Model-Glue 3 Roadshow

Joe Rinehart, Firemoss, LLC

FIREMOSS

Howdy!

- I'm Joe Rinehart, president of Firemoss, LLC, located just outside of Raleigh, NC.
- We've been providing ColdFusion, Flex, and Java consulting services since late 2006.

You'll want to be here if...

- You've got any interest in the Model-Glue framework

Model-Glue 3 Roadshow

Why 3?

FIREMOSS

Why do a version 3?

- When I released Model-Glue 1.0, I stated that I didn't see a 2.0 ever happening.
- Three years later, I'm working on version 3.
- Oops.

So what happened?

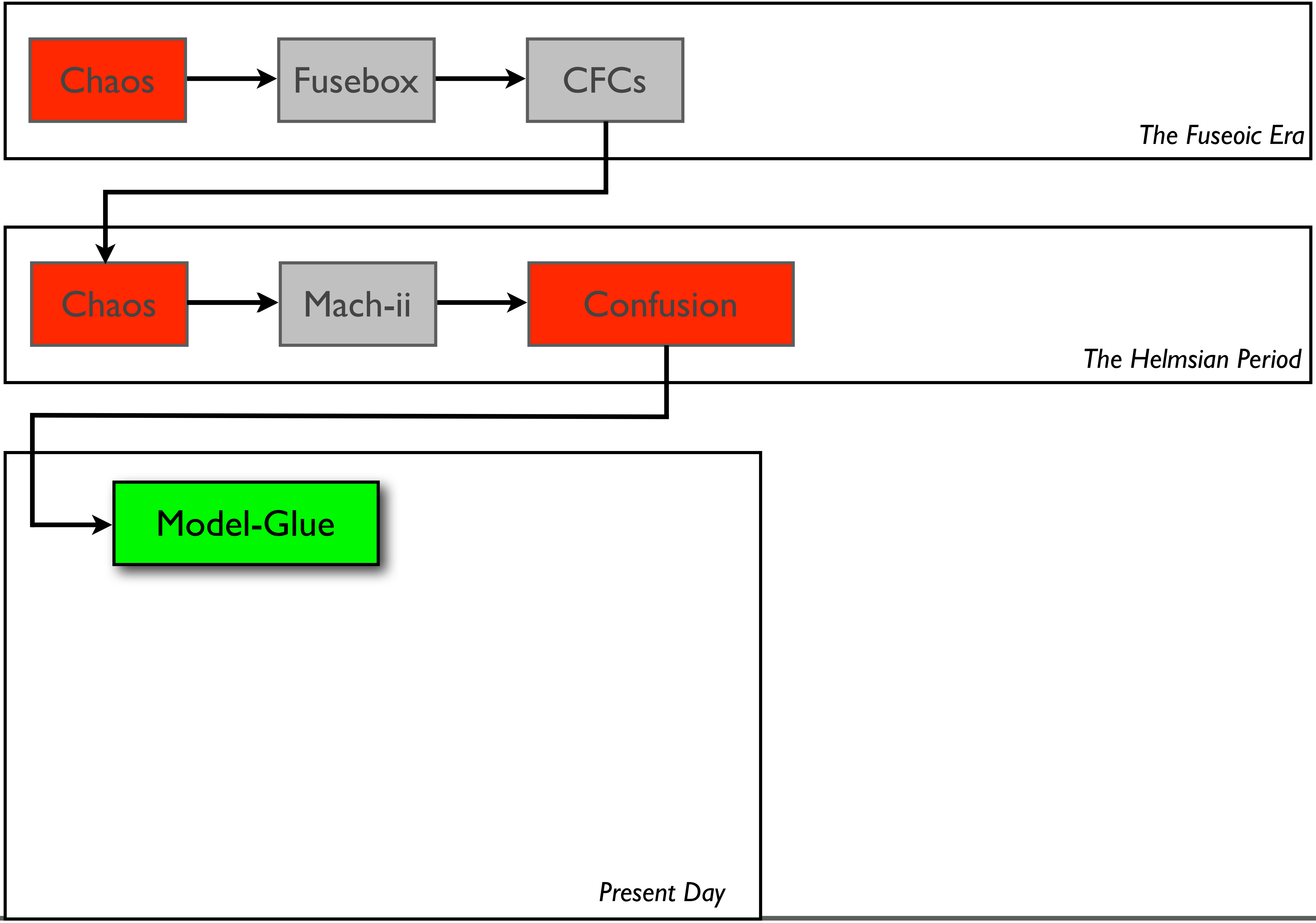
- Model-Glue 1.0 was the result of MVC framework evolution, and a product of its environment.
- The environment around OO and MVC ColdFusion development has changed, and it's causing Model-Glue to evolve.

