Adobe

ColdFusion Summit 2016

Bringing Order to the Chaos: Take the MVC Plunge

Carl Von Stetten / @cfvonner

About Me

- GIS Analyst for Central Contra Costa Sanitary District (www.centralsan.org)
  - Work with ColdFusion since 2002 (CF 4.5)
  - Lots of spatial and non-spatial data modeling, data management, intranet application development
- Tools I Use:
  - ColdFusion, JavaScript (incl. jQuery & Bootstrap), Python, Microsoft SQL Server, Esri ArcGIS Desktop and Server, Geocortex, Safe FME (spatial ETLs), etc…
- Manager of Bay Area ColdFusion User Group (BACFUG)– on hiatus
- Adobe Community Professional for ColdFusion

- Married 24 years, two adult children
- Unreformed DIY remodeler
- Passionate about craft beer! (should become obvious in a few minutes)

USER GROUP MANAGER

COMMUNITY PROFESSIONAL

Agenda

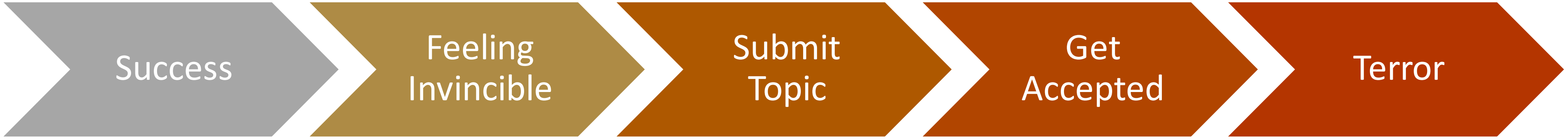
- Why am I talking about MVC?
- Prerequisites for this talk
- Brief review of procedural applications
    - Some code
- Introduction to Model-View-Controller concept
    - Some more code
- Review pros/cons of MVC

A little personal history…

- Writing/maintaining **procedural** ColdFusion apps since 2002
  - Mostly .cfm files,
  - Custom tags
  - Some components (.cfcs)

- Started a complete overhaul of our intranet map portal this spring
  - Rewrite ColdFusion portion from scratch
  - “We’re going to do it right” = go MVC
  - Working prototype within 3 weeks

Why am I speaking?


Success

Feeling Invincible

Submit Topic

Get Accepted

Terror

What do I need to know?

- CFML language (syntax, functions/tags, etc.)
- Basic understanding of components
  - How are they structured
  - How do I use them
    - CreateObject()
    - new
    - CFInvoke
    - CFObject
- ColdFusion request cycle

- Also helpful:
  - CFScript syntax (similar to JavaScript syntax)

