UNIVERSAL MIND

WEBMANIACS 2008
FUNDAMENTALS OF MACH-II

Andrew Powell

Email / GTalk: andrew.powell@universalmind.com

umAndy

Universal Mind

Introductions

UNIVERSAL MIND IS HIRING!!

Inc. 500

- Sr. Consultant, Universal Mind
- UM is a 2007 Inc. 500 Honoree
- 10+ Years Experience In Software Development
- ASP --> Java --> ColdFusion --> Flex & AJAX
- Multiple Publications on Flex, Java, & ColdFusion
- http://www.infoaccelerator.net

What Will We Learn Today?

- What's a Framework?
- What's Mach-II?
- How does Mach-II work, or, what's Mach-II's world view?
- "Events, Listeners, and the Framework That Loves Them"

UNIVERSAL MIND

Goals

- When you leave here you should be able to grab a friend and tell them:
  - What a Framework Is...
  - What Mach-II Is...
  - What the "II" in Mach-II Stands For...
  - How Mach-II Works...
  - How To Build A Simple App...

What’s a Framework?

- A core group of files that provide a subset of reusable functionality to the language in which the framework is written
    - Can be thought of as an application with an associated API inside which your application runs
- A consistent, logical way of creating and organizing application components
- The tools/methodology that allow the application components to communicate with one another
- A *very strongly encouraged* design pattern (in Mach-II’s case, this is MVC)

Let's Try That Again....
In English

- Frameworks are here to make our development lives easier by handling the basic “plumbing” of the application
  - Less rethinking and doing the same things over and over
- A good framework is (relatively) easy to learn
- A good framework helps, doesn’t hinder

What Is Mach-II?

- Mach-II is an ***event-driven*** framework based on the concept of ***implicit invocation***
- Mach-II is an ***object-oriented*** framework and encourages ***highly cohesive***, ***loosely coupled components***
- Mach-II is built around the ***Model-View-Controller (MVC)*** design pattern

Let’s Clarify “Event-Driven”

- The event object is at the heart of everything that happens in Mach-II
- In a Mach-II request ...
  - An event is announced
  - “Stuff happens” (more on this later)
  - Either additional events are announced or the request ends

Implicit Invocation

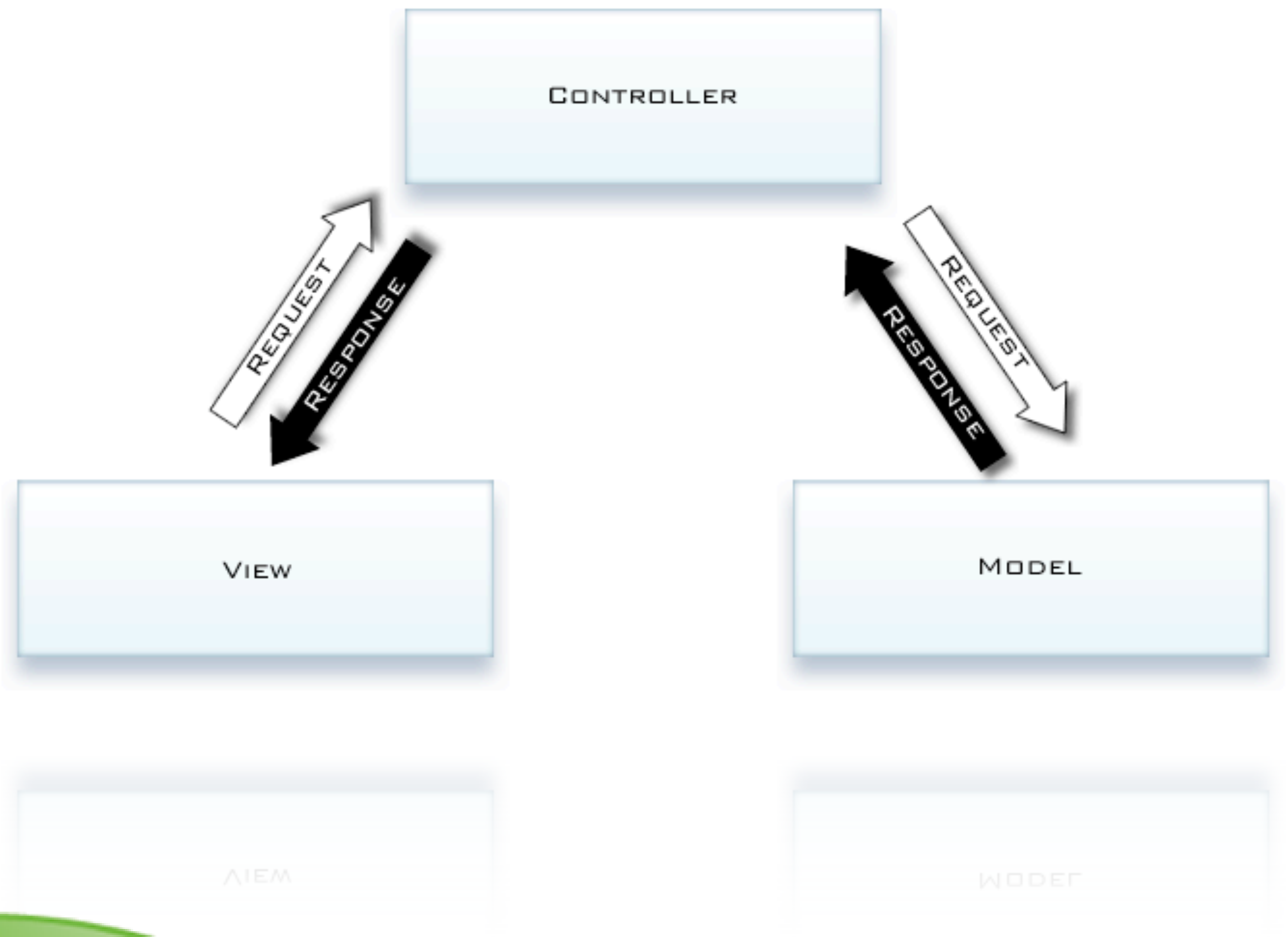
- Hence the "II" in "Mach-II"!
- Implicit invocation is a very anti-procedural or anti "top down" approach
- The event announcement implicitly causes other procedures to be invoked
- The event does not invoke procedures directly