Model-Glue 3 Roadshow

Model-Glue's Evolution: Past, Present, and Future

FIREMOSS

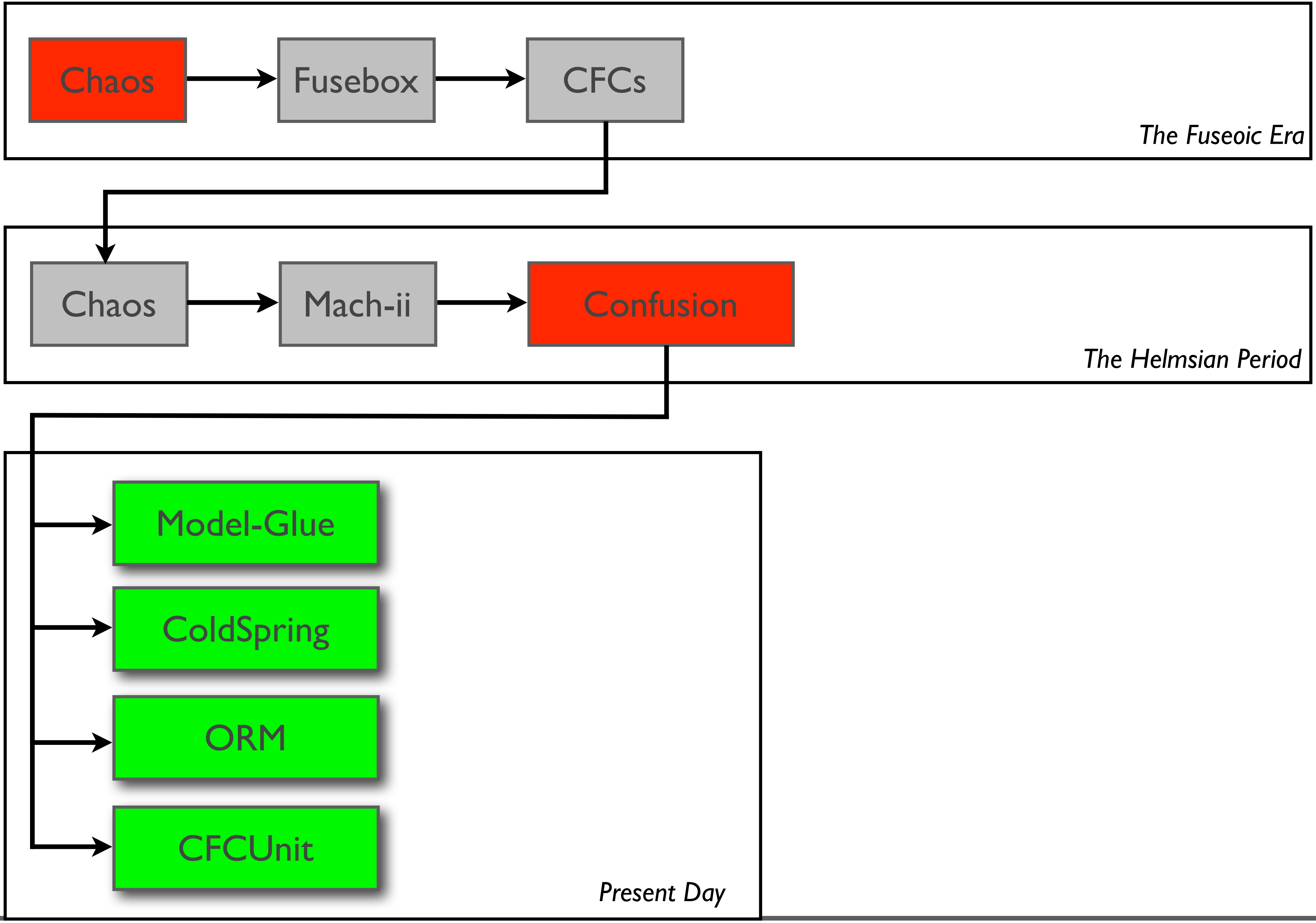
Model-Glue Thought Timeline



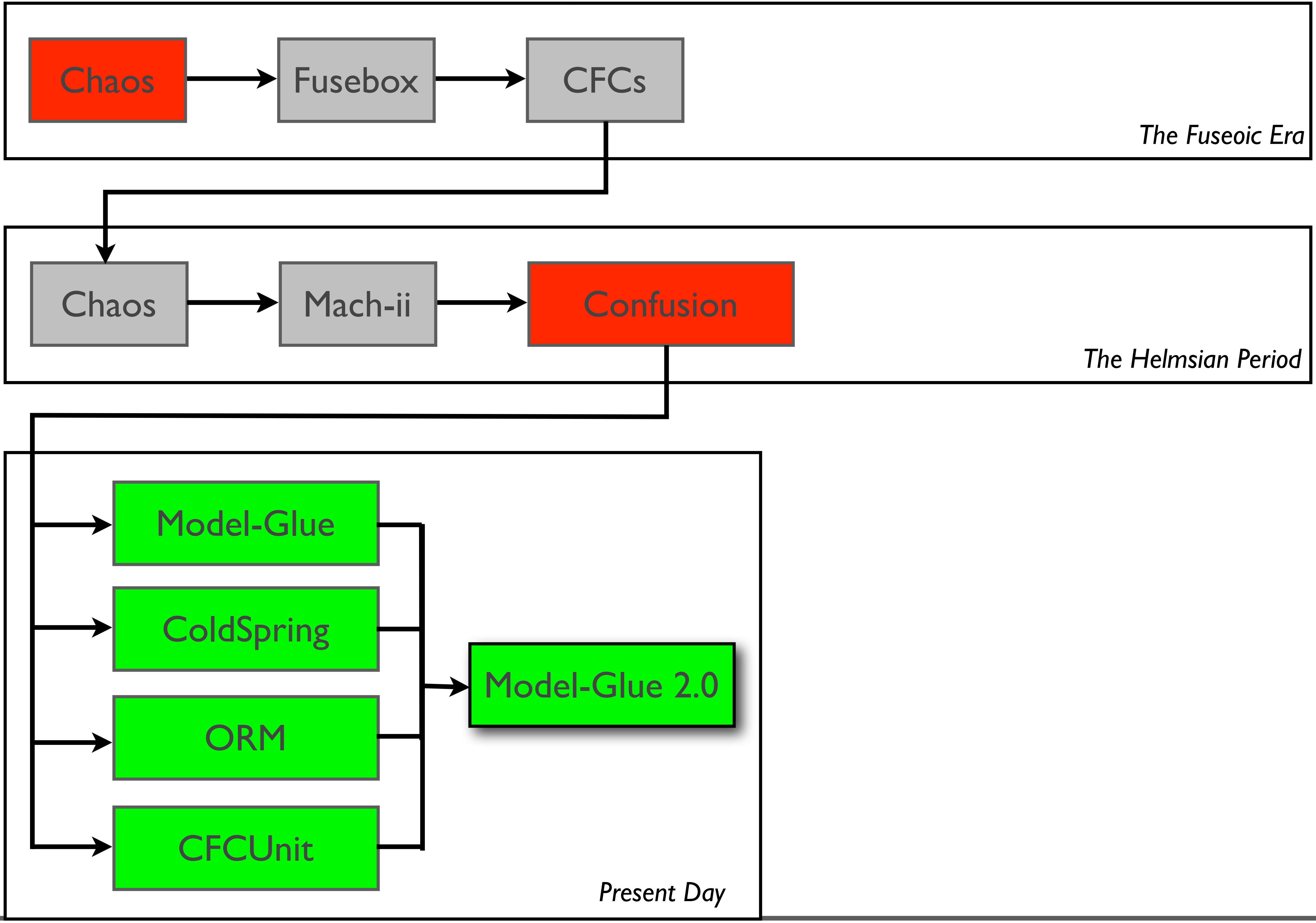
Model-Glue 1: Growing Legs

- Model-Glue 1.0 was released at a pivotal point in ColdFusion development.
- At the same time it introduced simple MVC and Implicit Invocation development, other supporting technologies were created.
- While I’ve been hesitant to change Model-Glue, this environment presented an opportunity for *true evolution*.

