Leveraging ColdFusion with Adobe Integrated Runtime (AIR) Applications

Rupesh Kumar (http://coldfused.blogspot.com)

Computer Scientist

Adobe Systems

2007 Adobe Systems Incorporated. All Rights Reserved.

Adobe

Agenda

- Introduction to AIR
- Where ColdFusion fits in
- ColdFusion as Data-Provider
- Flex with ColdFusion in AIR
- HTML/Ajax with ColdFusion in AIR
- ColdFusion as Presentation Layer
- Future Directions
- Q&A

2007 Adobe Systems Incorporated. All Rights Reserved.

Introduction to AIR

Adobe® Integrated Runtime (AIR™) is a cross-platform runtime that allows you to leverage your existing web development skills to build and deploy Rich Internet Applications (RIAs) to the desktop

2007 Adobe Systems Incorporated. All Rights Reserved.

What it provides

- Flash runtime + WebKit + pdf renderer
  - No browser dependency
- Leverages existing web development skills
  - HTML, Javascript, Ajax, Flex, Flash
  - Much easier than traditional desktop application development
- Application like any other desktop application
- Cross platform installer
- Rich Set of API to work with File, Network, local database, window, menu etc.
- Much more responsive than web application as UI is already present
- Ability to go offline and then sync data when it goes online.
  - Local storage on disk or local database SQLite

2007 Adobe Systems Incorporated. All Rights Reserved.

AIR application structure

- Its a zip file !
- Application Contents
  - HTML, javascript, css, images, swf, configuration files etc..
- Application descriptor

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://ns.adobe.com/air/application/1.0.M4"
      appId="samples.ScorpioMail" version="0.1">

