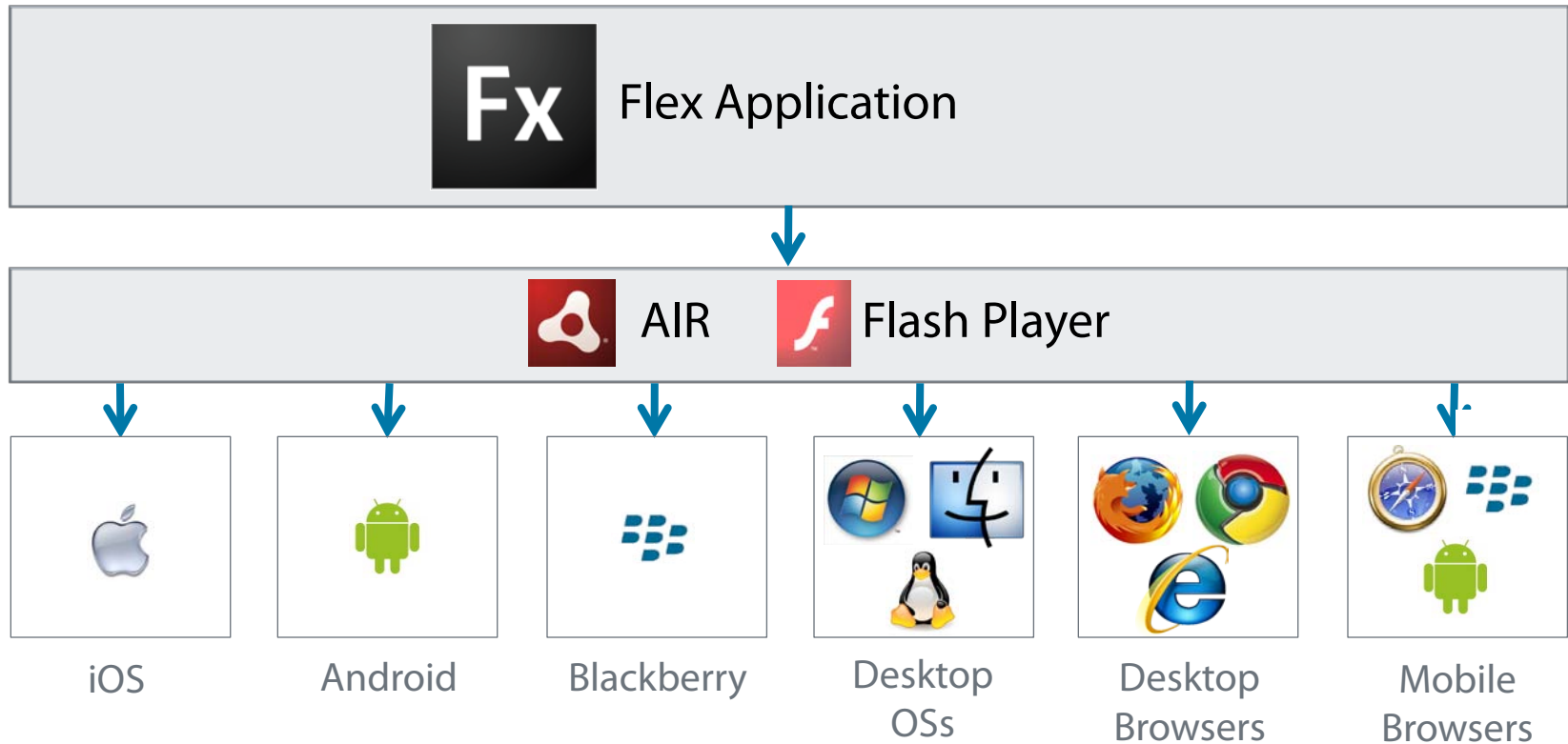


Native Apps and Web Apps - GALORE!



