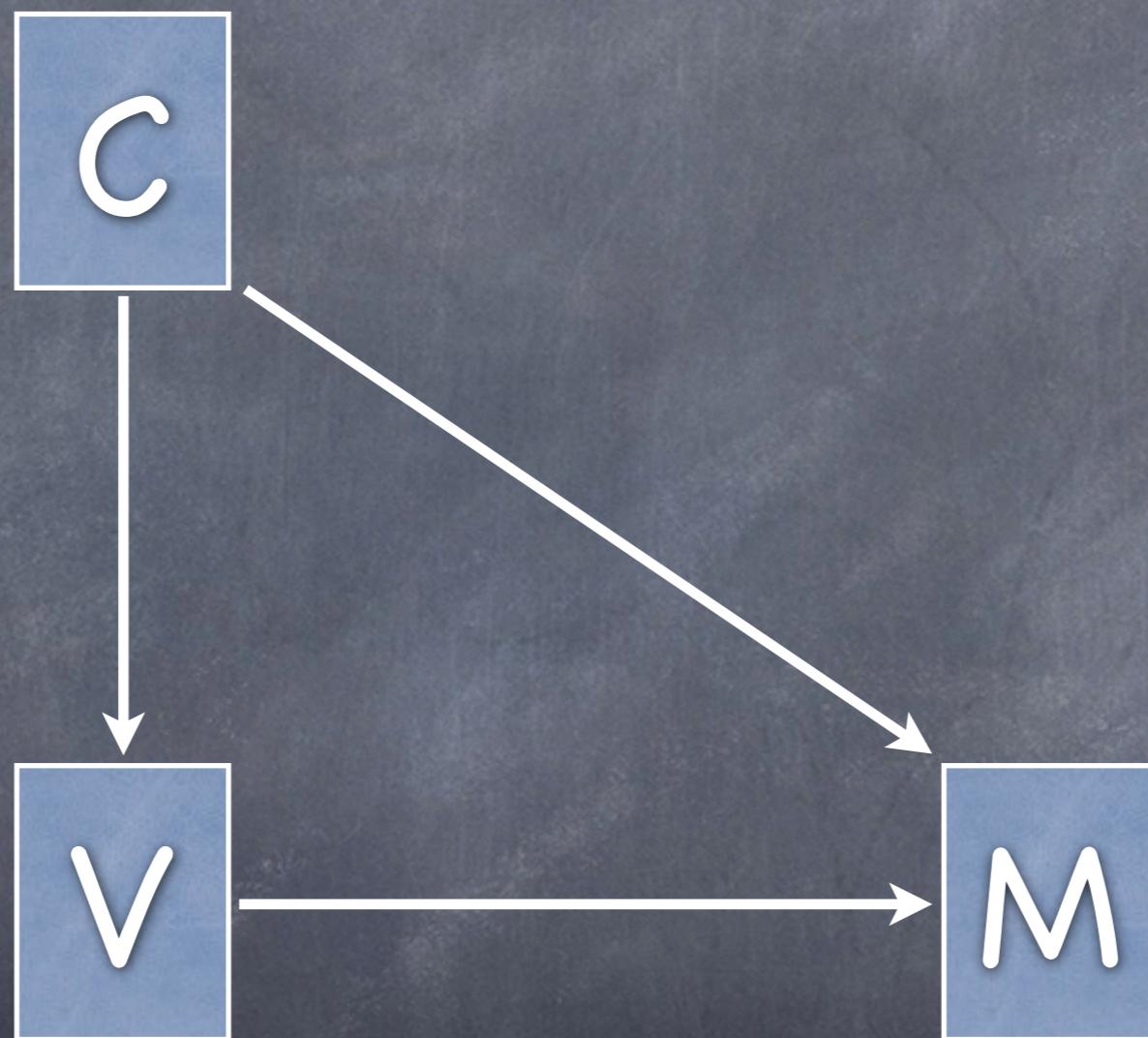


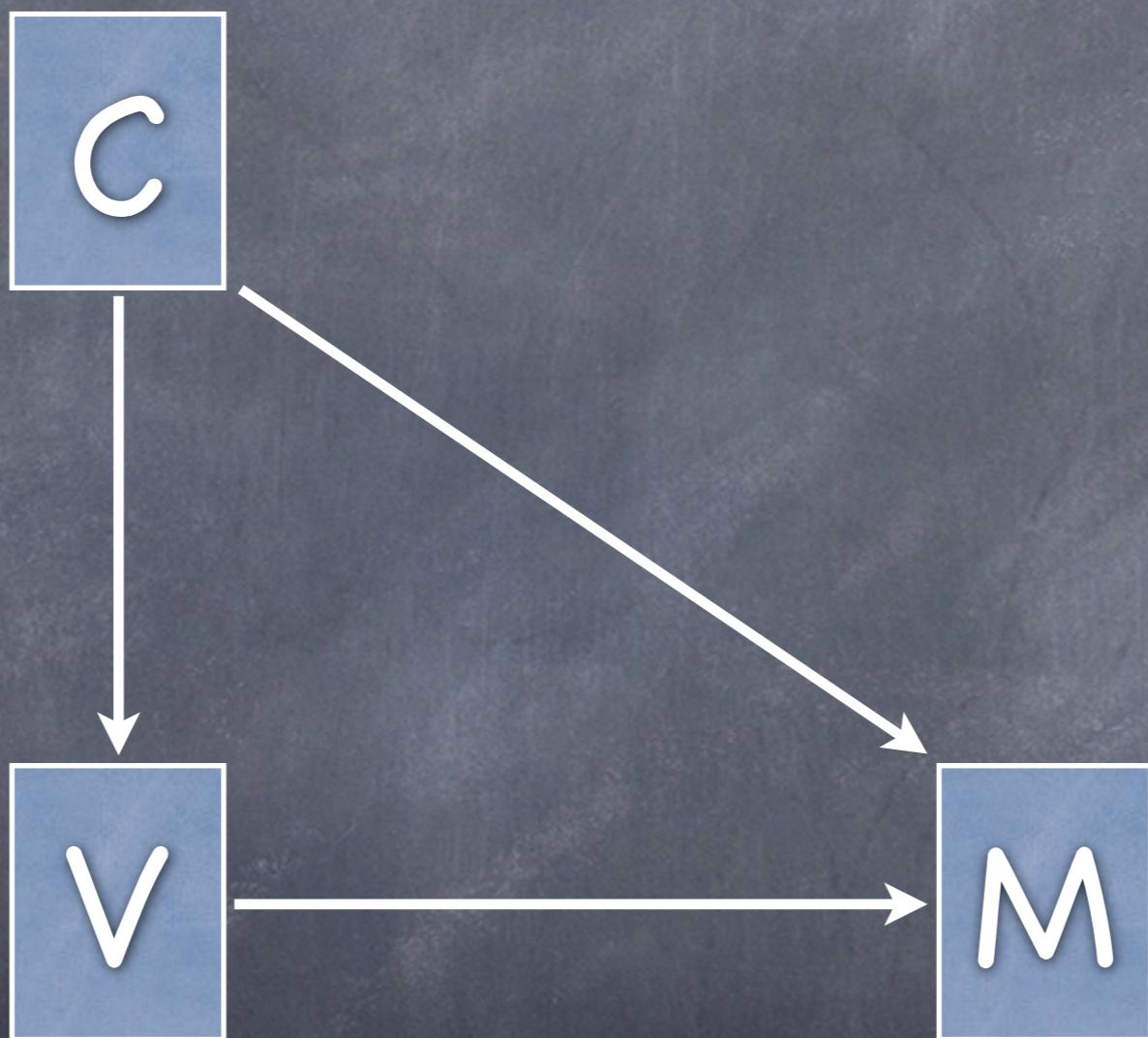
Refactoring III: Views