Model-View-Controller (MVC)

- Mach-II encourages good application architecture: MVC
  - **Model:** business objects in the application (CFCs)
  - **View:** pages the user sees and interacts with (CFML pages)
  - **Controller:** "traffic cop" for the application (Mach-II's index.cfm +

UNIVERSAL MIND

MVC Pattern



UNIVERSAL MIND

Cohesion and Coupling

- Mach-II encourages building highly cohesive, loosely coupled components
- Cohesion: the degree to which something does one thing and does it well
    - High cohesion is the goal
- Coupling: the degree to which components are dependent upon one another

Before we proceed ...
Any Question So Far?

Mach-II Framework Structure

- The Mach-II framework consists of 40 CFCs
- The great thing about Mach-II is that you DON’T have to worry about the “under the hood” stuff unless you want to

Mach-II Application Skeleton

Let’s take a look...

UNIVERSAL MIND

Mach-II Request Lifecycle

- Typical Mach-II URL:
    - index.cfm?event=doSomething
- All requests in Mach-II are routed through index.cfm
- The “doSomething” event name corresponds to an event that is defined in Mach-II’s XML configuration file

UNIVERSAL MIND

Remember this pattern!

- An event is announced, either via the URL or programmatically (e.g. index.cfm?event=saveEmployee)
- "Stuff happens," which in our case will be that a **`EmployeeManagerListener`** (more on this in a moment) is notified of request and the **`saveEmployee()`** method in the listener is called
- Either another event is announced, or the request comes to a conclusion, usually by displaying a view to the user