Goal of the Flash Platform

One Tool, One Language, One Codebase



Any Platform

Flash Platform - A complete system for web innovation

Applications, Content and Video



Tools to Design and Develop



Adobe® Flash® CS4 Professional



Adobe® Flash® Catalyst™



Adobe® Flash® Builder™

Framework



Adobe® Flex®

Runtimes



Adobe® AIR®



Adobe® Flash® Player

Services



Adobe Flash Platform Services

Servers

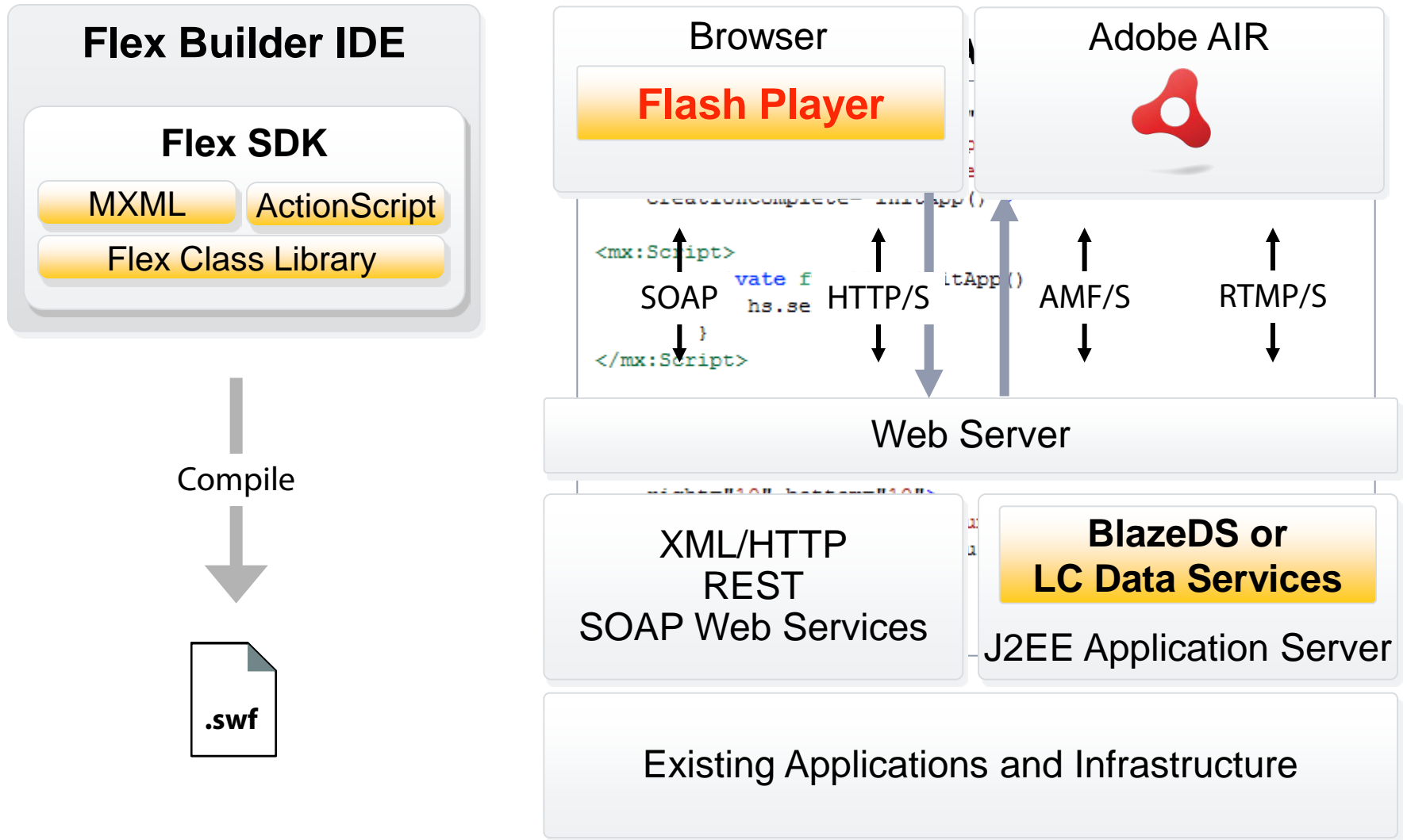


Adobe® Flash® Media Server Family



Adobe® LiveCycle® Data Services

How Flex Works



Deploying AIR Mobile Applications



App.apk



Android Market



Android Device



App.air



App.ipk



iTunes App Store



iOS Device



App.bar

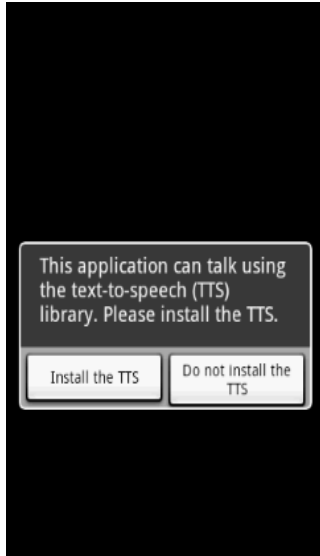


BlackBerry App World

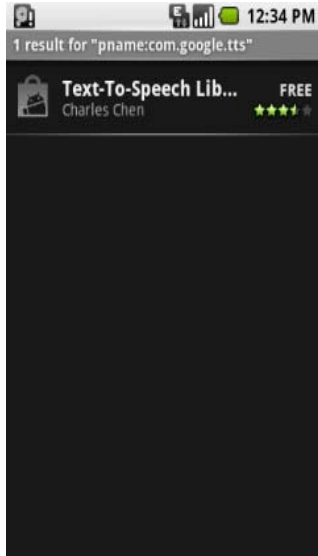


RIM Device

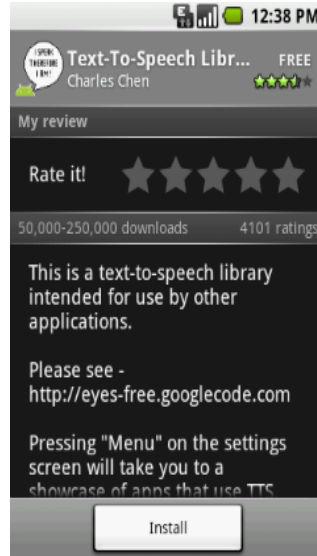
End User Experience for Android Users if AIR is not pre-installed



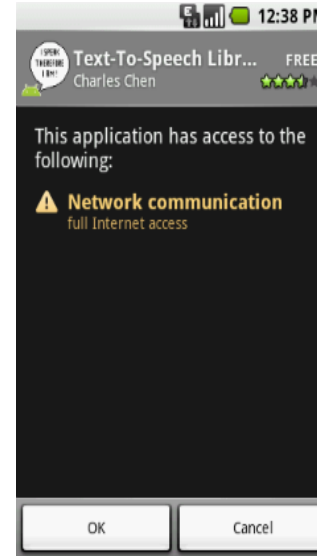
First time AIR app launches, user prompted to download shared library



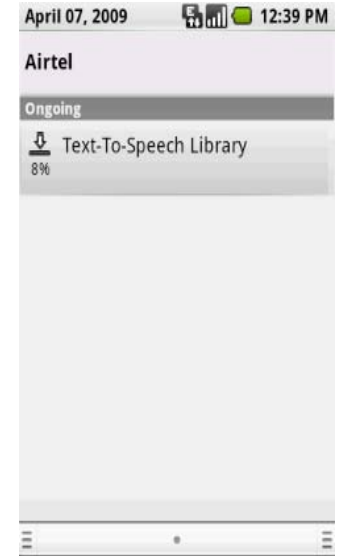
Android Market client directs user to AIR Runtime