      <name>Demo</name>
      <rootContent systemChrome="standard" visible="true“
         width="1024" height="800">helloworld.htm
      </rootContent>
</application>
```

- Packaged together as .air file
- A simple AIR application

2007 Adobe Systems Incorporated. All Rights Reserved.

Type of AIR Applications

- Flash only applications
- Flash based with HTML content
- HTML/Javascript only
- HTML/Javascript with flash content

2007 Adobe Systems Incorporated. All Rights Reserved.

Where does ColdFusion Fit in

- AIR does not have CFML engine
  - cfm/cfc can not be packaged inside AIR applications.
- Traditionally ColdFusion application generates the UI
- In AIR application, generally the UI is pre-designed and packaged

- How can it use ColdFusion then??
  - Use ColdFusion applications directly
  - ColdFusion as data provider
  - ColdFusion as data provider as well as UI provider

2007 Adobe Systems Incorporated. All Rights Reserved.

Use Web Applications directly

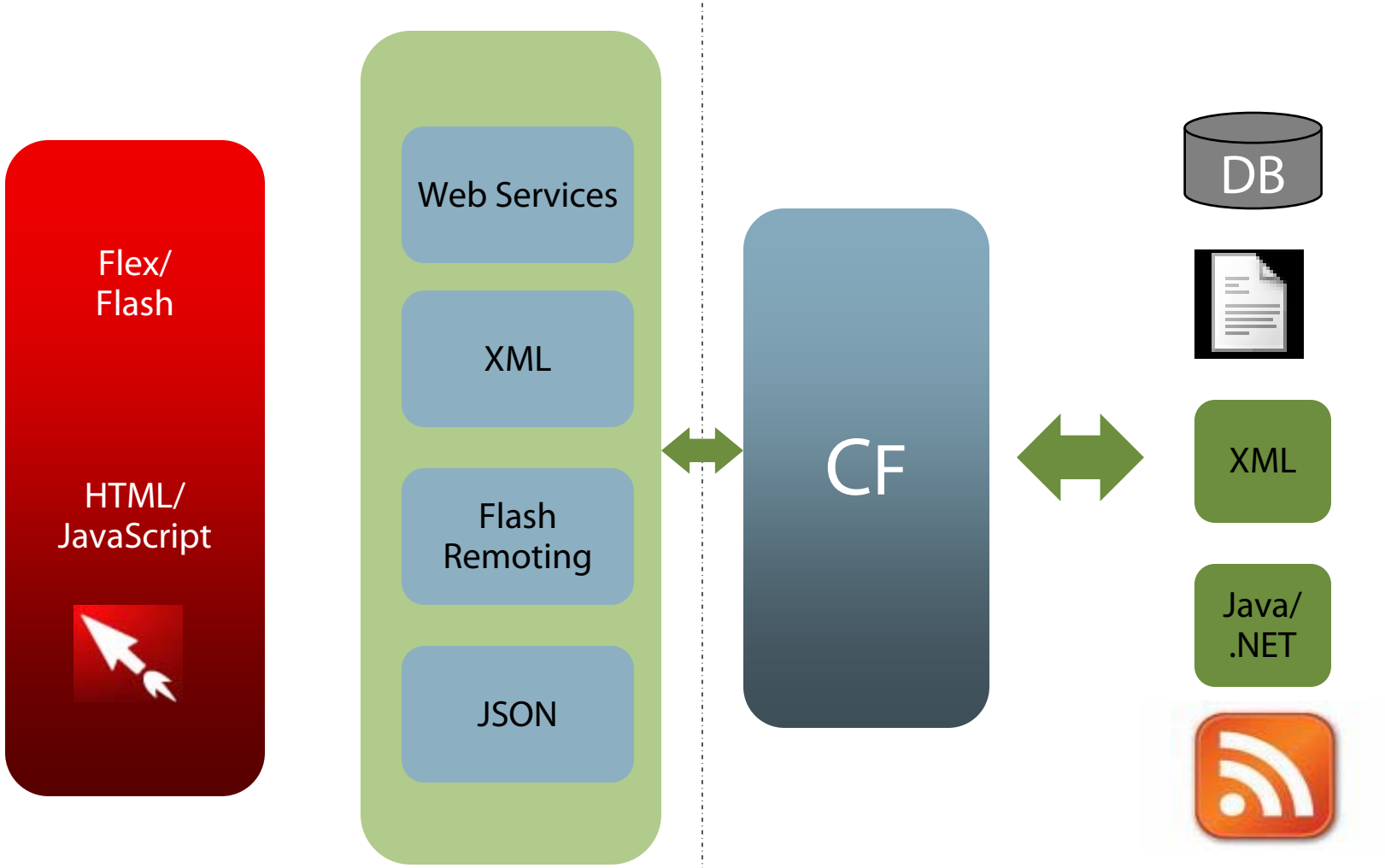
- ColdFusion is the fastest and easiest way to develop web application.
- Use web application directly without any modification.
- The root content of the AIR application points to online app's URL.

```
<html>
  <frameset cols="100%">
   <frame
      src=http://localhost:8500/CFIDE/administrator/index.cfm
      frameborder="0">
   </frameset>
</html>
```

