zero
downtime

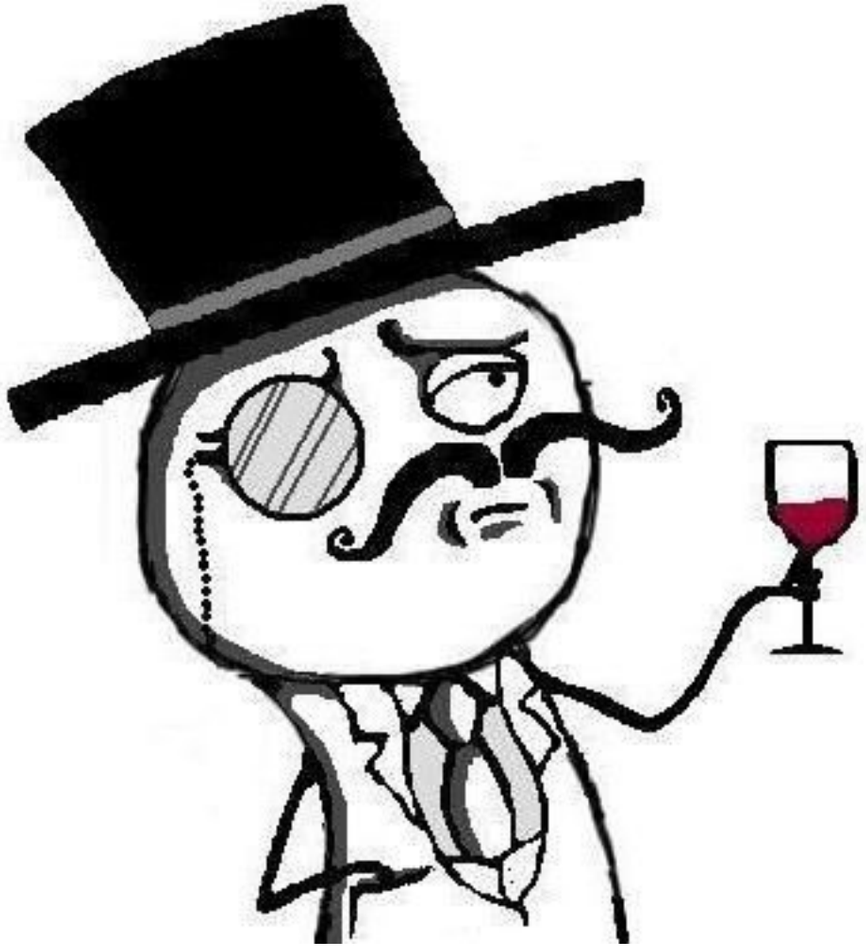
deploys
for Rails

photo by Chris
http://www.flickr.com/people/chrisinplymouth/

```ruby
class ApparentlyHarmlessMigration < ActiveRecord::Migration
  def self.up
    remove_column :users, :notes
  end
end
```

```ruby
class CompletelySafeMigration < ActiveRecord::Migration
  def self.up
    remove_column :users, :notes
  end

  def self.down
    add_column :users, :notes, :text
  end
end
```

```ruby
class CompletelySafeMigration < ActiveRecord::Migration
  def self.up
    remove_column :users, :notes
  end

  def self.down
    add_column :users, :notes, :text
  end
end
```



PGError: ERROR: column "notes" does not exist

```
INSERT INTO users(email, password, ..., notes)
```

MAKES YOU DEFINE
ATTRIBUTES IN THE DB

BLOWS UP WHEN YOU
NEED TO CHANGE THEM

heroku :: status

Resolved: Planned Maintenance: 22-June @ 20:00PDT

Please direct questions or concerns to Heroku support:
http://support.heroku.com/

JUN 21, 2011 – 21:26 UTC – 9 MONTHS AGO

ISSUE: The Heroku API will be unavailable for approximately 10 minutes beginning at 20:00 Pacific Daylight Time on Wednesday, June 22nd. During this maintenance window, the following services will be unavailable:

1. The Heroku command line client.
2. The api.heroku.com web site for managing applications.
3. git push heroku
4. Unidling of idle applications.

Running applications will be unaffected by this scheduled maintenance.

JUN 21, 2011 – 21:26 UTC – 9 MONTHS AGO


CHANGE SCHEMA ON RUNNING PROCESSES
EXPECT IT TO JUST WORK

Zero downtime?

Challenge accepted

Agenda

- Introduction
- Development practices
- Production environment setup
- Future
- Conclusion

Testable code + Test coverage	TDD

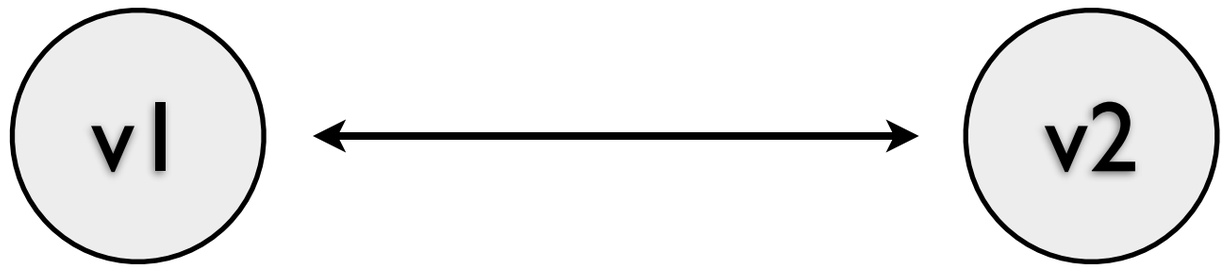
Testable code + Test coverage	TDD
Easy maintenance	DRY

Testable code + Test coverage	TDD
Easy maintenance	DRY
Zero downtime deploys	Hot compatibility

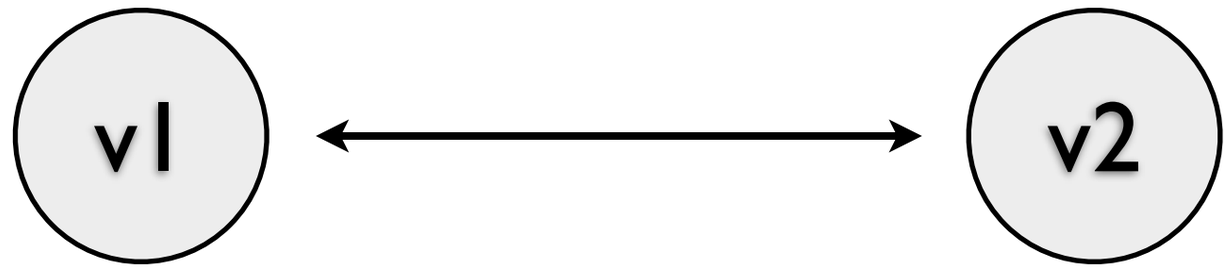
Hot compatibility

- Run concurrent versions of your application
- That are compatible between each other

