

# Applied Web Heresies

Avi Bryant

<http://smallthought.com/avi/heresy>

Seaside:  
a web framework in  
Smalltalk

<http://www.seaside.st>

Seaside:  
a web framework in  
Smalltalk

YOU AIN'T GONNA  
USE IT

# Seaside as a Recipe

Primo Levi's "onions in the varnish"

# Recipes have Structure

- Justification
- Inspiration
- Implementation
- Re-implementation
- Omissions

# Equipment list:

- OOP
- Servlet-style app server
- Blocks/closures

Cooking with discipline:  
Try not to taste it 'til  
it's done





First thing we do, let's  
kill all the templates

The Zen of HTML Generation

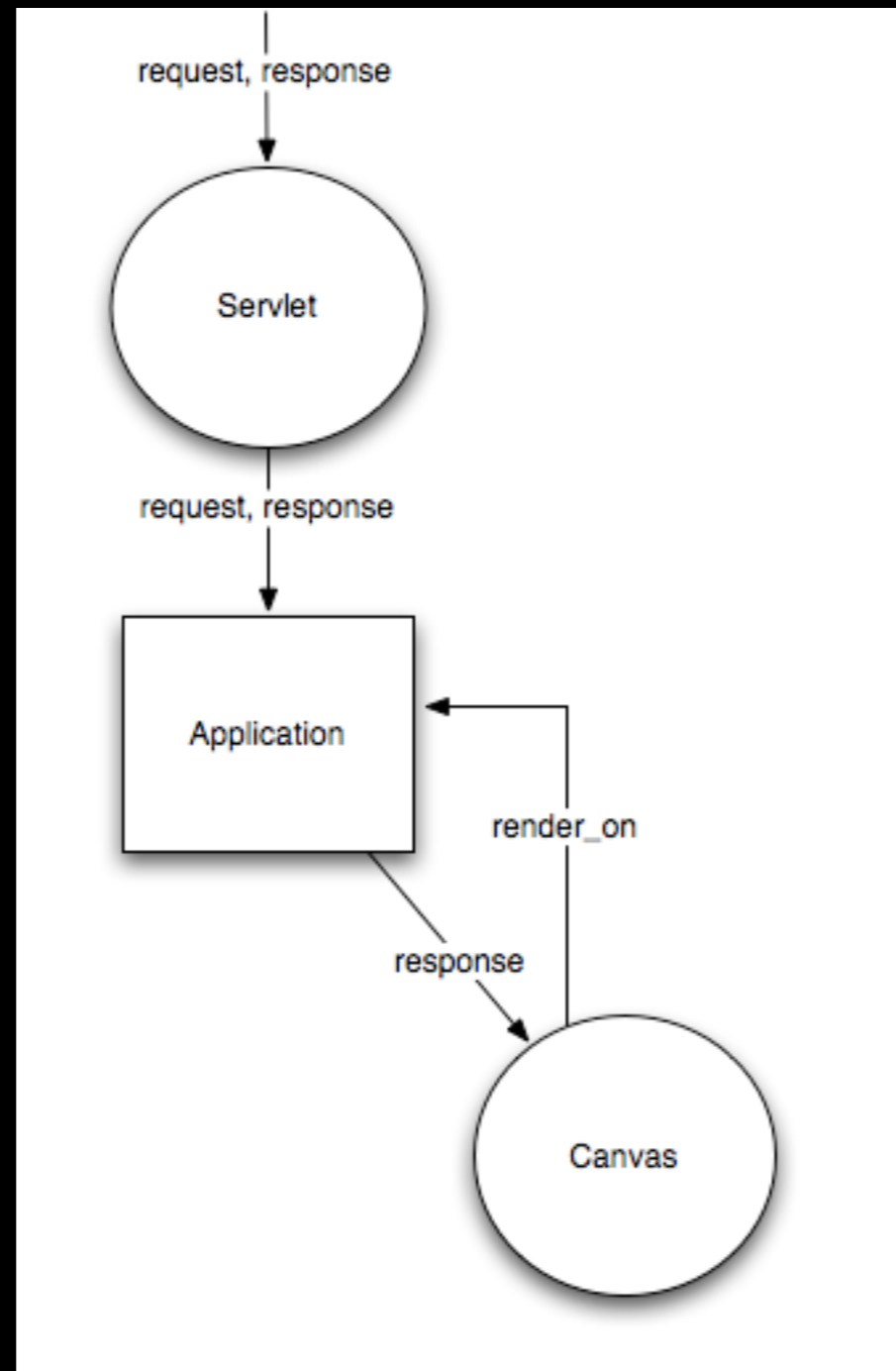
What about model/  
view separation? What  
about web designers?