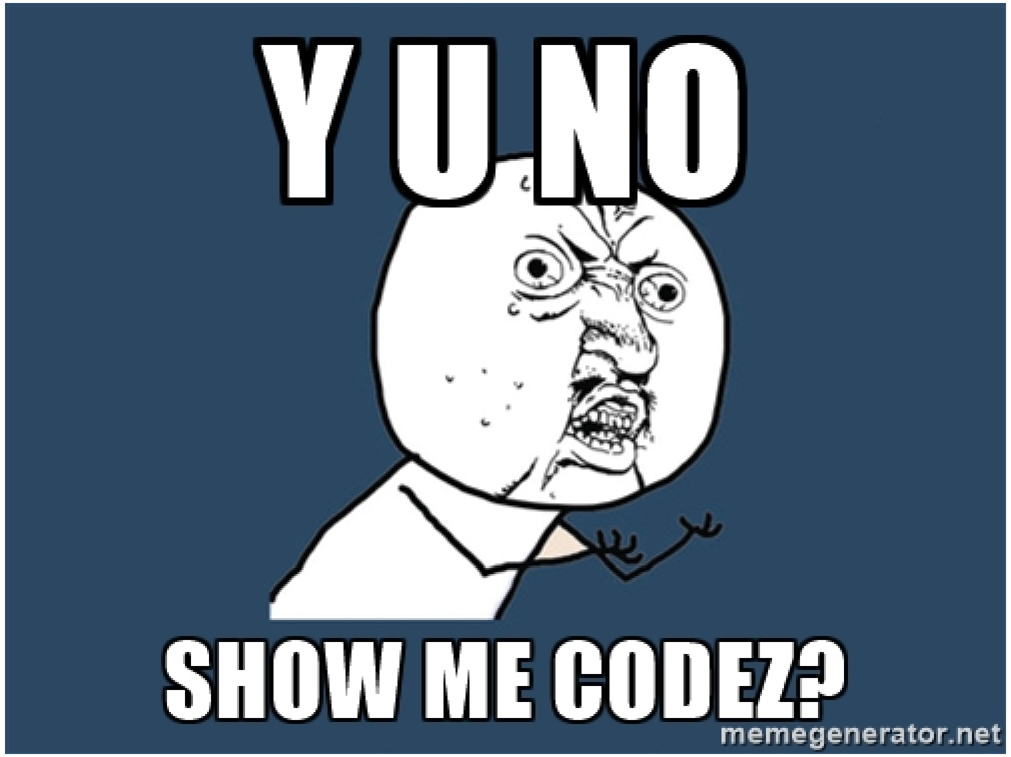
What is a procedural application?

- CFM page for each URL in app
- Pages execute from top to bottom
- Typically 100’s of lines of code per page
- Mix of business logic and display code

Disclaimer

“The code you about to see is simplified for clarity. It does not necessarily represent best practices, does not include security measures, and does not include user input validation/sanitation.” - Me

What does a procedural app look like?



Downsides of procedural code

- Increasingly hard to maintain, especially as app evolves
- Hard to distribute work to teams
- Potentially lots of code duplication
  - Violating DRY principle
- Can’t build an API from the business logic
  - Intertwined with display code

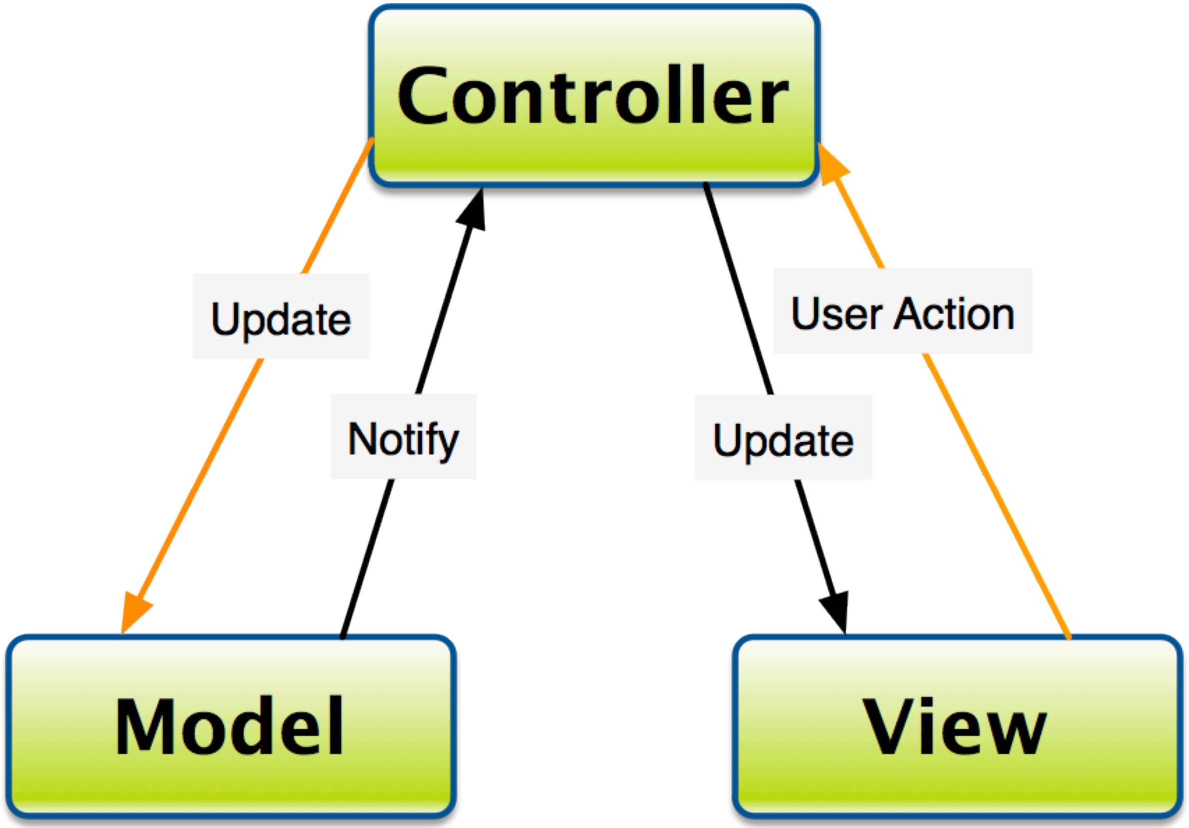
Is there a better way?