The Employee Form

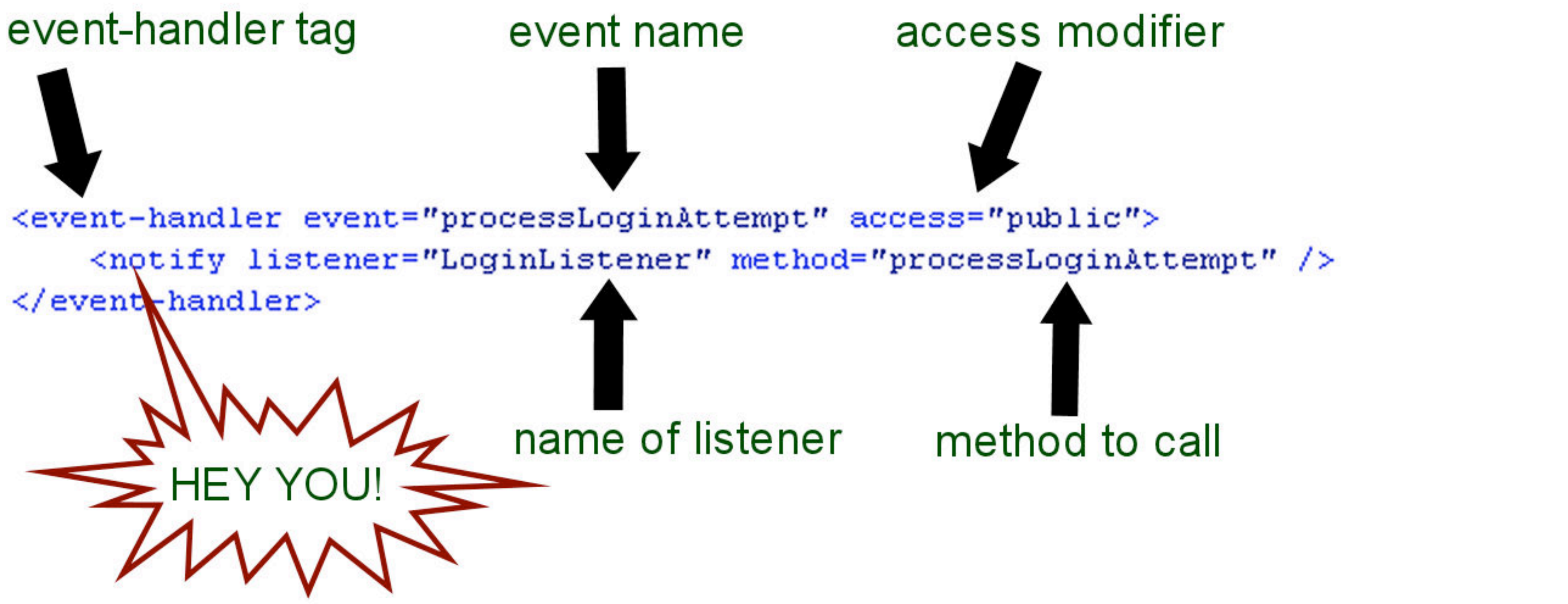
- Form fields: firstname, lastname, etc.
- Mach-II grabs both form and URL variables and puts them in the event object for you
- Let's look at the steps that occur when the save employee form is submitted by the user...

UNIVERSAL MIND

Step 1: The Event Is Announced

- The save process starts with a form post, the action attribute of which is **`index.cfm?event=saveEmployee`**
- The **`<event-handler>`** tag in the XML configuration file defines what happens in this event

The <event-handler> Tag


event-handler tag
event name
access modifier
name of listener
method to call
HEY YOU!

```
<event-handler event="processLoginAttempt" access="public">
    <notify listener="LoginListener" method="processLoginAttempt" />
</event-handler>
```

Step 2: Listener is Notified

- Listeners are developer-defined objects that “listen” for notifications from events
    - Developer-defined Listeners extend **MachII.framework.Listener**
- Mach-II notifies Listeners via the notify command when an event involves them
- Methods (functions) within Listeners are then called
    - Typically Listener methods will either return something or announce the next event

The EmployeeManagerListener

- Guess what? It listens for login attempts!
- We'll examine the **`saveEmployee`** event which is called when--you guessed it--a user attempts to save an employee

UNIVERSAL MIND

The story so far ...

- The **`saveEmployee`** is announced
- The **`EmployeeManagerListener`** is notified of the login attempt

Step 3: Listener Method is Called

- The notify command specifies a method to be called, which in this case is **`saveEmployee()`**
- Remember that Mach-II automatically puts all form and URL variables in the event object

Listener Method Arguments

- In the vast majority of cases, your Listener methods will take in a single argument, namely the Mach-II event object:

```
<cfargument name="event"
type="MachII.framework.Event" required="true" />
```

- You then get the data you want from within the event object, e.g.:

```
<cfset var myFormFieldData =
arguments.event.getArg("formFieldName") />
```

UNIVERSAL MIND

Saving The Record

- Once we have the data from the event object, we can persist it to a database
  - Already we're getting into tight cohesion and loose coupling

Step 4: Another Event Announcement

- After the save attempt is processed, the listener announces another event, in this case either **`saveSucceeded`** or **`saveFailed`**
- These events are PRIVATE so they can't be accessed directly via the URL

UNIVERSAL MIND

SHUT UP AND SHOW SOME CODE!!!!

