

Cheap, Fast, and Good

You can have it all with Ruby on Rails

Brian McCallister

brianm@chariotsolutions.com

<http://www.chariotsolutions.com/>

What is Ruby?

- Dynamic and Interpreted
- Strong support for OO programming
 - Everything is an object (`2.next == 3`)
- Strong support for functional-style programming
 - Blocks, closures, first-class functions
- Child of Perl and Smalltalk

What is Rails?

- Model-2 Web Framework
- Object/Relational Mapping Library
- SOAP Producer/Consumer Framework
- Email Templating/Mailing Library
- Code Generator

- Very Rapidly Evolving!

Principles Involved

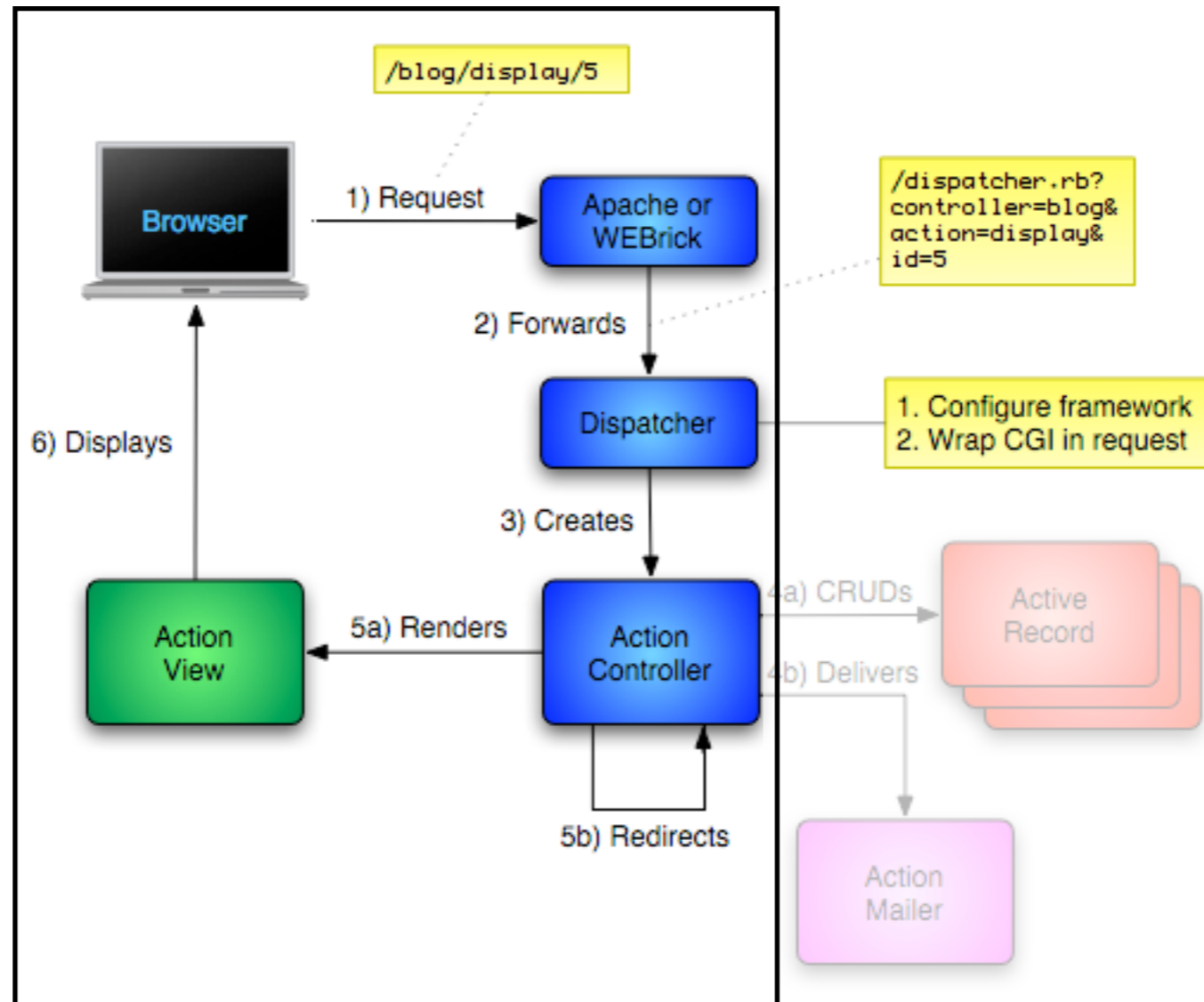
- Less Code
- Convention over Configuration
- Proximity
- Least Surprise

- Make components play nicely together!

