Frameworks for SOA Platforms

Sharing Data Between ColdFusion Applications

Elliott Sprehn

elliott@teratech.com

http://www.elliottsprehn.com/blog/

A Warning…

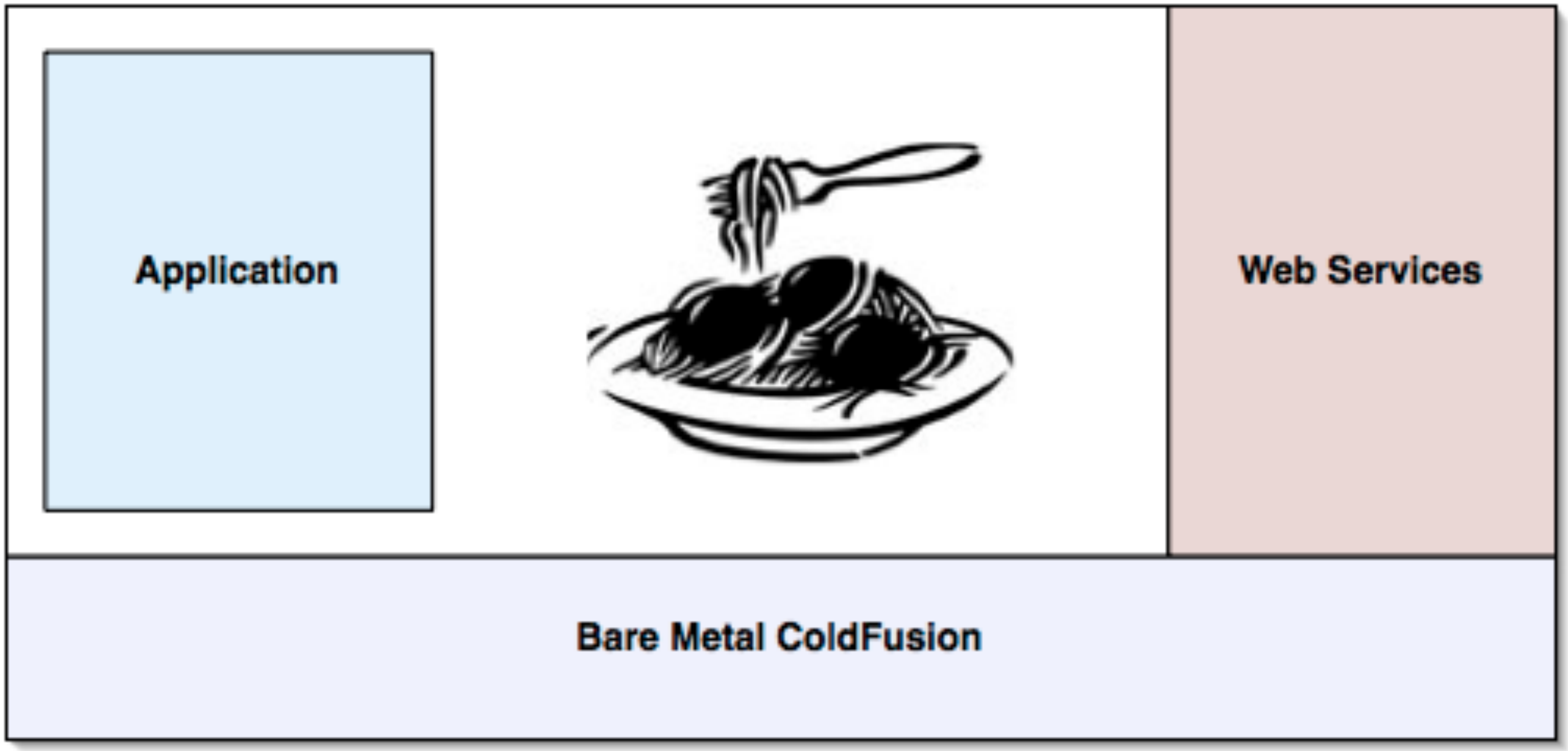
We're going to **totally rethink** how we build applications, it's totally new, **completely different** and nothing like what we're already doing….

What’s the Service Layer?