Hot compatibility



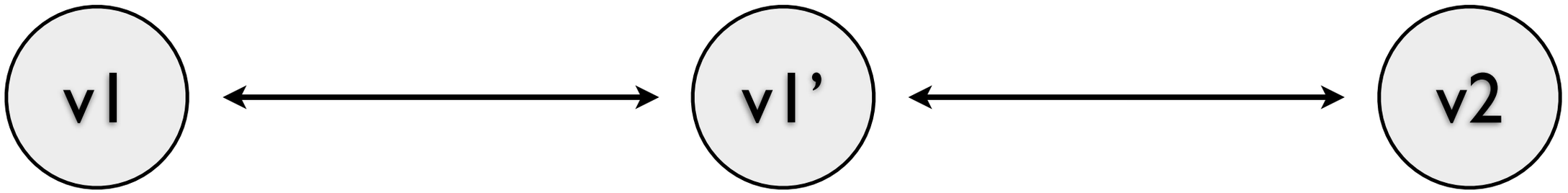
Hot compatibility



writes to `notes`

drop column `notes`

Hot compatibility

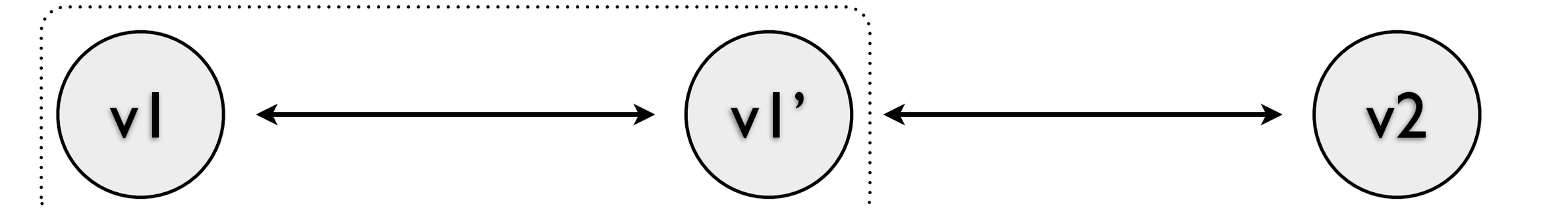


writes to `notes`

stop writing to `notes`

drop column `notes`

Hot compatibility

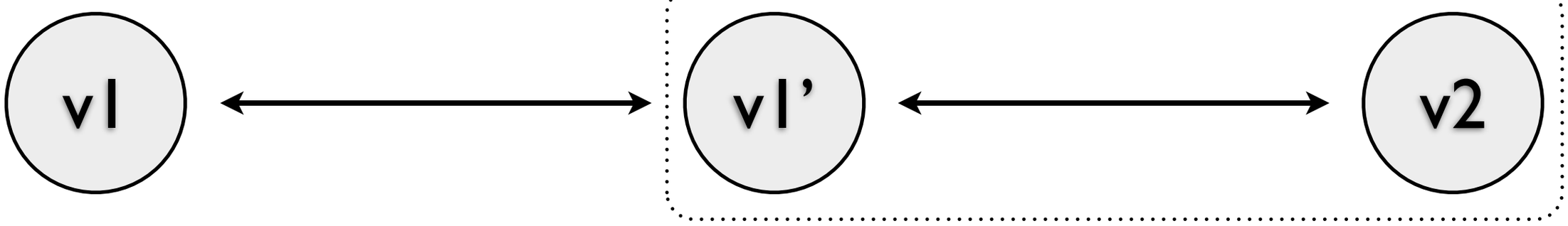


writes to `notes`

stop writing to `notes`

drop column `notes`

Hot compatibility

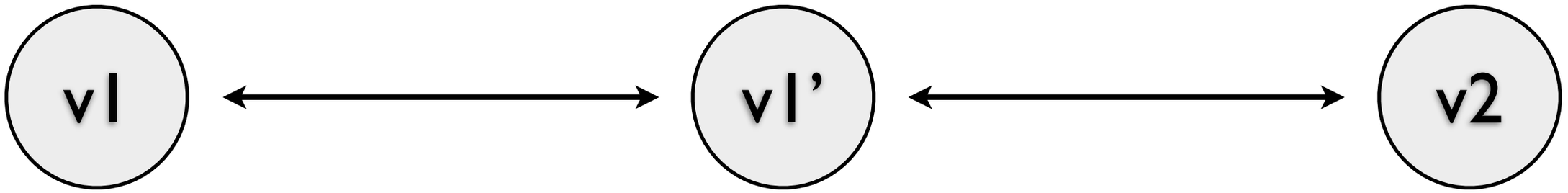


writes to `notes`

stop writing to `notes`

drop column `notes`

Hot compatibility

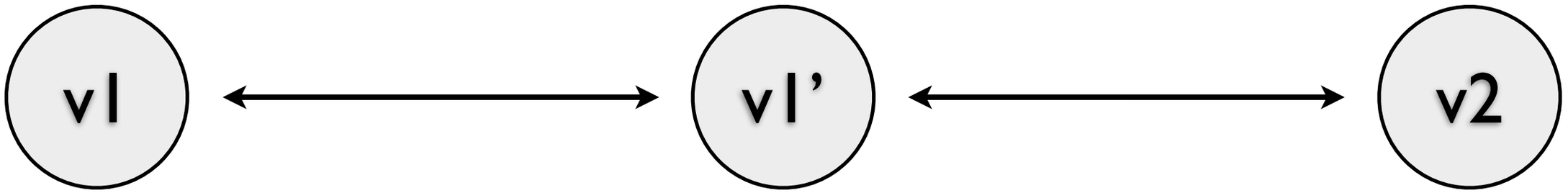


writes to `notes`

stop writing to `notes`

drop column `notes`

Hot compatibility



writes to `notes`

stop writing to `notes`

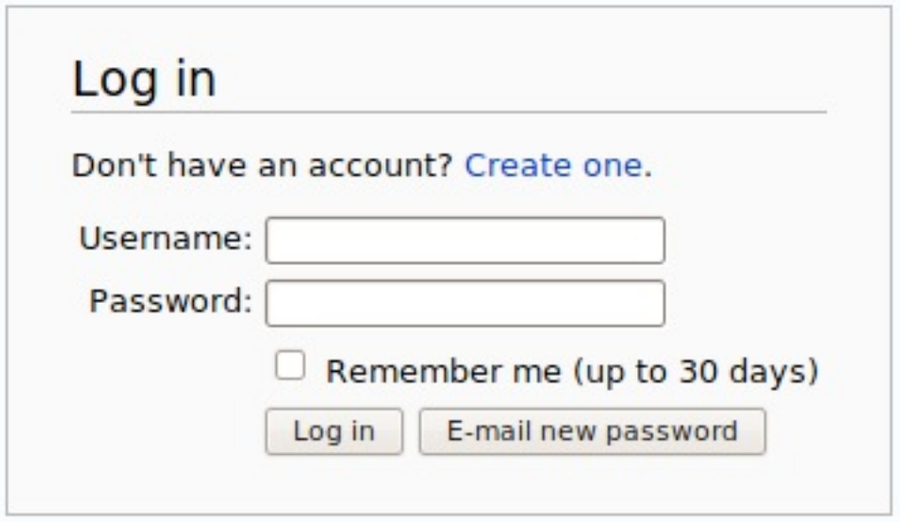
drop column `notes`

```
class User
  def self.columns
    super.reject { |c| c.name == "notes" }
  end
end
```

Self dependency

Self dependency

user loads form



Self dependency

new form is deployed

```
--- a/app/views/auth/login.rhtml
+++ b/app/views/auth/login.rhtml
@@ -7,8 +7,8 @@
   <form method="post" action="/login" class="panel-body">
     <fieldset>
       <div class="field">
-        <label>Username</label>
-        <input type="text" name="username" class="text" />
+        <label>Email</label>
+        <input type="text" name="email" class="text" />
       </div>
```

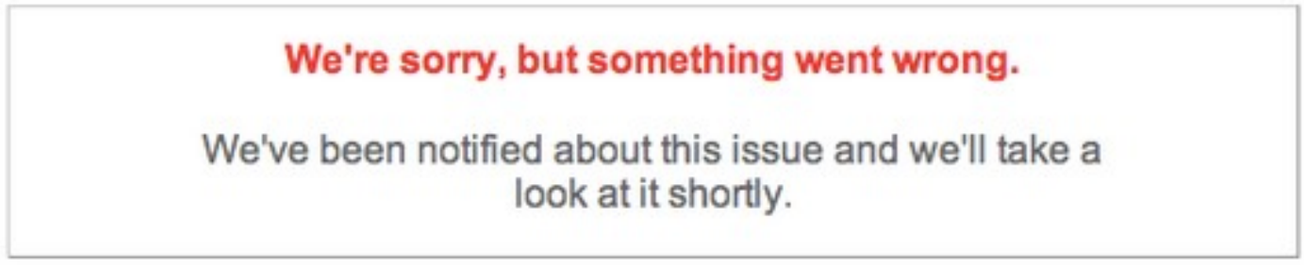
Self dependency

user submits the form

```
POST /login
username=bill@microsoft.com&password=q1w2e3
```