Action Pack

That Web Stuff

Request Cycle



Some Code

```
require 'date'

class AggregateController < ApplicationController

  model :entry

  def for_date
    date = DateTime.parse @params[:date]
    logger.debug "for_date: #{date}"
    @entries = Entry.on_date date
  end

  # Additional actions removed for slide's benefit`


end
```

Action

```
require 'date'

class AggregateController < ApplicationController

  model :entry

  def for_date ← 
    date = DateTime.parse @params[:date]
    logger.debug "for_date: #{date}"
    @entries = Entry.on_date date
  end

  # Additional actions removed for slide's benefit`

end
```


Controller

```
require 'date'
```

Controller



```
class AggregateController < ApplicationController
```

```
  model :entry
```

Action



```
  def for_date
```

```
    date = DateTime.parse @params[:date]
```

```
    logger.debug "for_date: #{date}"
```

```
    @entries = Entry.on_date date
```

```
  end
```

Action



```
  def list_entries
```

```
    @entries = Entry.find_all
```

```
  end
```

```
  # Additional actions removed for slide's benefit`
```

```
end
```

How We Got Here

```
ActionController::Routing::Routes.draw do |map|

  # map.connect ':controller/service.wsdL',
  #           :action => 'wsdl'

  map.connect 'wombat/is/friendly',
             :controller => :feeds,
             :action => :list

  map.connect '',
             :controller => :feeds,
             :action => :list

  # Default Route
  map.connect ':controller/:action/:id'
end
```

routes.rb

```
ActionController::Routing::Routes.draw do |map|
```

```
# map.connect ':controller/:action/:id'
#
```

<http://localhost/wombat/is/friendly>



```
map.connect 'wombat/is/friendly',
            :controller => :feeds,
            :action => :list
```

```
map.connect '',
            :controller => :feeds,
            :action => :list
```

```
# Default Route
map.connect ':controller/:action/:id'
end
```

routes.rb

```
ActionController::Routing::Routes.draw do |map|
```

```
# map.connect ':controller/service.wsdL',  
#           :action => 'wsdl'
```

```
map.connect 'wombat/is/friendly',  
           :controller => :feeds,  
           :action => :list
```

<http://localhost/>

```
map.connect '',  
           :controller => :feeds,  
           :action => :list
```

```
# Default Route  
map.connect ':controller/:action/:id'  
end
```

routes.rb

```
ActionController::Routing::Routes.draw do |map|
```

```
  # map.connect ':controller/service.wSDL',  
  #           :action => 'wSDL'
```

```
  map.connect 'wombat/is/friendly',  
             :controller => :feeds,  
             :action => :list
```

```
  map.connect '',  
             :controller => :feeds,  
             :action => :list
```

<http://localhost/feeds/list>

```
  # Default Route  
  map.connect ':controller/:action/:id'  
end
```

Show Us Something!

```
<h1>Recent Stories...</h1>
<table class="entryTable" >
<% for entry in @entries %>
  <tr>
    <td class="entryTitle">
      <%= entry.feed.title + ": " + entry.title %>
    </td>
  </tr>
  <tr>
    <td class="entryBody">
      <%= entry.body %>
    </td>
  </tr>
<% end %>
</table>
```

Directives

```
<h1>Recent Stories...</h1>
<table class="entryTable" >
  <% for entry in @entries %>
    <tr>
      <td class="entryTitle">
        <%= entry.feed.title + ": " + entry.title %>
      </td>
    </tr>
    <tr>
      <td class="entryBody">
        <%= entry.body %>
      </td>
    </tr>
  <% end %>
</table>
```