User is given information on AIR Runtime



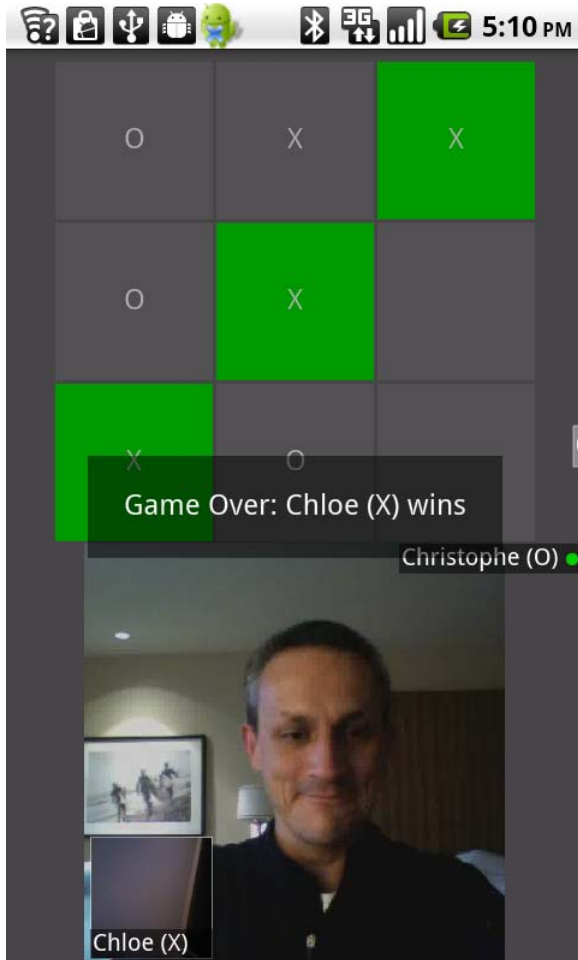
User confirms APIs used by AIR Runtime



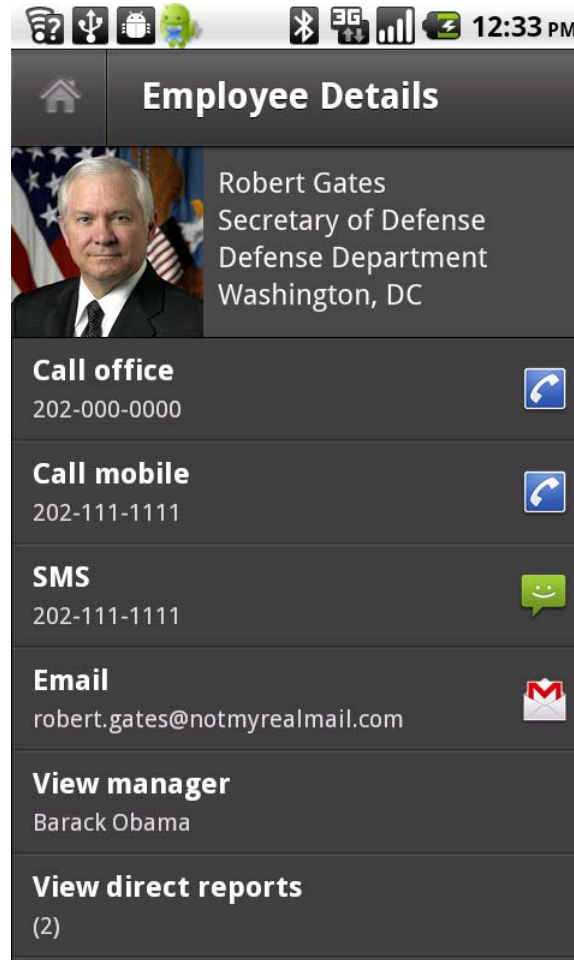
AIR Runtime is installed

Mobile Applications in "Hero"

s:Application



s:ViewNavigatorApplication

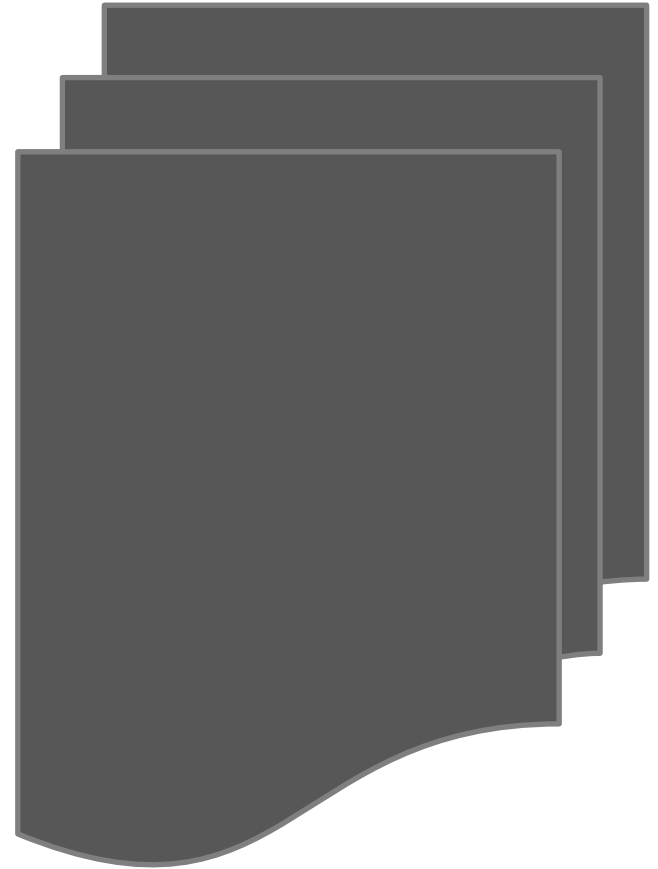


s:TabbedApplication



ViewNavigatorApplication and TabbedApplication

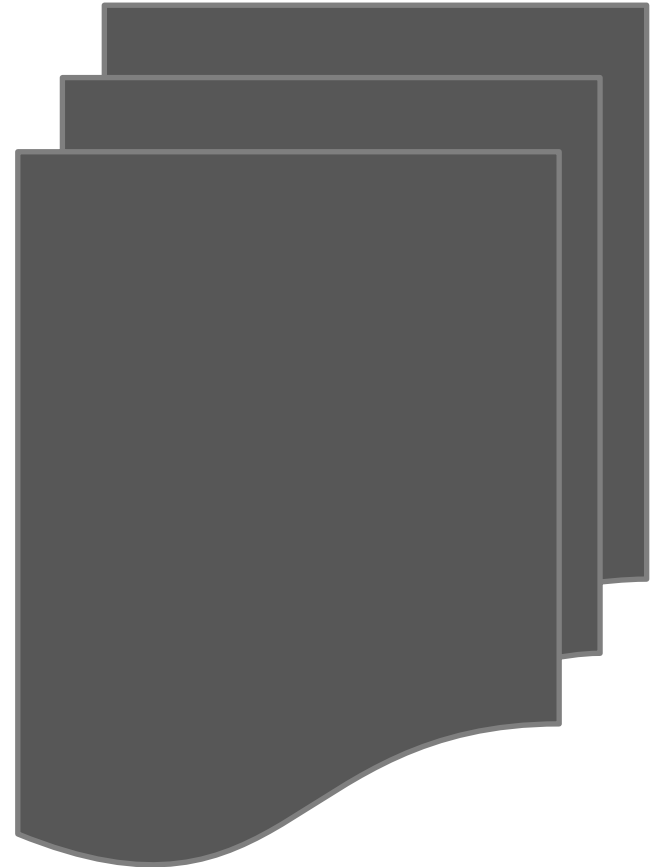
- View stack metaphor
 - View
 - ViewNavigator / TabbedViewNavigator
- Built-in mobile navigation experience
 - Action bar
 - Back button
 - Transitions
 - APIs



