

# ColdFusion Powered AIR

Ryan Stewart

Platform Evangelist



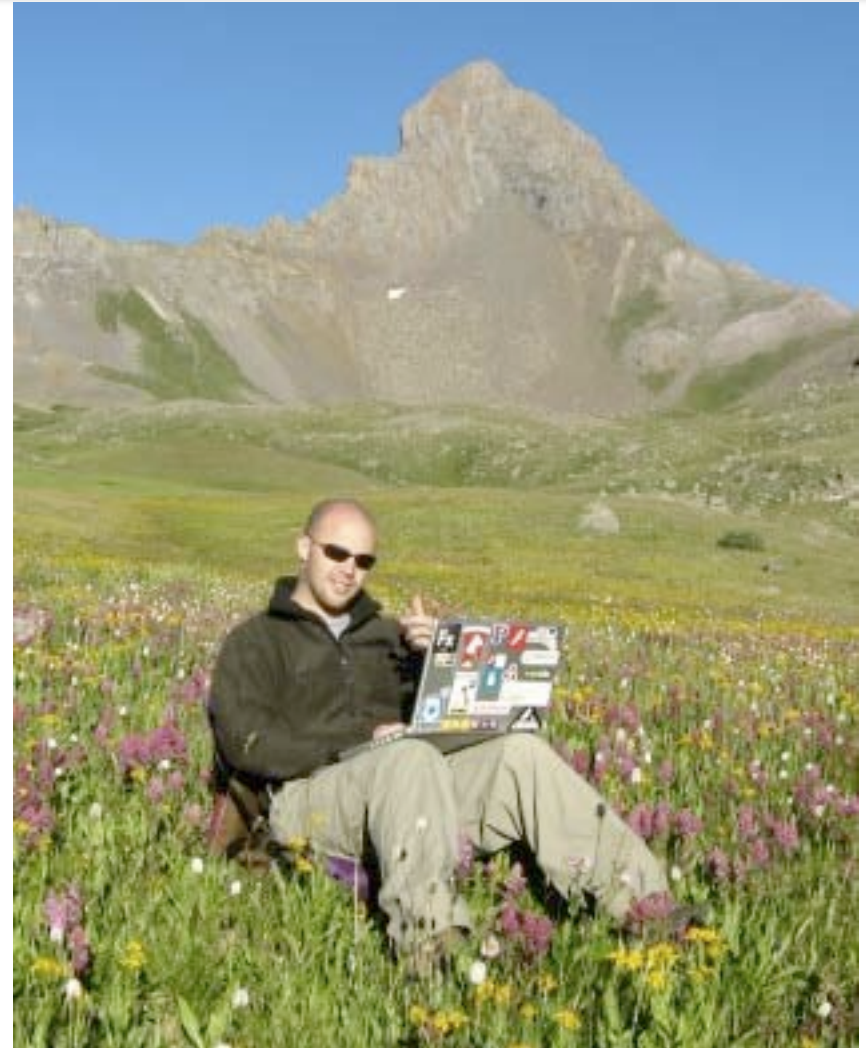
# About Ryan

- Platform Evangelist
- At Adobe for ~2 years
- ColdFusion, Flex developer
- From Seattle, Washington

[ryan@adobe.com](mailto:ryan@adobe.com)

[blog.digitalbackcountry.com](http://blog.digitalbackcountry.com)

Twitter/brightkite: ryanstewart



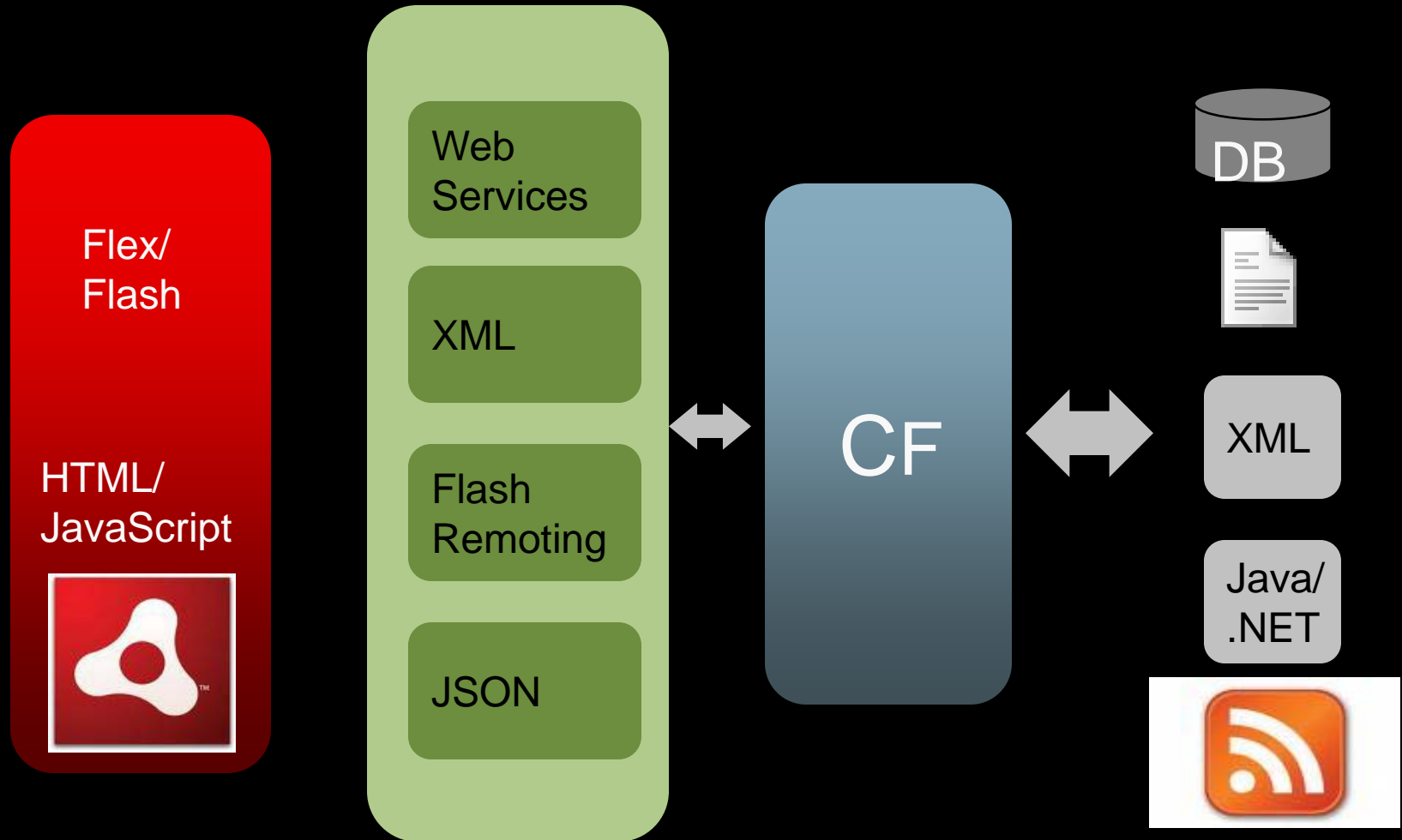
- A real quick intro to AIR
- The Big Picture
- Communication between AIR application and ColdFusion
- AIR Security Model
- Accessing ColdFusion from AIR
- Adobe Employee Directory AIR Application
- A Full fledged email AIR application with CF backend
- Sneaks on the future of AIR and CF