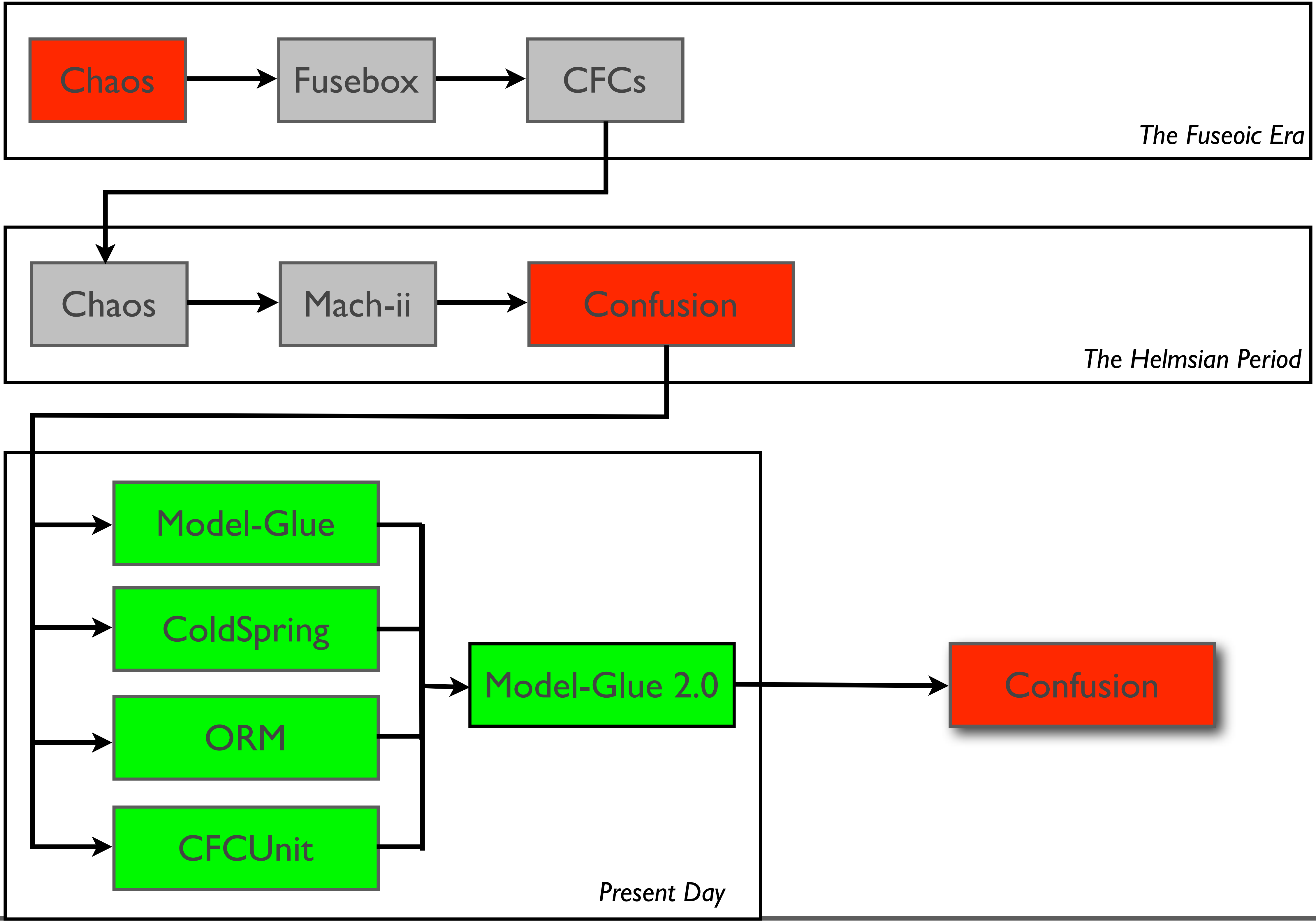
Model-Glue Thought Timeline



Model-Glue Thought Timeline

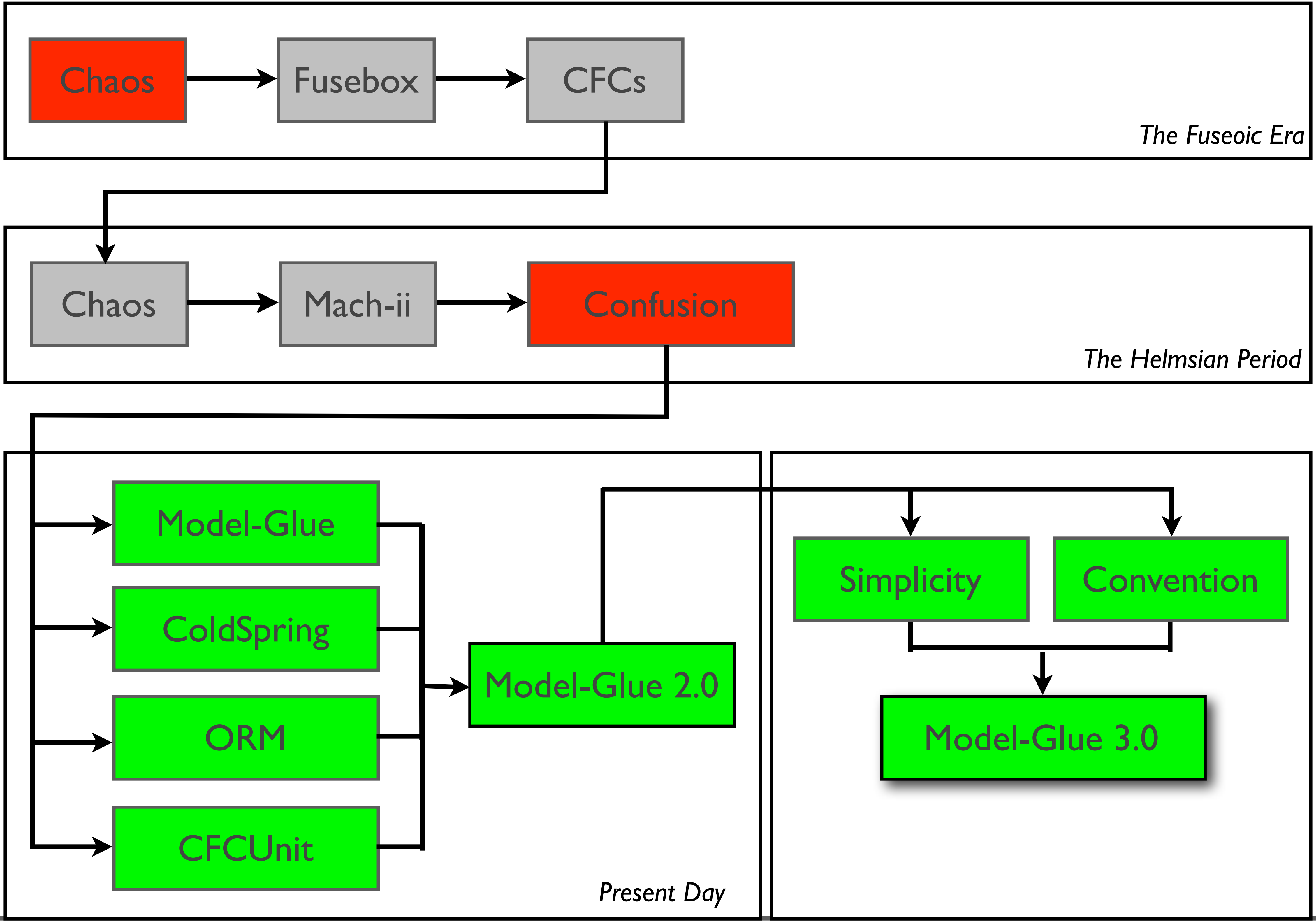


Model-Glue Thought Timeline



FIREMOSS

Model-Glue Thought Timeline



FIREMOSS

Model-Glue 3 Roadshow

Introducing "Gesture"

Model-Glue's Origins

- Over three years ago, I posted a ten-line *blog* entry introducing Model-Glue, calling it "My take on the MVC pattern."
- Model-Glue 1.0 focused on transforming the complicated and confusing world of *MVC+II* development into a simple and straightforward *development tool.*

Model-Glue's Influence

- In the past three years, the influence of Model-Glue's theme of simple, straightforward development powered by flexible, *IoC-driven architecture* has caused *change in its predecessors* and *shaped the design new frameworks*.