Self dependency

server blows up



Self dependency

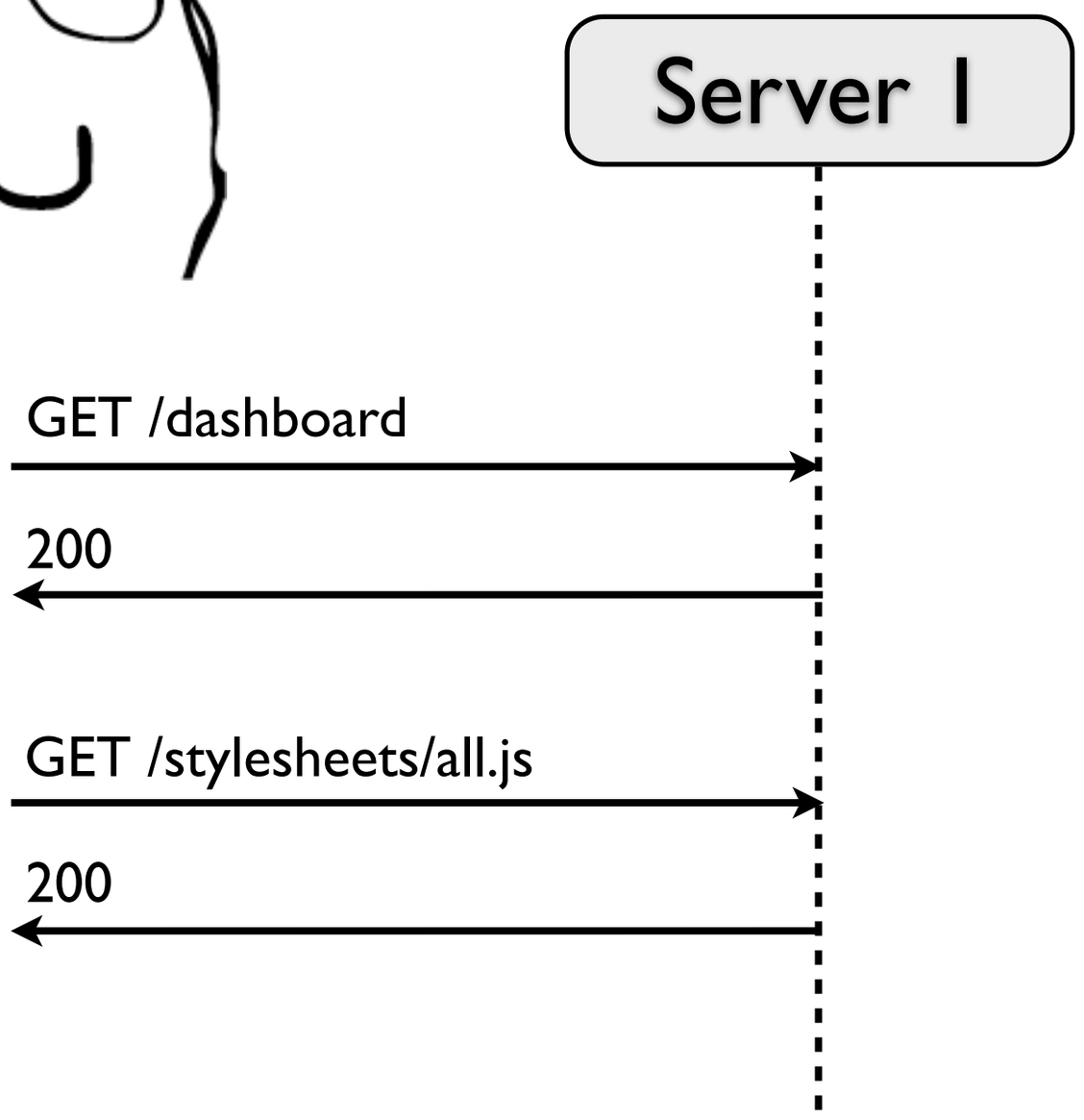
- Make controllers compatible:

Self dependency

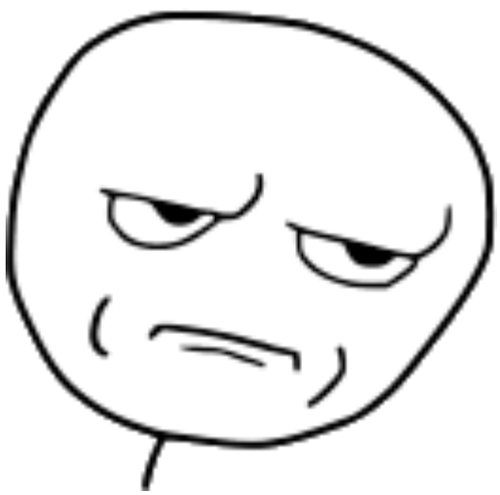
- Make controllers compatible:

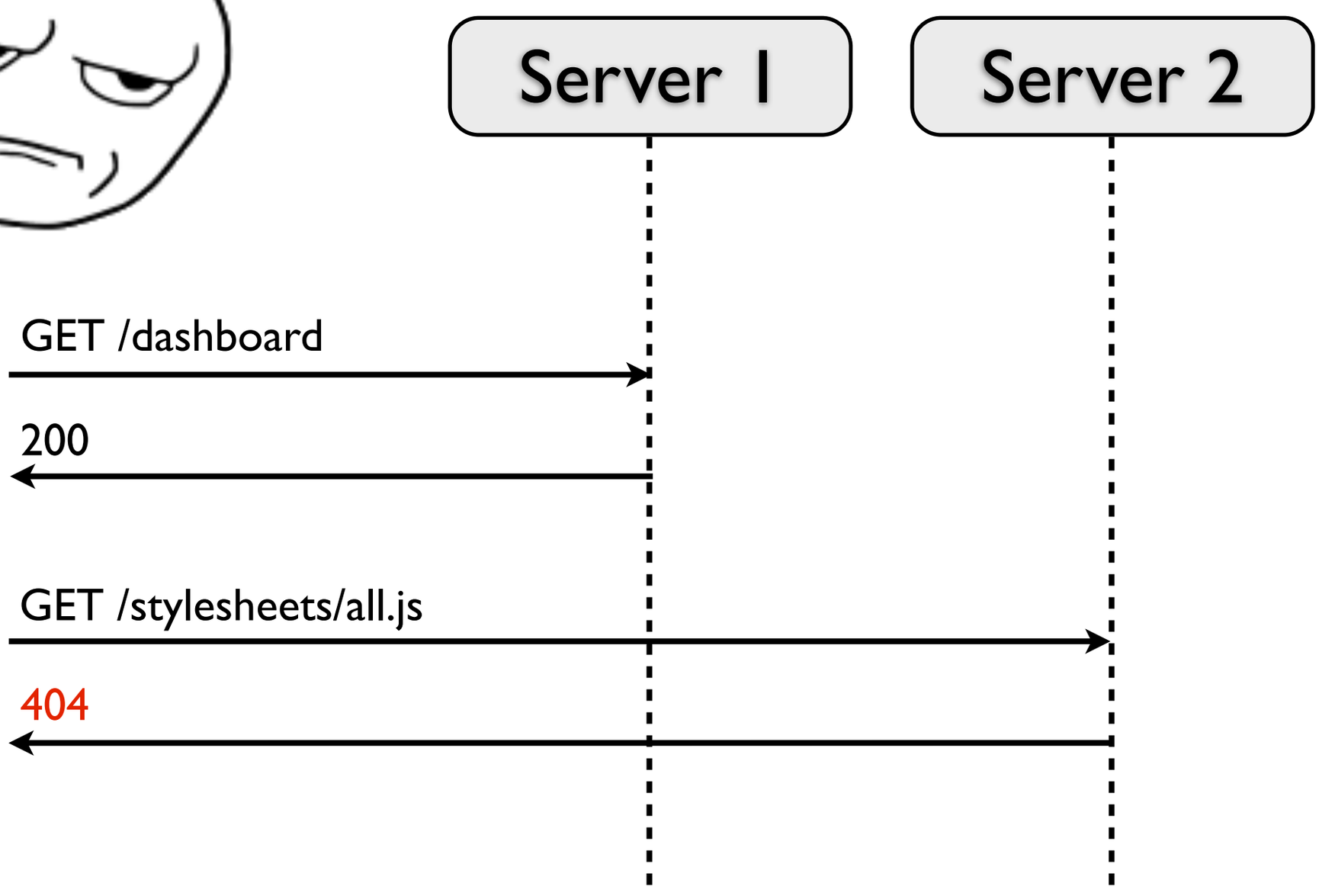
```
class AuthController < ApplicationController
  protected

  def filtered_params
    params.dup.tap do |p|
      p.merge!(:email => p.delete(:username))
    end
  end
end
```