Expressions

```
<h1>Recent Stories...</h1>
<table class="entryTable" >
  <% for entry in @entries %>
    <tr>
      <td class="entryTitle">
        <%= entry.feed.title + ": " + entry.title %>
      </td>
    </tr>
    <tr>
      <td class="entryBody">
        <%= entry.body %>
      </td>
    </tr>
  <% end %>
</table>
```


Helpers & Partials

```
<h1>Recent Stories...</h1>
<%=
  render_partial "entry_list",
                :stories => @entries
%>
```

my_view.rhtml

```
<table class="entryTable" >
<% for entry in stories %>
  <tr>
    <td class="entryTitle">
      <%= entry.feed.title + ": " + entry.title %>
    </td>
  </tr>
  <tr>
    <td class="entryBody">
      <%= entry.body %>
    </td>
  </tr>
<% end %>
</table>
```

_entry_list.rhtml

Helpers

```
<h1>Recent Stories...</h1>
<%=
  render_partial "entry_list",
                :stories => @entries
%>
```

my_view.rhtml

```
<table class="entryTable" >
<% for entry in stories %>
  <tr>
    <td class="entryTitle">
      <%= entry.feed.title + ": " + entry.title %>
    </td>
  </tr>
  <tr>
    <td class="entryBody">
      <%= entry.body %>
    </td>
  </tr>
<% end %>
</table>
```

_entry_list.rhtml

Parameterized Partials

```
<h1>Recent Stories...</h1>  
<%=  
  render_partial "entry_list",  
                :stories => @entries  
%>
```

my_view.rhtml

```
<table class="entryTable" >  
<% for entry in stories %>  
  <tr>  
    <td class="entryTitle">  
      <%= entry.feed.title + ": " + entry.title %>  
    </td>  
  </tr>  
  <tr>  
    <td class="entryBody">  
      <%= entry.body %>  
    </td>  
  </tr>  
<% end %>  
</table>
```

_entry_list.rhtml

Picking Views

- Default Views
- The View for that one over there...
- Named Views
- Something completely different?

Default Views

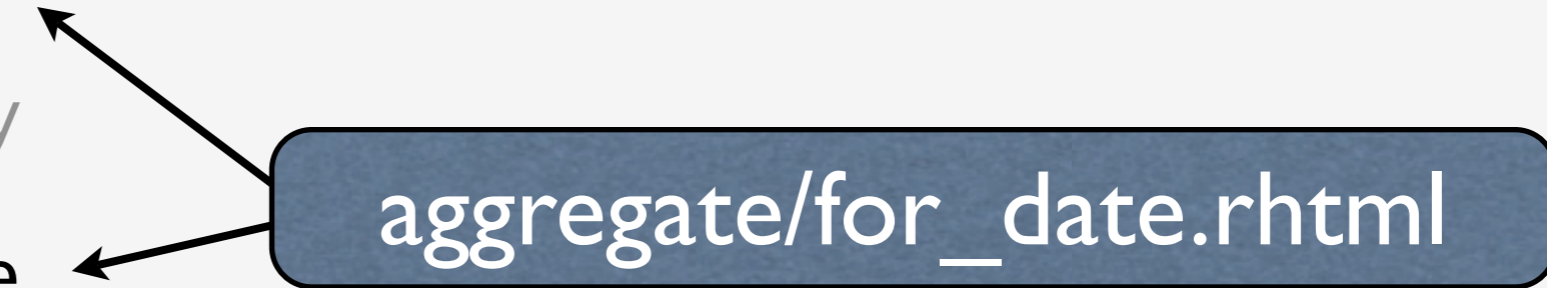
```
require 'date'

class AggregateController < ApplicationController
  model :entry

  def for_date
    date = DateTime.parse @params[:date]
    logger.debug "for_date: #{date}"
    @entries = Entry.on_date date
  end

  # Additional actions removed for slide's benefit`

end
```



The diagram consists of a dark blue rounded rectangle containing the text 'aggregate/for_date.rhtml' in white. Two black arrows originate from this box: one points to the 'for_date' method definition in the code, and the other points to the 'AggregateController' class name in the code.

render_*

```
require 'date'

class AggregateController < ApplicationController

  model :entry

  def today
    @entries = Entry.on_date Date.today
    render_action :list_entries
  end

  def list_entries
    @entries = Entry.find_all
  end

  # Additional actions removed for slide's benefit`

end
```

aggregate/list_entries.rhtml

Named Views

```
require 'date'


class AggregateController < ApplicationController

  model :entry

  def another_one
    @entries = Entry.find_all
    render :a_template
  end

  # Additional actions removed for slide's benefit`

end
```



Raw Rendering

```
require 'date'

class AggregateController < ApplicationController

  model :entry

  def direct_write
    render_text "Hello World!"
  end

  # Additional actions removed for slide's benefit

end
```

No template!