Model-Glue 2's Focus

- Model-Glue 2.0 was an *internal architectural release*, and did little to simplify the MVC+II portions of the framework. If anything, life was sometimes more complicated.

Model-Glue 3: Back to Basics

- ColdFusion Frameworks released since Model-Glue 1.0 such as *CFRails* and *ColdBox* have shown that Model-Glue's format isn't the endgame for simplifying MVC development.

- Model-Glue 3 is taking a back to basics approach in its conceptualization. Its goal is to provide the easiest workflow possible for doing MVC ColdFusion development *without sacrificing the benefits of adding Implicit Invocation*.

Model-Glue 3 Goal

Model-Glue 3 aims to be the easiest to use, most flexible, and most powerful MVC framework available for ColdFusion.

Model-Glue 3: Features and Architecture

- Model-Glue 3 is primarily a feature release.
- In order to provide these new features, "*some*" architectural changes are being made to internals.
- The features are all aimed at productivity, leading to the code name of "Gesture."

Model-Glue 3: New Features

So what's coming in 3?

(These may change.)

(You know what I like about this slide? I get to sound like Ben.)

Special for cf.Objective()

- Model-Glue 3's first public alpha is now available at http://www.model-glue.com. It's not well documented, and best suited for experienced Model-Glue developers.

Model-Glue 3 Roadshow

New Feature 1: Streamlined Workflow

Streamline? XML?

- Many current MVC frameworks use a shortcut to simplicity using a “Controller.Action” approach. Through a front controller mechanism, a URL parameter (e.g., “User.Save”) corresponds to a method (“save()”) on a Controller-tier class (“User” or “UserController”).

That’s not good news!

- This turns Controller methods into procedures that are often not much different from pre-framework “action” pages that processed a form! Often, these controller methods *spread tentacles* across many areas of concern: security, validation, persistence, resultant HTML rendering, etc.
- Model-Glue, Fusebox, and Mach-II all resist this by encouraging you to create small units of work: listener functions and fuses.

But units need wiring!

- Yes, they do, but isn't much of the wiring repetitive and predictable?
- Why not generate the predictable bits and allow you to fill in your customizations?

A History of Automation

- Model-Glue has always tried to ease the “getting started” tasks
- Model-Glue shipped with an application template
- Model-Glue 2 used Ant to configure a template that was already set up to connect to a database via Transfer or Reactor
- Gesture’s taking the idea a step further...

Event Generation

- When Model-Glue 3 runs in development mode with "event generation" turned on, you'll be able to enter the URL of a nonexistent event and receive a wealth of free code.

Event Generation: Details

- An event-handler tag will be added to your Model-Glue XML file.
- The event-handler tag will broadcast a like-named message.
- A controller will be generated (if necessary) and a like-named listener function added.
- A message-listener tag will be added to an appropriate controller (via convention).
- A view will be generated and added to the event-handler tag.

Event Generation: Big Picture

- Run Ant to create new Model-Glue application
- Type “index.cfm?event=user.login”
- Edit UserController.login() to perform authentication code.
- Edit /views/user/login.cfm to display results.
- No XML editing necessary!

But wait, there's more!

- When a Controller is created through Event Generation, a CFCUnit (or maybe MXUnit?) test case is created to test its listener functions.
- When methods are added, test functions will be added as well!

Model-Glue 3 Roadshow

New Feature 2: Event Types

FIREMOSS

Event-Handler XML is repetitive!

- When you’ve got 100 events, and 95 of them use a site-wide template, you’ll wind up with 95 identical XML tags in ModelGlue.xml.
- Almost every <event-handler> I create has a result that runs the overall template:

```
<result do=”view.template” />
```

Event-Handler XML is repetitive!

- This stinks.

Fixing Event Handlers

- Shouldn’t we be able to define what such an event *is* (“TemplatedEvent”) and re-use that definition?

Event Types: Details

- In Model-Glue 3, you can!
- By creating a subclass of the base EventHandler, you'll be able to create your own event types such as TemplatedEvent or LoginRequiredEvent that define their own broadcasts, results, and views.
- In your ModelGlue.xml, you'll then be able to state an alternate event type.