# CSS

(cf. <http://www.csszengarden.com>)

Inspiration:  
AWT

# Implementation



# Re-implementation

```
require 'webrick'  
require 'stringio'  
  
server = WEBrick::HTTPServer.new(:Port => 2000)  
  
server.mount_proc("/heresy"){|req, res| Application.new.handle(req, res)}  
server.mount_proc("/favicon.ico"){|req, res| res.status = 404}  
trap("INT"){ server.shutdown }  
server.start
```

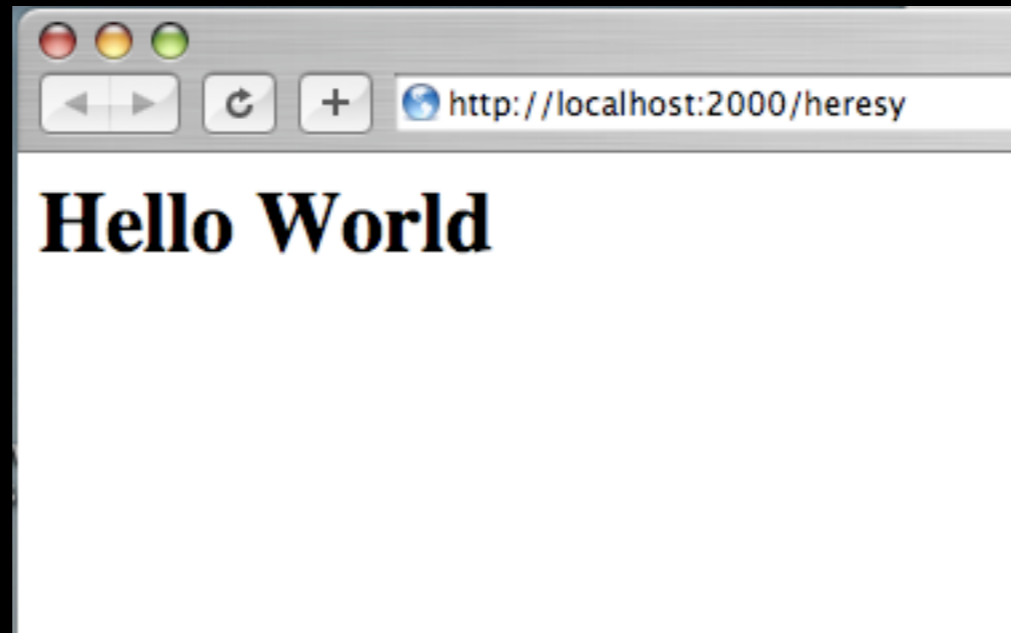
# Re-implementation

```
class Application
  def handle(req, res)
    canvas = Canvas.new(res)
    render_on(canvas)
  end

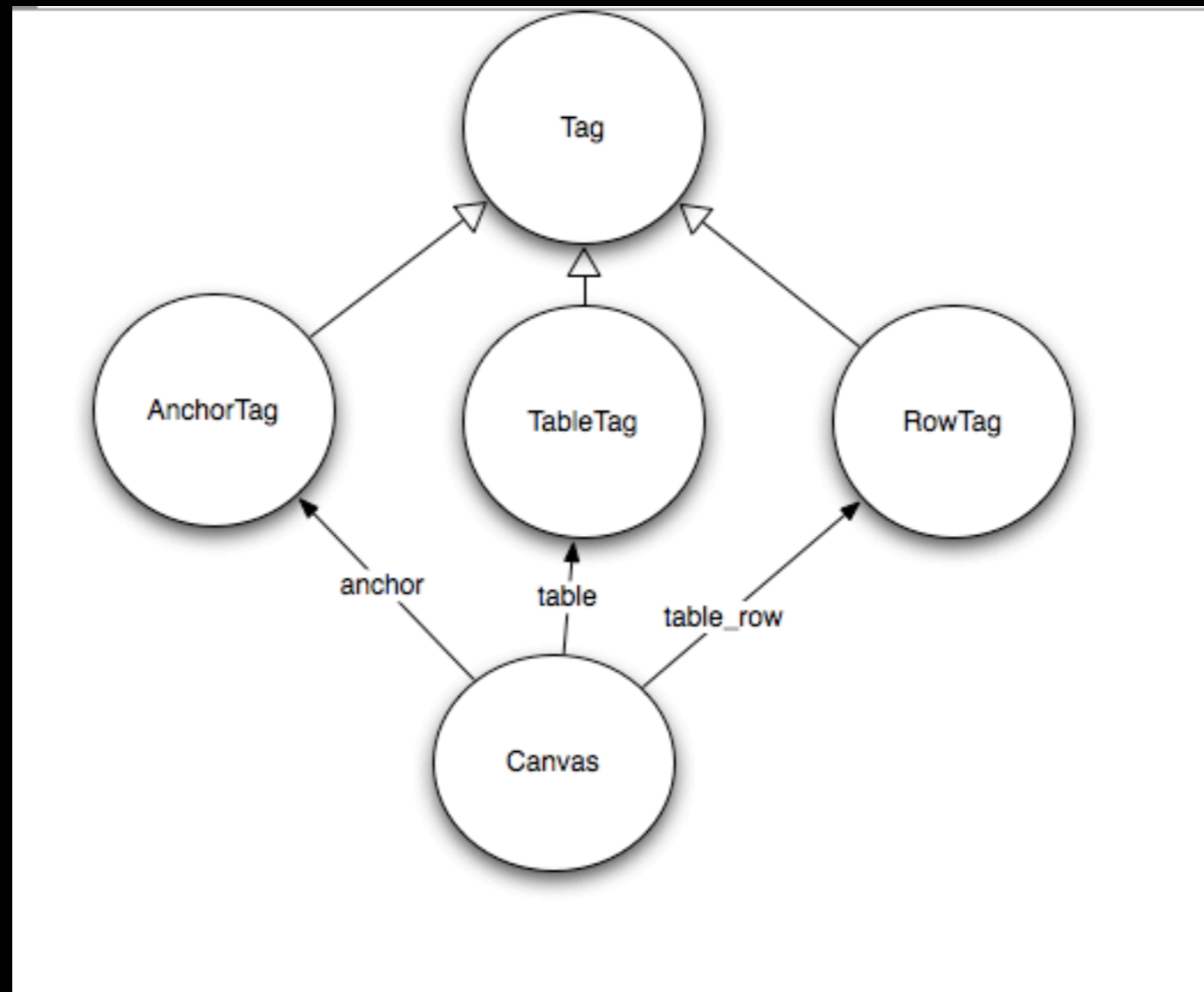
  def render_on(html)
    html.heading("Hello World")
  end
end

class Canvas
  def heading(str, level=1)
    ...
  end
end
```