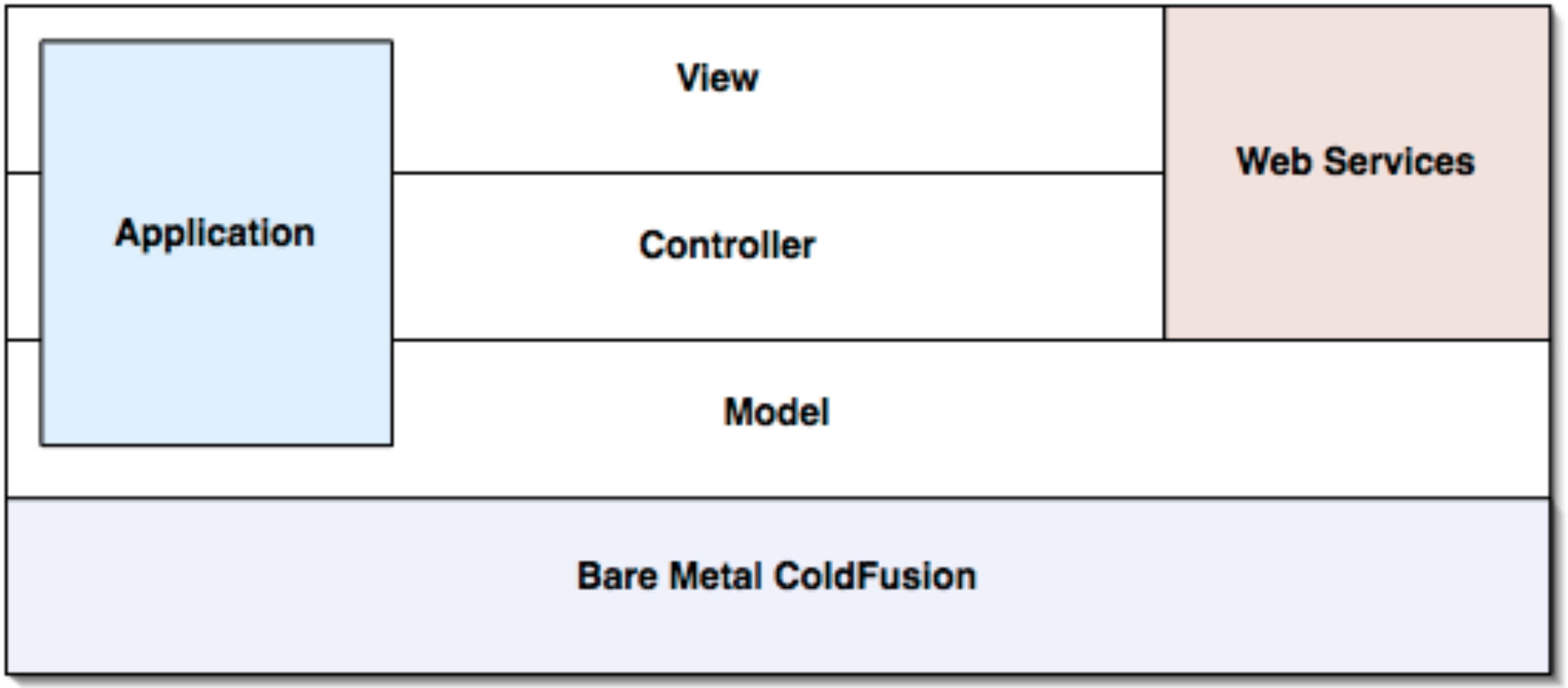
- Set of services usable as a foundation for applications.
- Provides services, but not application behavior.
- What’s the difference between the Model Layer?
  - In traditional ColdFusion application design… nothing.

