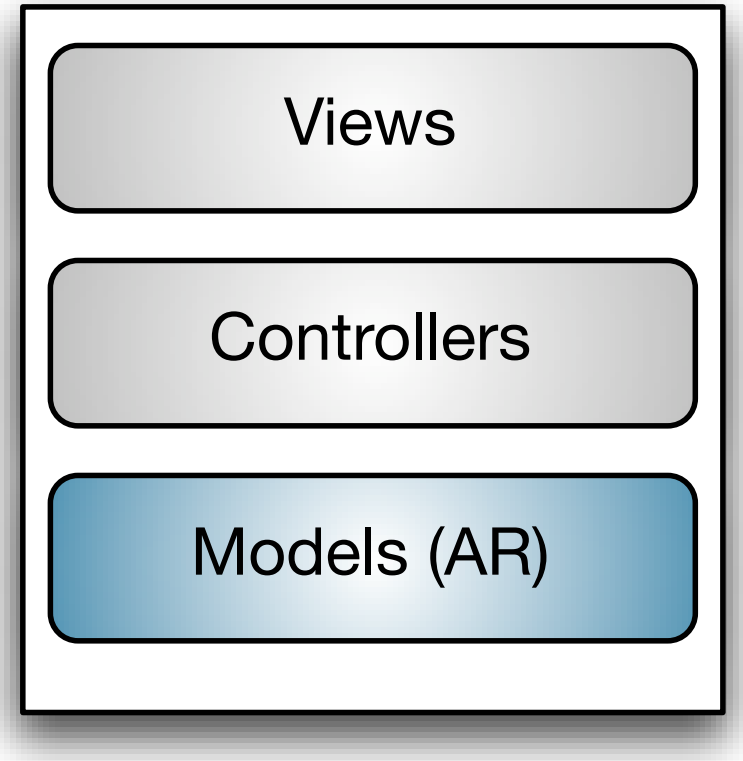
RAILS HYDRA

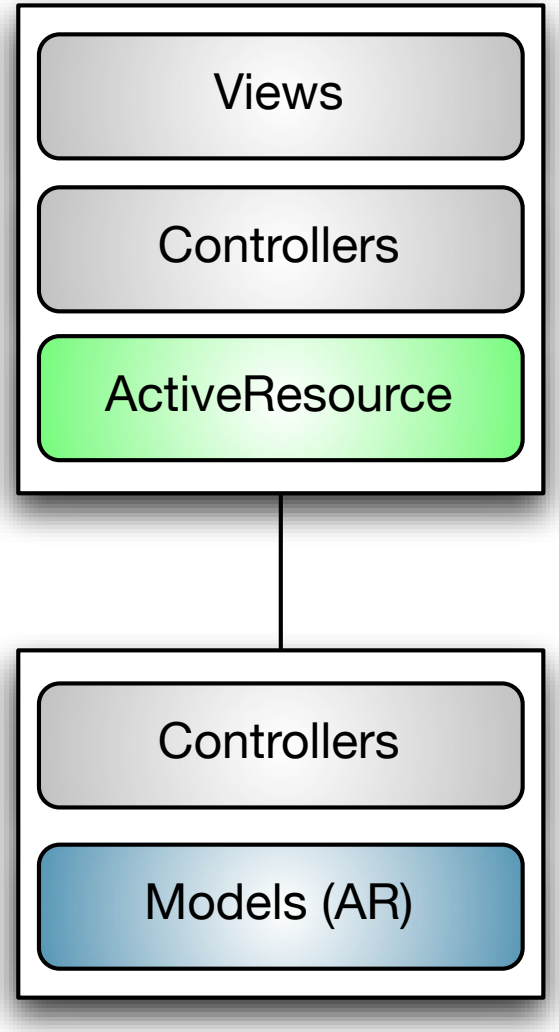
Synthesizing an Application out of Multiple Rails Codebases

