

# Ajax Development using ColdFusion 8

Rakshith

Adobe Systems



# Who Am I?

I am



I work as a **Computer Scientist** at



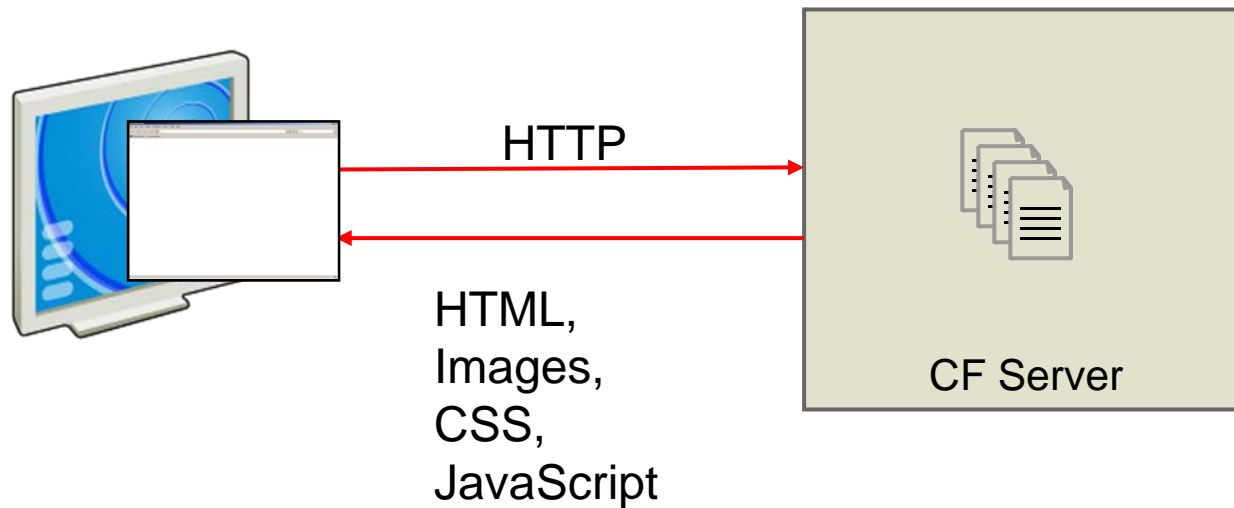
is what I build



# Agenda

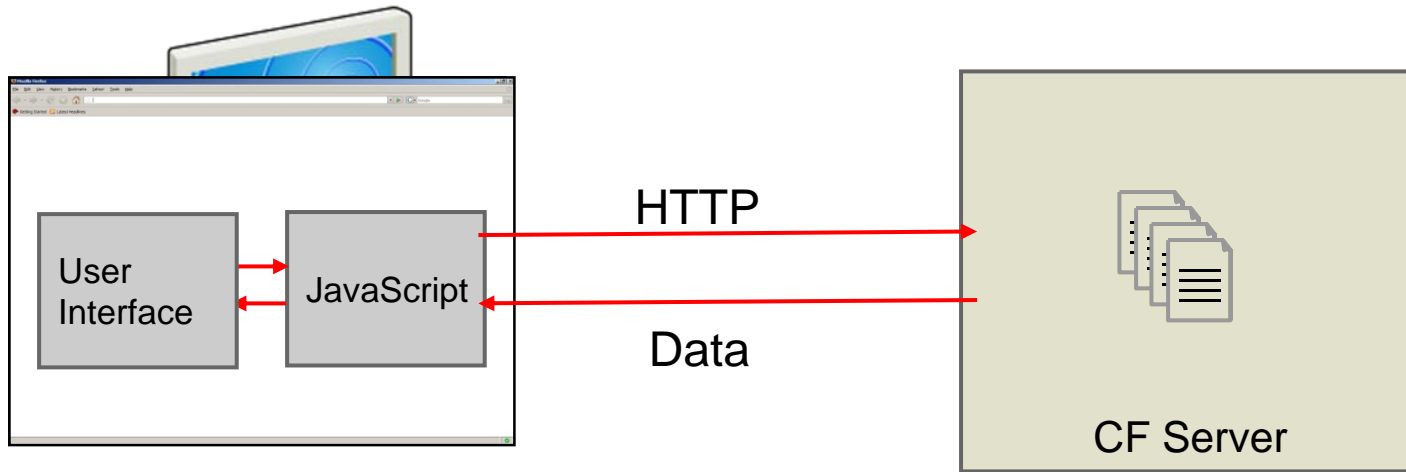
- AJAX overview
- JSON support
- AJAX plumbing
- AJAX Widgets
- Q & A