Assets - Rails 2



Assets - Rails 2

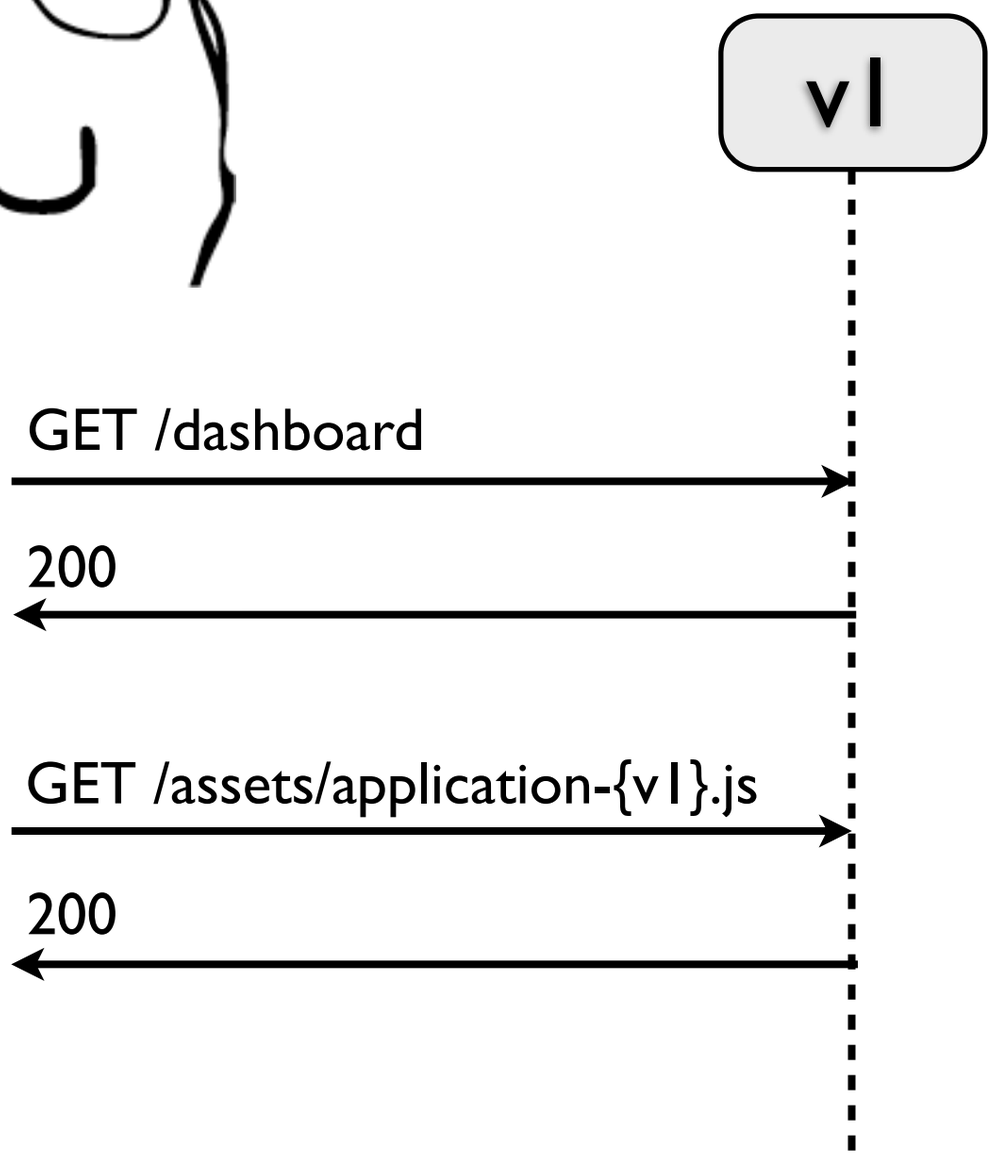




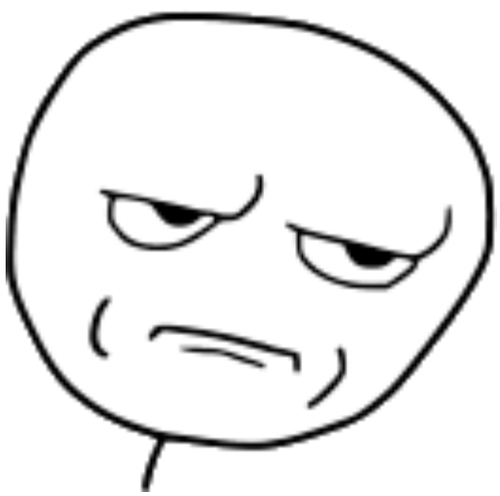
Assets - Rails 2

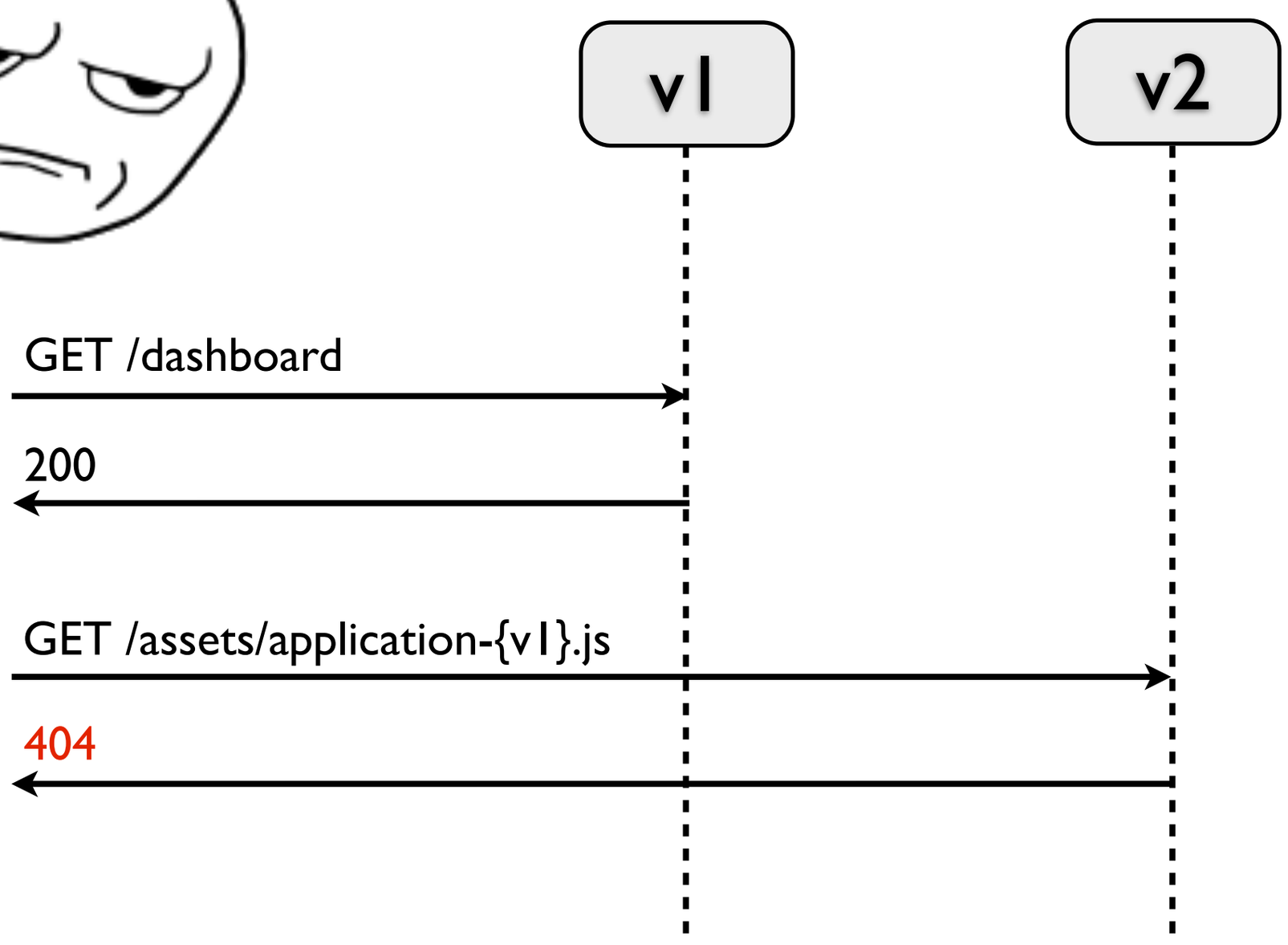
- https://github.com/ddollar/asset-resource