# Goal



# Omission: Tag Objects



Sessions are too  
valuable to persist

All the good stuff's in memcached anyway

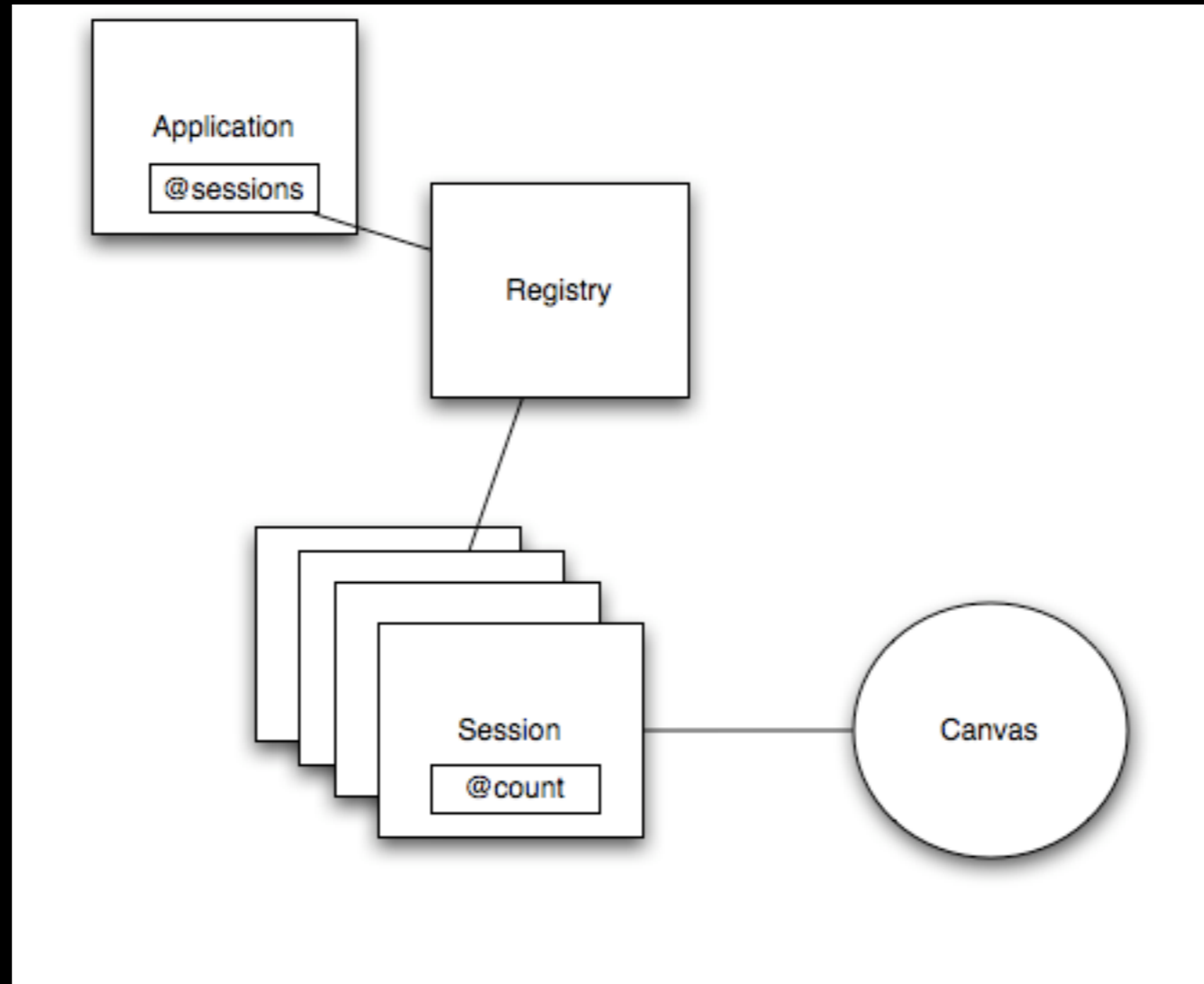
What about load  
balancing and fail-over?

YAGNI

(really)

# Inspiration: NeXT's WebObjects

# Implementation



# Re-implementation

```
class Registry
  def initialize
    @items = []
  end

  def register(item)
    @items << item
    (@items.size - 1).to_s
  end

  def find(key)
    @items[key.to_i]
  end
end
```

# Re-implementation

```
class Session
  def initialize
    @count = 0
  end

  def handle(req, res)
    ...
  end
end
```

# Goal



# Omissions

- Unguessable session keys
- Session keys as query params
- Session locks
- Expiration from registry