# Traditional Web Application Model



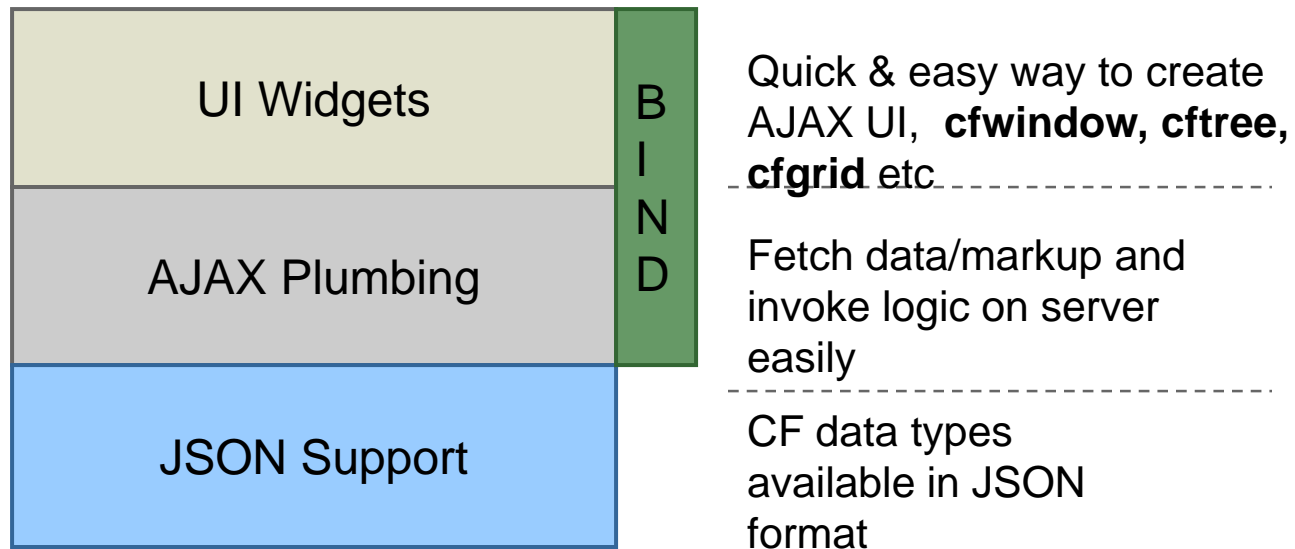
- Browser makes HTTP call
- Complete HTML updated

# AJAX Web Application Model

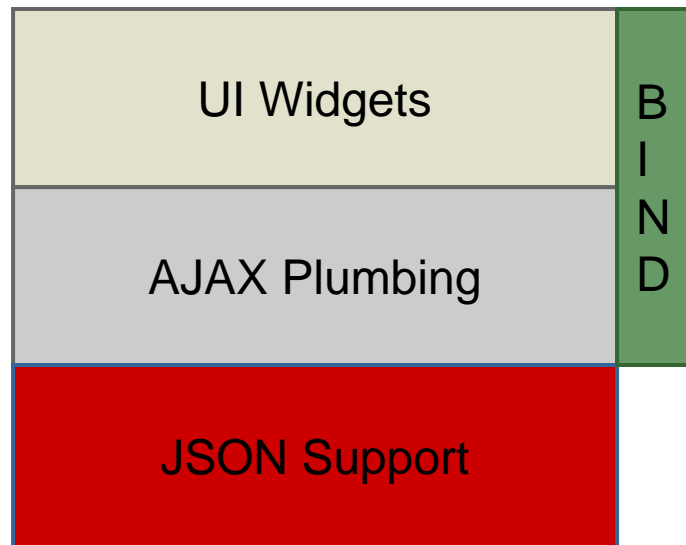


- JavaScript executes in browser & makes HTTP call
- User Interface (part-HTML) updated