ViewNavigator API

- `navigator.pushView();`
- `navigator.popView();`
- `navigator.popToFirstView();`

- `navigator.activeView;`



ActionBar



navigationContent

titleContent

actionContent

```
<s:View xmlns:fx=http://ns.adobe.com/mxml/2009 ...  
  title="Expenses">
```

```
  <s:navigationContent>
```

```
    <s:Button icon="@Embed('assets/home.png')"/>
```

```
  </s:navigationContent>
```

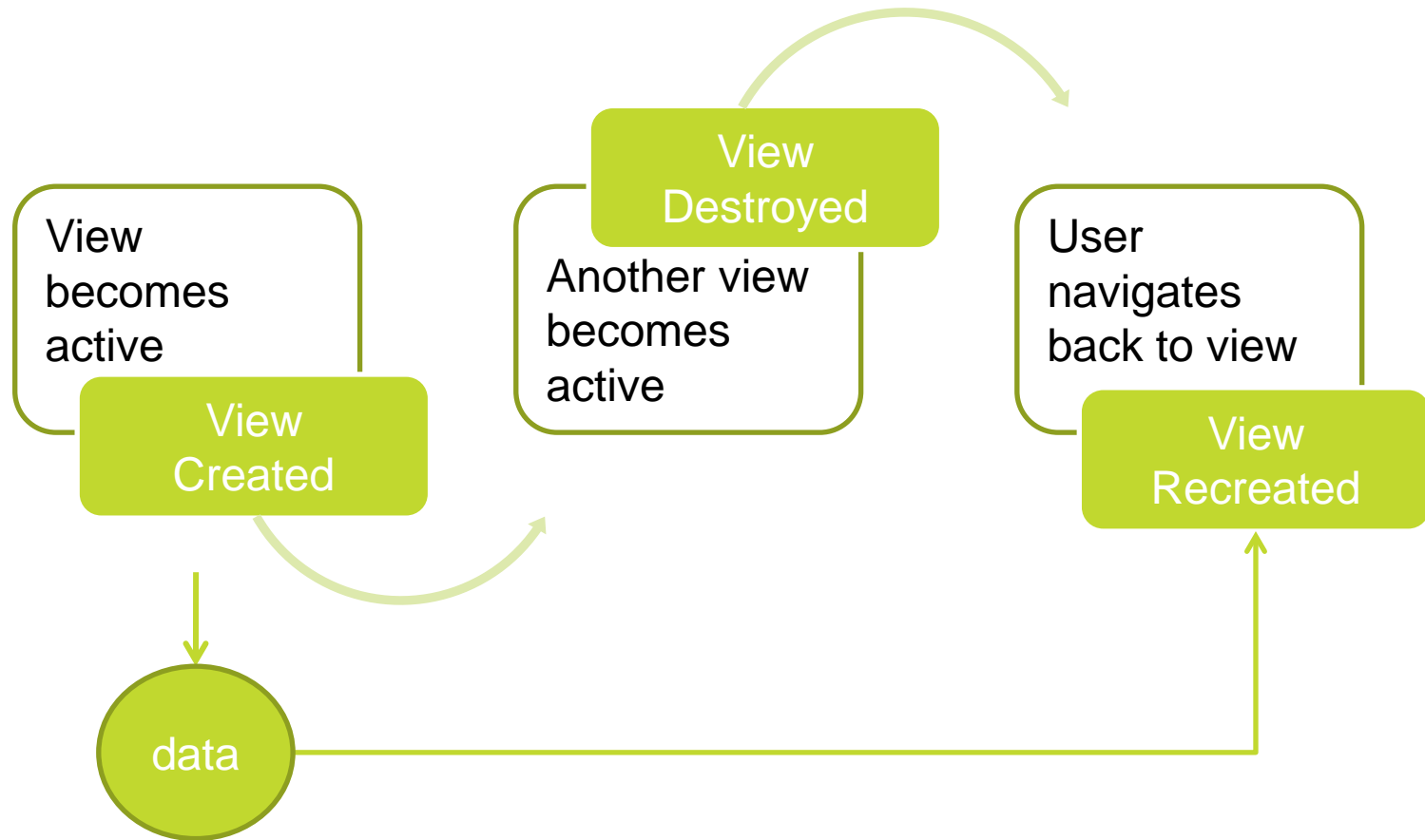
```
  <s:actionContent>
```

```
    <s:Button label="+"/>
```

```
  </s:actionContent>
```