Meaningful URLs don't  
carry enough meaning

Names make me think too much

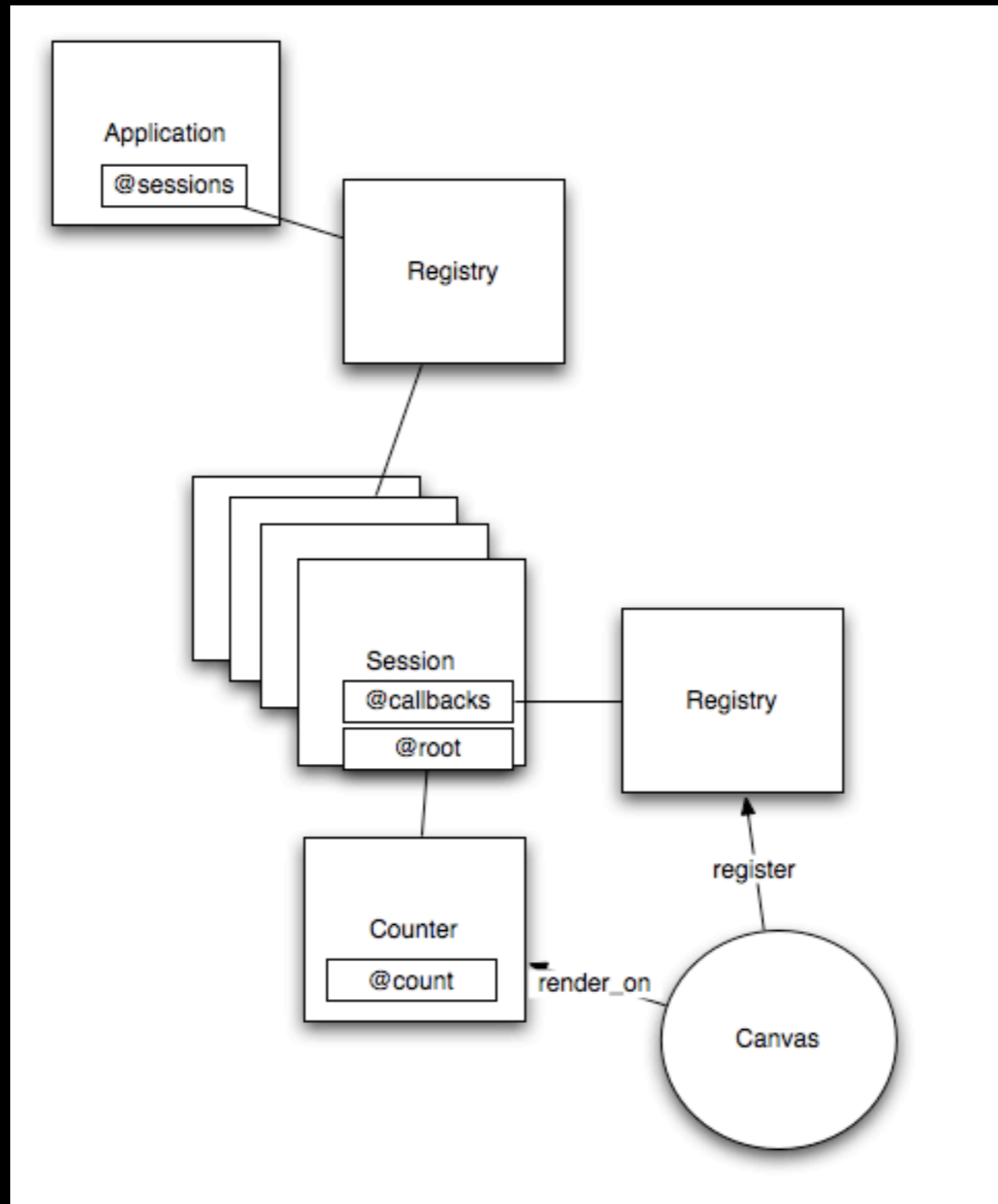
What acronym do we  
get this time?

# DRY

(not everything is an API)

Inspiration:  
TCL/TK

# Implementation



# Re-implementation

```
class Session
  def initialize
    @callbacks = Registry.new
    @root = Counter.new
  end

  ....

end
```

# Re-implementation

```
class Counter
  def initialize
    @count = 0
  end

  def render_on(html)
    html.heading("Hello World: #{@count}")
    html.link("--"){@count -= 1}
    html.space
    html.link("++"){@count += 1}
  end
end
```

# Goal



# Omissions

- Splitting callback registries up by page view
- Tracking the back-button
- Redirecting after side-effects
- Forms!