Additional Options

- `send_data`
- `send_file`
- `render_to_string`
- `render_nothing`
- `render_text` with a block
- `redirect_to`
- `redirect_to_url`
- `redirect_to_path`
- `builders`

- and more!

Helpers

- Programmatic output generation
- Global, Controller Specific, Importable
- Like tag libraries
 - kind of
- Lots of built-ins

#{Controller}Helper

```
module AggregateHelper
```

```
  def say_hello(name)
    "<h1>Hello, #{name}"
  end
```

```
end
```

```
<h1>Recent Stories...</h1>
<%= say_hello "Brian" %>
<table>
<% for entry in @entries %>
...

```

Helper

Usage

Rules, well Suggestions

- `#{Controller}Helper` applied from matching Controller
- `ApplicationHelper` available everywhere
- Declare reliance on a specific Helper from any Controller
- Rarely need to do this, though

Action Pack Magic 3

- Controllers
- Helpers
- Views

- Things not discussed:
 - Components
 - Caching
 - Filters

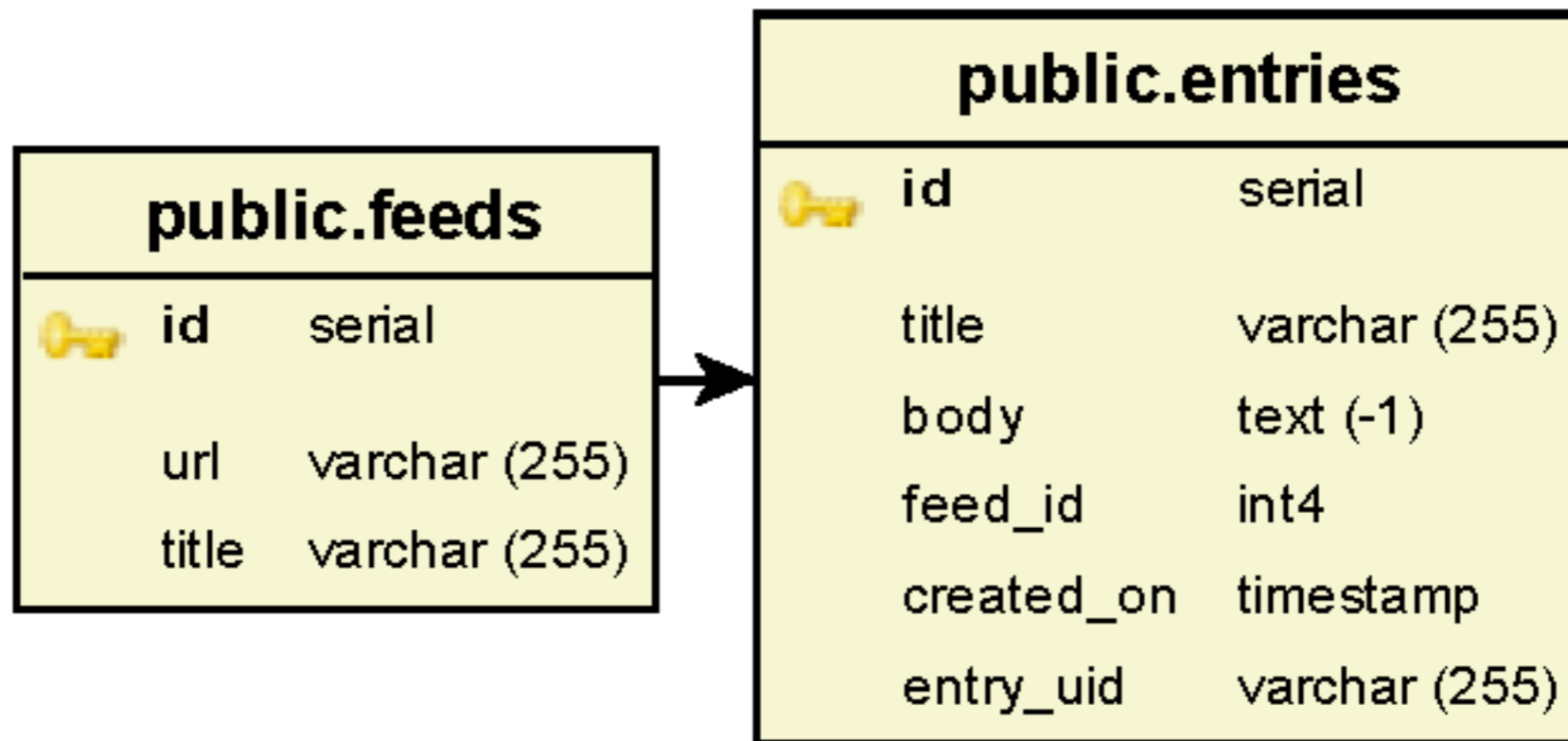
Active Record

You get the data from the database
and shake it all about...

Active Record Basics

- Not Required!
- One Row == One Instance
- Dynamic Properties by Default
- Query by SQL or Criteria
 - Including joins
- PostgreSQL, MySQL, Oracle, DB2 ... more

The Schema



The Classes

```
class Feed < ActiveRecord::Base
  has_many :entries

  validates_presence_of :title, :url
end

...

class Entry < ActiveRecord::Base
  belongs_to :feed

  validates_presence_of :title, :body, :entry_uid
  validates_uniqueness_of :entry_uid
end
```

Relations

```
class Feed < ActiveRecord::Base
  has_many :entries

  validates_presence_of :title, :url
end

...

class Entry < ActiveRecord::Base
  belongs_to :feed

  validates_presence_of :title, :body, :entry_uid
  validates_uniqueness_of :entry_uid
end
```

Queries

```
def Entry.recent(count)
  Entry.find :all,
            :order => 'created_on DESC',
            :limit => count,
            :include => [:feed]