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

- Flash runtime + WebKit + pdf renderer
- Cross platform installer
- Rich Set of API to work with File, Network, local database, window, menu etc.
- Ability to go offline and then sync data when it goes online.
  - Local storage on disk or local database SQLite

- Brings web applications to the desktop
- Leverages existing web development skills
- Much easier than traditional desktop application development
- AIR Application behaves like any other desktop application
- Rich Set of API to work with File, Network, local database, window etc
- Provides the ability for the applications to go offline and then sync data when it goes online

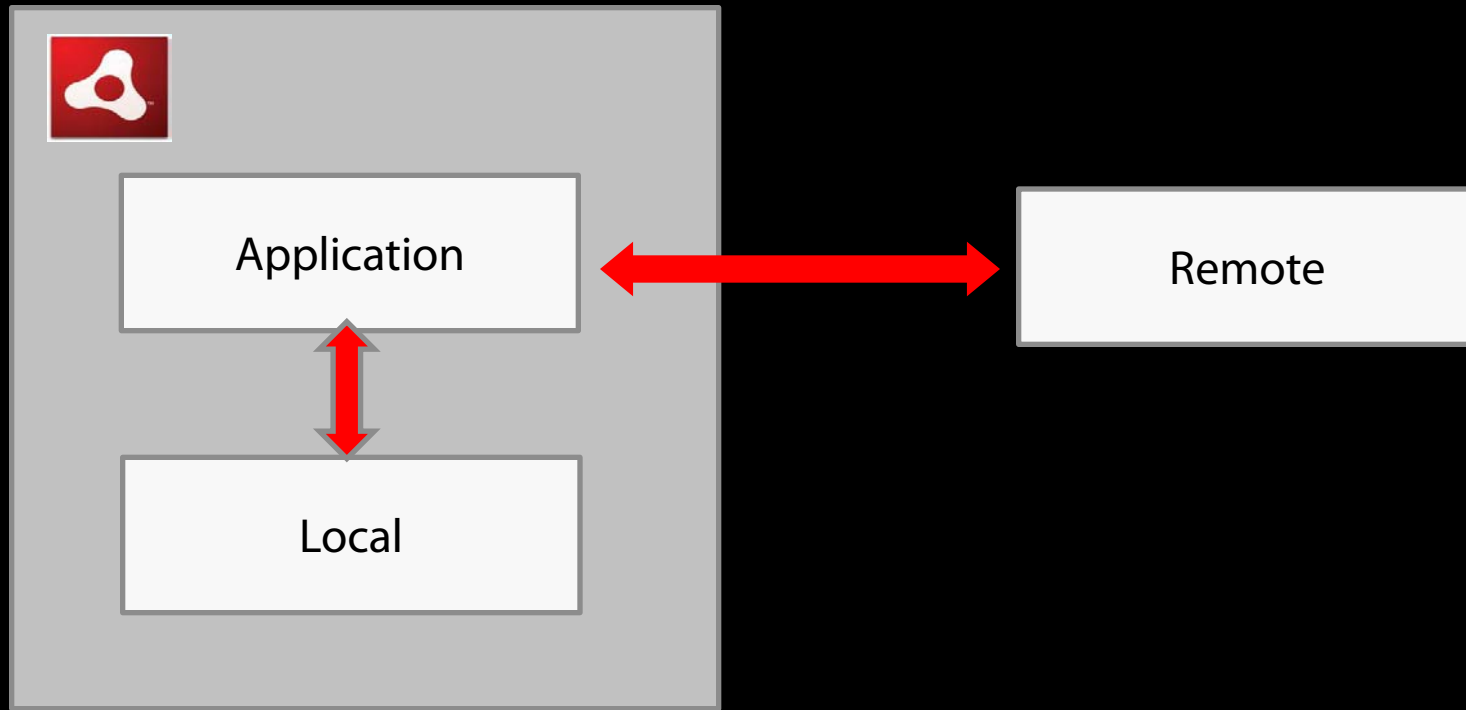
# Communication between AIR app and Server



- AIR security model is an evolution of the Flash Player security model
- AIR applications run with the same user privileges as native applications
- AIR runtime provides memory management



- Logical security groupings called sandboxes
- Based on the origin of the file, internal or external