Brian Hughes
@brianvhughes

MVC in Rails

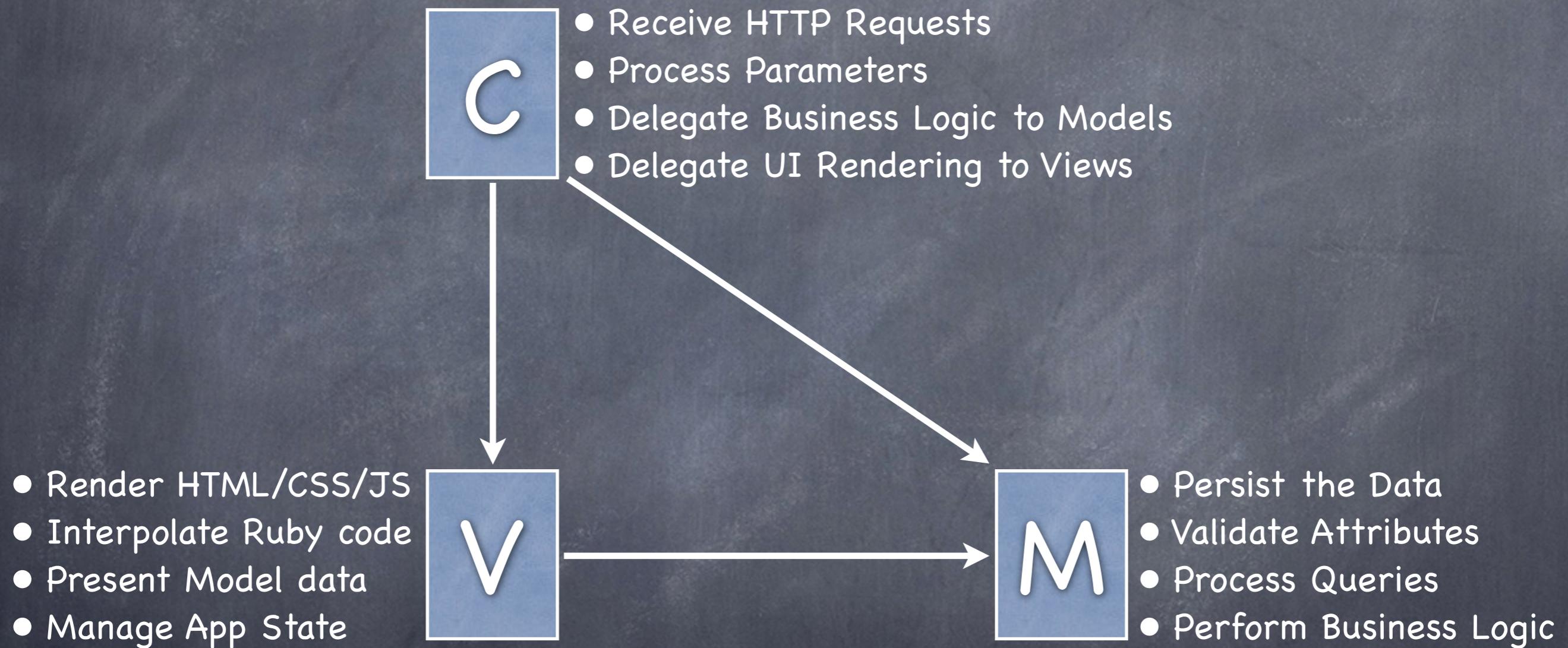


MVC in Rails