Asset pipeline - Rails 3



Asset pipeline - Rails 3





Asset pipeline - Rails 3

Asset pipeline - Rails 3

- When updating assets, keep the existing version around

Asset pipeline - Rails 3

- When updating assets, keep the existing version around
- We're still in need of tooling to help cleaning up old assets

Asset pipeline - Rails 3

- When updating assets, keep the existing version around
- We're still in need of tooling to help cleaning up old assets

```
rake assets:clean:outdated
```

Migration strategies

Migration strategies

- Read-only models

Migration strategies

- Read-only models

```
class Role < ActiveRecord::Base
  def readonly?
    true
  end
end
```

Migration strategies

- Read-only models
- Ignore cached columns

Migration strategies

- Read-only models
- Ignore cached columns

```
class User
  def self.columns
    super.reject { |c| c.name == "notes" }
  end
end
```

Migration strategies

- Read-only models
- Ignore cached columns
- Know your database

Migration strategies

- Read-only models
- Ignore cached columns
- Know your database
  - Beware of O(n) db locks!

MyISAM

MyISAM

stackoverflow Questions Tags Users Badges Unanswered

Rails transaction doesn't rollback on validation error

7

I have two models: user and company. They both get created from one form and I'm using a transaction like this:

```
User.transaction do

  @user.save!

  @company.user = @user
  @company.save!

  @user.reload
  @user.company = @company
  @user.save!

 flash[:notice] = "Thank you for your registration."
  redirect_to_index
end
```

The user gets saved to the database even when one of the company's validations fails. I've tried adding explicit error handling of ActiveRecord::RecordInvalid but it didn't help. I thought the validation would raise the error to rollback the transaction anyway. Any help is greatly appreciated.

InnoDB

InnoDB

- O(n) locks on:
  - Adding, updating or removing columns
  - Adding or removing indexes

InnoDB

1. Create a new table with the same structure
2. Apply changes to new table
3. Copy data over
4. Acquire lock on original table
5. Sync new/updated/deleted rows
6. Swap tables
7. Release lock

InnoDB

1. Create a new table with the same structure
2. Apply changes to new table
3. Copy data over

4. Acquire lock on original table
5. Sync new/updated/deleted rows
6. Swap tables
7. Release lock

InnoDB

https://github.com/freels/table_migrator

InnoDB

https://github.com/freels/table_migrator

```
class AddStuffToMyBigTable < TableMigration

  migrates :users

  change_table do |t|
    t.integer :foo, :null => false, :default => 0
    t.string  :bar
    t.index   :foo, :name => 'index_for_foo'
  end
end
```

InnoDB

https://github.com/freels/table_migrator

- Requires acts_as_paranoid
- Pretty old/stale codebase
  - Check forks for Rails 3 compatibility

Postgres

- Table locks on:
  - Adding, updating or removing columns
  - Adding or removing indexes

Postgres

- O(n) locks on:
  - ~~Adding~~, updating ~~or removing~~ columns
  - ~~Adding or removing indexes~~

Postgres

- O(n) lock
  - ~~Adding~~, updating ~~or removing~~ columns
  - ~~Adding or removing indexes~~

constant time if you avoid default values and constraints

Postgres

- O(n) locks on:
  - ~~Adding~~, updating ~~or removing~~ columns
  - ~~Adding or removing indexes~~

constant time

Postgres

- O(n) locks on:
  - ~~Adding~~, updating ~~or removing~~ columns
  - ~~Adding or removing indexes~~

can be done without any locking

Postgres

- O(n) locks on:
  - ~~Adding~~, updating ~~or removing~~ columns
  - ~~Adding or removing indexes~~

constant time

Postgres

- Adding fields with defaults

Postgres

- Adding fields with defaults

```
before_save :assign_defaults

def assign_defaults
  self.admin ||= false
end
```

Postgres

- Adding fields with defaults
- Create indexes concurrently

Postgres

- Adding fields with defaults
- Create indexes concurrently

```
CREATE INDEX CONCURRENTLY users_email_index ON users(email);
```

Postgres

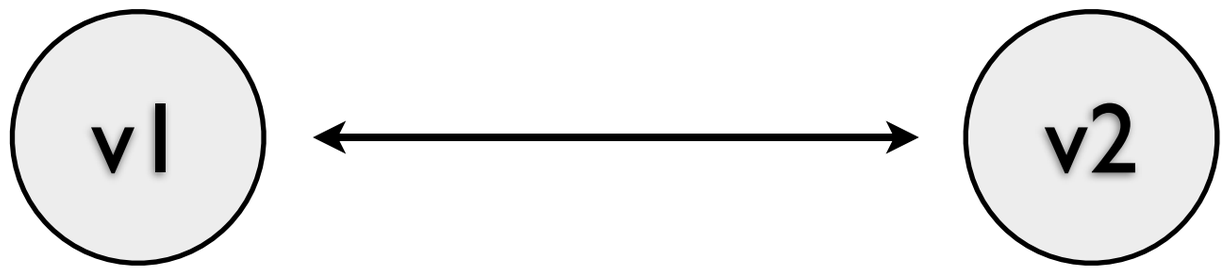
- Adding fields with defaults
- Create indexes concurrently
- Outside a transaction!

Postgres

- Adding fields with defaults
- Create indexes concurrently
- Outside a transaction!

```
class AddIndexToUsers < ActiveRecord::Migration
  def self.up
    return if indexes("users").map(&:name).include?("users_email_index")
    add_index :users, :email, :name => "users_email_index"
  end

  def self.down
    remove_index :users, :name => "users_email_index"
  end
end
```

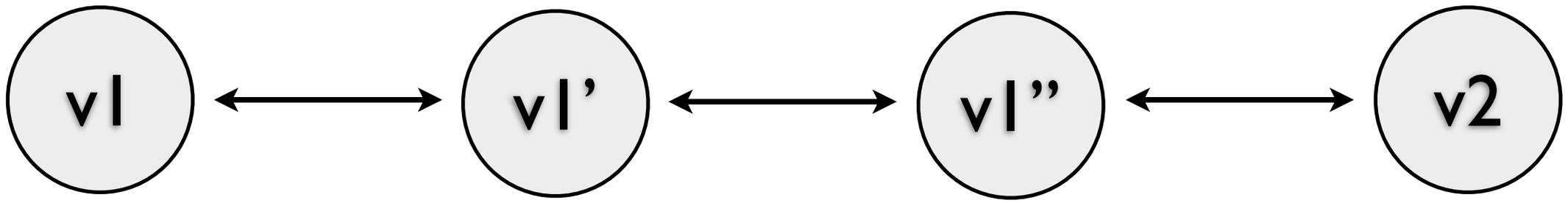
What about renaming?



reads and writes
to `foo`

reads and writes
to `bar`

What about renaming?



reads and writes
to `foo`

add column `bar`
+
reads from `foo`
writes to both
`foo` and `bar`

populate `bar`
where needed

reads and writes
to `bar`

What about renaming?

reads and writes
to `bar`

ignore `foo`

drop column `foo`

NoSQL

NoSQL

- Schema migrations / O(n) locks are just part of hot compatibility

NoSQL

- Schema migrations / O(n) locks are just part of hot compatibility
- You still need to put effort into data migration