UNIVERSAL MIND

Object-Oriented Programming (OOP)

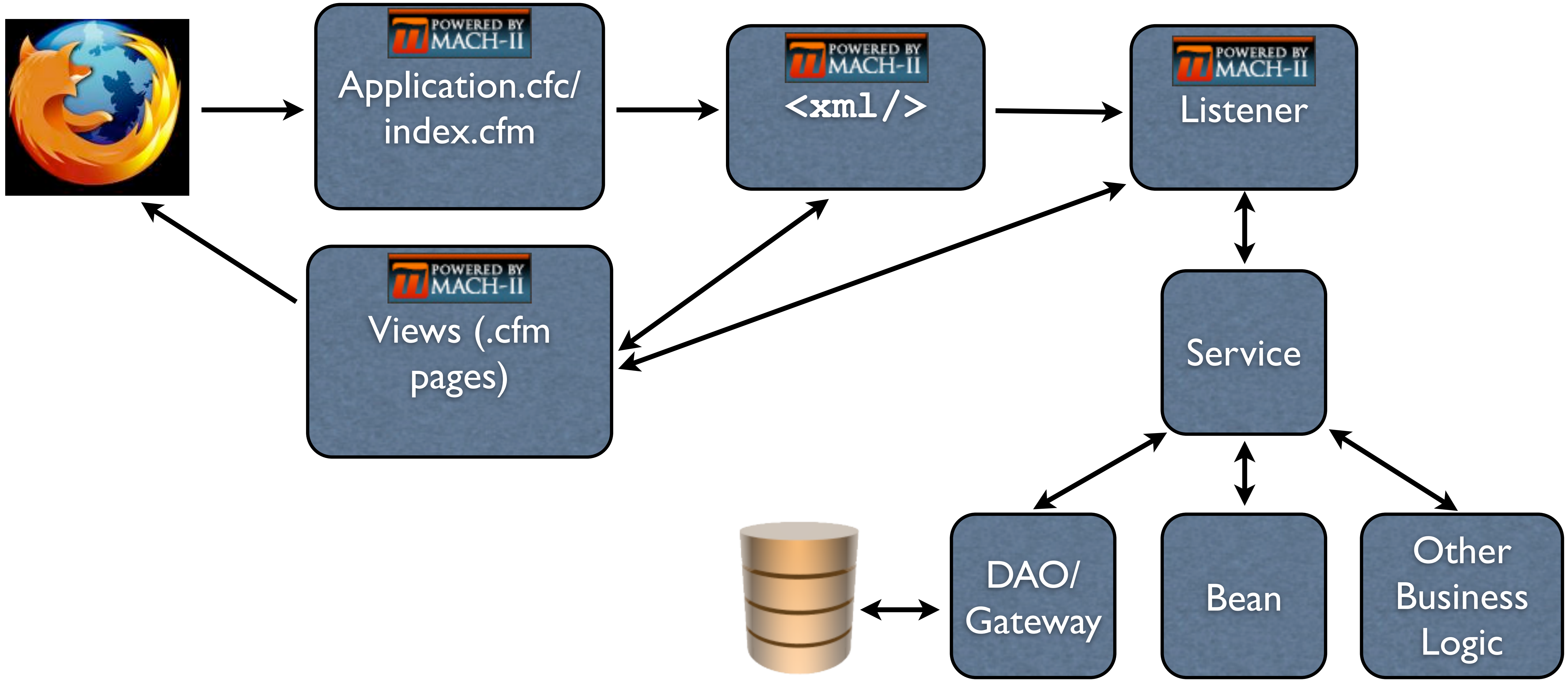
- Mach-II more or less mandates that you use OO development practices
- Best to learn OO principles first, and then take on Mach-II
  - Diving into Mach-II isn’t really the best way to learn OO
- Let’s take a brief look at a basic best-practice architecture

OO Architecture in Five Minutes Or Less

- Pay very close attention to what's aware of Mach-II and what isn't
- Your business logic (model) should NOT be aware of Mach-II
  - Listeners are aware of Mach-II, but something like an Account object is not aware of Mach-II
- Keeping your business logic agnostic of framework and front-end keeps your options open (e.g. Flex or AJAX)

UNIVERSAL MIND

Basic OO Architecture With Mach-II



UNIVERSAL MIND

Other Mach-II Niceties

- Beyond The Fundamentals Lie....
  - Filters
  - Plugins
  - XML Includes
  - Modules
  - Cool new stuff in Mach-II 1.6
    - Caching, debugging, admin dashboard
  - Even more cool new stuff coming in Mach-II 2.0

Let’s Review the Mach-II Request Lifecycle

1. Event is announced
2. Listener is notified (not mandatory, but typical)
3. “Stuff Happens”
4. Another event is announced, or the request ends

UNIVERSAL MIND

Resources

- Official Mach-II Web Site
  http://www.mach-ii.com
- Google Group/Mailing List
  http://groups.google.com/group/mach-ii-for-coldfusion
- Peter Farrell's Blog
  http://blog.maestropublishing.com
- Matt Woodward's Blog
  http://www.mattwoodward.com/blog
- My Blog
  http://www.infoaccelerator.net

UNIVERSAL MIND

Questions?