end

def Entry.after(date)
  Entry.find :all,
            :conditions => ['created_on >= ?', date],
            :order => 'created_on DESC',
            :include => [:feed]
end

def Entry.before(date)
  Entry.find_by_sql ["select e.* from entries e where
                    created_on < ?", date]
end
```

Criteria

```
def Entry.recent(count)
  Entry.find :all,
            :order => 'created_on DESC',
            :limit => count,
            :include => [:feed]
end

def Entry.after(date)
  Entry.find :all,
            :conditions => ['created_on >= ?', date],
            :order => 'created_on DESC',
            :include => [:feed]
end

def Entry.before(date)
  Entry.find_by_sql ["select e.* from entries e where
                    created_on < ?", date]
end
```

SQL

```
def Entry.recent(count)
  Entry.find :all,
             :order => 'created_on DESC',
             :limit => count,
             :include => [:feed]
end

def Entry.after(date)
  Entry.find :all,
             :conditions => ['created_on >= ?', date],
             :order => 'created_on DESC',
             :include => [:feed]
end

def Entry.before(date)
  Entry.find_by_sql ["select e.* from entries e where
                    created_on < ?", date]
end
```

Code Generator

This stuff rocks!

Creating Rails Project

```
brianm@kite:~/Sites$ rails apachecon
create  app
create  app/apis
create  app/controllers
create  app/helpers
create  app/models
...
create  log/test.log
```

```
brianm@kite:~/Sites$ ls -F apachecon/
```

```
CHANGELOG  Rakefile  components/  db/  lib/
public/    test/     README      app/  config/
doc/       log/     script/     vendor/
```

```
brianm@kite:~/Sites$
```

This Created...

- Project Hierarchy
- Config file (for database connection)
- Rakefile (Makefile)
- System Test Harness
- Unit Test Harness
- Apache HTTPD Configs (.htaccess)
- Additional Code Generation Scripts...