# ColdFusion AJAX Architecture



# JSON Support

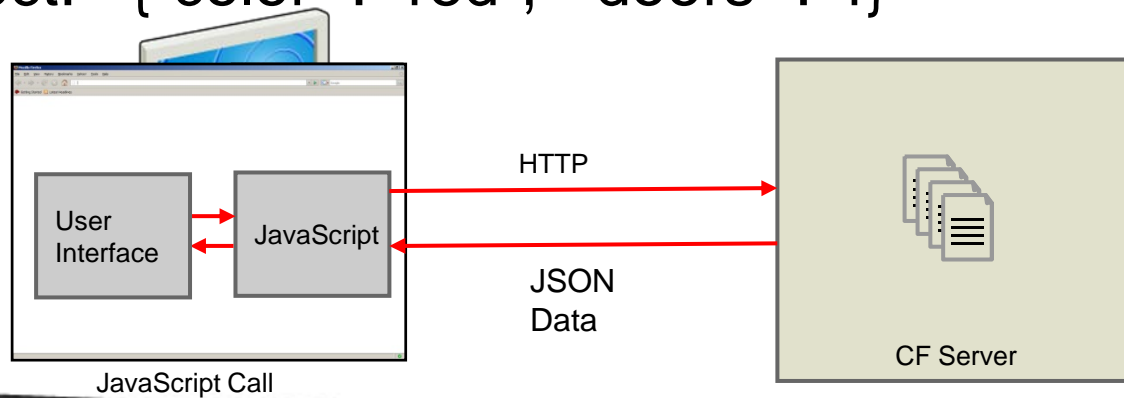


# What is JSON?

- **JavaScript Object Notation**
  - Representation of data in JavaScript
  - JavaScript can understand JSON, it's a native data format
- **Based on array and object literals of JavaScript**

Array: ["Benjamin", "Michael", "Scott"]

Object: {"color": "red", "doors": 4}



# XML Vs. JSON

- **XML**

```
<books>
```

```
  <book isbn="000230" title="xyz"/>
```

```
  <book isbn="000231" title="abc"/>
```

```
</books>
```

- **JSON - Fat free XML!**

```
{ books:{book: [
  {isbn:'000230', title:'xyz'}
  ,{isbn:'000231', title:'abc'}
  ]}
}
```

# JSON Support in ColdFusion 8

- **SerializeJSON ()**
  - Converts ColdFusion objects into their JSON data format
- **DeserializeJSON ()**
  - Converts JSON data into ColdFusion objects
- **IsJSON()**
  - Checks if the data passed to the function is in the JSON format

# CFML ↔ JSON type mapping

- **Simple types**

String: "Adobe"

Number: 1982.0

- **Array**

Array: ["Adobe","ColdFusion8"]

- **Struct**

Struct: {"PRODUCT":"ColdFusion8","NAME":"Adobe"}

- **Query**

- Row format Query:

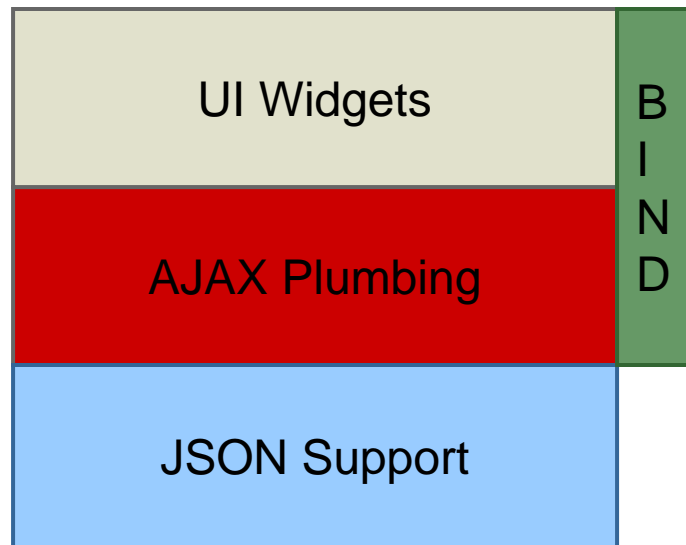
`{"ROWCOUNT":2,"COLUMNS":["NAME"],"DATA":{"name":["Adobe","ColdFusion8"]}}`

- Column format Query:

Query: {"COLUMNS":["NAME"],"DATA":[[["Adobe"],["ColdFusion8"]]]}



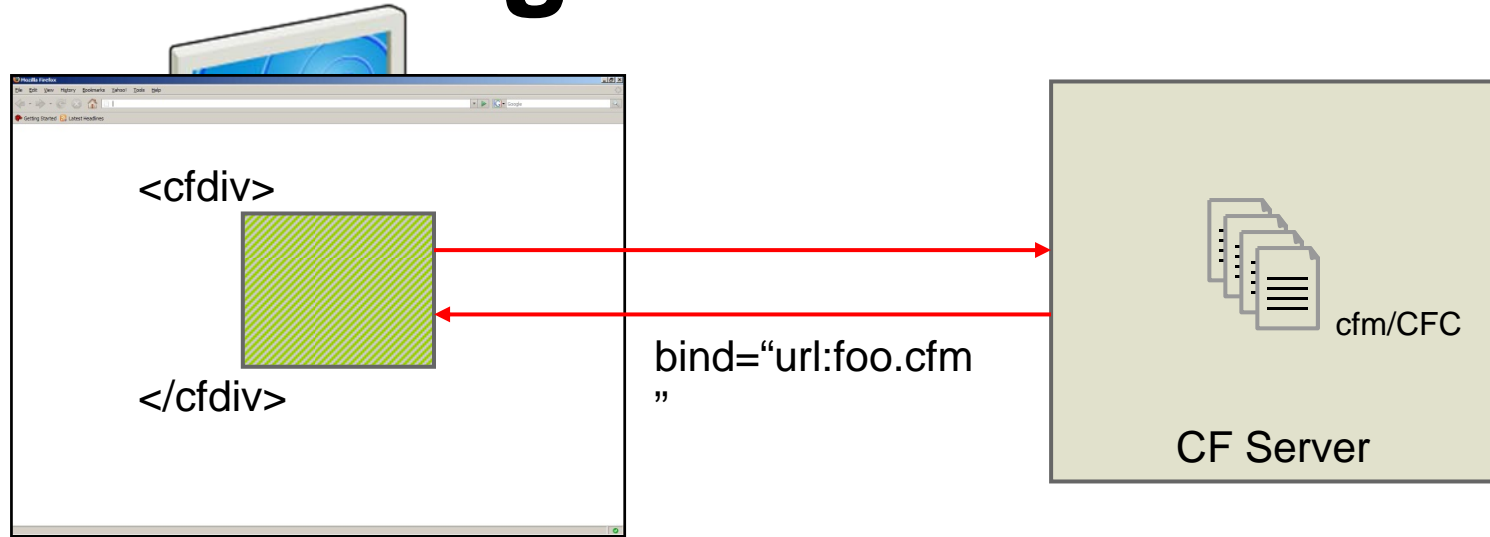
# AJAX Plumbing



# AJAX Plumbing

- **What you expect from AJAX application**
  - Some area of the page gets refreshed, no full page refresh
- **For that to happen, you need**
  - New HTML markup for that area
  - To invoke logic on the server to fetch the data

# AJAX region & data fetch



Define AJAX region using `<cfdiv>`

Fetch data for the region using  
BIND expressions

```
<cfdiv bind="url:foo.cfm">
```

# CFDIV – Form submission

- **CFFORM inside CFDIV is submitted asynchronously**

```
<cfdiv>
```

```
  <cform action="onlysubmitthisform.cfm">
```

```
    <cfinput name="isbn">
```

```
  </cform>
```

```
</cfdiv>
```

# BIND

`bind="cfc:bookData.getBookDetails({bookForm:isbn.value@change})"`

- React to UI control events

`{bookForm:isbn.value@change}`

- Retrieves UI control values

`{bookForm:isbn.value@change}`

- Invoke logic and retrieve markup or data

`cfc:bookData.getBookDetails({...`

- Four types

- CFC

- Javascript - `bind="javascript:getBookDetails(`{bookForm:isbn.value@change}`)"`

- url - `bind=url:viewBookDetails.cfm?isbn={bookForm:isbn.value@change}`

- Plain - `bind="The ISBN is:{bookForm:isbn.value@change}"`



UI Widgets	B
AJAX Plumbing	N
JSON Support	D

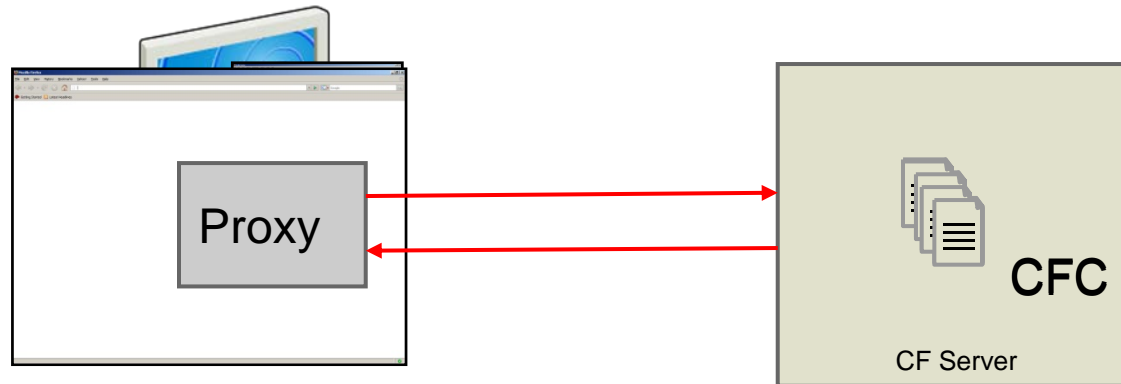
# Yahoo! Live Search



# Get your CFC functions in JavaScript

- **CFAJAXPROXY**

- 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>
```

# cfajaxproxy – built-in functions

- setHTTPMethod
- setAsyncMode, setSyncMode
- setCallbackHandler
- setErrorHandler
- setQueryStringFormat
- setReturnFormat
- setForm

# Fetch Markup – ColdFusion.navigate()

- “Navigates” the contents of a tag
- Specify HTTP method, custom callback and error handler JS functions
- Submit a form

ColdFusion.navigate(url, id, callback, errorHandler, httpMethod, formId)