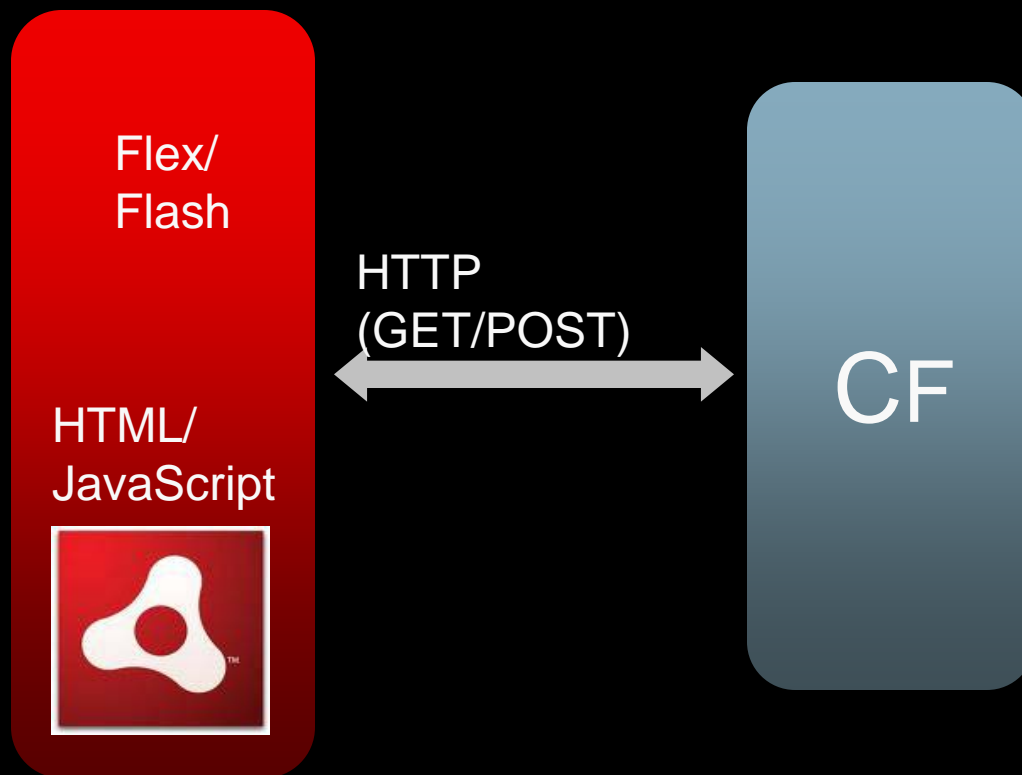


- All files within the application directory tree are assigned to the application sandbox when the application is run.
- Content in the application sandbox is blessed with the full privileges
- Content in the application security sandbox has access to AIR APIs

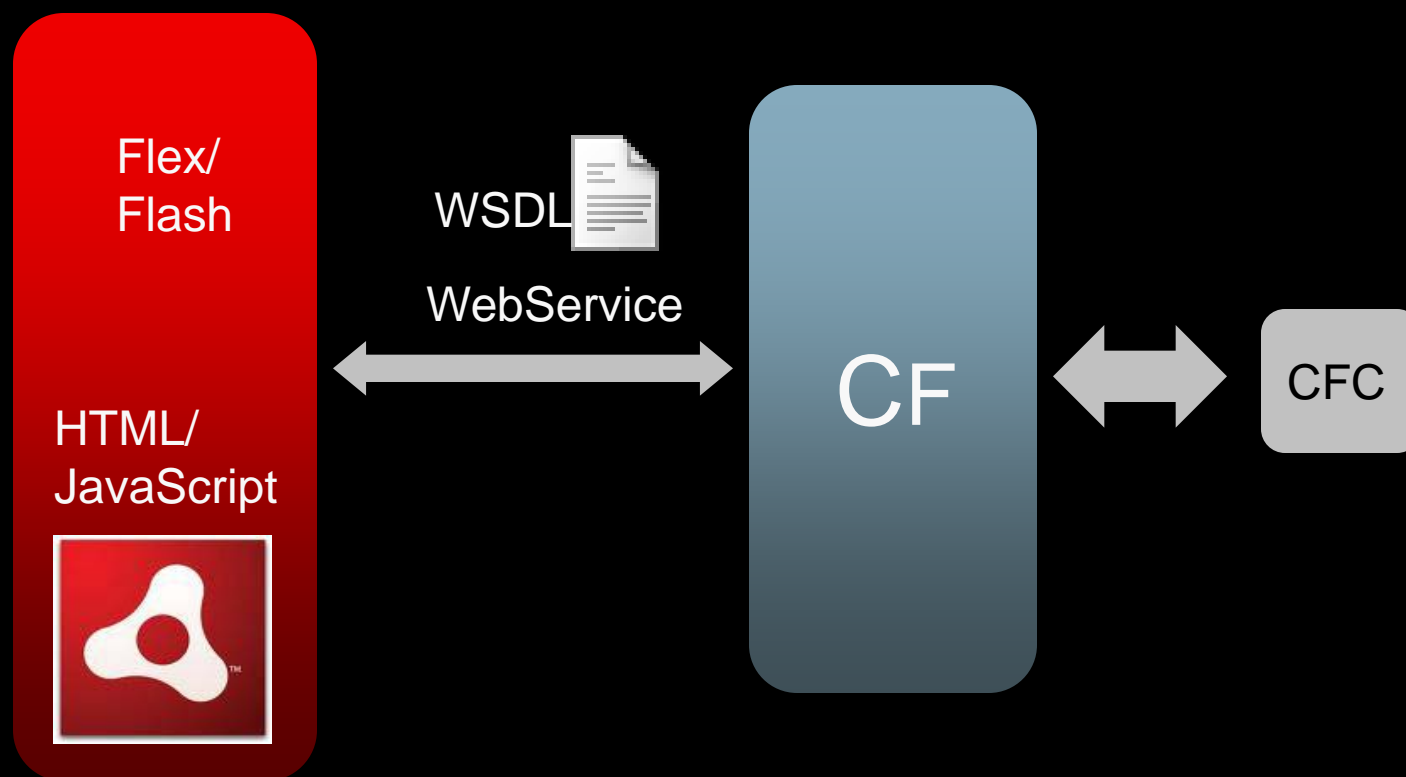
- Files loaded from a network or Internet location are assigned to the remote sandbox
- Local files fall into the local sandbox.
- No access to AIR APIs