```
<mx:HTML id="html" width="100%" height="100%"
  location="http://www.adobe.com/" />
```

2007 Adobe Systems Incorporated. All Rights Reserved.

ColdFusion as data provider



2007 Adobe Systems Incorporated. All Rights Reserved.

Flex Applications in AIR

Interaction with ColdFusion can happen through

- Flash Remoting
- Web Service
- HTTP Service

2007 Adobe Systems Incorporated. All Rights Reserved.

Flex-CF in AIR – Flash Remoting

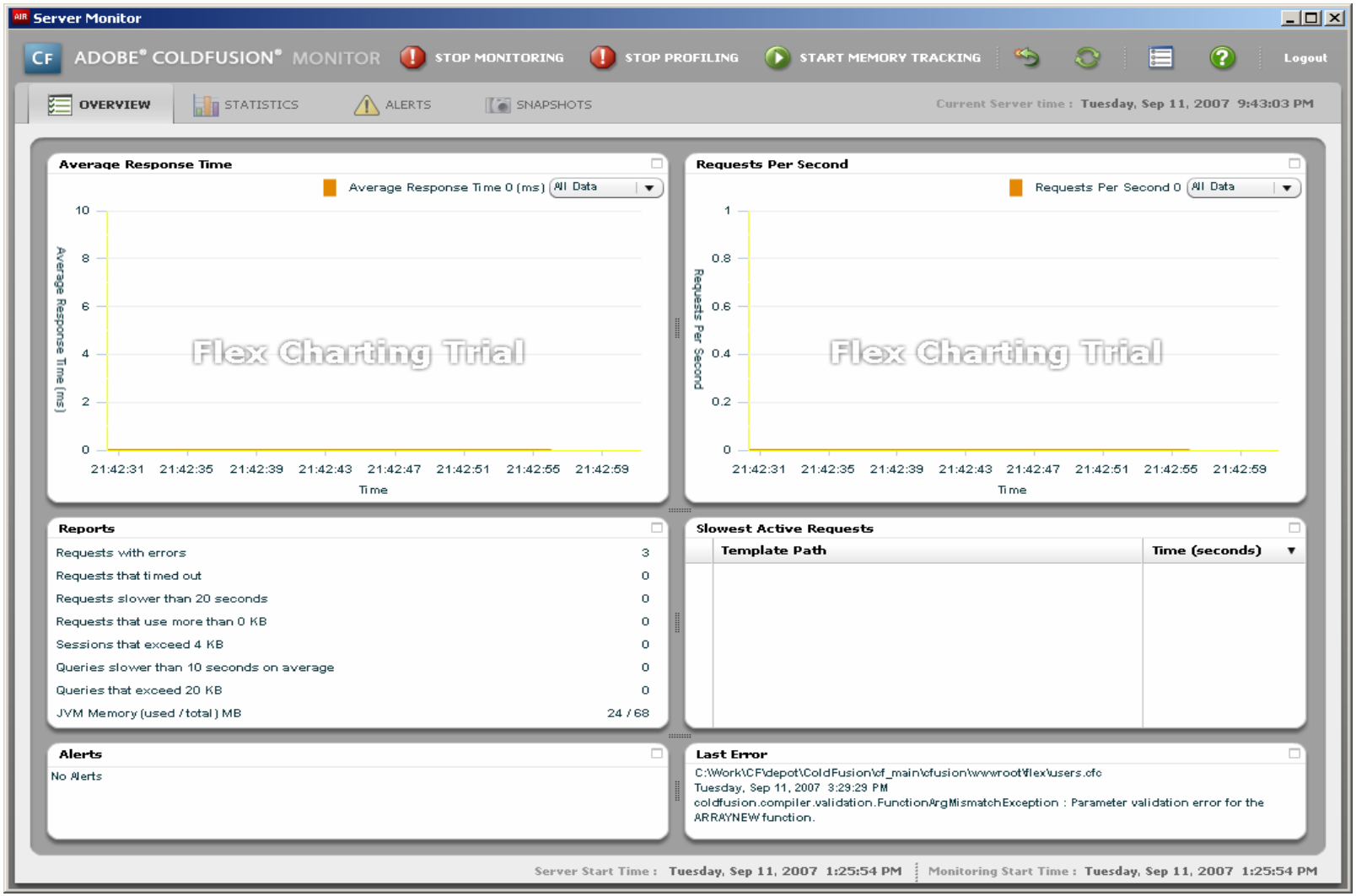
```
<mx:RemoteObject id="userRequest"
      destination="ColdFusion"
      source="flex.users">
  <mx:method name="returnRecords"
      result="returnCFCHandler(event)"
      fault="mx.controls.Alert.show(event.fault.faultString)"
  />

  <mx:method name="insertRecord" result="insertCFCHandler()“
      fault="mx.controls.Alert.show(event.fault.faultString)“
  />