```
<a href="
  javascript:ColdFusion.navigate(
    'getBookDetails.cfm', 'bookdetails' )"
>Navigate</a>
<div id="bookdetails"/>
```

# Fetch Markup – Server Functions

- AjaxLink(url)

Reloads the link's container with the url

**<cfdiv>**

**First Page**

**<a href="#AjaxLink('nextpage.cfm')#">Next Page</a>**

**</cfdiv>**

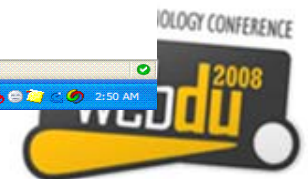
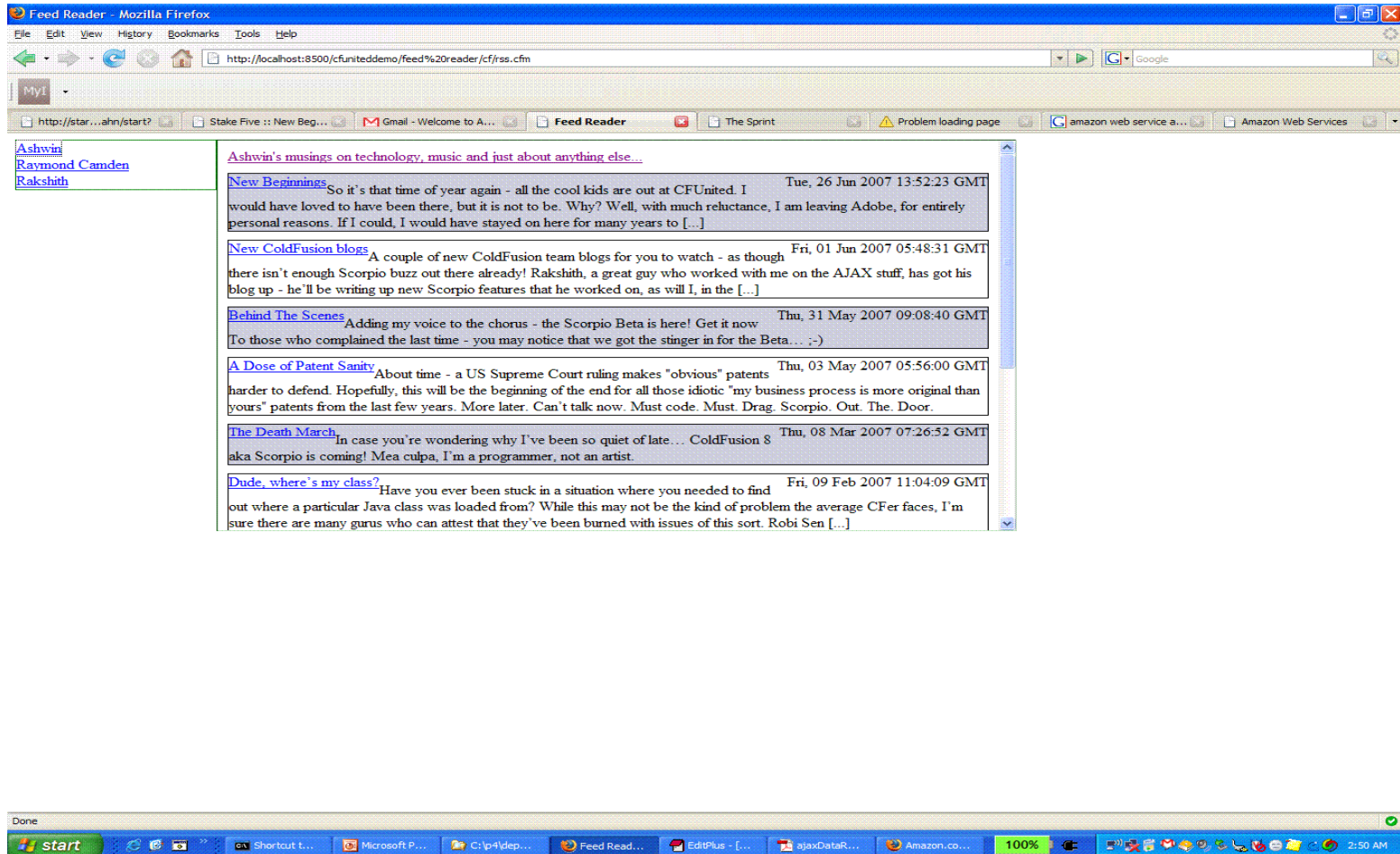
- AjaxOnLoad(functionName)

Calls a JS function when markup is loaded, after all CF components are initialized

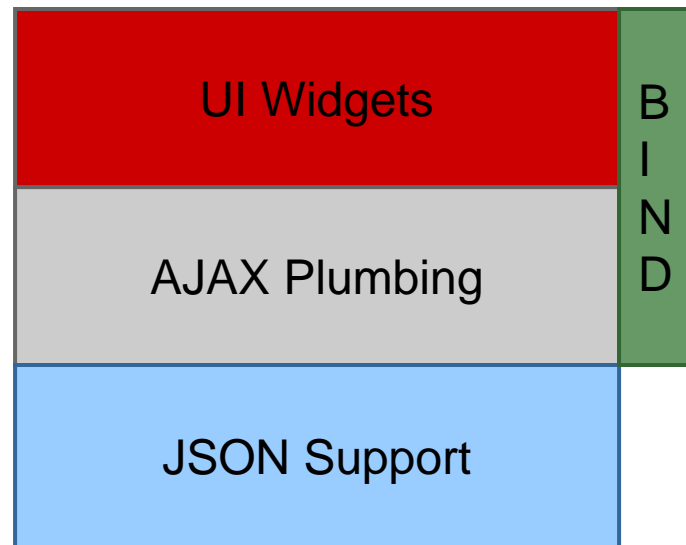
# Basic AJAX Functions

- `ColdFusion.Ajax.submitForm(formId, url, callbackHandler, errorHandler, httpMethod, async)`
- `ColdFusion.getElementValue(elementname, formname, attribute)`
- Logger functions
  - `ColdFusion.Log.debug(message, component)`
  - `ColdFusion.Log.info(message, component)`
  - `ColdFusion.Log.error(message, component)`
  - `ColdFusion.Log.dump(object, component)`
  - `?cfdebug`
- CF Administrator Settings for Logger
  - Globally enable/disable
  - Restrict by IP address

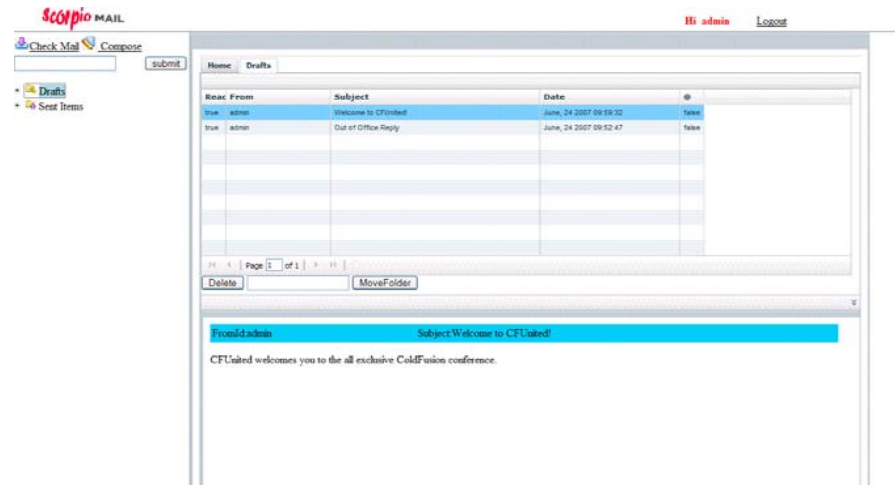
# Feed Reader Application



# AJAX Widgets



# The Mail Application



A screenshot of the Scorpio Mail web application interface. The top left shows the "scorpio MAIL" logo and navigation links for "Check Mail" and "Compose". The top right shows the