The Evolution of Application Design…

Spaghetti Application

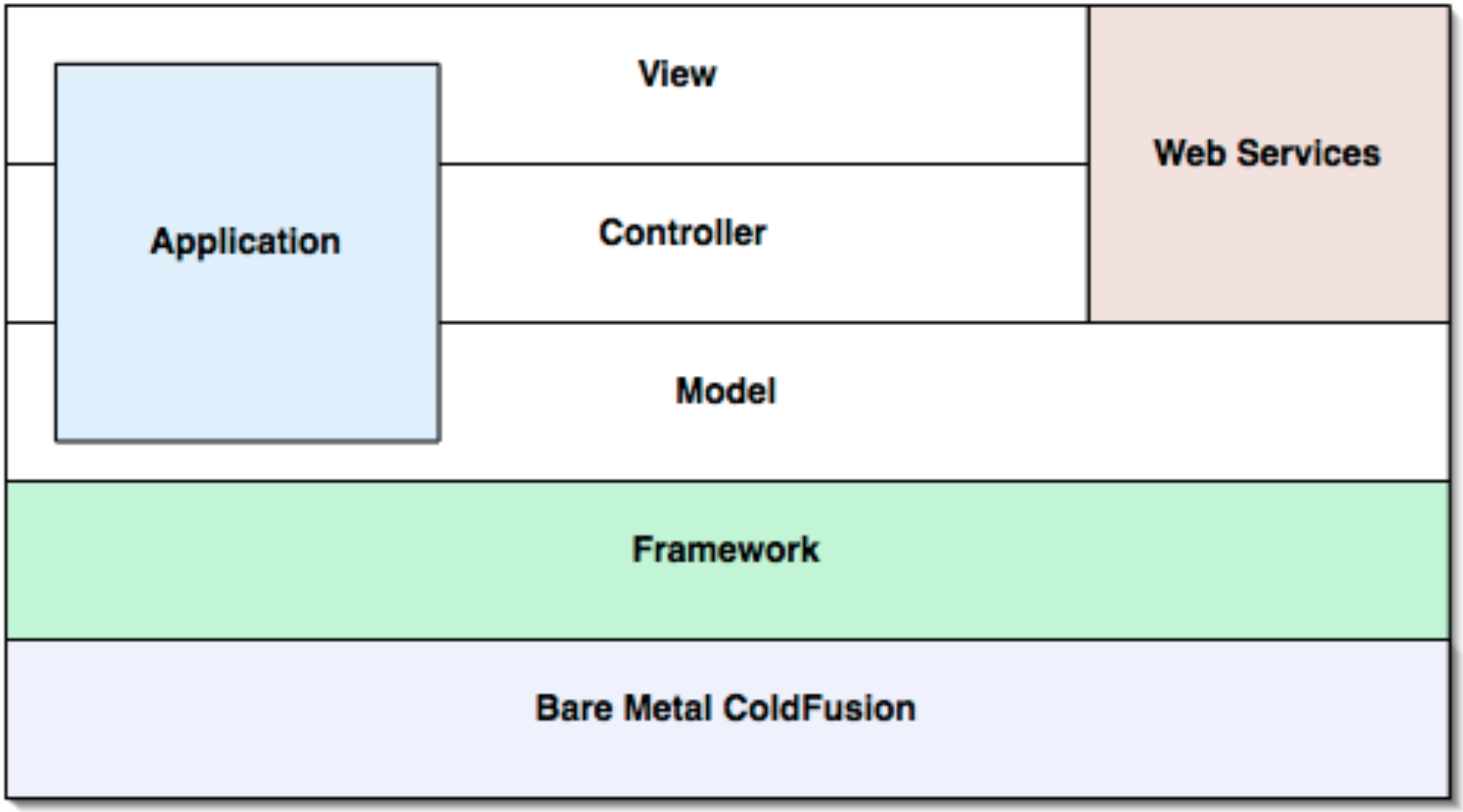


MVC Application