Workflow

Workflow

- Make sure pull requests are addressing hot compatibility

Workflow

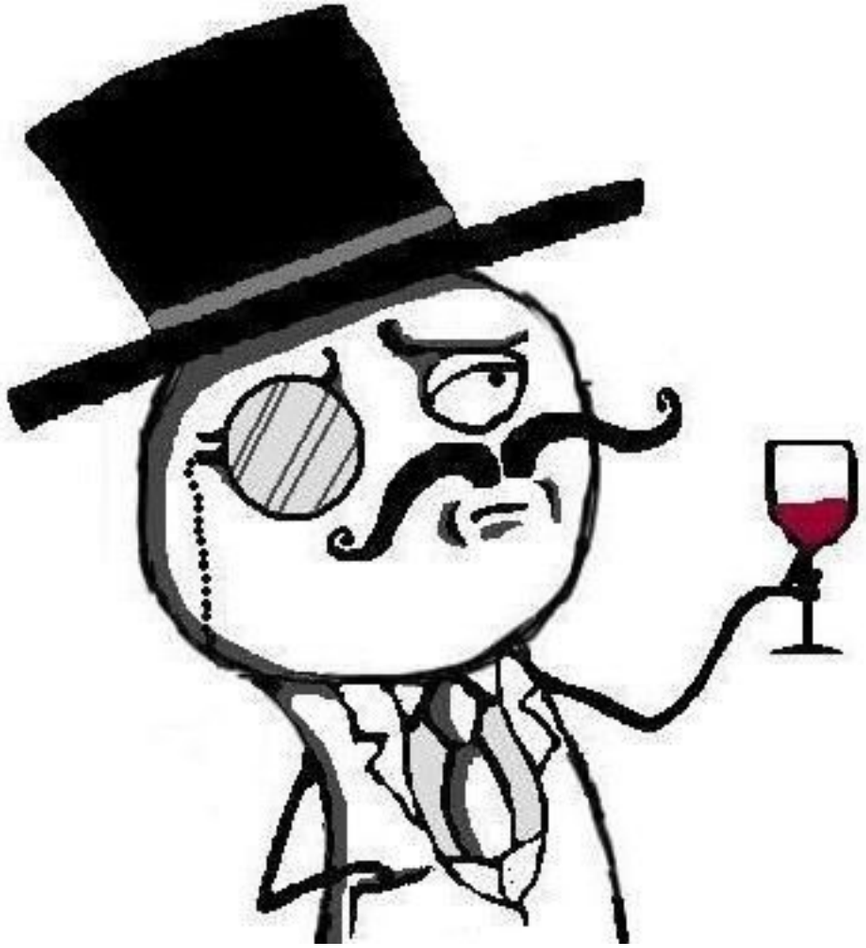
- Make sure pull requests are addressing hot compatibility
- Work after code is merged

Meanwhile, in production...

```
$ thin -C config/server.yml restart
```

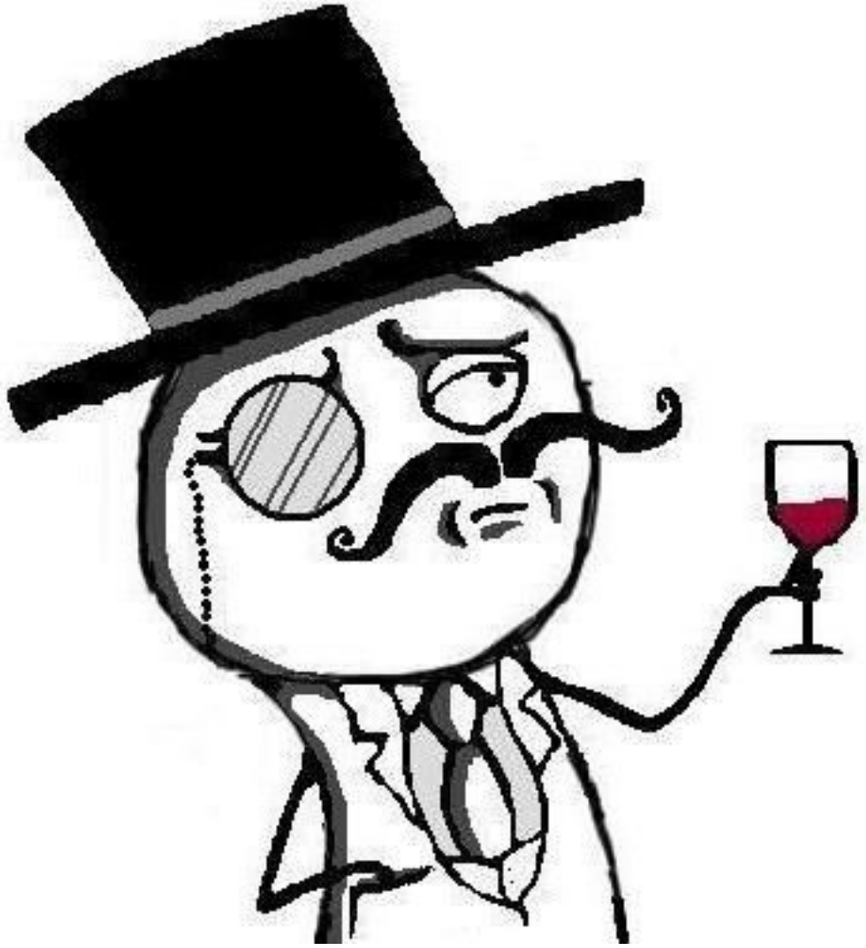
Meanwhile, in production...

```
$ thin -C config/server.yml restart
```



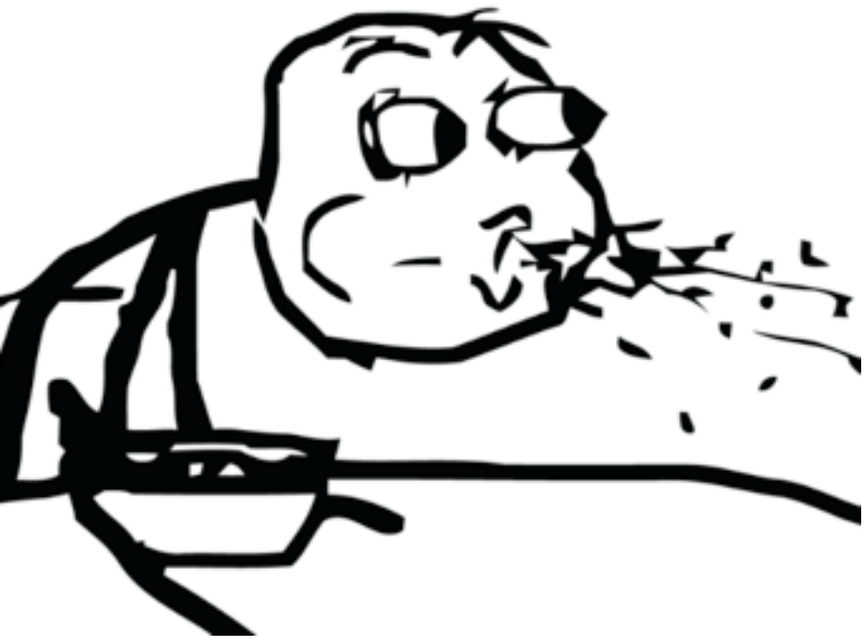
Meanwhile, in production...

```
$ thin -C config/server.yml restart
Stopping server on 0.0.0.0:5000 ...
Sending QUIT signal to process 3463 ...
Stopping server on 0.0.0.0:5001 ...
Sending QUIT signal to process 3464 ...
Stopping server on 0.0.0.0:5002 ...
Sending QUIT signal to process 3465 ...
```



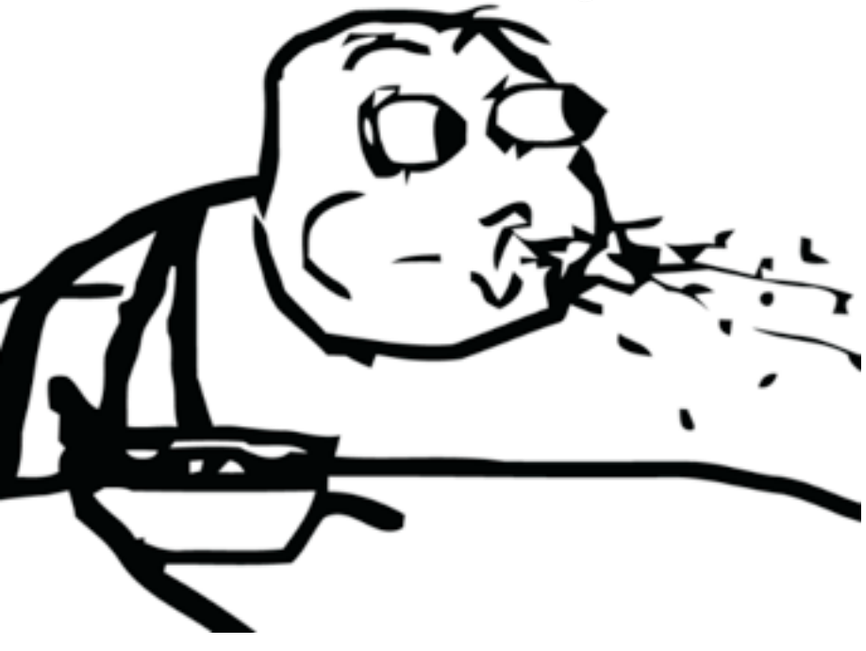
Meanwhile, in production...

```
$ thin -C config/server.yml restart
Stopping server on 0.0.0.0:5000 ...
Sending QUIT signal to process 3463 ...
Stopping server on 0.0.0.0:5001 ...
Sending QUIT signal to process 3464 ...
Stopping server on 0.0.0.0:5002 ...
Sending QUIT signal to process 3465 ...
```