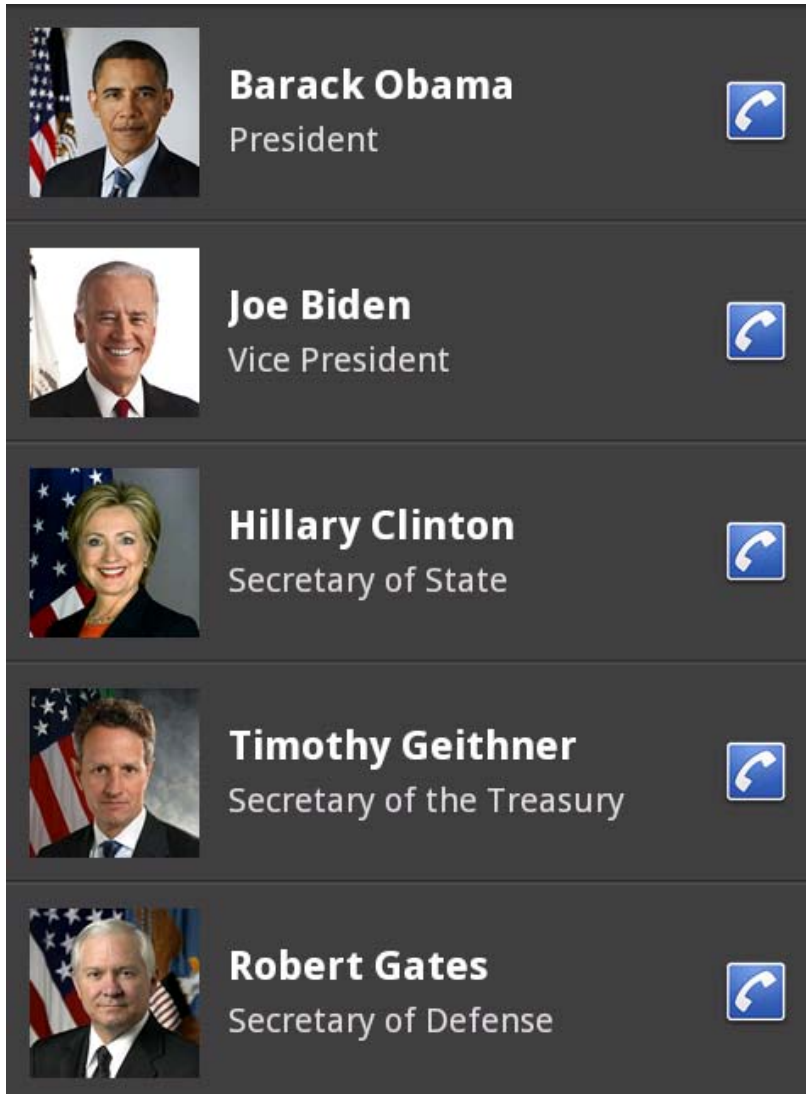
```
</s:View>
```

View Lifecycle



- Alternative option: `destructionPolicy="none"`

Mobile ItemRenderers



- MobileItemRenderer
- MobileIconItemRenderer

<s:MobileIconItemRenderer

labelField="lastName"

messageField="title"

iconField="picture"

decoratorClass="phoneIcon">

APIs

- GPS
- Accelerometer
- Multi-Touch / Gestures
- Full Screen / Screen Orientation
- Camera
- Native App Integration
- Local Database
- Stage Web View
- Remote Data

- Android Permission Required:

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />  
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

- Geolocation.isSupported

- g.addListener(GeolocationEvent.UPDATE, callbackFunction);

- GeolocationEvent:

- altitude
- heading
- horizontalAccuracy
- latitude
- longitude
- speed
- timestamp
- verticalAccuracy

Accelerometer APIs

- Accelerometer.isSupported
- a.addEventListener(AccelerometerEvent.UPDATE, callbackFunction);
- AccelerometerEvent:
 - accelerationX
 - accelerationY
 - accelerationZ
 - timestamp

Multi-Touch APIs

- Android manifest change:

```
<uses-feature android:required="true" android:name="android.hardware.touchscreen.multitouch"/>
```

- Gesture OR Multitouch --- Multitouch.inputMode

- MultitouchInputMode.GESTURE

- TransformGestureEvent

- GESTURE_PAN
- GESTURE_ROTATE
- GESTURE_SWIPE
- GESTURE_ZOOM

- GestureEvent.GESTURE_TWO_FINGER_TAP

- PressAndTapGestureEvent.GESTURE_PRESS_AND_TAP

- MultitouchInputMode.TOUCH_POINT

- TouchEvent

- TOUCH_BEGIN
- TOUCH_END
- TOUCH_MOVE
- TOUCH_OUT
- TOUCH_OVER
- TOUCH_ROLL_OUT
- TOUCH_ROLL_OVER
- TOUCH_TAP

Full Screen & Orientation

- `stage.displayState = StageDisplayState.NORMAL`
- `stage.displayState = StageDisplayState.FULL_SCREEN`
- `stage.displayState = StageDisplayState.FULL_SCREEN_INTERACTIVE`

- `StageOrientationEvent`:

```
stage.addEventListener(StageOrientationEvent.ORIENTATION_CHANGE, callbackFunction);
```

- `foo-app.xml`:

```
<initialWindow>
```

```
    <autoOrients>true</autoOrients>
```

Camera Access

- Camera - Raw camera feed

```
<uses-permission android:name="android.permission.CAMERA" />
```

```
<uses-feature android:name="android.hardware.camera" android:required="true"/>
```

```
<uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
```

```
<uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
```

- CameraUI - Native Camera App

```
var cameraUI:CameraUI = new CameraUI();
```

```
cameraUI.launch(MediaType.IMAGE);
```

- CameraRoll - Choose photos from the device's camera roll

```
var cameraRoll:CameraRoll = new CameraRoll();
```

```
cameraRoll.browseForImage();
```

```
cameraRoll.addBitmapData(bd);
```

Native App Integration

- Open Email, Browser, Maps, Phone, or SMS
 - `navigateToURL(new URLRequest('mailto:foo@bar.com'));`
 - `navigateToURL(new URLRequest('http://www.jamesward.com'));`
 - `navigateToURL(new URLRequest('http://maps.google.com/'));`
 - `navigateToURL(new URLRequest('tel:1234567890'));`
 - `navigateToURL(new URLRequest('sms:1234567890'));`

Local DB APIs

- SQLite - Nothing different than AIR for the Desktop
- SQLConnection
 - Sync and/or Async connections
- SQLStatement
 - Prepared Statements:

```
var stmt:SQLStatement = new SQLStatement();
```

```
stmt.sqlConnection = FlexGlobals.topLevelApplication['sqlConnection'];
```

```
stmt.text = "INSERT into giberish values(:giberish)";
```

```
stmt.parameters[":giberish"] = g.text;
```

```
stmt.execute();
```

StageWebView

- Requires Android INTERNET permission:

```
<uses-permission android:name="android.permission.INTERNET" />
```