Configure Database

```
brianm@kite:~/Sites/apachecon$ cat config/  
database.yml  
development:  
  adapter: postgresql  
  database: ruby_blogs_dev  
  host: localhost  
  username: blogs  
  password: *****(*)  
  
...  
  
production:  
  adapter: postgresql  
  database: ruby_blogs  
  host: localhost  
  username: blogs  
  password: *****(*)  
brianm@kite:~/Sites/apachecon$
```

Some CRUD

```
brianm@kite:~/Sites/apachecon$ ./script/generate scaffold Talk
dependency model
exists      app/models/
exists      test/unit/
exists      test/fixtures/
create      app/models/talk.rb
create      test/unit/talk_test.rb
create      test/fixtures/talks.yml
exists      app/controllers/
exists      app/helpers/
create      app/views/talks
exists      test/functional/
create      app/controllers/talks_controller.rb
create      test/functional/talks_controller_test.rb
create      app/helpers/talks_helper.rb
create      app/views/layouts/talks.rhtml
create      public/stylesheets/scaffold.css
create      app/views/talks/list.rhtml
create      app/views/talks/show.rhtml
create      app/views/talks/new.rhtml
create      app/views/talks/edit.rhtml
create      app/views/talks/_form.rhtml
brianm@kite:~/Sites/apachecon$
```

A Table

```
apachecon=> create table talks (  
  id serial primary key,  
  name varchar(255) not null,  
  presenter varchar(255) not null,  
  description text not null);
```

```
NOTICE: CREATE TABLE will create implicit sequence  
        "sessions_id_seq" for serial column "sessions.id"
```

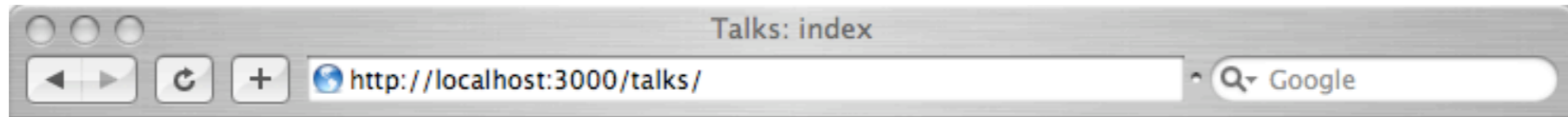
```
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index  
        "sessions_pkey" for table "sessions"
```

```
CREATE TABLE  
apachecon=>
```

Start the Server...

```
brianm@kite:~/Sites/apachecon$ ./script/server  
=> Rails application started on http://0.0.0.0:3000  
[2005-05-25 18:00:12] INFO WEBrick 1.3.1  
[2005-05-25 18:00:12] INFO ruby 1.8.2 (2004-12-25)  
[2005-05-25 18:00:12] INFO WEBrick::HTTPServer#start:  
pid=26996 port=3000
```

List



Listing talks

Description Presenter Name

[New talk](#)

TalksController

```
class TalksController < ApplicationController
  def index
    list
    render_action 'list'
  end

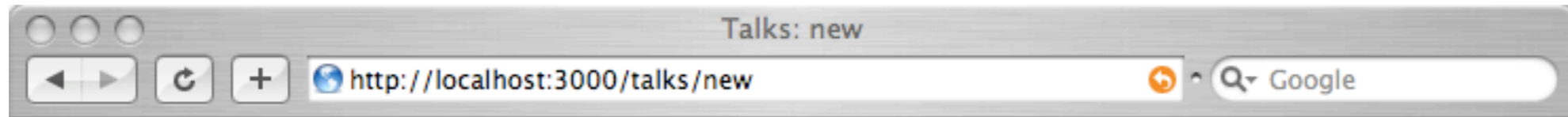
  def list
    @talk_pages, @talks = paginate :talk, :per_page => 10
  end

  def show
    @talk = Talk.find(@params[:id])
  end

  ...

end
```

Create



New talk

Description

A whirlwind tour of Rails!

Presenter

Brian McCallister

Name

Cheap, Fast, Good!