Meanwhile, in production...

```
$ thin -C config/server.yml restart
Stopping server on 0.0.0.0:5000 ...
Sending QUIT signal to process 3463 ...
Stopping server on 0.0.0.0:5001 ...
Sending QUIT signal to process 3464 ...
Stopping server on 0.0.0.0:5002 ...
Sending QUIT signal to process 3465 ...
Starting server on 0.0.0.0:5000 ...
Starting server on 0.0.0.0:5001 ...
Starting server on 0.0.0.0:5002 ...
```



Agenda

- Introduction
- Development practices
- Production environment setup
- Future
- Conclusion

Production

Production

- Run multiple server processes

Production

- Run multiple server processes
- Cycle them individually, or in batches

Production

- Run multiple server processes
- Cycle them individually, or in batches
    - Watch out for capacity!

Production

- Run multiple server processes
- Cycle them individually, or in batches
  - Watch out for capacity!
- Cycle them gracefully

Cycling techniques

- Above the stack
- Within the stack
- Platform

Above the stack

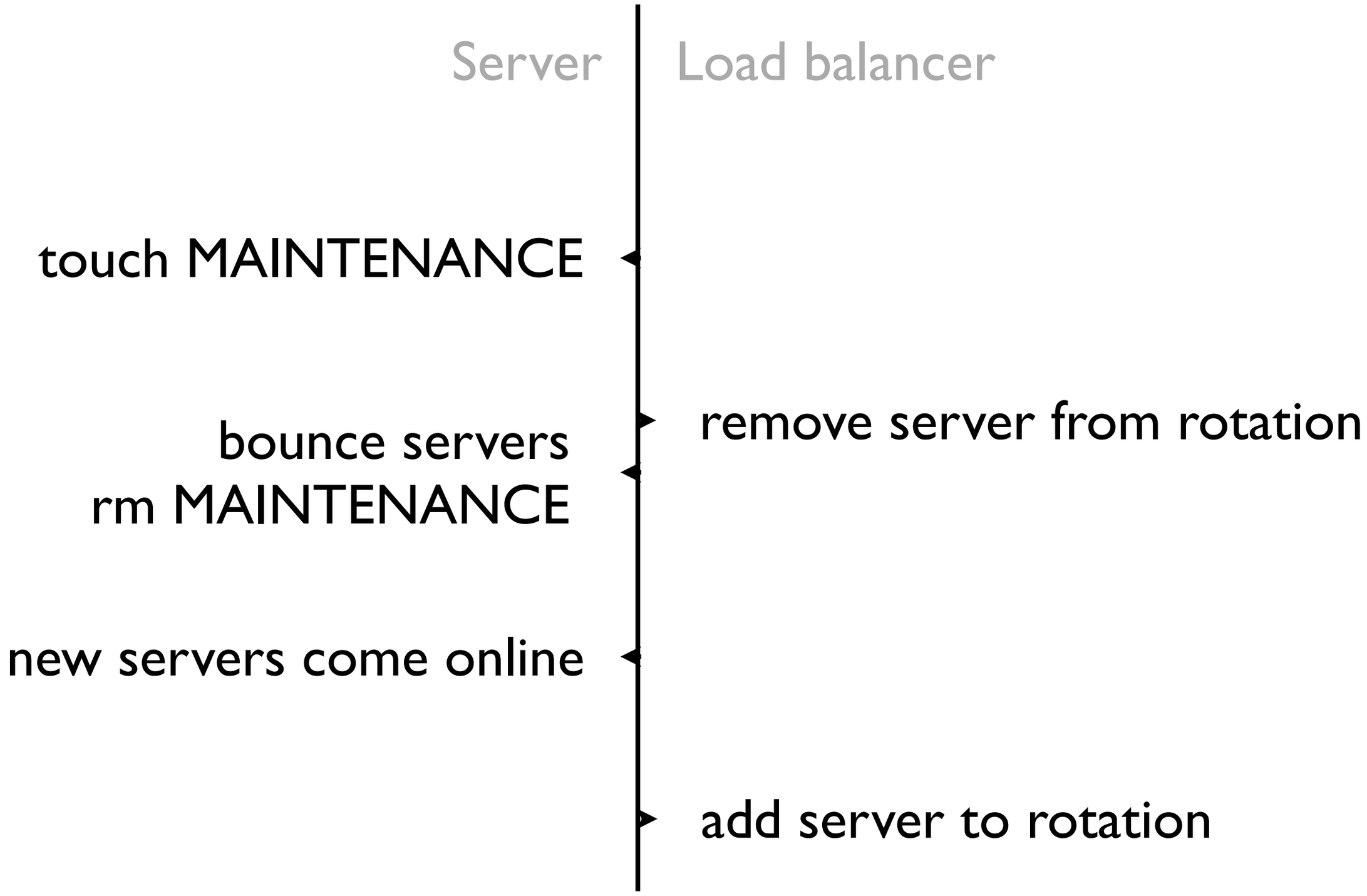
- Use your favorite web server
- Setup a load balancer above them, route around servers that are cycling

Above the stack

- PROTIP: default setup won't work
- Configure load balancer to poll for your application health:

```
class HealthController < ApplicationController
  def index
    if File.exists?(Rails.root.join("MAINTENANCE"))
      render :text => "Forget me!", :status => 503
    else
      render :text => "I'm alive!"
    end
  end
end
```

Cycling + Load balancer



HAProxy

```
option httpchk GET /health

listen myapp :9000
  server myapp-1 1.0.0.1:5000 maxconn 1 check inter 2000 fall 1 rise 1
  server myapp-2 1.0.0.2:5000 maxconn 1 check inter 2000 fall 1 rise 1
```

HAProxy

tell HAProxy where to check for the health

```
option httpchk GET /health

listen myapp :9000
  server myapp-1 1.0.0.1:5000 maxconn 1 check inter 2000 fall 1 rise 1
  server myapp-2 1.0.0.2:5000 maxconn 1 check inter 2000 fall 1 rise 1
```

HAProxy

```
option httpchk GET /health

listen myapp :9000
  server myapp-1 1.0.0.1:5000 maxconn 1 check inter 2000 fall 1 rise 1
  server myapp-2 1.0.0.2:5000 maxconn 1 check inter 2000 fall 1 rise 1
```

do not let your web server queue requests

HAProxy

```
option httpchk GET /health

listen myapp :9000
  server myapp-1 1.0.0.1:5000 maxconn 1 check inter 2000 fall 1 rise 1
  server myapp-2 1.0.0.2:5000 maxconn 1 check inter 2000 fall 1 rise 1
```

tell HAProxy to run the health check specified above

HAProxy

```
option httpchk GET /health

listen myapp :9000
  server myapp-1 1.0.0.1:5000 maxconn 1 check inter 2000 fall 1 rise 1
  server myapp-2 1.0.0.2:5000 maxconn 1 check inter 2000 fall 1 rise 1
```

every 2s

HAProxy

```
option httpchk GET /health

listen myapp :9000
  server myapp-1 1.0.0.1:5000 maxconn 1 check inter 2000 fall 1 rise 1
  server myapp-2 1.0.0.2:5000 maxconn 1 check inter 2000 fall 1 rise 1
```

stop routing to it once the health check failed once

HAProxy

```
option httpchk GET /health

listen myapp :9000
  server myapp-1 1.0.0.1:5000 maxconn 1 check inter 2000 fall 1 rise 1
  server myapp-2 1.0.0.2:5000 maxconn 1 check inter 2000 fall 1 rise 1
```

and bring it back once it worked once

ELB

```
$ elb-configure-healthcheck MyLoadBalancer \
--target "HTTP:5000/health" \
--interval 2 \
--unhealthy-threshold 1 \
--healthy-threshold 1
```