- WebStageView.isSupported

```
var swv:StageWebView = new StageWebView();
```

```
swv.viewPort = new Rectangle(0, stage.height - height, width, height);
```

```
swv.stage = stage;
```

```
swv.loadURL("http://www.jamesward.com");
```

Remote Data

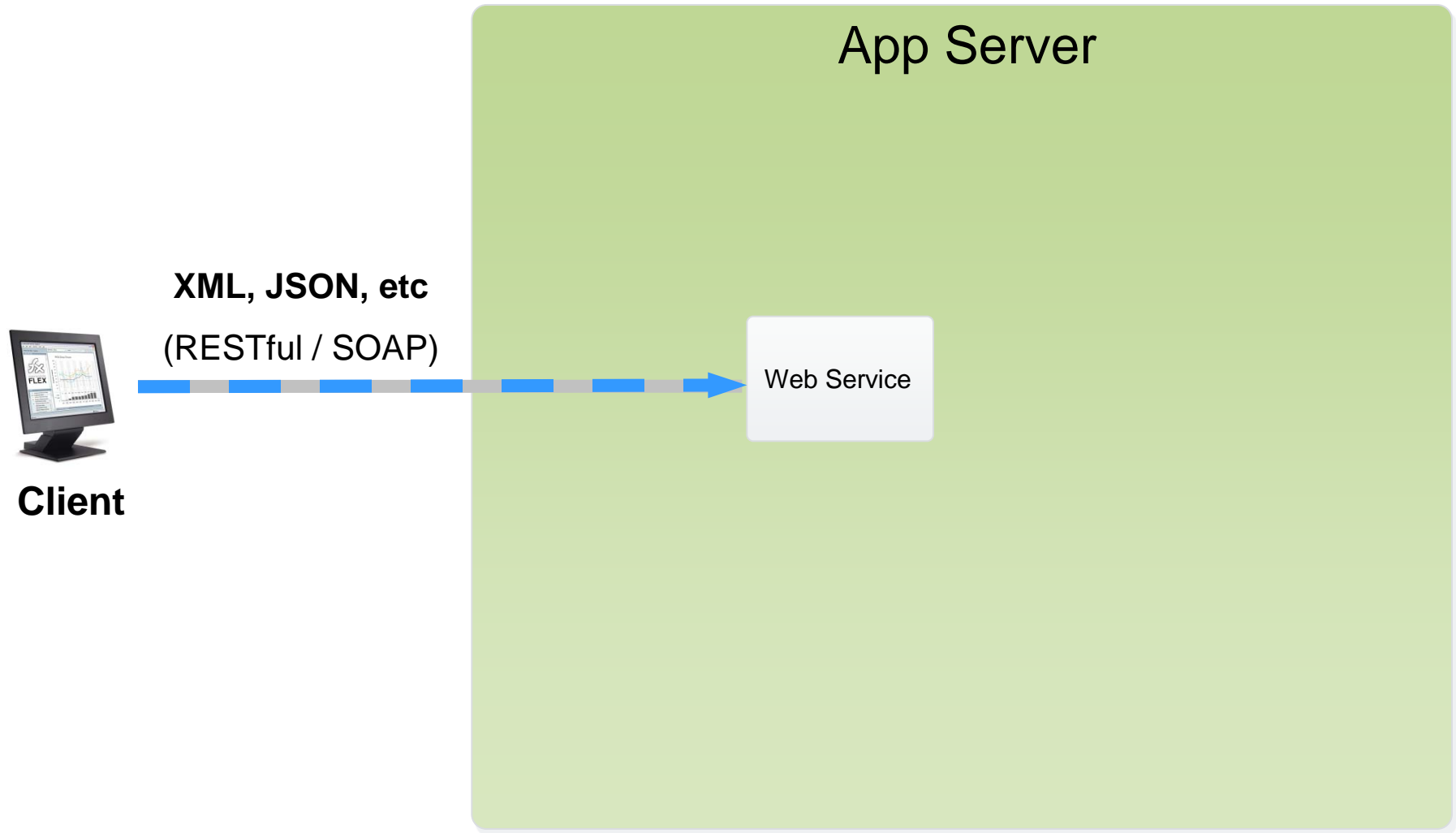
- Android Permission Required:

```
<uses-permission android:name="android.permission.INTERNET"/>
```

- Standard Flex Networking Libraries:

- HTTPService
- WebService
- RemoteObject

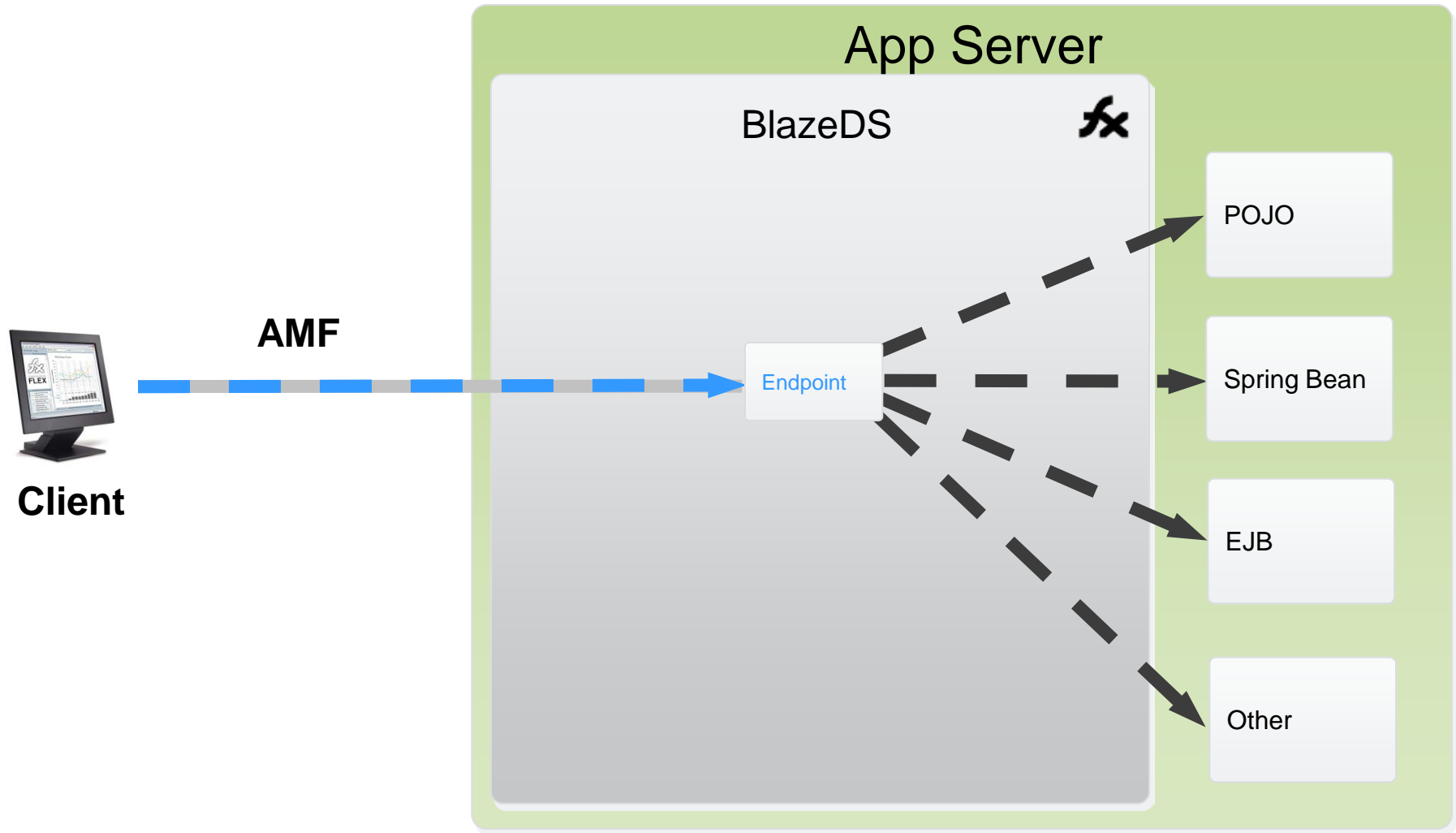
Flex with Java via Text-based Serialization