Event Types: Show the code!

- Old:

```
<event-handler name=”do.this”>
  <results>
    <result do=”view.template” />
  </results>
</event-handler>
```

- New:

```
<event-handler
  name=”do.this”
  type=”events.TemplatedEvent”
/>
```

But wait, there's more!

- By specifying a *type* in an event generation URL, you'll generate a new event of that type.

Model-Glue 3 Roadshow

New Feature 3: Application.cfc integration

Application.cfc Support

- Want to know when the application starts, a session starts, or a session ends?

Application.cfc support

- Just listen for it:

```
<message-listener message=”onApplicationStart” .../>
<message-listener message=”onSessionStart” .../>
<message-listener message=”onSessionEnd” .../>
```

Model-Glue 3 Roadshow

New Feature 4: SES Urls and UrlManager

FIREMOSS

SES Url Support

- SES Urls will be supported out of the box in an index.cfm/eventname/key1/value1/key2/value2 format.
- Thanks to an overhauled EventContext population architecture, we'll be able to *truly* plug in our own SES Url implementations via the *URLManager*.

SES Url Support

- To build views that are SES Url safe, a new function on the event context builds Urls:

  ```
  <a href=”#event.linkTo(“user.profile”, “userId,profileId”) />
  ```

  Might result in

  ```
  index.cfm?userId=2&profileId=42
  ```

  or:

  ```
  index.cfm/userId/2/profileId/42
  ```

Model-Glue 3 Roadshow

New Feature 5: Helpers

FIREMOSS

Ever used UDFs in Model-Glue?

- It's kind of a pain. While you can include a UDF library through a view, it's an awkward place for this configuration and doesn't help the Controller tier.

Helpers in Brief: Include-Style

- Drop CFLib.org’s dateLib.cfm into /helpers and all of its functions become available in the “helpers” scope inside of Controllers and Views:

```
<cfoutput>#helpers.dateLib.daysTilXmas()#</

cfoutput>
```

Helpers in Brief: CFC-Style

- Dropping a CFC that acts as a function library into /helpers works just like an include: its functions are copied to the "helpers" scope!

Model-Glue 3 Roadshow

New Feature 6: Bean Injection

FIREMOSS

Autowiring was a beginning!

- Autowiring allowed Model-Glue to automatically "set" defined beans into Controllers through a like-named setter function.

Bean “Injection”

- Through bean injection, you’ll no longer need to even write a setter function.

“Beans scope”

- Just wire your Controller:

```
<cfcomponent beans=”someDAO” />
```

- And use your bean:

```
<cfset beans.SomeDAO.save(SomeThing) />
```

Model-Glue 3 Roadshow

New Feature 7: Content Caching

FIREMOSS

Caching's Popular

- Most MVC frameworks now provide some form of caching mechanism.
- Gesture will not provide an object cache. That's a job of your service layer.
- Gesture will provide a dead-simple to use content cache.

Want to cache an entire event?

- Just say so:

```
<event-handler name=”page.home”
  cache=”true”
/>
```

- Outcome: Application-wide cache of content under the key “page.home”

What about a custom key or timeout?

- Just say so:

```
<event-handler name=”page.home”
  cache=”application” cacheKey=”homepage”
  timeout=”300”
/>
```

- Outcome: Application-wide cache of content under the key “homepage” for 300 seconds.

What if it differs by values?

- Use cacheKeyValues to dynamically create a cache key from event values!

```
<event-handler name=”user.profile”
  cacheKeyValues=”sessionId”
/>

or

<event-handler name=”product.details”
  cacheKeyValues=”productId”
/>
```

What if the cache becomes stale?

- Any listener function can invalidate a key:

```
<cfset beans.cacheAdapter.purge(“page.home”) />
```

Want to roll your own cache?

- MG3 ships with a very simple, intentionally weak caching system.
- If you're serious about caching, you'll likely want to implement your own.
- I'd like to work on JDBM and memcached adapters for serious performance. Volunteers welcome.

Model-Glue 3 Roadshow

*(Possible)*New Feature 8: Formats

FIREMOSS

Web apps aren’t just pages anymore.