- HTTPService
- WebService
- RemoteObject

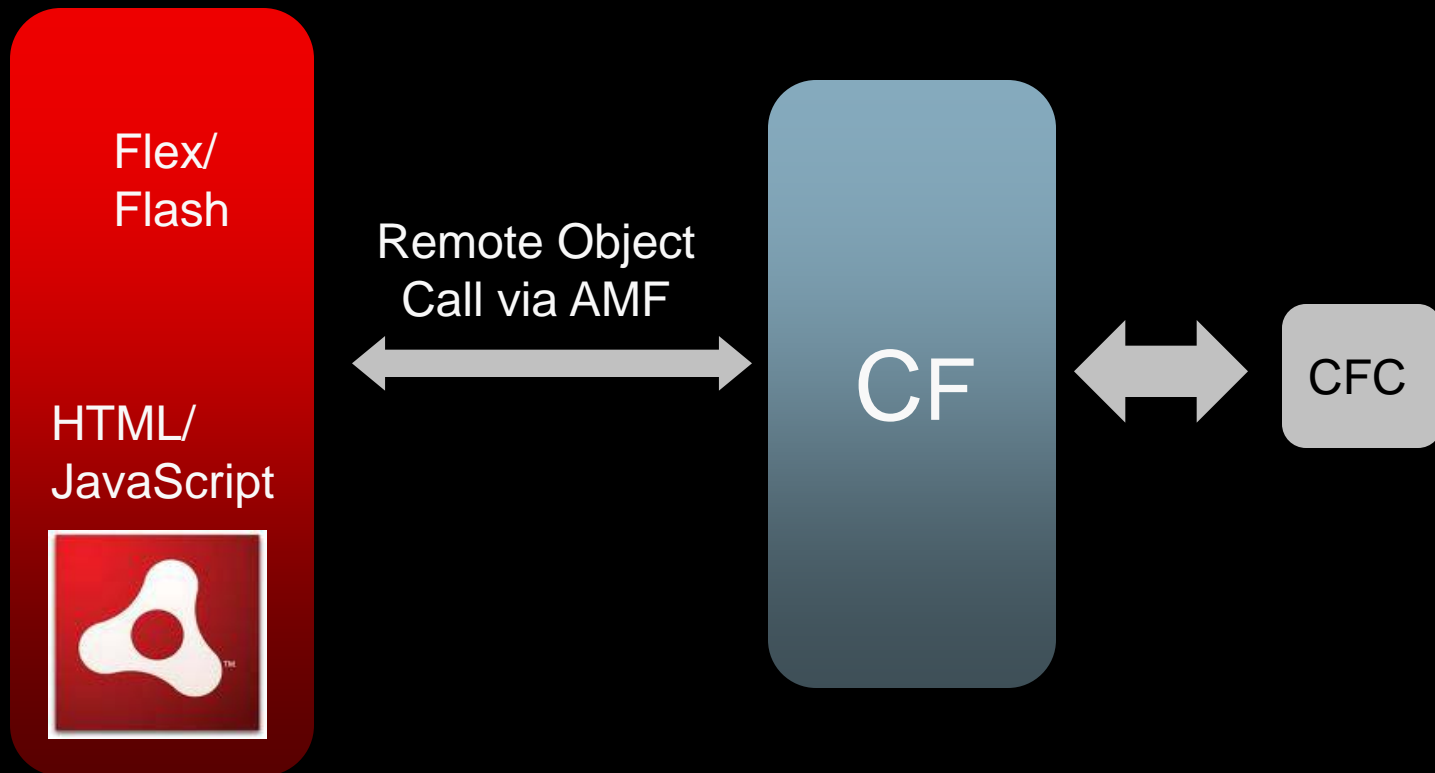
- The HTTPService component can be used to make a GET or a POST request to ColdFusion
- Useful for accessing server side accessible over HTTP



AIR Flex applications can interact with web services that define their interfaces in a Web Services Description Language 1.1 (WSDL 1.1)



RemoteObject component can be used to call methods on a ColdFusion component



# Accessing ColdFusion from Ajax client

- XMLHttpRequest
- CFAJAXPROXY



The popular Adobe Employee directory uses ColdFusion as backend



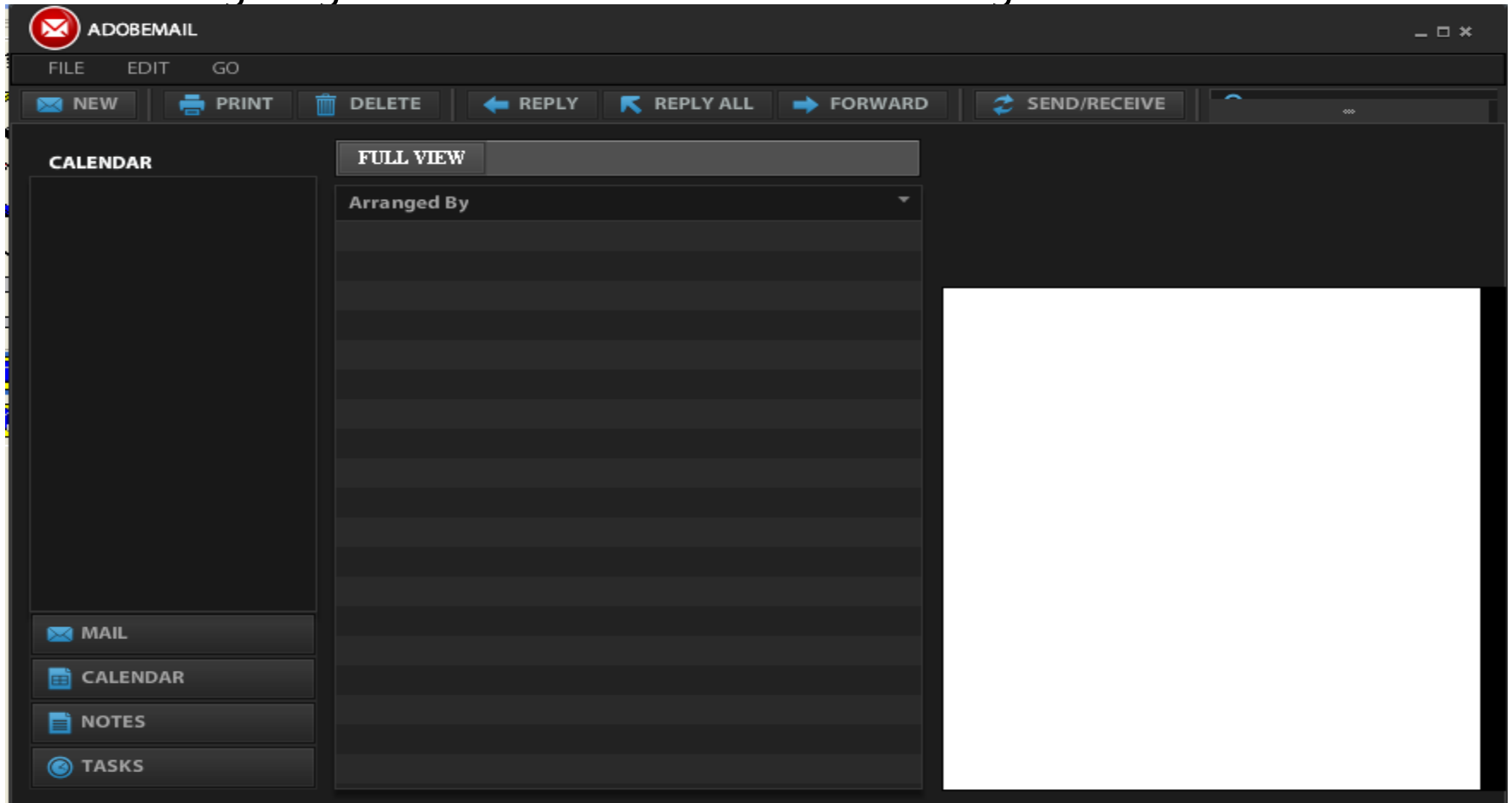
ColdFusion is used to connect to MS Exchange to retrieve employee and conference room free/busy information