Remoting Performance Example



Flex with Java via Binary Serialization



Open Source BlazeDS

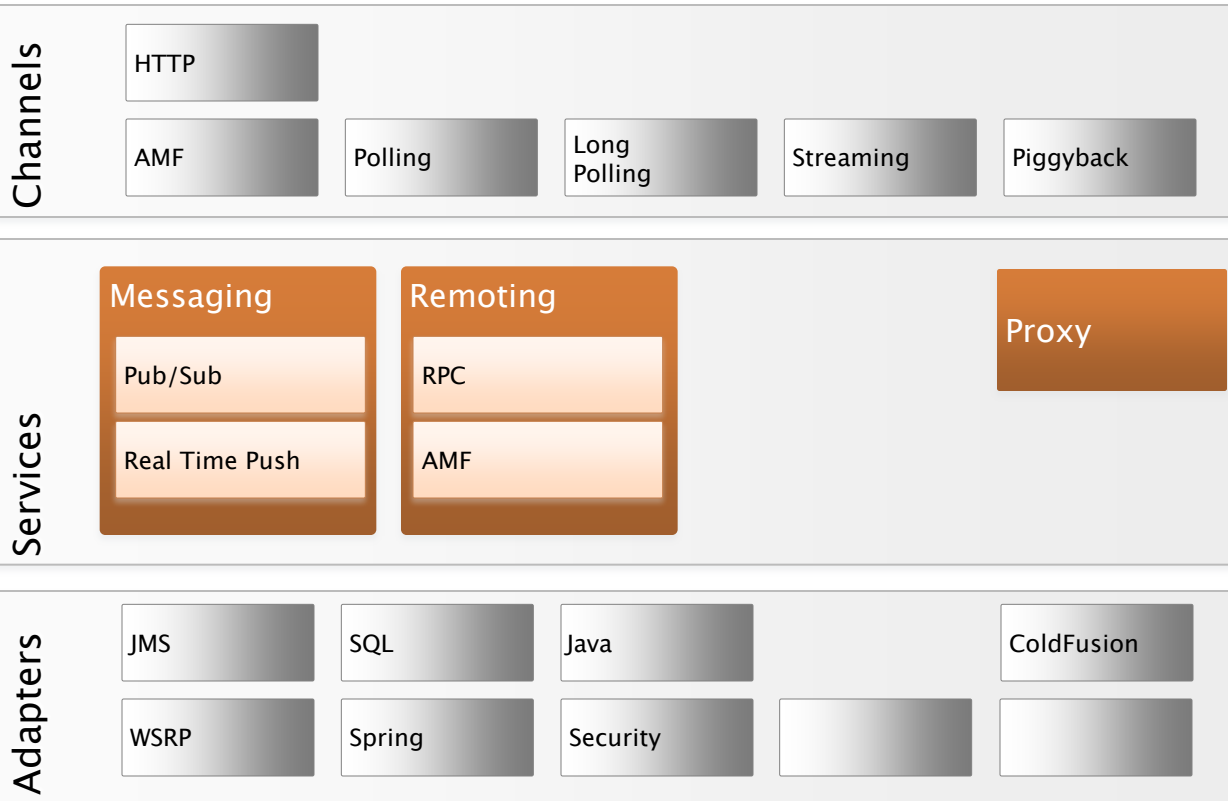
Client-side APIs

RemoteObject

Producer

Consumer

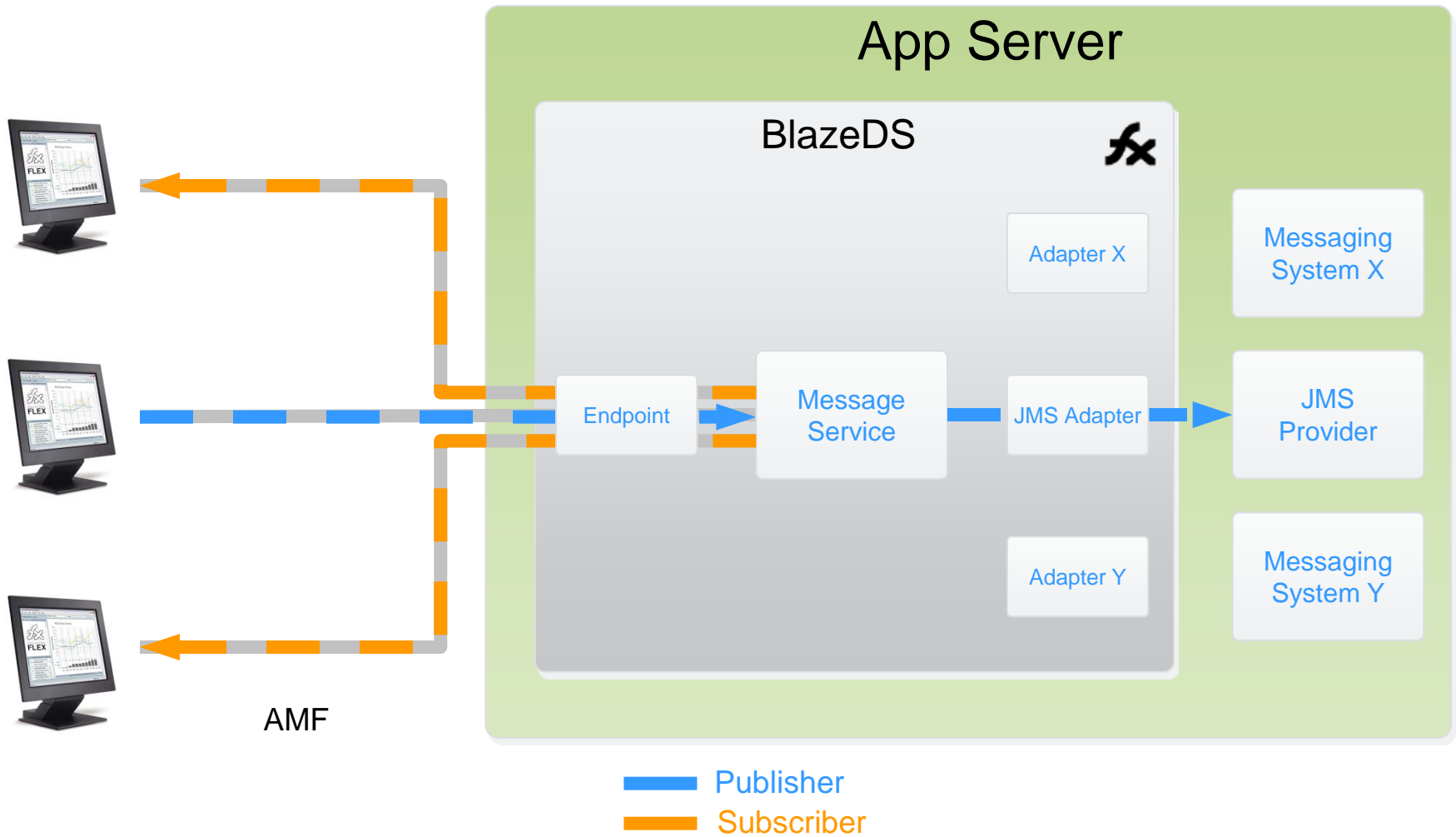
Server-side Infrastructure



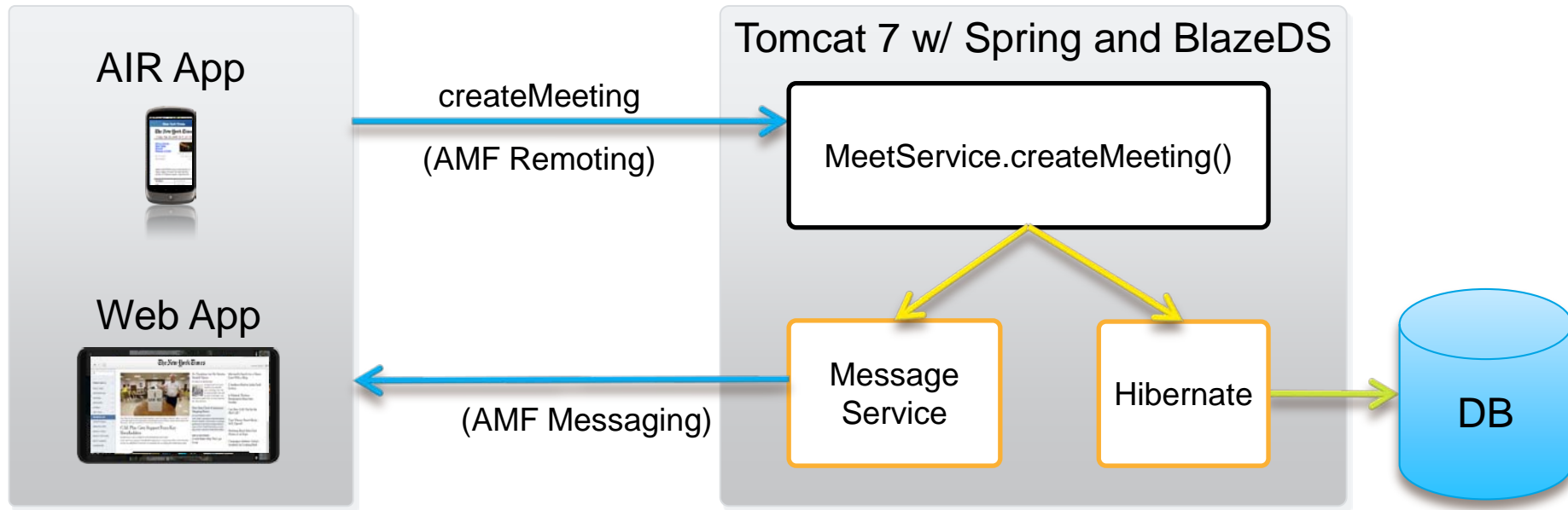
Setup Spring + BlazeDS

- Add dependencies to WEB-INF/lib
- web.xml org.springframework.web.servlet.DispatcherServlet
- Spring config:
 - <flex:message-broker>
 - <flex:remoting-service default-channels="my-amf" />
- BlazeDS channel config: WEB-INF/flex/services-config.xml
- Configure beans:
 - @RemotingDestination

Flex with Java via Messaging



Putting it All Together - Middle Meetup



www.middlemeetup.com