- YES!

- MVC
  - (you knew I was going to say that)

What is MVC?

- Model-View-Controller design pattern
- Separates concerns (business logic vs. view logic)
- Common design pattern in OO languages
- Usually leverage MVC framework
  - Rails for Ruby
  - Django for Python
  - ASP.Net MVC
  - Express/Sails for Node.js
  - Laravel/CakePHP for PHP

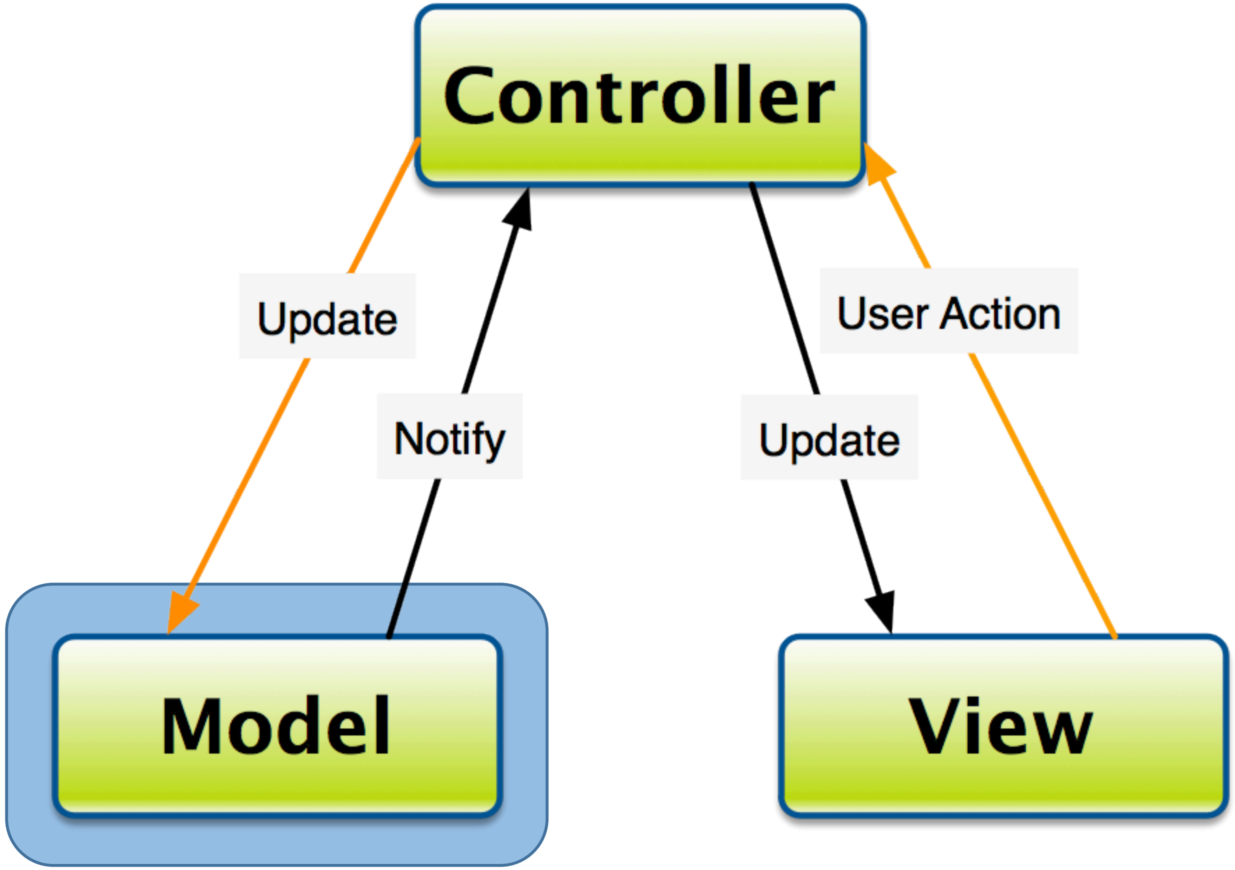


Carl Von Stetten / @cfvonner

Model Layer

- Business logic
- Database interactions (CRUD)
- External web service access
- Validation
- Helper services

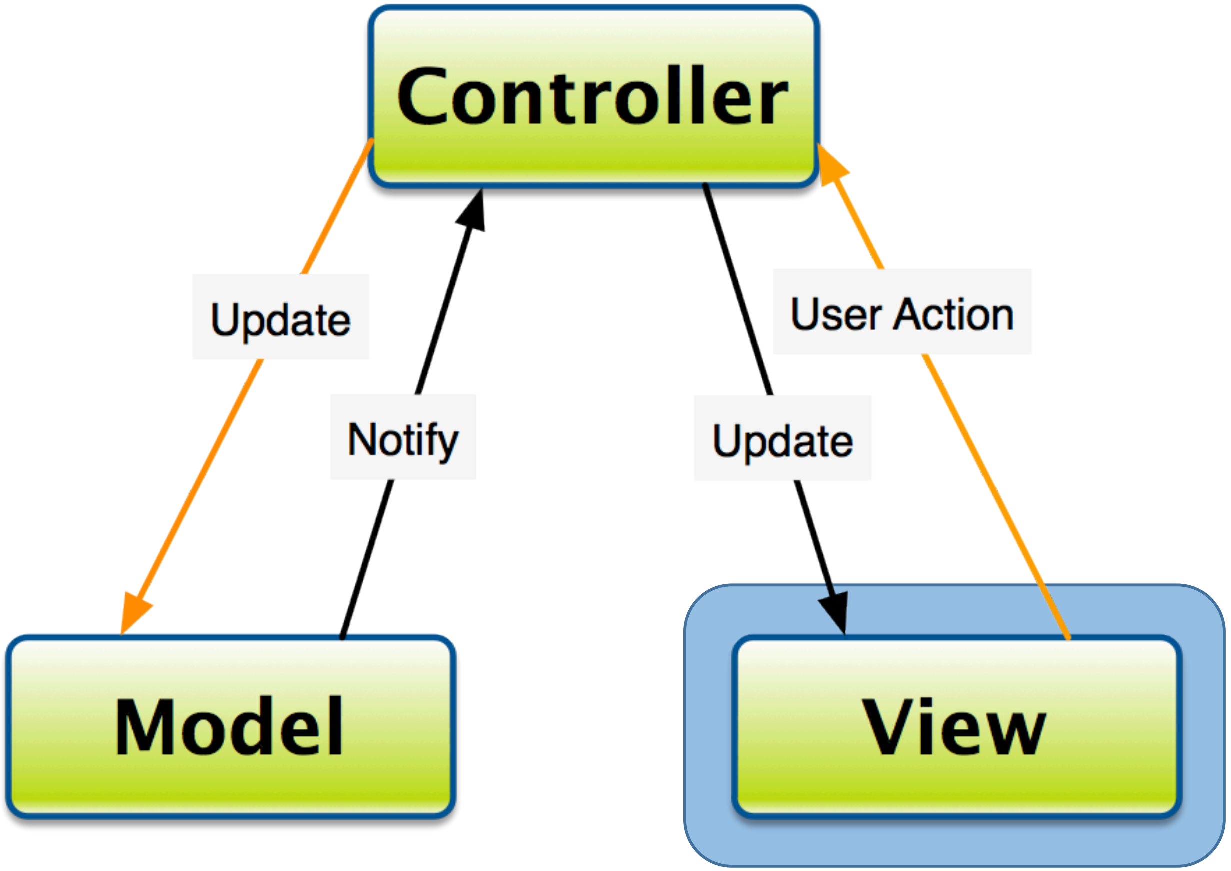
- Should not know anything about the framework (including controllers and views)
- Responds to requests from the controller