How does the Adobe Employee Directory leverage ColdFusion as backend?

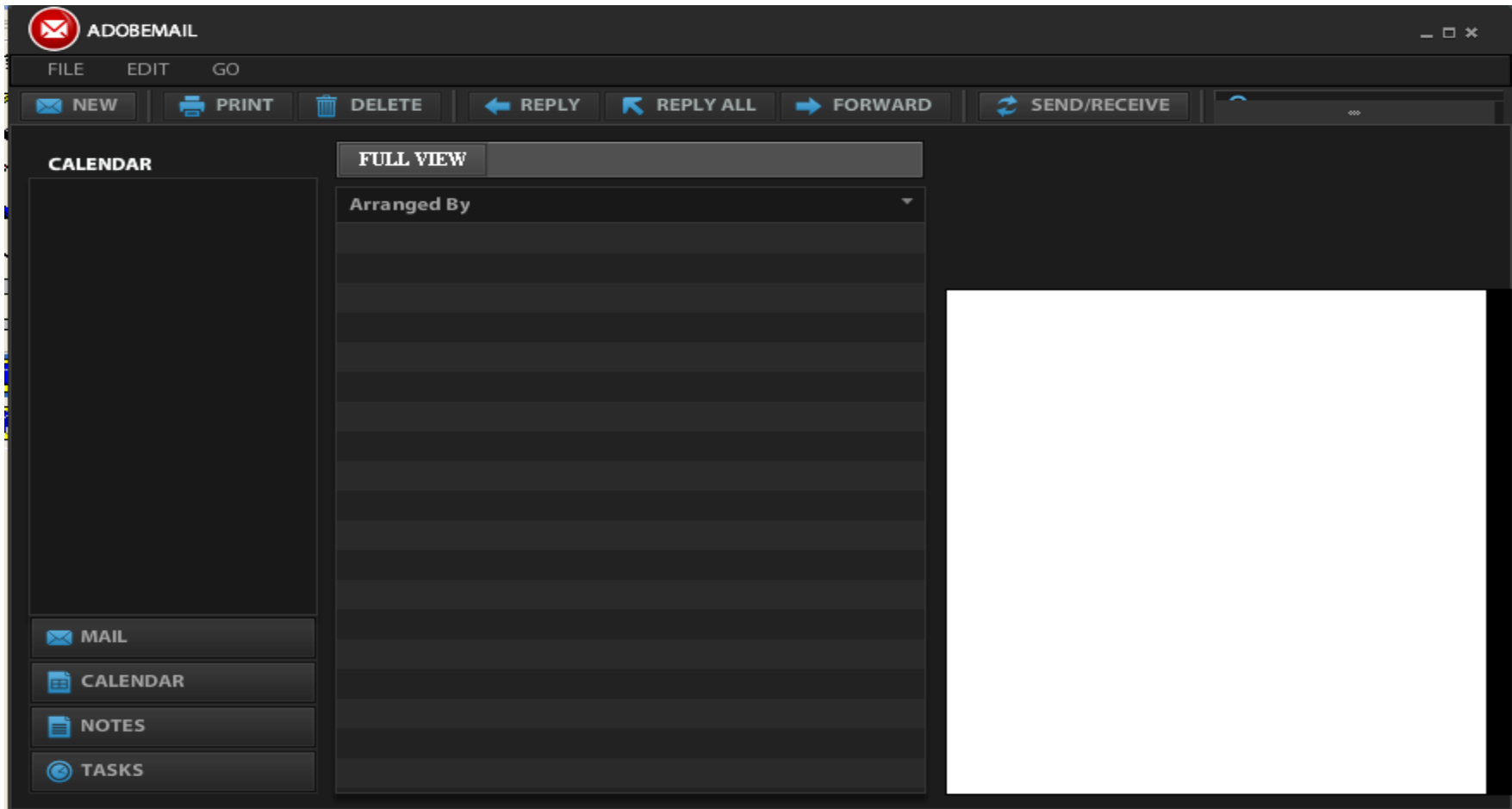


Adobe Employee Directory makes a WebService call to ColdFusion which then uses the exchange tag to fetch information from the Exchange Server

This application uses ColdFusion as backend and makes use of the popular cfexchange tag to fetch the data from the exchange server