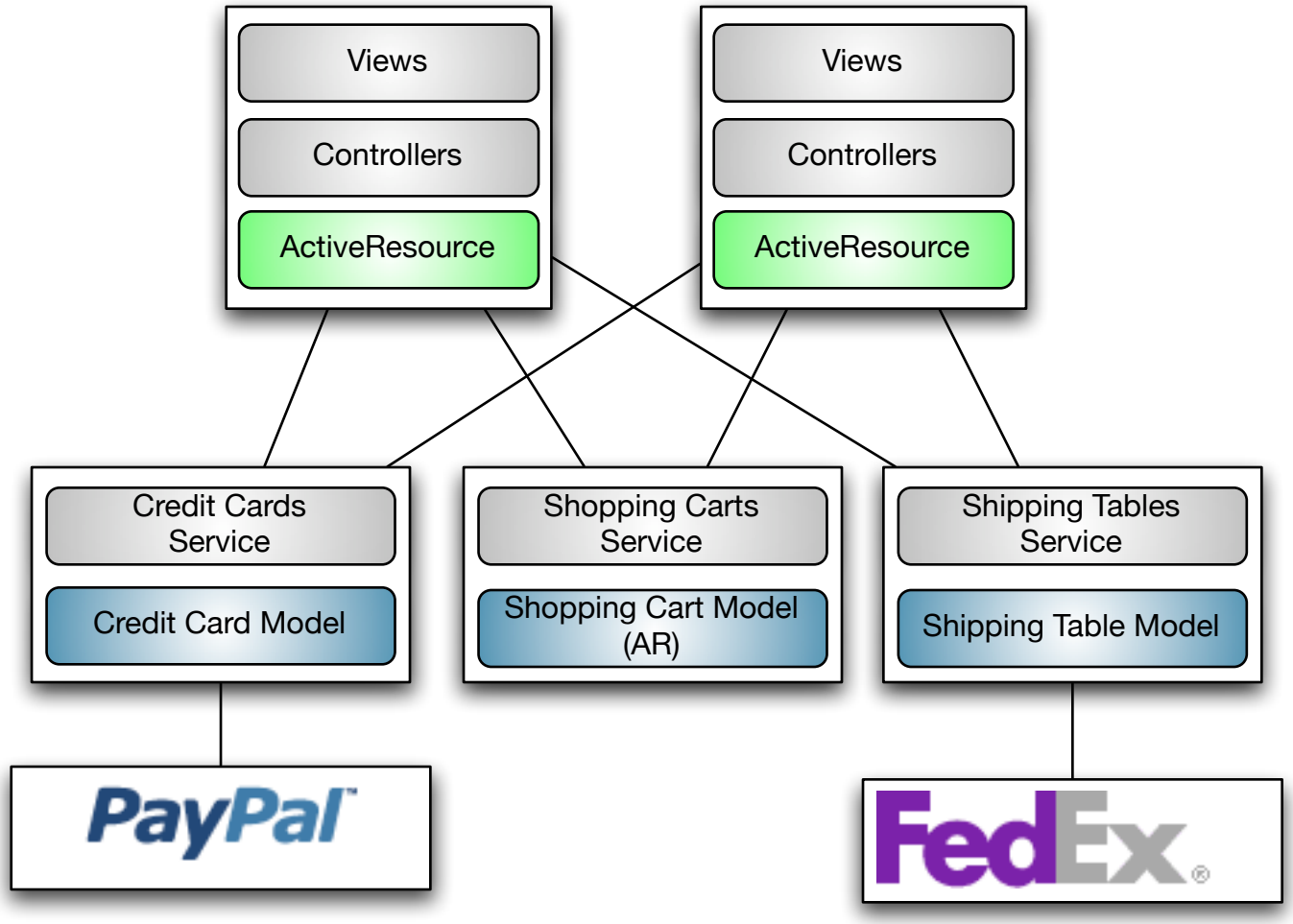
RailsConf Europe 2007

Craig McClanahan
Nick Sieger
Sun Microsystems, Inc.

Huh?




Views
Controllers
ActiveResource
Controllers
Models (AR)


Views
Controllers
ActiveResource
Views
Controllers
ActiveResource
Credit Cards Service
Credit Card Model
Shopping Carts Service
Shopping Cart Model (AR)
Shipping Tables Service
Shipping Table Model
PayPal
FedEx

Advantages

- Internal mashups
- Flexible deployment
- Reduce coupling
- Runtime code reuse
- Talk to non-Rails services (XML...)
- REST + Etags & cache-control = Caching
  - (HTTP reverse proxies)

Uniform Interface

http://media.rubyonrails.org/presentations/worldofresources.pdf

Duck Typed Models

http://flickr.com/photos/loramir/203476026/

ActiveResource

```
class Model < ActiveResource::Base
  self.site = "http://localhost:3000"
end
```

vs

```
class Model < ActiveRecord::Base
end
```

```
$ script/generate scaffold posts
...
$ egrep "(Post|@post)\." app/controllers/posts_controller.rb
5:    @posts = Post.find(:all)
9:      format.xml  { render :xml => @posts.to_xml }
16:   @post = Post.find(params[:id])
20:     format.xml  { render :xml => @post.to_xml }
26:   @post = Post.new
31:   @post = Post.find(params[:id])
37:   @post = Post.new(params[:post])
40:     if @post.save
46:       format.xml  { render :xml => @post.errors.to_xml }
54:   @post = Post.find(params[:id])
57:     if @post.update_attributes(params[:post])
63:       format.xml  { render :xml => @post.errors.to_xml }
71:   @post = Post.find(params[:id])
72:   @post.destroy
```

```
Post.new (Post.create)
Post.find
@post.save
@post.update_attributes
@post.errors
@post.destroy
```

Demos

- 3-tier resource with ActiveResource + Scaffold
- Change scaffold controller to make_resourceful
- ActiveAtomResource?
- Optimizing service chatter

Demos

```
rails app
rails web
require 'active_resource'
require 'make_resourceful'
require 'atom/feed'
require 'rails_utils'
```

Lessons Learned

- ActiveResource Considerations
  - Presumes server serializes data structures the Rails way
  - Presumes XML data formats
  - Chatty HTTP communication
  - New model instances don't know the valid attributes without getting loaded

Lessons Learned

- Scaling Considerations
  - > Keep view-app tier communications stateless!
  - > Cache where and when necessary

- Deployment Considerations
  - > .war deployments to Glassfish instead of Mongrels