View Layer

- User Interface
- HTML/CSS/JavaScript
- Minimal Server-Side logic for display control
- NO Business logic!!!
  - NO SQL code here!!!

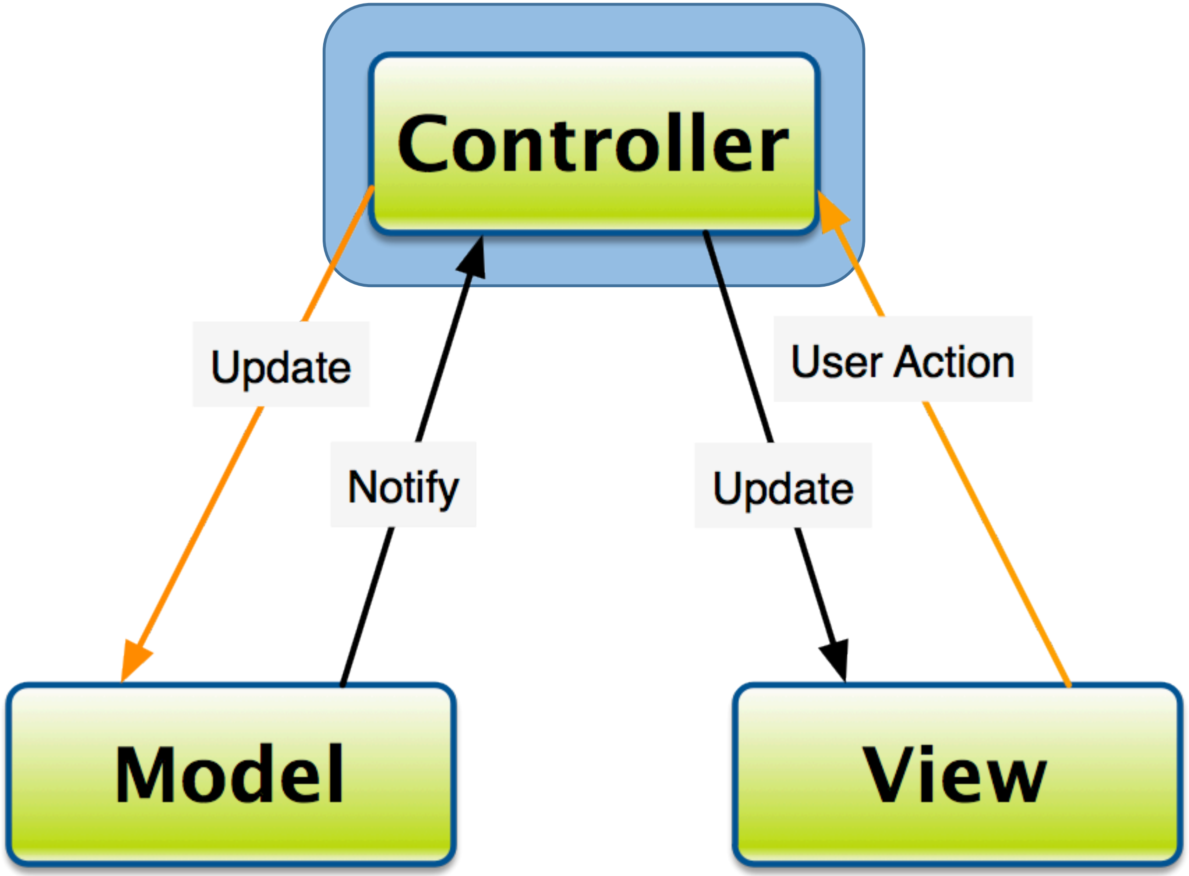
- Should not know anything about the model
- Relies on Controller to get data