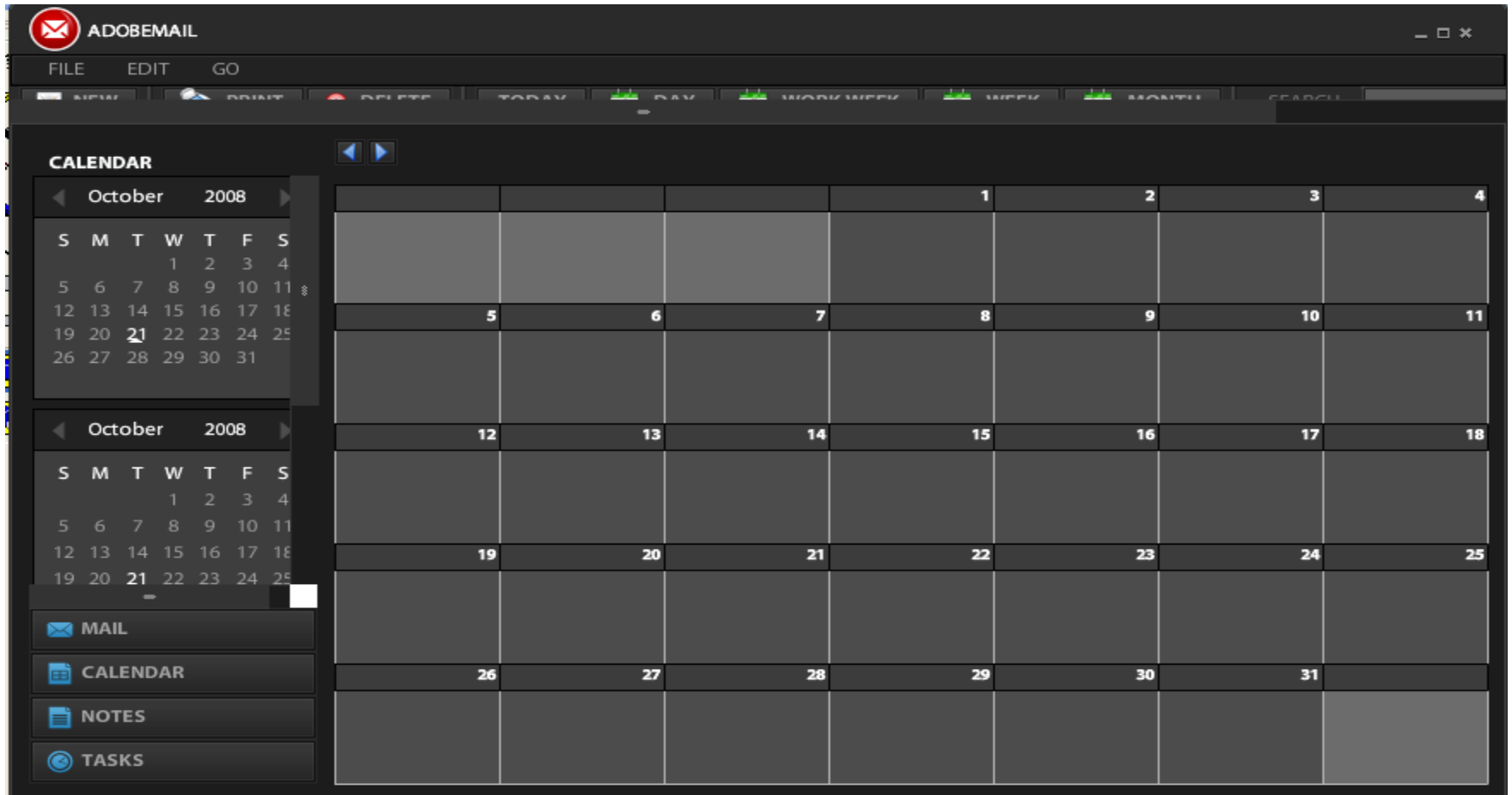


Manages your mail by asking you setting up your account



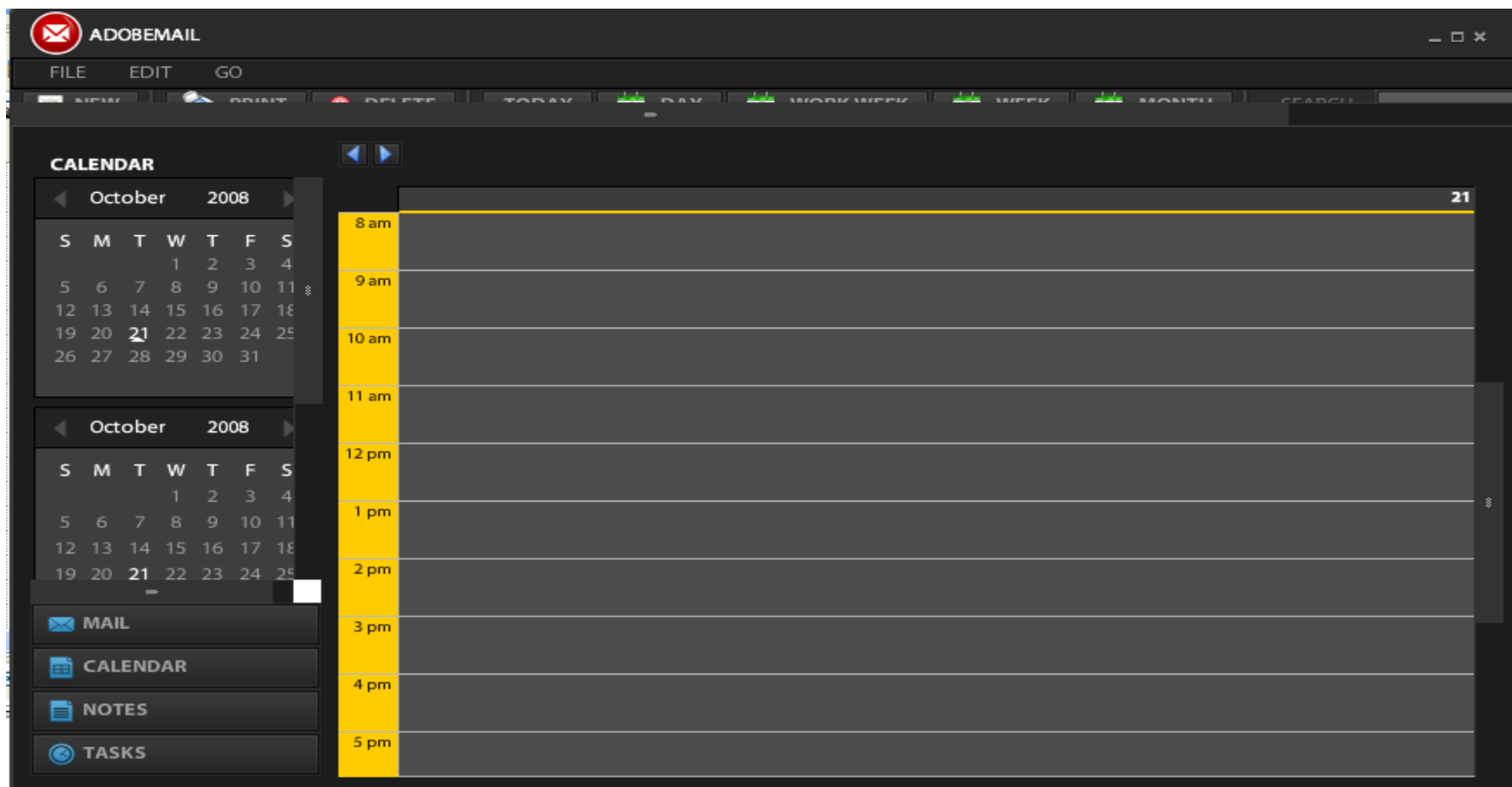
# Features of the application

Handles your calendar/notes/tasks just as outlook does



# Email Client – How does it work?

The application connects to ColdFusion using the HTTPService



CFCs are written on the server side to,

- Open connection with the exchange server
- Fetch the folders
- Fetch mail
- Send mail
- Get attachments
- Fetch calendar information for the account
- Manage notes and tasks