user "Hi admin" and a "Logout" link. Below the navigation is a "submit" button and a sidebar with "Drafts" and "Sent Items" folders. The main area displays an inbox table with columns for "Reac", "From", "Subject", "Date", and a status icon. Below the table are "Delete" and "MoveFolder" buttons. The selected email is displayed below the table, showing "From: admin" and "Subject: Welcome to CFUnited!". The email body text reads: "CFUnited welcomes you to the all exclusive ColdFusion conference."

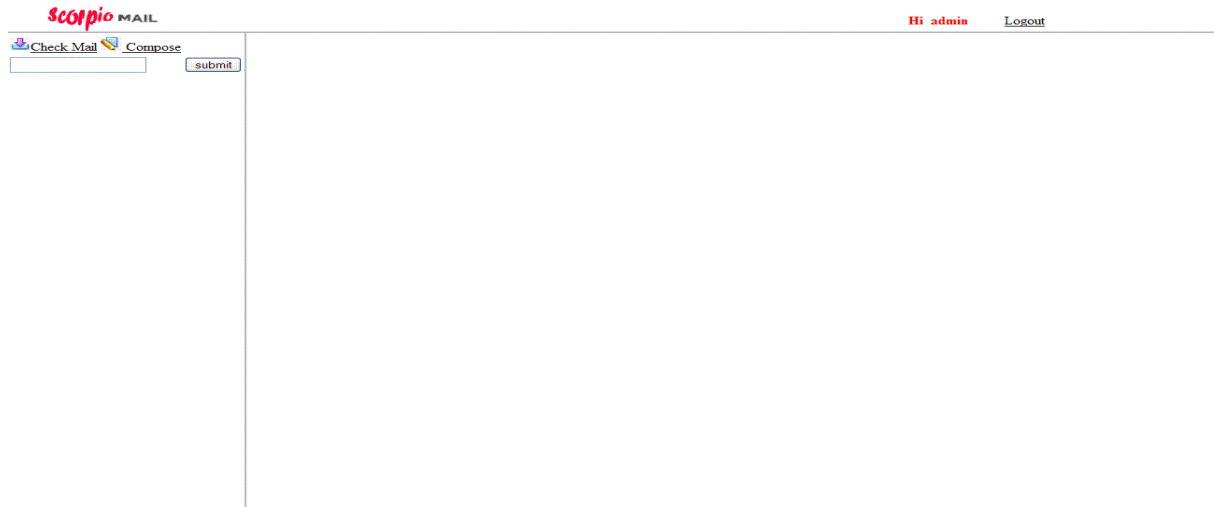
Reac	From	Subject	Date	
true	admin	Welcome to CFUnited!	June, 24 2007 09:59:32	false
true	admin	Out of Office Reply	June, 24 2007 09:52:47	false



# Stage 1

## Layout for the Application...

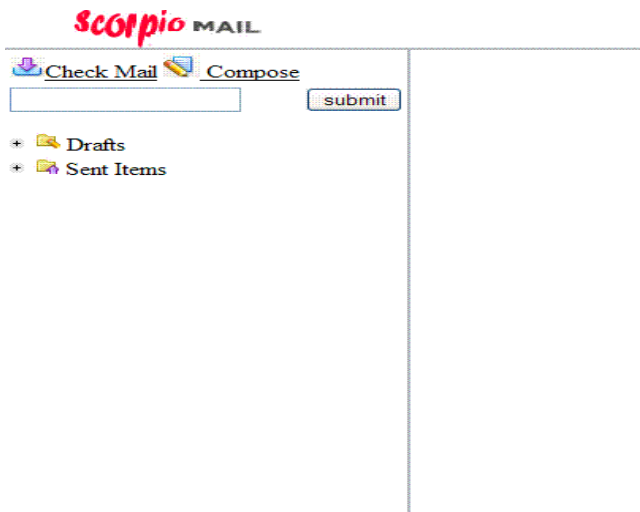
```
<cflayout type="border">
  <cflayoutarea position="top" size="40">
  </cflayoutarea>
  <cflayoutarea position="left" name="left" collapsible="true"
  splitter="false" >
  </cflayoutarea>
  <cflayoutarea position="center" name="center" >
  </cflayoutarea>
</cflayout>
```



# Stage 2

## The folder tree on the app...

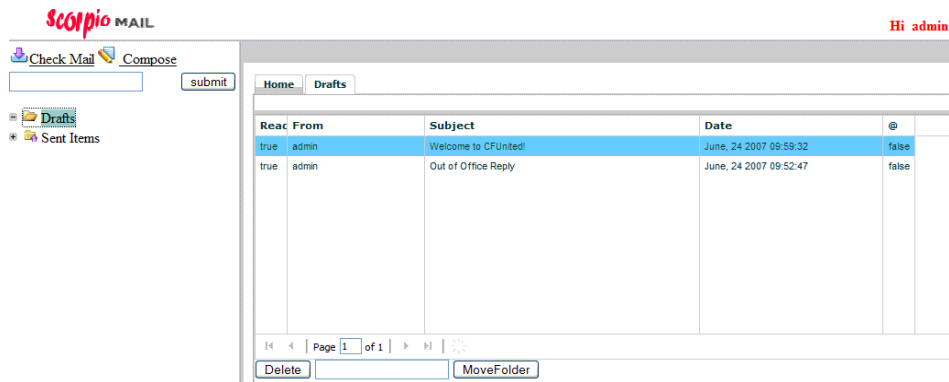
```
<cftree name="mail" format="html" completePath="yes">
  <cftreeitem
    bind="cfc:exchange.getNodes({cftreeitemvalue},{cftreeitempa
    th})" >
</cftree>
```



# Stage 3

## A dynamic grid to list the mails...