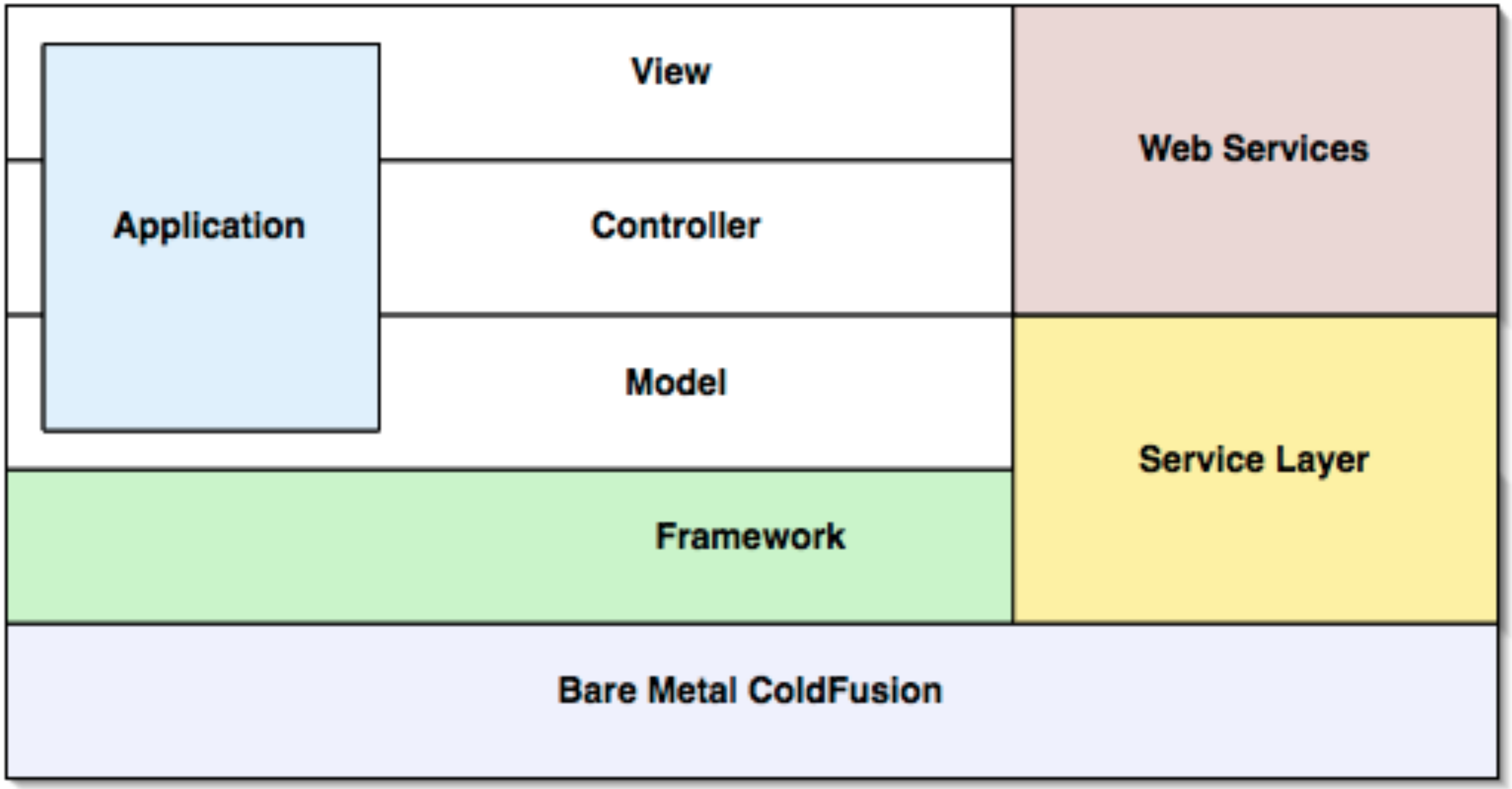


Not as tasty, unfortunately, but you're here for the pizza anyway….