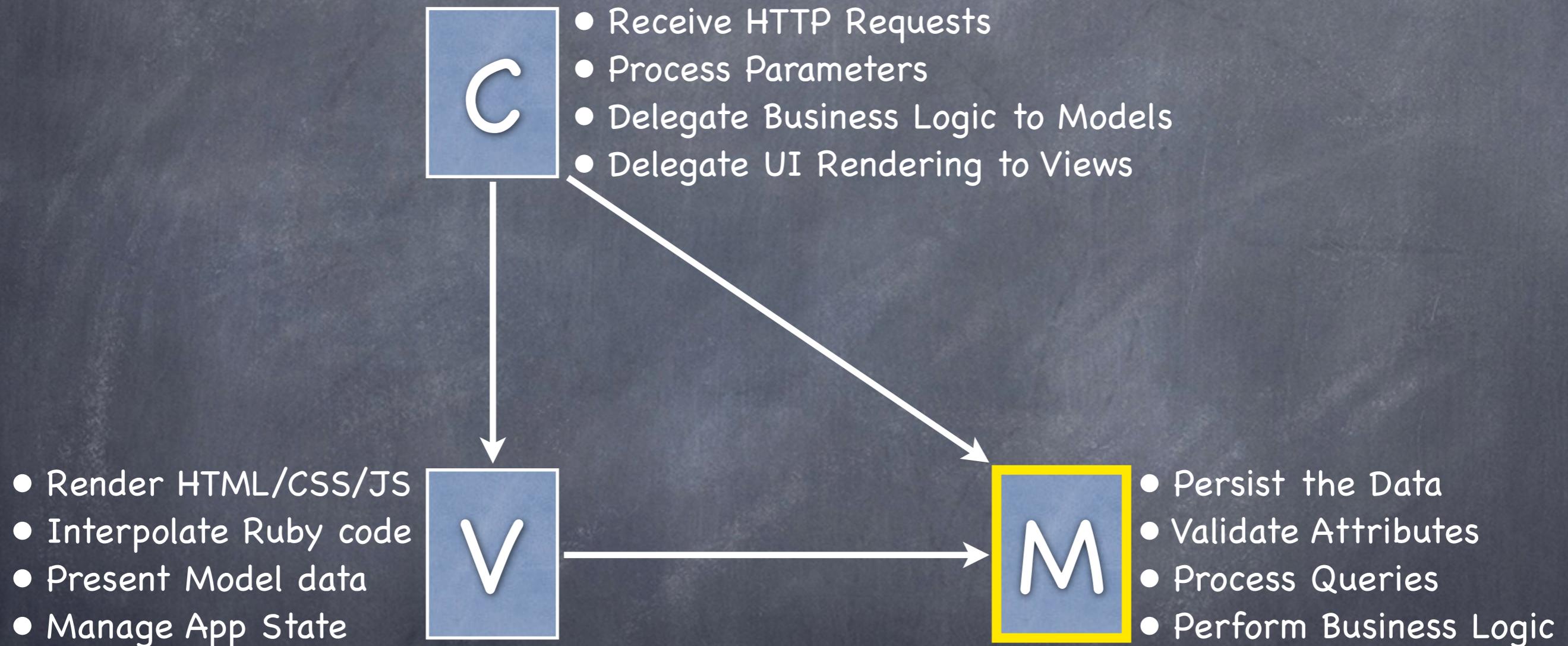
Has a Problem!

MVC in Rails



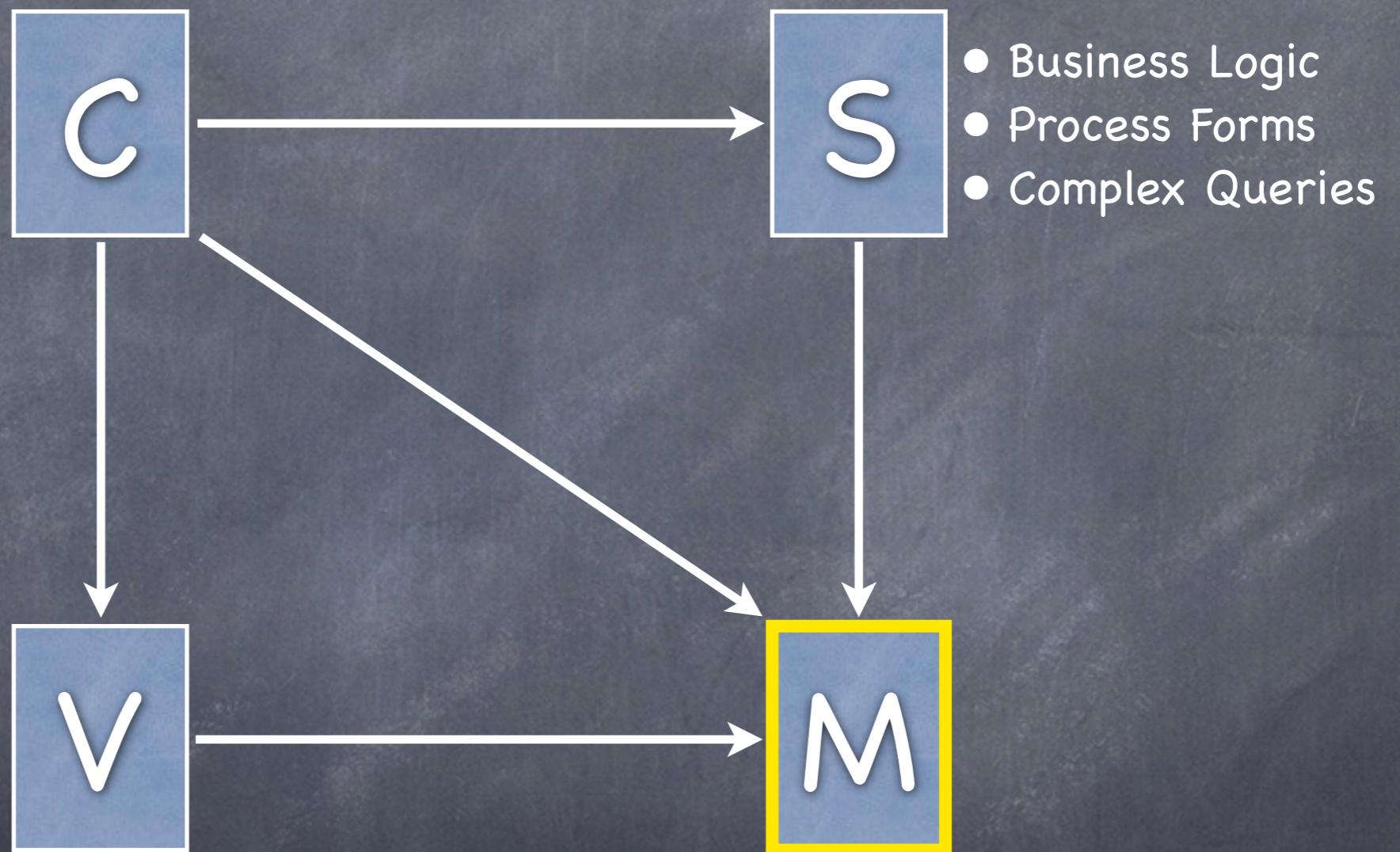
It's Anti-SRP!