- An MVC framework for modern Web applications must be able to present itself in a variety of ways: HTML, XML data, JSON data, and partial pages.

Duplicating work for this stinks.

- Providing the same data in *N* "formats" requires *N* event-handlers.

Getting a user list in HTML

- HTML:

```
<event-handler name="user.list.page"
  <broadcasts>
    <message name="needUserList" />
  </broadcasts>
  <views>
    <include template="dspUserList.cfm" name="body" />
  </views>
  <results>
    <result do="view.template" />
  </results>
</event-handler>
```

Getting a user list in HTML

- Partial HTML for <div> replacement:

```
<event-handler name=”user.list.page”
  <broadcasts>
    <message name=”needUserList” />
  </broadcasts>
  <views>
    <include template=”dspUserList.cfm” name=”body” />
  </views>
</event-handler>
```

Getting a user list in HTML

- In XML:

```
<event-handler name=”user.list.page”
  <broadcasts>
    <message name=”needUserList” />
  </broadcasts>
  <views>
    <include template=”dspDataAsXML.cfm” name=”body”>
      <value name=”data” name=”userQuery” />
    </include>
  </views>
</event-handler>
```

Getting a user list in HTML

- In JSON:

```
<event-handler name=”user.list.page”
  <broadcasts>
    <message name=”needUserList” />
  </broadcasts>
  <views>
    <include template=”dspDataAsJson.cfm” name=”body”>
      <value name=”data” name=”userQuery” />
    </include>
  </views>
</event-handler>
```

Enough of that!

- In Gesture, the broadcasts, views, and results blocks can be instructed to run only for certain request "formats."
- Format is simply a URL or Form value of a configurable name (default is "requestFormat").
- Now, we can combine all four into a single event!

Getting a user list

- In Gesture:

```
<event-handler name=”user.list.page”
  <broadcasts>
    <message name=”needUserList” />
  </broadcasts>
  <views format=”HTML,HTMLPartial”>
    <include template=”dspUserList.cfm” name=”body” />
  </views>
  <views format=”JSON”>
    <include template=”dspDataAsJson.cfm” name=”body”>
      <value name=”data” name=”userQuery” />
    </include>
  </views>
  <results format=”HTML”>
    <result do=”view.template” />
  </results>
</event-handler>
```

Model-Glue 3 Roadshow

*(Possible)*New Feature 9: Model-Glue Remoting

Web apps aren’t just HTML anymore.

- A typical ColdFusion application may consist of both HTML and rich Flex or Flash-based widgets.

Flex / Flash Support

- In Gesture, your app will contain both an index.cfm template as its HTML entry point as well as a ModelGlueGateway CFC as an entry point for Flex or Flash widgets it may contain.

Getting our User List in Flex

- XML is the same:

```
<event-handler name=”user.list.page”
  <broadcasts>
    <message name=”needUserList” />
  </broadcasts>
  <views format=”HTML,HTMLPartial”>
    <include template=”dspUserList.cfm” name=”body” />
  </views>
  <views format=”JSON”>
    <include template=”dspDataAsJson.cfm” name=”body”>
      <value name=”data” name=”userQuery” />
    </include>
  </views>
  <results format=”HTML”>
    <result do=”view.template” />
  </results>
</event-handler>
```

Getting our User List in Flex

- MXML:

```
<mx:RemoteObject
  id=”modelGlueGateway”
  destination=”ColdFusion”
  source=”ModelGlueGateway”
/>
```

- AS3:

```
var event:ModelGlueEvent =
  new ModelGlueEvent(“user.list”, [“userQuery”]);
modelGlueGateway.runEvent(event);
```

Conclusion

- Model-Glue 1.0 raised the bar in ease-of-use for ColdFusion frameworks.
- Since then, the bar's inched higher.
- Model-Glue 3.0 aims to raise it by a few feet, without sacrificing architectural advantages it provides.

What's the status?

- The new core is done, and pretty well baked. It runs the Model-Glue 2.0 application template and legacy Model-Glue 1.0 applications.
- The ORM controller is in place (and therefore the GDMs), but no scaffolding yet.
- Flex Remoting is not yet in place.

Model-Glue 3 will be available...

- Final release? When it's ready. Mid-2008 is likely.
- Alpha? Right now. http://www.model-glue.com .