What do you like to see?



# What's coming up in Centaur?

# Proxy MXML tags for the popular ColdFusion tags

- Proxy MXML tags gives you the ability of getting access to ColdFusion goodness from within ActionScript
- Continue coding in ActionScript and access server side functionality directly in ActionScript

Ex: MXML code snippet

```
<cf:Pdf action="addwatermark" source="sourceURL"  
image="imageURL" ...>
```

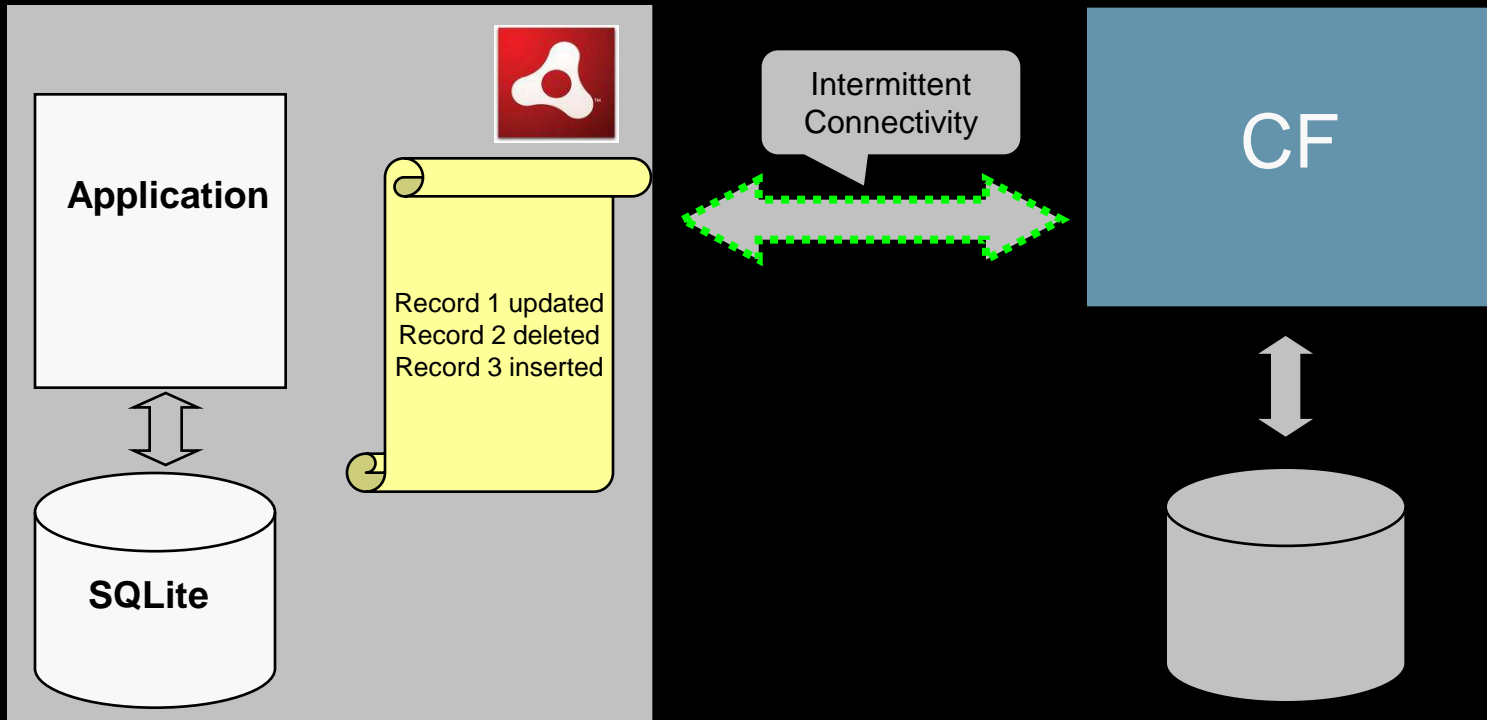
```
<cf:Mail to="toAddress" from="toFrom" server="server"  
port="port" ...>
```