MVC in Rails



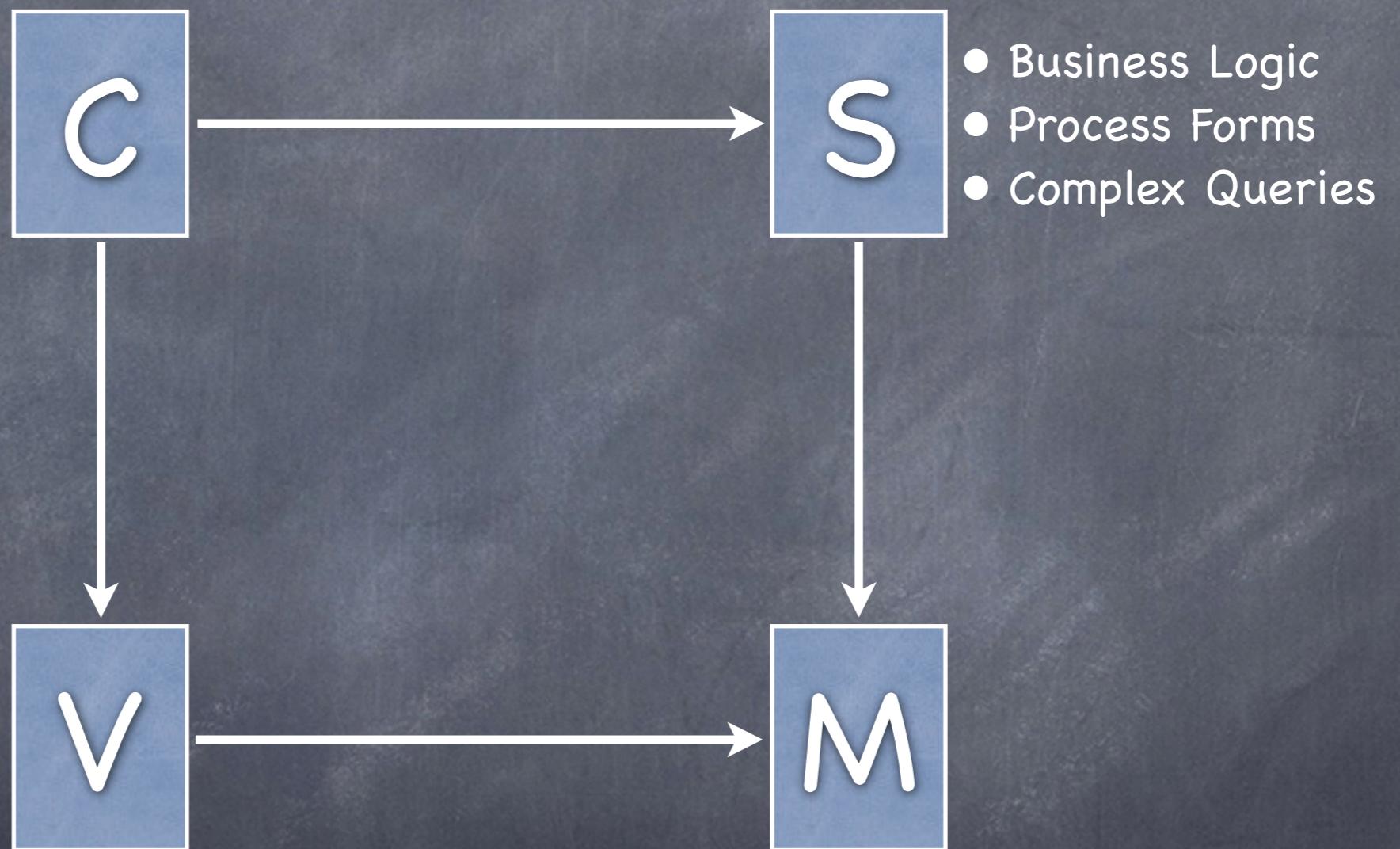
Skinny Controllers,
Fat Models

MVC in Rails



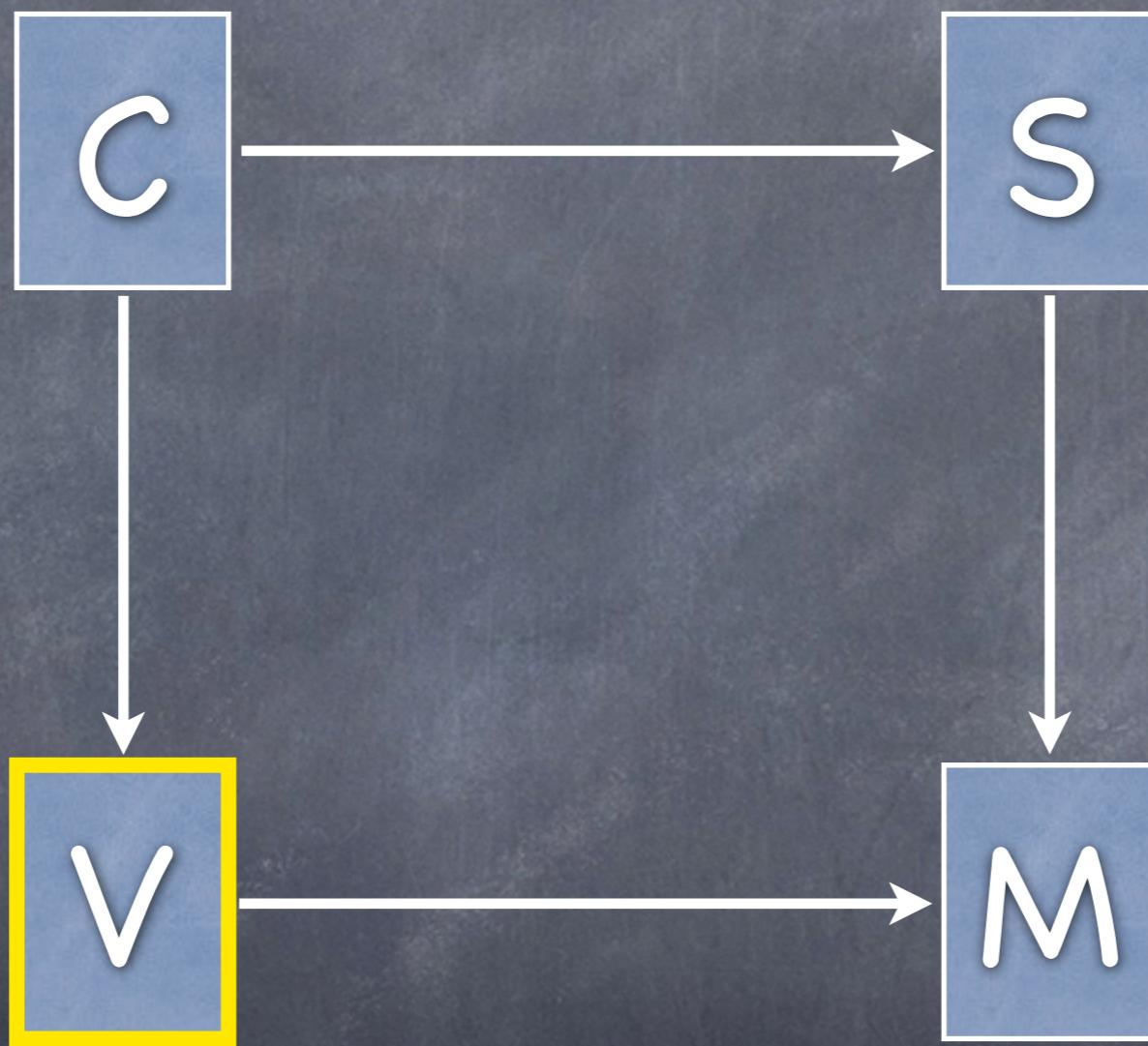
Service Classes can help
make our models thin.

MVC in Rails



Service Classes can help
make our models thin.

MVC in Rails



How do we deal with Views,
when they get fat?



Dealing with Fat Views

- ⦿ The Rails Way: Helpers
 - ⦿ Methods in modules. Not regular classes
 - ⦿ Can have name collisions
 - ⦿ Procedural in nature
 - ⦿ Difficult to test

We Don't Need that kind of Help



Objects, not Helpers

- ⦿ Standard classes, with standard methods
- ⦿ Knows the state of our Model objects
- ⦿ Provides presentation-specific data, when needed
- ⦿ Knows how to render sub-templates
- ⦿ Can be tested, just like our other classes



I'll bet he's talking about Design Patterns!

- ⦿ Decorator Pattern
 - ⦿ One of the original GoF patterns
- ⦿ Exhibit Pattern
 - ⦿ Objects on Rails, Avdi Grimm
- ⦿ Presenter Pattern
 - ⦿ Introduced by Jay Fields, 2007



How do I use these in my Rails app?

- ⦿ Roll your own Decorator
 - ⦿ Using Ruby stdlib's SimpleDelegator
- ⦿ Display Case - Exhibits
 - ⦿ github.com/objects-on-rails/display-case
- ⦿ Draper
 - ⦿ github.com/drapergem/draper



What is Draper?