Controller Layer

- Controls the flow of the application
- Examines each incoming request (URL/Form variables or path)
- Calls relevant business logic (model objects/components)
- Passes results from model to view layer

- Minimal validation
- Short controller functions



MVC in ColdFusion

- MVC can be done without a framework
  - Unless your hobby is reinventing the wheel, why would you?
- Frameworks offer lots of functionality you don’t have to reinvent
  - Dependency Injection/Inversion of control
    - Stay in this room for Nolan Erck: *Dependency Injection: Why is it awesome and why should I care?*
  - URL Routes
  - Layout/view templating
  - Data Rendering (JSON, XML, text, custom)
  - Modules
- Frameworks often improve code organization

Modern MVC Frameworks

- Framework-One (FW/1)
- ColdBox

- Some older frameworks still in use
  - Model-Glue
  - Mach II
  - Fusebox
  - CFWheels
  - FarCry Core

What is Framework-One (FW/1)?

- Created by Sean Corfield
- Version 4.0 recently released
- Small, lightweight, **convention-over-configuration** MVC framework
  - MVC portion is one file, one.cfc (130KB)
  - Two related files:
    - ioc.cfc – dependency injection/inversion of control
    - aop.cfc – aspect-oriented programming

What is “convention over configuration”?

- Structure your application folders per recommendations and the framework figures out the rest for you
- You only need to specify configuration settings if you:
  - Need to override defaults
  - Deviate from the recommended structure
  - Deviate from the file naming conventions

What the heck is “rc”?

- **rc** variable = alias for request context, or request.context
- Sort of another scope (like application, session, request, etc.)
- Mechanism for passing data between framework, controller, and views
- FW/1 automatically populates **rc** with URL and FORM variables
- You write controller methods to insert anything else your views need

What does a FW/1 MVC app look like?

Talk is cheap. Show me the code.

– Linus Torvalds

MVC - the good and bad

Pros

- Promotes DRY (code reuse)
- Assists team efforts
    - Different parts of model, controllers, views can be worked on simultaneously
- Common pattern/terminology
- Enforces better code organization

Cons

- Have to learn some new concepts, even if excellent procedural programmer
- Might seem to end up with more files to manage
    - But the files will be more focused and on-point

Migrating Legacy Apps

- Don’t have to “eat the elephant” all at once
- Choose logical sections/modules of an app to migrate OR start new work in MVC pattern
- MVC code can coexist with procedural code
  - With combination of configuration settings and web server rewrite rules

Key takeaways

- MVC isn’t as hard as it seems
- It will make application code maintenance easier for the future

Learn more at ColdFusion Summit!