Pages are a lousy unit  
of reuse

And partials ain't much better

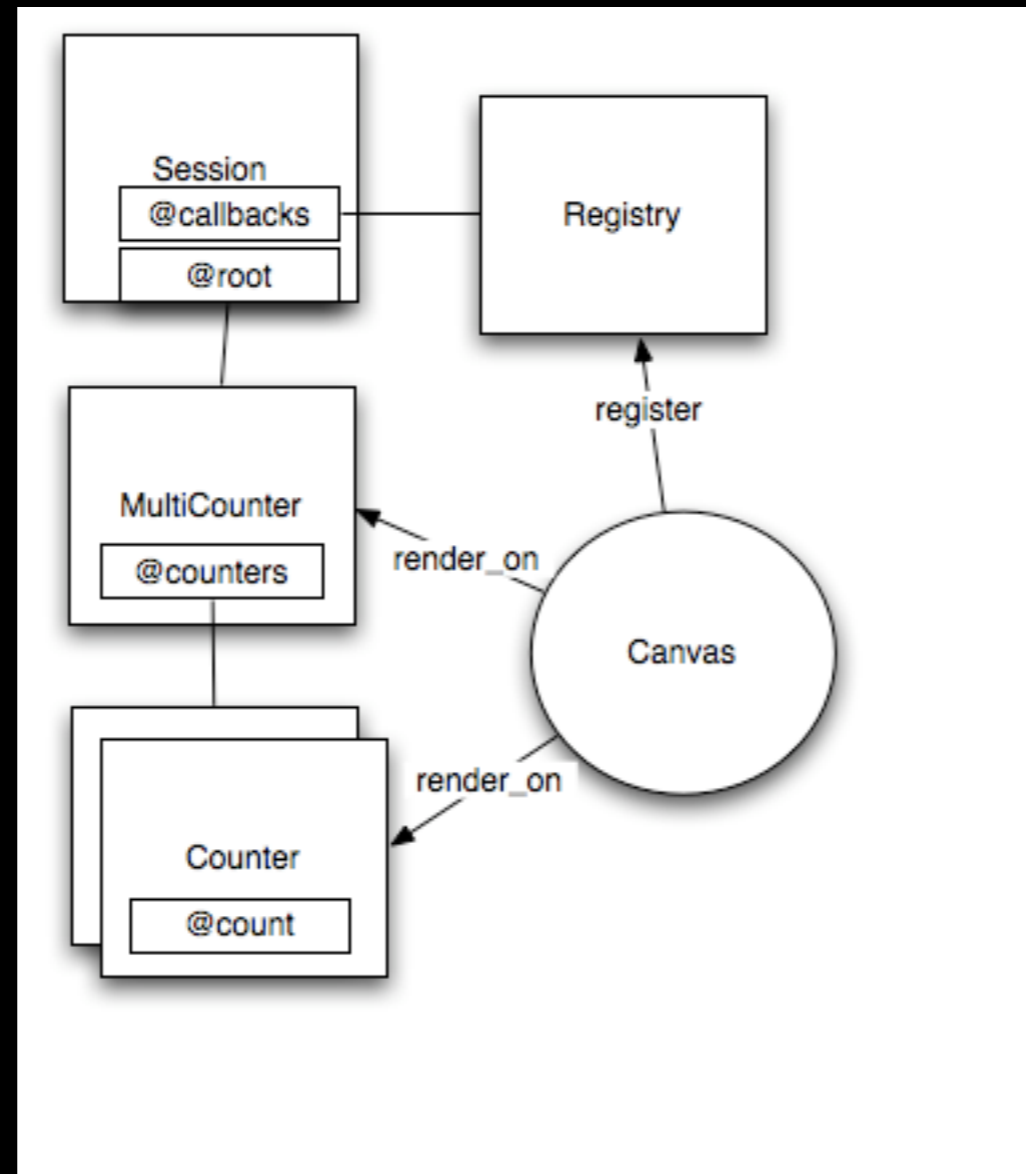
But every piece of my application is a beautiful and unique snowflake.

# CSS

(cf. <http://css.maxdesign.com.au/listamatic/>)

Inspiration:  
Can I say WebObjects  
again?