- ⦿ The most popular gem for View refactoring
- ⦿ A bit confused about design patterns
- ⦿ Almost a pure implementation of Presenter
- ⦿ Calls itself a View-Model
- ⦿ Stores classes in /app/decorators



When should we refactor our Views?

- ⦿ When we find Code Smells!
 - ⦿ Use of conditionals: if, else, unless, case
 - ⦿ Performing an ActiveRecord query
 - ⦿ Processing Arrays/Hashes: each, map
 - ⦿ Formatting model data for presentation
 - ⦿ Improper use of partial templates
 - ⦿ Instantiating local variables



Example View Template

```
%center-
  %h1 Meetup for #{l @meeting.date, format: :long}-
  -
  - if @meeting.kudos_available?(Time.current, current_user)-
    %br-
    .alert.alert-info-
      %center-
        %h4 To give a Kudo, click on the presentation you liked best.-.
    .kudos-
      %center-
        - @meeting.time_slots.each do |slot|-
          = link_to(slot.topic.title, '#', class: 'btn btn-large btn-success btn-kudo',
data: {url: give_kudo_topic_path(slot.topic)})-
        -
        = render 'points_awarded' if @meeting.closed?-
        -
        %h3 Topics Presented-
        = render(TimeSlotDecorator.decorate(@meeting.time_slots))-
        -
        = current_user.edit_meeting_link(@meeting)-
        = current_user.finalize_meeting_link(@meeting)-
```

Contains at least 6 code smells!



Code Smell #1

```
%center~  
  %h1 Meetup for #{l @meeting.date, format: :long}~  
~  
- if @meeting.kudos_available?(Time.current, current_user)~  
%br~  
.alert.alert-info~  
%center~  
  %h4 To give a Kudo, click on the presentation you liked best.~  
.kudos~  
%center~  
  - @meeting.time_slots.each do |slot|~  
    = link_to(slot.topic.title, '#', class: 'btn btn-large btn-success btn-kudo',  
data: {url: give_kudo_topic_path(slot.topic)})~  
~  
= render 'points_awarded' if @meeting.closed?~  
~  
%h3 Topics Presented~  
= render(TimeSlotDecorator.decorate(@meeting.time_slots))~  
~  
= current_user.edit_meeting_link(@meeting)~  
= current_user.finalize_meeting_link(@meeting)~
```

Formatting the meeting date for presentation



Code Smell #2

```
%center~  
  %h1 Meetup for #{l @meeting.date, format: :long}~  
  ~  
  - if @meeting.kudos_available?(Time.current, current_user)  
    %br~  
    .alert.alert-info~  
    %center~  
      %h4 To give a Kudo, click on the presentation you liked best.~  
    .kudos~  
    %center~  
      - @meeting.time_slots.each do |slot|~  
        = link_to(slot.topic.title, '#', class: 'btn btn-large btn-success btn-kudo',  
data: {url: give_kudo_topic_path(slot.topic)})~  
      ~  
      = render 'points_awarded' if @meeting.closed?~  
      ~  
    %h3 Topics Presented~  
    = render(TimeSlotDecorator.decorate(@meeting.time_slots))~  
    ~  
    = current_user.edit_meeting_link(@meeting)~  
    = current_user.finalize_meeting_link(@meeting)~
```

Conditional check to render more content



Code Smell #3

```
%center-
  %h1 Meetup for #{l @meeting.date, format: :long}-
  -
  - if @meeting.kudos_available?(Time.current, current_user)-
    %br-
    .alert.alert-info-
      %center-
        %h4 To give a Kudo, click on the presentation you liked best.-.
      .kudos-
        %center-
          - @meeting.time_slots.each do |slot|-
            = link_to(slot.topic.title, '#', class: 'btn btn-large btn-success btn-kudo',
data: {url: give_kudo_topic_path(slot.topic)})-
        -
        = render 'points_awarded' if @meeting.closed?-
        -
%h3 Topics Presented-
= render(TimeSlotDecorator.decorate(@meeting.time_slots))-
-
= current_user.edit_meeting_link(@meeting)-
= current_user.finalize_meeting_link(@meeting)-
```

Conditional content should move to a partial



Code Smell #4

```
%center-
  %h1 Meetup for #{l @meeting.date, format: :long}-
  -
  - if @meeting.kudos_available?(Time.current, current_user)-
    %br-
    .alert.alert-info-
      %center-
        %h4 To give a Kudo, click on the presentation you liked best.-.
      .kudos-
        %center-
          - @meeting.time_slots.each do |slot|-
            = link_to(slot.topic.title, '#', class: 'btn btn-large btn-success btn-kudo',
data: {url: give_kudo_topic_path(slot.topic)})-
  -
  = render 'points_awarded' if @meeting.closed?-
  -
%h3 Topics Presented-
= render(TimeSlotDecorator.decorate(@meeting.time_slots))-
  -
= current_user.edit_meeting_link(@meeting)-
= current_user.finalize_meeting_link(@meeting)-
```

Iterating over array should call to render



Code Smell #5

```
%center-
  %h1 Meetup for #{l @meeting.date, format: :long}-
  -
  - if @meeting.kudos_available?(Time.current, current_user)-
    %br-
    .alert.alert-info-
      %center-
        %h4 To give a Kudo, click on the presentation you liked best.-.
    .kudos-
      %center-
        - @meeting.time_slots.each do |slot|-
          = link_to(slot.topic.title, '#', class: 'btn btn-large btn-success btn-kudo',
data: {url: give_kudo_topic_path(slot.topic)})-
        -
        = render 'points_awarded' if @meeting.closed?-
  -
  %h3 Topics Presented-
  = render(TimeSlotDecorator.decorate(@meeting.time_slots))-
  -
  = current_user.edit_meeting_link(@meeting)-
  = current_user.finalize_meeting_link(@meeting)-
```