Create

[Back](#)

The Edit View

```
<h1>Editing talk</h1>
<%= start_form_tag :action => 'update', :id => @talk %>
  <%= render_partial "form" %>
  <%= submit_tag "Edit" %>
<%= end_form_tag %>

<%= link_to 'Show', :action => 'show', :id => @talk %> |
<%= link_to 'Back', :action => 'list' %>
```

talks/edit.rhtml

```
<%= error_messages_for 'talk' %>

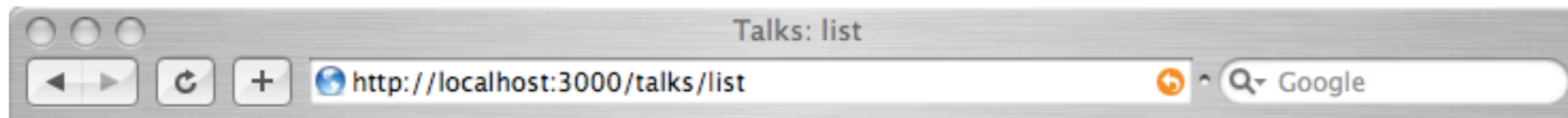
<!--[form:talk]-->
<p><label for="talk_description">Description</label><br/>
<%= text_area 'talk', 'description' %></p>

<p><label for="talk_presenter">Presenter</label><br/>
<%= text_field 'talk', 'presenter' %></p>

<p><label for="talk_name">Name</label><br/>
<%= text_field 'talk', 'name' %></p>
<!--[eoform:talk]-->
```

talks/_form.rhtml

List Again



Listing talks

Description	Presenter	Name
A whirlwind tour of Rails!	Brian McCallister	Cheap, Fast, Good! Show Edit Destroy

[New talk](#)

Unit and System Tests

```
brianm@kite:~/Sites/apachecon$ rake
(in /Users/brianm/Sites/apachecon)
ruby -Ilib:test "/usr/local/lib/ruby/gems/1.8/gems/rake-0.5.4/lib/
rake/rake_test_loader.rb" "test/unit/talk_test.rb"
Loaded suite /usr/local/lib/ruby/gems/1.8/gems/rake-0.5.4/lib/rake/
rake_test_loader
Started
.
Finished in 0.11654 seconds.

1 tests, 1 assertions, 0 failures, 0 errors
ruby -Ilib:test "/usr/local/lib/ruby/gems/1.8/gems/rake-0.5.4/lib/
rake/rake_test_loader.rb" "test/functional/
talks_controller_test.rb"
Loaded suite /usr/local/lib/ruby/gems/1.8/gems/rake-0.5.4/lib/rake/
rake_test_loader
Started
.....
Finished in 0.456943 seconds.

8 tests, 26 assertions, 0 failures, 0 errors
brianm@kite:~/Sites/apachecon$
```

.htaccess

```
# General Apache options
AddHandler fastcgi-script .fcgi
AddHandler cgi-script .cgi
Options +FollowSymLinks +ExecCGI

# If you don't want Rails to look in certain directories,
# use the following rewrite rules so that Apache won't ... <snip />
#
# Example:
# RewriteCond %{REQUEST_URI} ^/notrails.*
# RewriteRule .* - [L]

# Redirect all requests not available on the filesystem to Rails
# By default the cgi dispatcher is used which is very slow
#
# For better performance replace the dispatcher with the fastcgi one
#
# Example:
# RewriteRule ^(.*)$ dispatch.fcgi [QSA,L]
RewriteEngine On
RewriteRule ^$ index.html [QSA]
RewriteRule ^([\^.]*)$ $1.html [QSA]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ dispatch.cgi [QSA,L]
...
```

Extensible Generator

- Some 3rd Party Generators:
 - Login Generator
 - Salted Hash Login Generator
 - Tabbed Navbar Generator
 - Search Generator
 - Configuration Generator
 - Postback Generator

Deployment Time

Stuff to be aware of...