Above the stack

Above the stack

- Minimal impact

Above the stack

- Minimal impact
- Extended availability comes for free

Above the stack

- Minimal impact
- Extended availability comes for free

- Complicated setup

Above the stack

- Minimal impact
- Extended availability comes for free

- Complicated setup
- Another component you have to track and maintain

Within the stack

Within the stack

Unicorn

- Replaces your web server
- Bounces gracefully

Unicorn

```
63480 0.0  1.4  2549796  unicorn_rails master -c config/unicorn.rb
63489 0.0  0.2  2549796  unicorn_rails worker[0] -c config/unicorn.rb
63490 0.0  0.2  2549796  unicorn_rails worker[1] -c config/unicorn.rb
63491 0.0  0.2  2549796  unicorn_rails worker[2] -c config/unicorn.rb
```

Unicorn

```
$ kill -USR2 63480

63480 0.0  1.4  2549796  unicorn_rails master -c config/unicorn.rb
63489 0.0  0.2  2549796  unicorn_rails worker[0] -c config/unicorn.rb
63490 0.0  0.2  2549796  unicorn_rails worker[1] -c config/unicorn.rb
63491 0.0  0.2  2549796  unicorn_rails worker[2] -c config/unicorn.rb
```

Unicorn

```
$ kill -USR2 63480

63480 0.0  1.4  2549796  unicorn_rails master (old) -c config/unicorn.rb
63489 0.0  0.2  2549796  unicorn_rails worker[0] -c config/unicorn.rb
63490 0.0  0.2  2549796  unicorn_rails worker[1] -c config/unicorn.rb
63491 0.0  0.2  2549796  unicorn_rails worker[2] -c config/unicorn.rb
63521 0.0  1.4  2549796  unicorn_rails master -c config/unicorn.rb
63535 0.0  0.2  2549796  unicorn_rails worker[0] -c config/unicorn.rb
63536 0.0  0.2  2549796  unicorn_rails worker[1] -c config/unicorn.rb
63537 0.0  0.2  2549796  unicorn_rails worker[2] -c config/unicorn.rb
```

Unicorn

```
63480 0.0  1.4  2549796  unicorn_rails master (old) -c config/unicorn.rb
63489 0.0  0.2  2549796  unicorn_rails worker[0] -c config/unicorn.rb
63490 0.0  0.2  2549796  unicorn_rails worker[1] -c config/unicorn.rb
63491 0.0  0.2  2549796  unicorn_rails worker[2] -c config/unicorn.rb
63521 0.0  1.4  2549796  unicorn_rails master -c config/unicorn.rb
63535 0.0  0.2  2549796  unicorn_rails worker[0] -c config/unicorn.rb
63536 0.0  0.2  2549796  unicorn_rails worker[1] -c config/unicorn.rb
63537 0.0  0.2  2549796  unicorn_rails worker[2] -c config/unicorn.rb
```

Unicorn

```
$ kill -QUIT 63480

63480 0.0  1.4  2549796  unicorn_rails master (old) -c config/unicorn.rb
63489 0.0  0.2  2549796  unicorn_rails worker[0] -c config/unicorn.rb
63490 0.0  0.2  2549796  unicorn_rails worker[1] -c config/unicorn.rb
63491 0.0  0.2  2549796  unicorn_rails worker[2] -c config/unicorn.rb
63521 0.0  1.4  2549796  unicorn_rails master -c config/unicorn.rb
63535 0.0  0.2  2549796  unicorn_rails worker[0] -c config/unicorn.rb
63536 0.0  0.2  2549796  unicorn_rails worker[1] -c config/unicorn.rb
63537 0.0  0.2  2549796  unicorn_rails worker[2] -c config/unicorn.rb
```

Unicorn

```
$ kill -QUIT 63480

63521 0.0  1.4  2549796  unicorn_rails master -c config/unicorn.rb
63535 0.0  0.2  2549796  unicorn_rails worker[0] -c config/unicorn.rb
63536 0.0  0.2  2549796  unicorn_rails worker[1] -c config/unicorn.rb
63537 0.0  0.2  2549796  unicorn_rails worker[2] -c config/unicorn.rb
```

Unicorn

```
63521 0.0  1.4  2549796  unicorn_rails master -c config/unicorn.rb
63535 0.0  0.2  2549796  unicorn_rails worker[0] -c config/unicorn.rb
63536 0.0  0.2  2549796  unicorn_rails worker[1] -c config/unicorn.rb
63537 0.0  0.2  2549796  unicorn_rails worker[2] -c config/unicorn.rb
```

Within the stack

Within the stack

- Easy to setup and deploy

Within the stack

- Easy to setup and deploy
- Optimal performance to cycle the server

Within the stack

- Easy to setup and deploy
- Optimal performance to cycle the server

- Changes your application stack

Within the stack

- Easy to setup and deploy
- Optimal performance to cycle the server

- Changes your application stack
- Doesn't address availability issues

Platform

Heroku

```
Process  State               Command
-------  ------------------  ------------------------------
web.1    up for 10h          bundle exec thin start -p $PORT
```

Heroku

```
$ git push heroku master
```

```
Process  State               Command
-------  ------------------  ------------------------------
web.1    up for 10h          bundle exec thin start -p $PORT
```

Heroku

```
$ git push heroku master
```

```
Process   State               Command
-------   -----------------   ------------------------------
web.1     starting for 1s     bundle exec thin start -p $PORT
```

Heroku

```
Process  State               Command
-------  ------------------  ------------------------------
web.1    starting for 5s     bundle exec thin start -p $PORT
```

Heroku

```
Process  State               Command
-------  ------------------  ------------------------------
web.1    up for 1s           bundle exec thin start -p $PORT
```

Heroku

BETA

```
Process  State               Command
-------  ------------------  ------------------------------
web.1    up for 10h          bundle exec thin start -p $PORT
```

Heroku

BETA

```
$ heroku labs:enable preboot

Process  State              Command
-------  -----------------  ------------------------------
web.1    up for 10h         bundle exec thin start -p $PORT
```

Heroku

BETA

```
Process  State               Command
-------  ------------------  ------------------------------
web.1    up for 10h          bundle exec thin start -p $PORT
```

Heroku

BETA

```
$ git push heroku master
```

```
Process   State               Command
-------   -----------------   ------------------------------
web.1     up for 10h          bundle exec thin start -p $PORT
```

Heroku

BETA

```
$ git push heroku master
```

```
Process  State              Command
-------  -----------------  ------------------------------
web.1    up for 10h         bundle exec thin start -p $PORT
web.1    starting for 5s    bundle exec thin start -p $PORT
```

Heroku

BETA

```
Process  State               Command
-------  ------------------  ------------------------------
web.1    up for 10h          bundle exec thin start -p $PORT
web.1    up for 1s           bundle exec thin start -p $PORT
```

Heroku

BETA

```
Process  State               Command
-------  ------------------  ------------------------------
web.1    up for 5s           bundle exec thin start -p $PORT
```

Heroku

- Migrations

Heroku

- Migrations

```
$ git push pre-production master
$ heroku run rake db:migrate --remote pre-production
$ git push production master
```

Agenda

- Introduction
- Development practices
- Production environment setup
- Future
- Conclusion

Future

Future

- Asset pipeline

Future

- Asset pipeline
- Mysql tooling

Future

- Asset pipeline
- Mysql tooling
- Can Rails help making the development of hot compatible applications easier?

Future

- Asset pipeline
- Mysql tooling
- Can Rails help making the development of hot compatible applications easier?
- Can databases?

Future

Future

- Hot compatibility beyond zero downtime deploys

Conclusion

Conclusion

- Zero downtime is a pain in the ass

Conclusion

- Zero downtime is a pain in the ass
- Hot compatibility to the rescue

Conclusion

- Zero downtime is a pain in the ass
- Hot compatibility to the rescue
- Make sure your production env is ready

Thank you!

http://pedro.heroku.com

@ped