Conditional check to render more content



Code Smell #6

```
%center-
  %h1 Meetup for #{l @meeting.date, format: :long}-
  -
  - if @meeting.kudos_available?(Time.current, current_user)-
    %br-
    .alert.alert-info-
      %center-
        %h4 To give a Kudo, click on the presentation you liked best.-.
    .kudos-
      %center-
        - @meeting.time_slots.each do |slot|-
          = link_to(slot.topic.title, '#', class: 'btn btn-large btn-success btn-kudo',
data: {url: give_kudo_topic_path(slot.topic)})-
        -
        = render 'points_awarded' if @meeting.closed?-
        -
%h3 Topics Presented-
= render(TimeSlotDecorator.decorate(@meeting.time_slots))-
-
= current_user.edit_meeting_link(@meeting)-
= current_user.finalize_meeting_link(@meeting)-
```

Formatting time_slots for presentation



Code Smell #7-ish

```
%center-
  %h1 Meetup for #{l @meeting.date, format: :long}-
  -
  - if @meeting.kudos_available?(Time.current, current_user)-
    %br-
    .alert.alert-info-
      %center-
        %h4 To give a Kudo, click on the presentation you liked best.-.
    .kudos-
      %center-
        - @meeting.time_slots.each do |slot|-
          = link_to(slot.topic.title, '#', class: 'btn btn-large btn-success btn-kudo',
data: {url: give_kudo_topic_path(slot.topic)})-
        -
        = render 'points_awarded' if @meeting.closed?-
        -
%h3 Topics Presented-
= render(TimeSlotDecorator.decorate(@meeting.time_slots))-
  -
  = current_user.edit_meeting_link(@meeting)-
  = current_user.finalize_meeting_link(@meeting)-
```

Delegating to object other than @meeting



Draper to the Rescue!

```
class MeetingDecorator < Draper::Base
  decorates :meeting
end
```

/app/decorators/meeting_decorator.rb



Draper to the Rescue!

```
class MeetingDecorator < Draper::Base
  decorates :meeting
  decorates_association :time_slots

  def meetup_date
    I18n.localize(date, format: :long)
  end

  def display_kudo_vote_links(current_user)
    return '' unless kudos_available?(Time.current, current_user)
    h.render partial: 'meetings/kudo_votes', locals: {meeting: self}
  end

  def display_points_awarded
    return '' unless closed?
    h.render partial: 'meetings/points_awarded', locals: {meeting: self}
  end
end
```

/app/decorators/meeting_decorator.rb



Draper to the Rescue!

```
class MeetingDecorator < Draper::Base
  decorates :meeting
  decorates_association :time_slots

  def meetup_date
    I18n.localize(date, format: :long)
  end

  def display_kudo_vote_links(current_user)
    return '' unless kudos_available?(Time.current, current_user)
    h.render partial: 'meetings/kudo_votes', locals: {meeting: self}
  end

  def display_points_awarded
    return '' unless closed?
    h.render partial: 'meetings/points_awarded', locals: {meeting: self}
  end
end
```

Addresses Code Smell #1



Draper to the Rescue!

```
class MeetingDecorator < Draper::Base
  decorates :meeting
  decorates_association :time_slots

  def meetup_date
    I18n.localize(date, format: :long)
  end

  def display_kudo_vote_links(current_user)
    return '' unless kudos_available?(Time.current, current_user)
    h.render partial: 'meetings/kudo_votes', locals: {meeting: self}
  end

  def display_points_awarded
    return '' unless closed?
    h.render partial: 'meetings/points_awarded', locals: {meeting: self}
  end
end
```

Addresses Code Smell #2



Draper to the Rescue!

```
class MeetingDecorator < Draper::Base
  decorates :meeting
  decorates_association :time_slots

  def meetup_date
    I18n.localize(date, format: :long)
  end

  def display_kudo_vote_links(current_user)
    return '' unless kudos_available?(Time.current, current_user)
    h.render partial: 'meetings/kudo_votes', locals: {meeting: self}
  end

  def display_points_awarded
    return '' unless closed?
    h.render partial: 'meetings/points_awarded', locals: {meeting: self}
  end
end
```

Addresses Code Smells #3 & 4



Refactoring w/Partials

```
- if @meeting.kudos_available?(Time.current, current_user)-
  %br-
  .alert.alert-info-
    %center-
      %h4 To give a Kudo, click on the presentation you liked best.-.
  .kudos-
    %center-
      - @meeting.time_slots.each do |slot|-
        = link_to(slot.topic.title, '#', class: 'btn btn-large btn-success btn-kudo',
          data: {url: give_kudo_topic_path(slot.topic)})-
```



Refactoring w/Partials

```
- if @meeting.kudos_available?(Time.current, current_user)
  = render partial: 'meetings/kudo_votes', locals: {meeting: @meeting}
```

Replace HAML block with partial call

```
%br
.alert.alert-info
%center
  %h4 To give a Kudo, click on the presentation you liked best.
.kudos
%center
- @meeting.time_slots.each do |slot|
  = link_to(slot.topic.title, '#', class: 'btn btn-large btn-success btn-kudo',
  data: {url: give_kudo_topic_path(slot.topic)})
```

app/views/meetings/_kudo_votes.html.haml



Refactoring w/Partials

```
- if @meeting.kudos_available?(Time.current, current_user)
  = render partial: 'meetings/kudo_votes', locals: {meeting: @meeting}
```