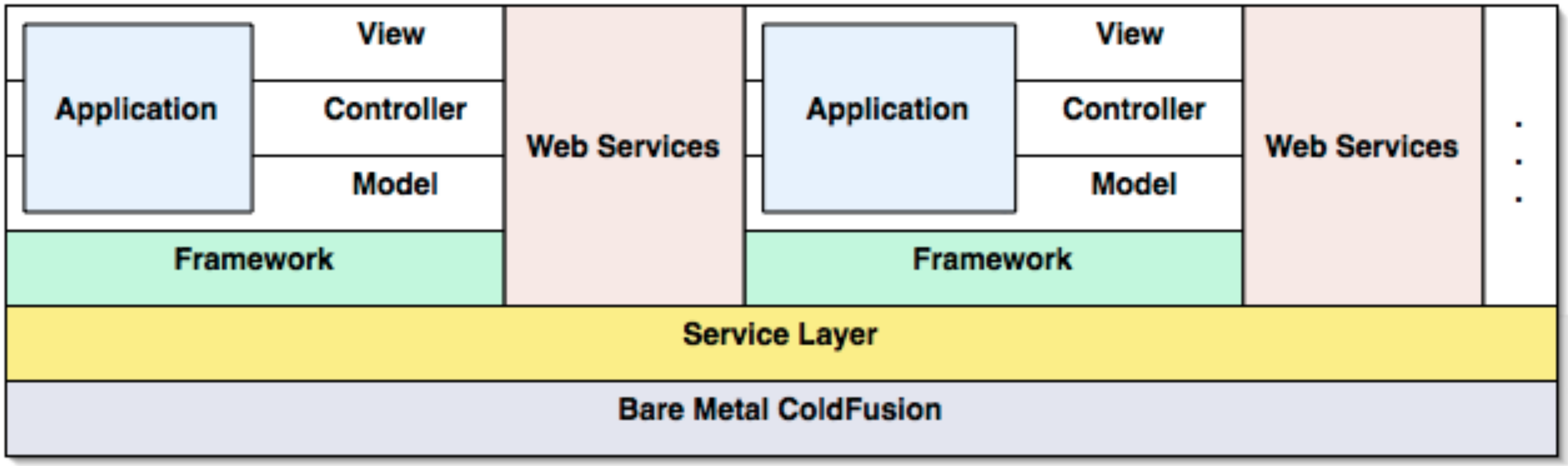
Framework Application



Framework + Service Layer



Shared Service Layer



Application
View
Controller
Model
Framework
Web Services
Application
View
Controller
Model
Framework
Web Services
. . .
Service Layer
Bare Metal ColdFusion

- Many applications share the same Services
    - Login, Models, Utilities…
- Framework Agnostic
    - No Framework, Model-Glue, *XYZ* Framework

Benefits of Sharing

- Code and Behavior Reuse
  - Easily achieved by shared code base… but no coordination or instance reuse.
- Simplifies Adding new features and modules
- Allows rapid “scripting”
  - Quick tasks don’t need entire applications.

Modules are Applications

- Core Features EQ Individual Applications
  - Wikis, Blogs, Schedulers, …
- Scales well to large projects
  - Developers can work on applications separately.
  - Only worry about shared services changing
  - Prevents Stale Service Code

What Tools?

- ColdFusion’s Application.cfc
  - Transparent Front Controller
  - Supports “extension” already.
- ColdSpring
  - Singleton Objects/ Services
  - Nice Configuration
    - Imports are awesome.
  - Extension mechanism for transparent usage
- Model-Glue (Fusebox??)
  - ColdSpring support built in.
  - Provides MVC with little work.

Not Special!

Scripting

- Often overkill to create an entire application to solve a simple request
  - On unix you write “shell scripts”

- Why not CF Scripts?
  - With this scripting is transparent.
  - No extra leg work.
  - Save your time for better things…

This is Craziness…

- All conference applications are a single CF application.
  - cfunited.com
  - blog.cfunited.com
  - europe.cfunited.com
  - express.cfunited.com
  - scheduler.cfunited.com
  - ugmm.cfunited.com
  - wiki.cfunited.com
  - callforspeakers.cfconf.org
- All share ***LIVE*** Models, Services, Login, …

Lets See it!

Disadvantages?

- Can’t scale to huge load if you’re not careful.
  - Bad to depend too much on shared data.
  - Encapsulate as much as possible.
  - Facades around scopes.
- Must always remember thread safety… everywhere!
- More testing
  - Subsystem changes may require testing every application all over again…

It’s All Good.

- Saves tons of time.
- Allows really neat things.
- Simplified maintenance and upgrades.

- Finds the ultimate question to the ultimate answer about the universe, life and everything… okay maybe not.

Questions?

Can also contact me at:

Elliott Sprehn
TeraTech, Inc.
elliott@teratech.com

See you at CFUnited 2009

cfunited
the premier coldfusion technical conference
The Premiere Coldfusion Technical Conference
CF
Fx
AIR