```
<cfgrid format="html" name="mailGrid" pagesize=10 sort="false"
  bind="cfc:exchange.getMails({cfgridpage},{cfgridpagesize},{cfgridsor
  tcolumn},{cfgridsortdirection},'#url.foldername#')" selectMode="row"
  width="930" delete="yes" striperows="yes" >
  <cfgridcolumn name="ISREAD" display="true" header="Read"
  width="35"/>
  <cfgridcolumn name="FROMID" display="true" header="From"
  width="150"/>
  <cfgridcolumn name="SUBJECT" display="true" header="Subject"
  width="300"/>
</cfgrid>
```



The screenshot shows a webmail interface for 'scopio MAIL'. The user is logged in as 'Hi admin'. The interface includes a navigation bar with 'Check Mail' and 'Compose' buttons, and a sidebar with 'Drafts' and 'Sent Items' folders. The main content area displays a table of emails with columns for 'Read', 'From', 'Subject', 'Date', and a status column. The table contains two rows of email data.

Read	From	Subject	Date	@
true	admin	Welcome to CFUnited	June, 24 2007 09:59:32	false
true	admin	Out of Office Reply	June, 24 2007 09:52:47	false

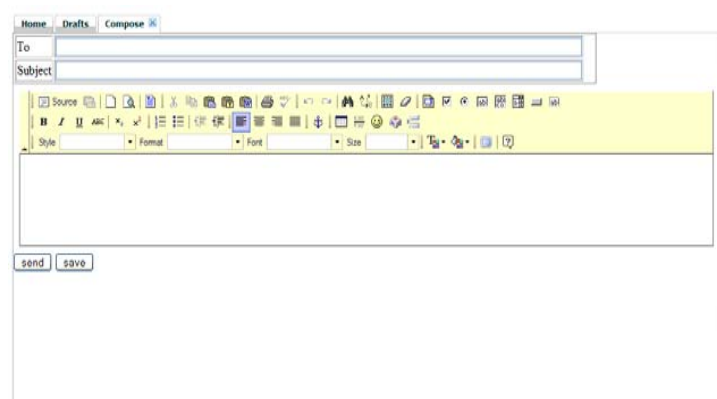
At the bottom of the interface, there are navigation controls for the grid, including 'Page 1 of 1' and buttons for 'Delete' and 'MoveFolder'.



# Stage 4

## The compose tab...