Environment

- Development
- Test
- Production

Web Servers

- Webrick
- Apache Web Server
 - CGI
 - mod_ruby
 - fastcgi
- lighttpd
 - fastcgi

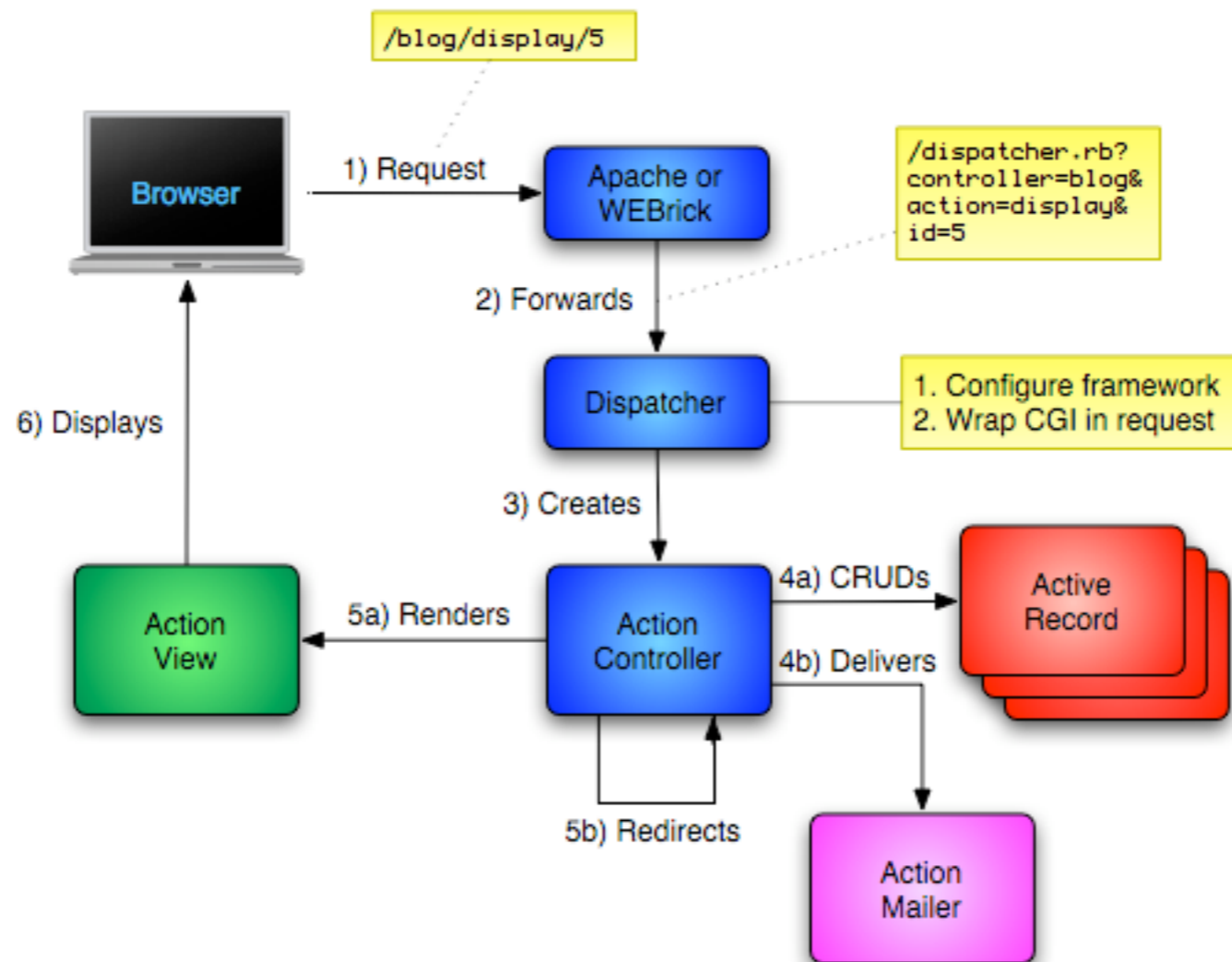
Session Storage

- Serialize to /tmp
- DRb Server
- Database
- memcached
- Custom
- No Sessions

Scaling Up

- Multi-processing model (like prefork)
- DB Connection per FCGI Process
- Remote FCGI instances
- Static and Dynamic Caching
- Easy to interface with C
 - (Almost as easy to interface with OCaml)

The Diagram



Web Resources

- Ruby
 - <http://www.ruby-lang.org/>
- Why's Poignant Guide to Ruby
 - <http://poignantguide.net/>
- Ruby on Rails
 - <http://www.rubyonrails.com/>
- Ruby Documentation
 - <http://www.ruby-doc.org/>

That's all folks!

Brian McCallister

brianm@chariotsolutions.com

<http://www.chariotsolutions.com/>