Replace HAML block with partial call

```
%br
.alert.alert-info
%center
  %h4 To give a Kudo, click on the presentation you liked best.
.kudos
%center
  = render 'kudo_link', collection: meeting.topics
```

app/views/meetings/_kudo_votes.html.haml

```
= link_to(kudo_link.title, '#', class: 'btn btn-large btn-success btn-kudo', data:
{url: give_kudo_topic_path(kudo_link)})
```

app/views/meetings/_kudo_link.html.haml



Draper to the Rescue!

```
class MeetingDecorator < Draper::Base
  decorates :meeting
  decorates_association :time_slots

  def meetup_date
    I18n.localize(date, format: :long)
  end

  def display_kudo_vote_links(current_user)
    return '' unless kudos_available?(Time.current, current_user)
    h.render partial: 'meetings/kudo_votes', locals: {meeting: self}
  end

  def display_points_awarded
    return '' unless closed?
    h.render partial: 'meetings/points_awarded', locals: {meeting: self}
  end
end
```

Addresses Code Smells #3 & 4



Draper to the Rescue!

```
class MeetingDecorator < Draper::Base
  decorates :meeting
  decorates_association :time_slots

  def meetup_date
    I18n.localize(date, format: :long)
  end

  def display_kudo_vote_links(current_user)
    return '' unless kudos_available?(Time.current, current_user)
    h.render partial: 'meetings/kudo_votes', locals: {meeting: self}
  end

  def display_points_awarded
    return '' unless closed?
    h.render partial: 'meetings/points_awarded', locals: {meeting: self}
  end
end
```

Addresses Code Smell #5



Draper to the Rescue!

```
class MeetingDecorator < Draper::Base
  decorates :meeting
  decorates_association :time_slots

  def meetup_date
    I18n.localize(date, format: :long)
  end

  def display_kudo_vote_links(current_user)
    return '' unless kudos_available?(Time.current, current_user)
    h.render partial: 'meetings/kudo_votes', locals: {meeting: self}
  end

  def display_points_awarded
    return '' unless closed?
    h.render partial: 'meetings/points_awarded', locals: {meeting: self}
  end
end
```

Addresses Code Smell #6



Example View Template

```
%center-
  %h1 Meetup for #{l @meeting.date, format: :long}-
  -
  - if @meeting.kudos_available?(Time.current, current_user)-
    %br-
    .alert.alert-info-
      %center-
        %h4 To give a Kudo, click on the presentation you liked best.-.
    .kudos-
      %center-
        - @meeting.time_slots.each do |slot|-
          = link_to(slot.topic.title, '#', class: 'btn btn-large btn-success btn-kudo',
data: {url: give_kudo_topic_path(slot.topic)})-
        -
        = render 'points_awarded' if @meeting.closed?-
        -
        %h3 Topics Presented-
        = render(TimeSlotDecorator.decorate(@meeting.time_slots))-
        -
        = current_user.edit_meeting_link(@meeting)-
        = current_user.finalize_meeting_link(@meeting)-
```

Before Applying Draper



Example View Template

```
%center
  %h1 Meetup for #{@meeting.meetup_date}

  = @meeting.display_kudo_links(current_user)
  = @meeting.display_points_awarded

  %h3 Topics Presented
  = render @meeting.time_slots

  = current_user.edit_meeting_link(@meeting)
  = current_user.finalize_meeting_link(@meeting)
```

After Applying Draper



What about Smell #7?

```
%center
  %h1 Meetup for #{@meeting.meetup_date}
  = @meeting.display_kudo_links(current_user)
  = @meeting.display_points_awarded

  %h3 Topics Presented
  = render @meeting.time_slots

  = current_user.edit_meeting_link(@meeting)
  = current_user.finalize_meeting_link(@meeting)
```

@meeting should display these links



Extend our Decorator

```
class MeetingDecorator < Draper::Base
  decorates :meeting
  decorates_association :time_slots

  def meetup_date
  end

  def display_kudo_vote_links(current_user)
  end

  def display_points_awarded
  end

  def edit_link(current_user)
    current_user.edit_meeting_link(self)
  end

  def finalize_link(current_user)
    current_user.finalize_meeting_link(self)
  end
end
```

/app/decorators/meeting_decorator.rb



Extend our Decorator

```
class MeetingDecorator < Draper::Base
  decorates :meeting
  decorates_association :time_slots

  def meetup_date
  end

  def display_kudo_vote_links(current_user)
  end

  def display_points_awarded
  end

  def edit_link(current_user)
    current_user.edit_meeting_link(self)
  end

  def finalize_link(current_user)
    current_user.finalize_meeting_link(self)
  end
end
```

/app/decorators/meeting_decorator.rb



Update View Template

```
%center
  %h1 Meetup for #{@meeting.meetup_date}
  = @meeting.display_kudo_links(current_user)
  = @meeting.display_points_awarded

  %h3 Topics Presented
  = render @meeting.time_slots

  = current_user.edit_meeting_link(@meeting)
  = current_user.finalize_meeting_link(@meeting)
```

Before the meeting display links



Final View Template

```
%center
  %h1 Meetup for #{@meeting.meetup_date}

  = @meeting.display_kudo_links(current_user)
  = @meeting.display_points_awarded

  %h3 Topics Presented
  = render @meeting.time_slots

  = @meeting.edit_link(current_user)
  = @meeting.finalize_link(current_user)
```

After the meeting display links