- Directly related:
  - *Dependency Injection: Why is it awesome and why should I care?* – Nolan Erck
    - Banyan B – 11:30am-12:30pm

- Easier development environments:
  - *Instant ColdFusion Servers with Vagrant* – Trip Ward
    - Banyan B – 1:30pm-2:30pm
  - *Herding Cats: A new way to manage all your Adobe servers on one dev machine* – Brad Wood
    - Jasmine F – 4pm-5pm

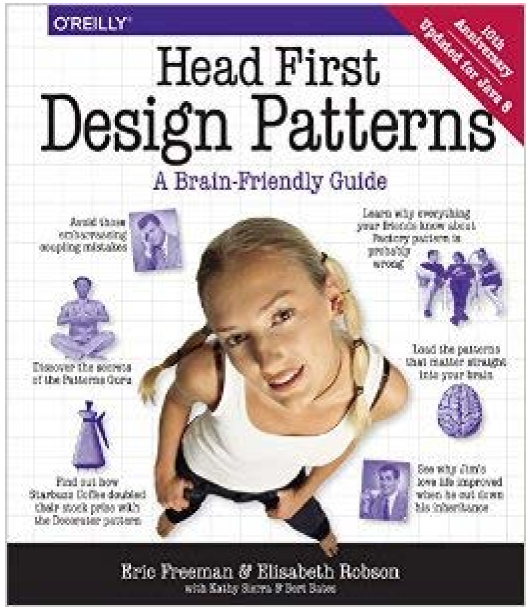
MVC framework resources

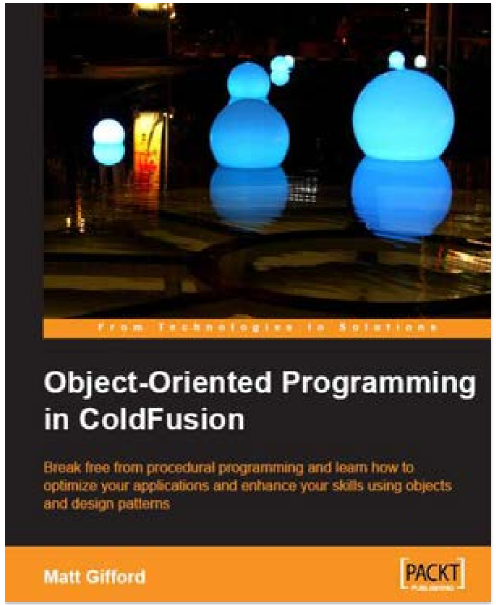
- Documentation & Downloads
  - Framework-One
    - Docs: http://framework-one.github.io/
    - Download: https://github.com/framework-one/fw1
  - ColdBox - https://www.coldbox.org/
- Discussion Groups
  - Framework-One Google Group
  - ColdBox Google Group

- Real-Time Assistance
  - Slack Team
    - FW/1 channel
    - Box-products channel

Some more resources

- Books:


O'REILLY
Head First Design Patterns
A Brain-Friendly Guide
10th Anniversary Updated for Java 8
Eric Freeman & Elisabeth Robson
with Kathy Sierra & Bert Bates


From Technologies to Solutions
Object-Oriented Programming in ColdFusion
Break free from procedural programming and learn how to optimize your applications and enhance your skills using objects and design patterns
Matt Gifford
PACKT

- Get Help:
  - Adobe ColdFusion forum:
    - http://adobe.ly/2eg63QJ
  - StackOverflow.com

- Documentation:
  - Adobe ColdFusion docs:
    - http://adobe.ly/2eg7nTD
  - Community docs:
    - http://cfdocs.org/
      - Links back to ACF and Lucee docs
      - Great CFScript info!!

- Real-Time Assistance
  - CFML Slack Team

Thank you!!!

- Questions???

- Contact me:
  - Email: carl.vonstetten1@gmail.com
  - Twitter: @cfvonner
  - GitHub: cfvonner
  - CFML Team on Slack: @cfvonner

- Don’t forget to fill out session evaluation
  - You can do it in the Mobile App on iOS and Android!

- Code/slides will be on Github: https://github.com/cfvonner/CFSummit2016-MVC

Thank You!