</mx:RemoteObject>
```

2007 Adobe Systems Incorporated. All Rights Reserved.

Example : Server Monitor in AIR



2007 Adobe Systems Incorporated. All Rights Reserved.

Flex-CF in AIR – Web Service

```
<mx:WebService id="userRequest"
      wsdl="http://localhost:8500/flex/users.cfc?wsdl">

  <mx:operation name="returnRecords" resultFormat="object"
     fault="mx.controls.Alert.show(event.fault.faultString)"
     result="remotingCFCHandler(event)"/>

  <mx:operation name="insertRecord"
     result="insertCFCHandler()"
     fault="mx.controls.Alert.show(event.fault.faultString)"
  />

</mx:WebService>
```

2007 Adobe Systems Incorporated. All Rights Reserved.

Flex-CF in AIR – HttpService

```
<mx:HTTPService id="userRequest"
  url="http://localhost:8500/flex/flexapp.cfm"
  useProxy="false" method="GET">
</mx:HTTPService>

<mx:DataGrid id="dgUserRequest" x="22" y="128"
dataProvider="{userRequest.lastResult.users.user}">

   <mx:columns>
      <mx:DataGridColumn headerText="Mail ID"
                         dataField="emailaddress"/>
      <mx:DataGridColumn headerText="User Name"
                         dataField="username"/>
  </mx:columns>

</mx:DataGrid>
```

- Result coming in XML.

2007 Adobe Systems Incorporated. All Rights Reserved.

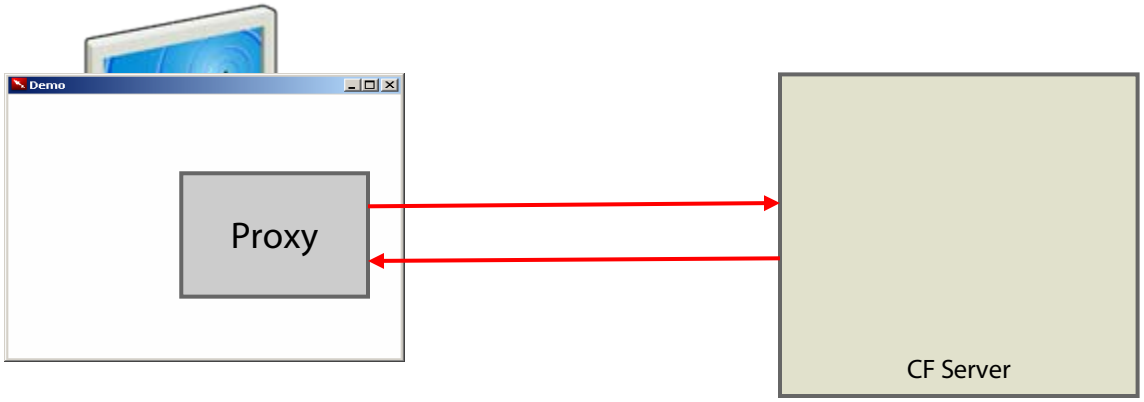
HTML/JavaScript (AJAX) in AIR

- Most of the UI packaged inside AIR app.
- Uses ColdFusion as data provider
  - CFAjaxProxy
  - XMLHttpRequest
  - Plain HTTP (cfm returning the data in XML, JSON, text format)
- Data formats
  - Plain Text
  - XML
  - JSON
    - Var aNames = [“Benjamin”, “Micheal”, “Scott”];
    - Var aCar = {“color” : “red”,“doors” : 4};
- Example – SerializeJSON()

2007 Adobe Systems Incorporated. All Rights Reserved.

CFAjaxProxy in AIR

- Makes cfc available inside AIR application
- Creates a JavaScript proxy for a ColdFusion component
- Generates a proxy function for every remote function on the cfc