# Implementation



# Re-implementation

```
class Session
  def initialize
    @root = MultiCounter.new
    .....
end

class Canvas
  def render(obj)
    obj.render_on(self)
  end
end
```

# Re-implementation

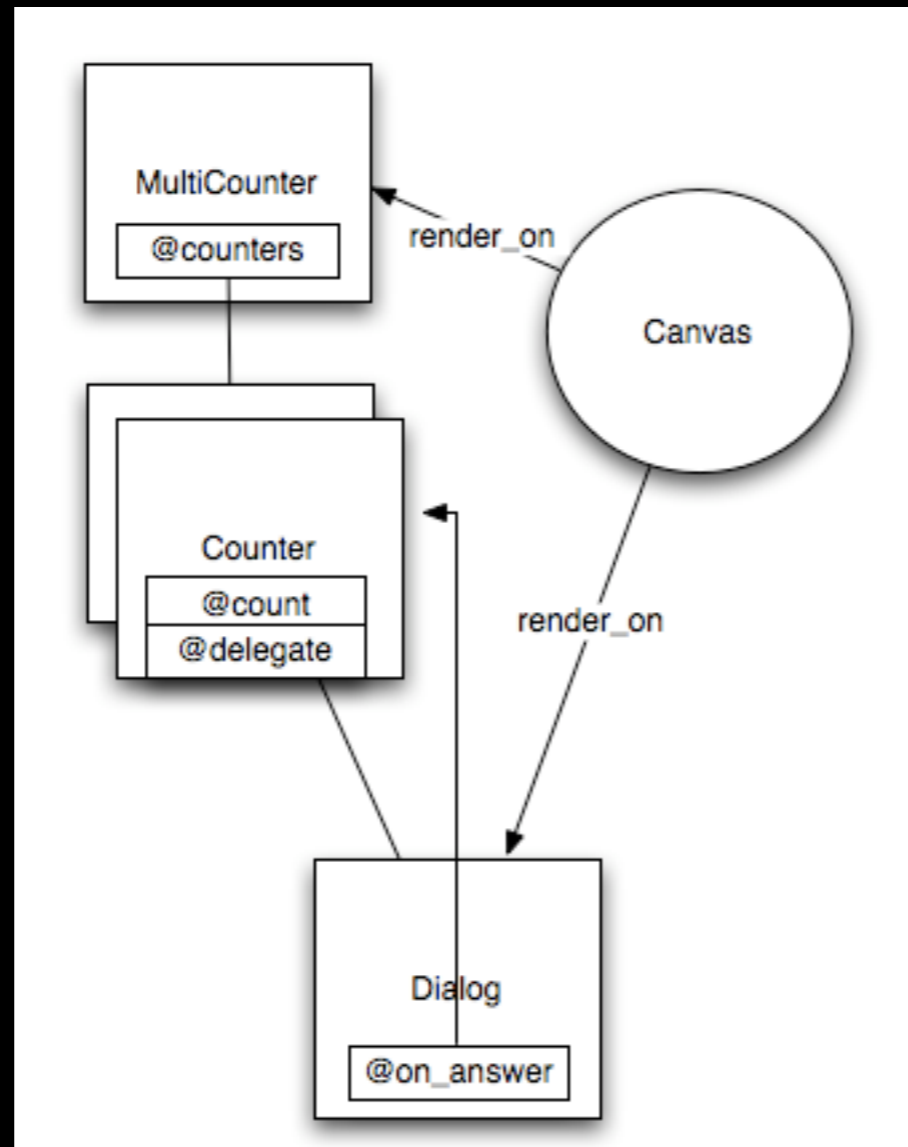
```
class MultiCounter
  def initialize
    @counters = [Counter.new, Counter.new, Counter.new]
  end

  def render_on(html)
    @counters.each{|ea| html.render(html)}
  end
end
```

# Goal



# Omission: Call/Answer



**Extra Credit: Halos**

# Inspiration: Squeak, Tapestry

# Implementation

```
class Session
  def render_on(html)
    html.render(@root)
    html.p
    if @canvas_class == DebugCanvas
      html.link("Stop Debugging"){@canvas_class = Canvas}
    else
      html.link("Start Debugging"){@canvas_class = DebugCanvas}
    end
  end
end
end
```

# Goal



The screenshot shows a web browser window with the address bar displaying `http://localhost:2000/heresy?76`. The main content area contains three stacked panels, each representing a counter object. Each panel has a title bar with a unique ID and a count, followed by a large bold text display and a pair of blue links labeled `--` and `++`.

Counter ID	Count	Display
#<Counter:0x227dbc @count=-4>	-4	Hello World: -4
#<Counter:0x227da8 @count=4>	4	Hello World: 4
#<Counter:0x227d94 @count=2>	2	Hello World: 2

At the bottom of the browser window, there is a blue link labeled [Stop Debugging](#).