# Demo of the proxy tags

# Offline support in ColdFusion to build AIR Applications

## An offline application works

- By persisting the data that the client fetched from the server into the local database
- By working with the data in the local database when the connection with the server is offline
- By sending to server all the updates that happened to the local database when the connection was offline



But here's what a developer will complain about building offline applications

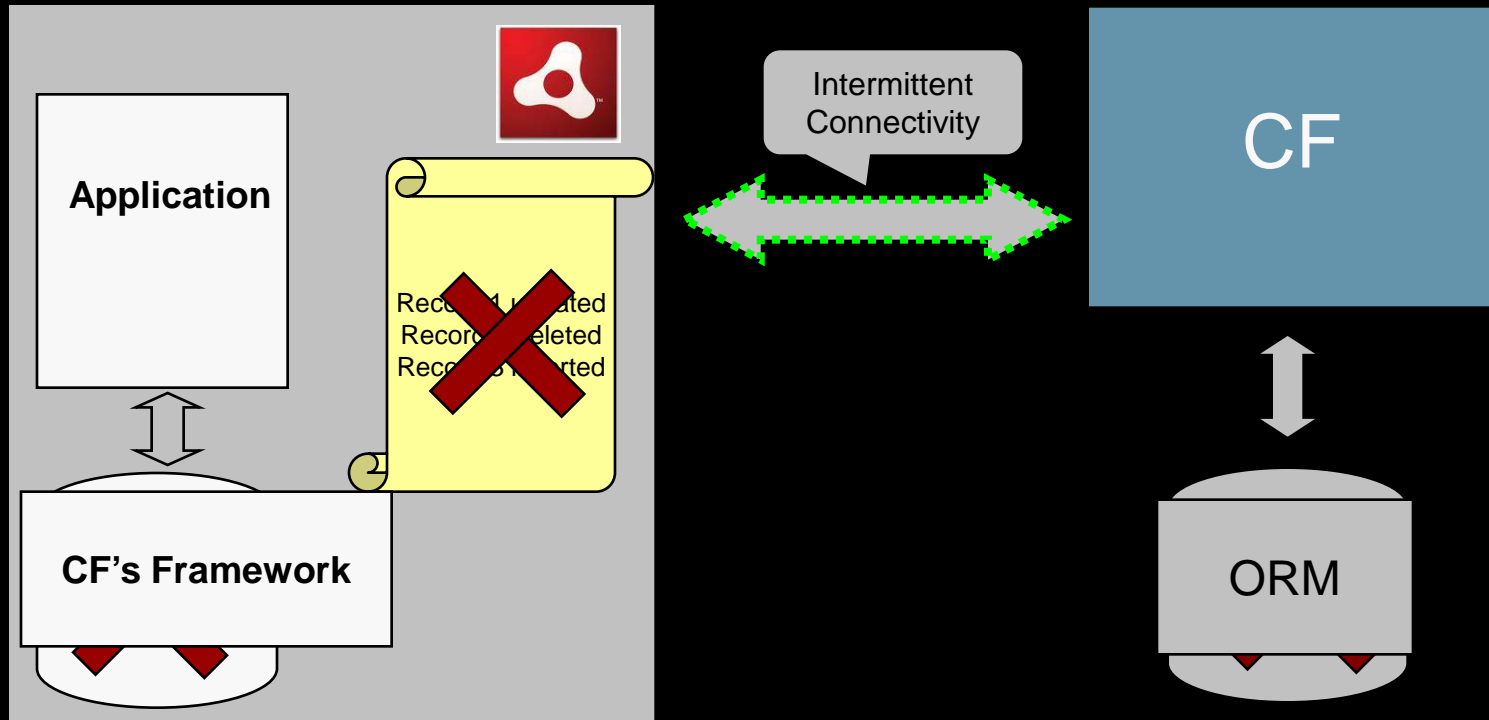
1. "I am apprehensive about the complexity because of the SQL mess involved. I will have to deal with database design and SQL statements on the client side as well"
2. "I will also have to deal with sending the updates to the server and manage conflicts"



## ColdFusion's offline support does the following

1. ColdFusion has an ActionScript persistent framework that requires developers to work only with ActionScript objects \*without\* any SQL
2. ColdFusion also keeps track of the updates on the local database to sync the data back to the server and eases conflict management

# ColdFusion AIR Integration



Makes use of the ActionScript meta data tags

```
Package {
```

```
  [Entity]
```

```
  Public class Employee {
```

```
    [Id]
```

```
    Public var eid:uint;
```

```
    Public var name:String;
```

```
  }
```

```
}
```

Employee	
eid (PK, INTEGER)	name(TEXT)

All ActionScript objects of type Employee can now persisted, retrieved or updated into the database without any SQL

```
var emp:Employee = new Employee();
```

```
emp.eid = 1;
```

```
emp.name = "Tom";
```

```
session.save(emp);
```

```
...
```

```
...
```

```
session.update(emp);
```

```
...
```

```
session.remove(emp);
```

The framework also handles relationship between entities (OneToOne, OneToMany, ManyToOne, ManyToMany)

# Mapping between ActionScript and CFC

## Employee.as

```
[RemoteClass(alias="a.b.employee")]
```

```
[Entity]
```

```
Public class Employee
```

```
{
```

```
    [Id]
```

```
    public var eid:uint;
```

```
    Public var name:String;
```

```
}
```

## employee.cfc

```
<cfcomponent>
```

```
    <cfproperty name="eid" type="numeric">
```

```
    <cfproperty name="name" type="string">
```

```
</cfcomponent>
```

# Setup required on the server

- A manager cfc forms the point of contact on the server side for the AIR application
- The manager cfc has the fetch methods defined in it using which data is fetched by the AIR application
- The manager cfc also has a sync function that is invoked when the data is synced from the AIR application

- The commit call on the AIR application, sends all offline updates of the application
- The server can raise a conflict which invokes the conflict handler defined on the AIR application.

- Presentation at my blog : <http://www.rakshith.net/blog/>

Thank you!