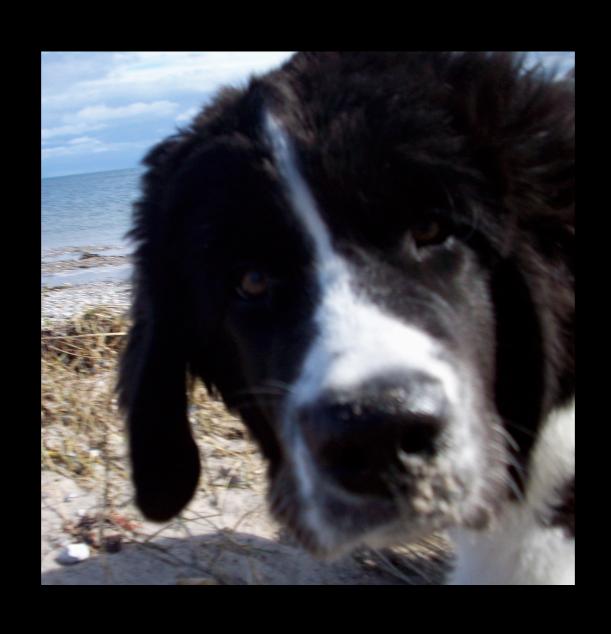# Why Merb?

Kim Dalsgaard

Co-owner of, and Software Designer at Trifork Athene
Co-founder of Aarhus Ruby Brigade

What is Merb?

Web framework for the VC in MVC

ORM agnostic

Merb is thread-safe

No CGI.rb!

A small core, enhanced with plug-ins

Merb is a hacker's framework

Stay small
Stay hackable

Kitchen sink not included

Some Historical Facts

Merb was started in October of 2006 by Ezra Zygmuntowicz

Originally designed as a hack for uploading files for Ruby on Rails

... but has since grown into a full-fledged framework

Merb == Mongrel + Erb

Merb == Rack + Template Language Agnostic

The Merb Philosophy

No code is faster than no code

Simplicity and clarity trumps magic every time

... so prefer simple code over magic code

Don't be clever!

Things I like about Merb

Compiled routes

Merb::Router.
compiled_statement
outputs the compiled
match method

Some controller stuff

Merb renders the return value from each action

- String objects
- IO objects
- Proc objects

```ruby
class Hello < Application

  def index
    "Hello RubyFools!"
  end

end
```

```ruby
class Pid < Application

  def index
    File.open Merb.root +
      "/log/merb.#{request.port}.pid"
  end

end
```

The render method
returns a String

```ruby
class ChineseBox < Application

  def index
    txt = "Chinese Box Set"
    red = render txt, :layout => "red"
    green = render red, :layout => "green"
    blue = render green, :layout => "blue"
    render blue
  end

end
```

Chinese Box Set

The `provides` and `display` methods for rendering objects

The `provides` method for registering mime-types to render

The `display` method for rendering objects

```ruby
class Regions < Application
  provides :yaml, :json

  def index
    @regions = Region.all
    display @regions
  end

  def show
    @region = Region[params[:id]]
    display @region
  end
```

The `display` method returns a String

The render_then_call methods do work after the respond is send

```ruby
class ThenCall < Application

  def index
    content = render
    render_then_call content do
      worker = Worker.new
      worker.work
    end
  end

end
```

Merb is build on top of Rack

Rack provides an minimal interface between web servers and web frameworks.

A Rack application is a Ruby object (not a class) that responds to `call`.

It takes exactly one argument, the environment

... and returns an Array of exactly three values: The status, the headers, and the body.

```
# config/rack.rb

run Merb::Rack::Application.new
```

```ruby
class MyRackHandler

  def initialize(app)
    @app = app
  end

  def call(env)
    @app.call env
  end

end

use MyRackHandler
run Merb::Rack::Application.new
```

```ruby
def call(env)
  request = Merb::Request.new env
  if request.path =~ %r{/api/(.*)}
    [200,
      {"Content-Type" => "text/json"},
      Api.create_json($1)]
  else
    @app.call env
  end
end
```

Supported Handlers

- Mongrel

- evented_mongrel

- Ebb

- Thin

- WEBrick

- and many more

Supported Adapters

- Ruby on Rails

- Camping

- Ramaze

- Merb

- and many more

Merb Structure

- Merb Core

- Merb More

- Merb Plug-ins

Merb Core is the essential parts of Merb

Merb More is a collection of very often used plug-ins

- merb-action-args

- merb-assets

- merb-gen

- merb-haml

- merb-builder

- merb-mailer

- merb-parts

- merb-cache

merb-action-args

```ruby
class Regions < Application
  provides :yaml, :json

  def show(id)
    @region = Region[id]
    display @region
  end

end
```

merb-assets

- require_css

- require_js

- include_required_css

- include_required_js

- and more

Official Plug-ins

- merb_activerecord

- merb_datamapper

- merb_sequel

- merb_helpers

- merb_param_protection

- merb_stories

- merb_test_unit

The dependency and dependencies methods includes dependent gems

They try to load the file via ROOT/gems first before moving off to the system gems

```
# These are some examples of how you might
# specify dependencies.
#
# dependencies "RedCloth", "merb_helpers"
# OR
# dependency "RedCloth", "> 3.0"
# OR
# dependencies "RedCloth" => "> 3.0",
#               "ruby-aes-cext" => "= 1.0"

dependency "merb-parts"
```

Using an ORM

Merb supports three different ORM's through official plug-ins

- Active Record

- DataMapper

- Sequel

The use_orm method is used to select an ORM

```
### Uncomment for DataMapper ORM
# use_orm :datamapper

### Uncomment for ActiveRecord ORM
# use_orm :activerecord

### Uncomment for Sequel ORM
use_orm :sequel
```

Why Sequel?

Sequel provides thread safety, connection pooling

... and a concise DSL for constructing queries and table schemas.

Sequel also includes a lightweight but comprehensive ORM layer for mapping records to Ruby objects using the ActiveRecord pattern.

```ruby
class PlayerMigration < Sequel::Migration

  def up
    create_table :players do
      primary_key :id
      varchar :name
      foreign_key :team_id, :table => :teams
    end
  end

  def down
    drop_table :players
  end

end
```

```ruby
class Player < Sequel::Model
  many_to_one :team
  one_to_many :goals
end
```

```ruby
class Players < Application
  provides :yaml, :json

  def index
    @players = Player.all
    display @players
  end

  def show(id)
    @player = Player[id]
    display @player
  end

end
```

- http://www.merbivore.org/

- http://brainspl.at

- http://github.com/drnic/merb-tmbundle/tree/master

Questions?