```
<cfajaxproxy cfc = “books" jsclassname = “jsbooks“>
<script>
	var b = new jsbooks();
	var details = b.getBookDetails();
</script>
```


2007 Adobe Systems Incorporated. All Rights Reserved.

Using CFAjaxProxy

```
ColdFusion.Ajax.importTag('CFAJAXPROXY');
 var _cf_states=ColdFusion.AjaxProxy.init(
      'http://localhost:8500/ajax/ajaxproxy/states.cfc','states');

_cf_states.prototype.getCities=function(state)
  { return ColdFusion.AjaxProxy.invoke(
              this, "getCities", {state:state});
  };

_cf_states.prototype.getStates=function()
  { return ColdFusion.AjaxProxy.invoke(this, "getStates", {});};



function getStates(){
     var s = new states();
     s.setCallbackHandler(fillStates);
     s.setErrorHandler(errorHandler);
     s.getStates();
}
```

2007 Adobe Systems Incorporated. All Rights Reserved.

Using ColdFusion AJAX for AIR

- Use ColdFusion AJAX Tags to generate the UI
- Helps in developing Ajax application without writing much Javascript
- Readymade controls like layout, panels, pods, window, grid, tree.

- Write the cfm using CF ajax tags.
  - Can also use the CF-Ajax wizard
  - Should not have any business logic in it
- Execute the page and save the htm
- In this htm, change the bind urls to absolute URLs
  - Relative URLs are treated as local to AIR app
- Add the custom code required
- Package this htm along with the ajax scripts shipped with CF in AIR application

2007 Adobe Systems Incorporated. All Rights Reserved.

Taking Applications offline

2007 Adobe Systems Incorporated. All Rights Reserved.

Taking applications offline

- Identify features of application that can be made offline
- Identify what data and how much data should be made available offline
- Storing offline data
  - File system
  - Local DB (SQLite)
- Modality
  - Modal – Distinct offline/online mode. User driven. Simple to implement.
  - Modeless – Seamless offline/online. Most of the data stored locally. Better user experience.
- Synchronization
  - Manual Synch
  - Background client driven synch
  - Server driven synch
    - LiveCycle Data Services

2007 Adobe Systems Incorporated. All Rights Reserved.

Using SQLite

Checking application status

```
var request = new air.URLRequest("http://localhost:8500/");
var loader = new air.URLLoader();
loader.addEventListener(air.Event.COMPLETE, online);
loader.addEventListener(air.IOErrorEvent.IO_ERROR,
offline);
```

Populate data

```
if(online){
    var mailcfc = new MailCFC();
    fillGrid(mailcfc.getInboxMails());
}else{
    stmt = new air.SQLStatement();
    stmt.addEventListener( air.SQLEvent.RESULT, fillGridDB);
    stmt.addEventListener(air.SQLErrorEvent.ERROR,
handleErr);
    stmt.sqlConnection = db;
    stmt.text = "SELECT * FROM mail";
    stmt.execute();
}
```

2007 Adobe Systems Incorporated. All Rights Reserved.

Taking Applications offline…

More at Sneak Peek

2007 Adobe Systems Incorporated. All Rights Reserved.

Future Possibilities ?

- Some random thoughts !!!
- Application to be written using CFAjax tags with different binds.
- Either a wizard or utility
- Can take user inputs as in
  - which cfms to include, which files to include
  - What data to persist locally and how to synch it.
  - User's implementation of persistence and synching.
  - Application name and configuration
- Smartly generate the appropriate code for AIR.
- Standard javascript files that CF uses can be made part of AIR runtime.
- Wizard for creating CRUD application with offline capabilities.
- Ideas/Suggestions??

2007 Adobe Systems Incorporated. All Rights Reserved.

Q&A

rukumar@adobe.com

http://coldfused.blogspot.com

2007 Adobe Systems Incorporated. All Rights Reserved.

Better by Adobe™

2007 Adobe Systems Incorporated. All Rights Reserved.