```
<cfajaxproxy cfc="exchange" jsclassname="exchangeCFC" >  
function savemail()  
{  
var m = new exchangeCFC();  
var to = ColdFusion.getElementValue('To');  
var subject = ColdFusion.getElementValue('Subject');  
var content = ColdFusion.getElementValue('mailcontent');  
m.sendMail(to,subject,content,false);  
}
```



# Slide 5

## The contacts Autosuggest...

```
<cfinput autosuggest="cfc:exchange.getContacts({cfautosuggestvalue})"
  showautosuggestloadingicon=false type="text" name="To" />
<cffunction name="getContacts" access="remote" returnType="any">
  <cfargument name="autosuggestvalue" type="String" required="false"
    default="/">

  <!--- Query --->
  <cfset contactsArray = ArrayNew(1)>
  <cfloop query="mail">
    <cflog text="fromid = #fromid#">
    <cfset contactsArray[CurrentRow] = '#fromid#'>
  </cfloop>
</cfreturn contactsArray>
```

Home	Drafts	Compose <input type="button" value="x"/>
To	a	
Subject	admin	



# ColdFusion Ajax Security

# Security concern in Ajax applications

- The same origin policy - Any page loaded from this origin may access, download, and interact with any other resource from the same origin.
- Way to get around same origin policy  
<script src="can access location from different origin">
- The browser evals JSON data returned from src
- It is possible to extract sensitive JSON data from a known url as an attempt to get access to the data

# Security support in ColdFusion

- JSON Prefix – Adds a prefix to the JSON data returned by the server. By default the value is //. Configured from admin, application.cfc and cffunction tag
- Client Verification – Prevents unauthorized party from performing actions on the server by attaching an encrypted security token with each request. Verifyclient attribute can be set at the function level or by calling verifyclient function on the page.
- Enable client or session management for client verification. Should be set only for sensitive functions that are called from the Ajax client

# Summary

- Ajax in ColdFusion is available at three different levels: JSON Support, Ajax Plumbing, Ajax Widgets
- Can use JSON support and the Ajax Plumbing without using the Ajax Widgets: Allows other Ajax Frameworks to integrate nicely
- Integration with Spry: CFSPRYDATASET
- Presentation and sample applications  
: <http://www.rakshith.net/blog/>

# Q & A



# Thank you!



Email: [rakshith@adobe.com](mailto:rakshith@adobe.com)  
Blog: [www.rakshith.net/blog/](http://www.rakshith.net/blog/)