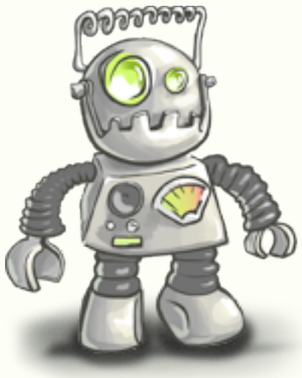
Chronicle of a SOA foretold
(Ruby in the enterprise)

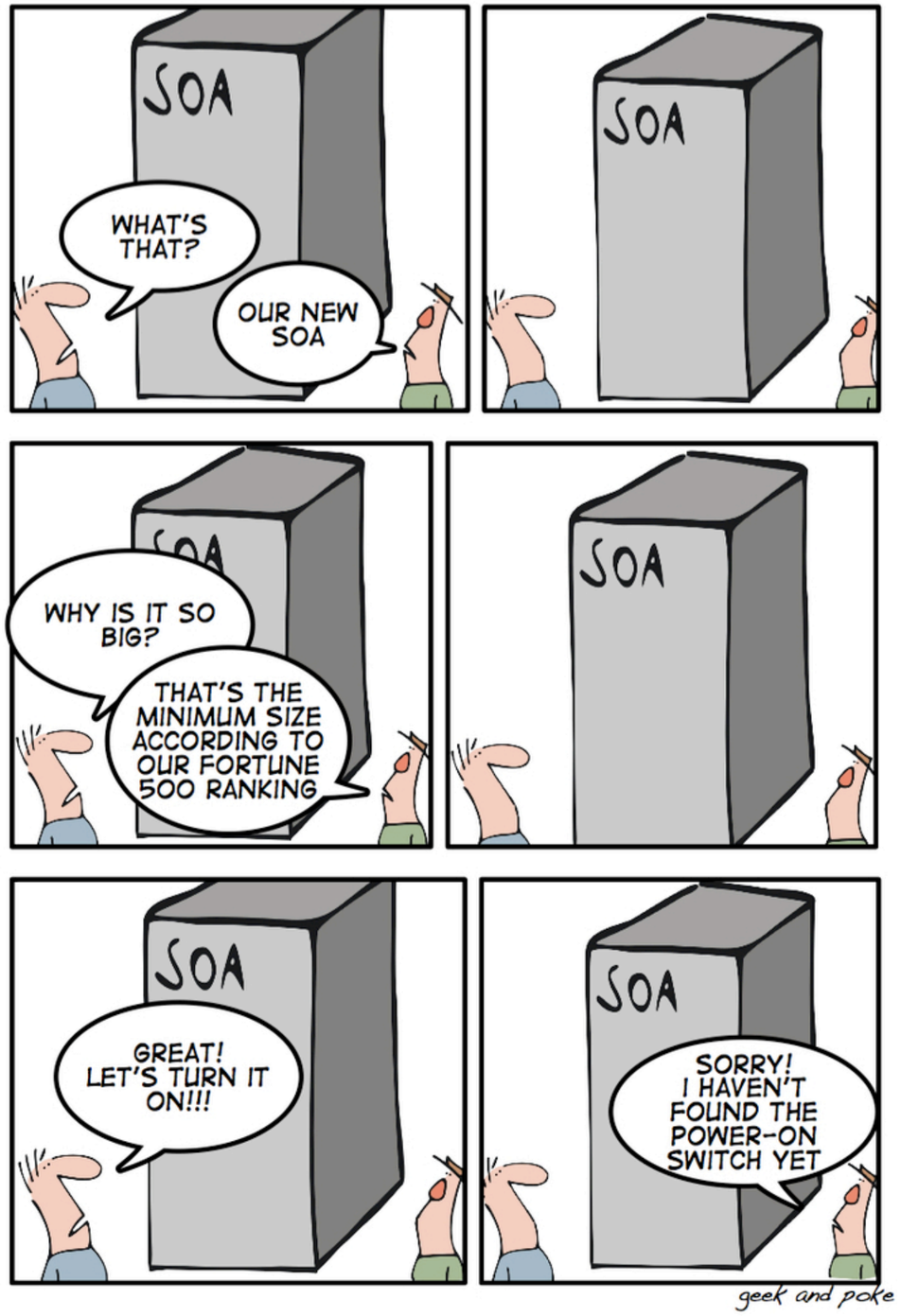
Me • @kidpollo

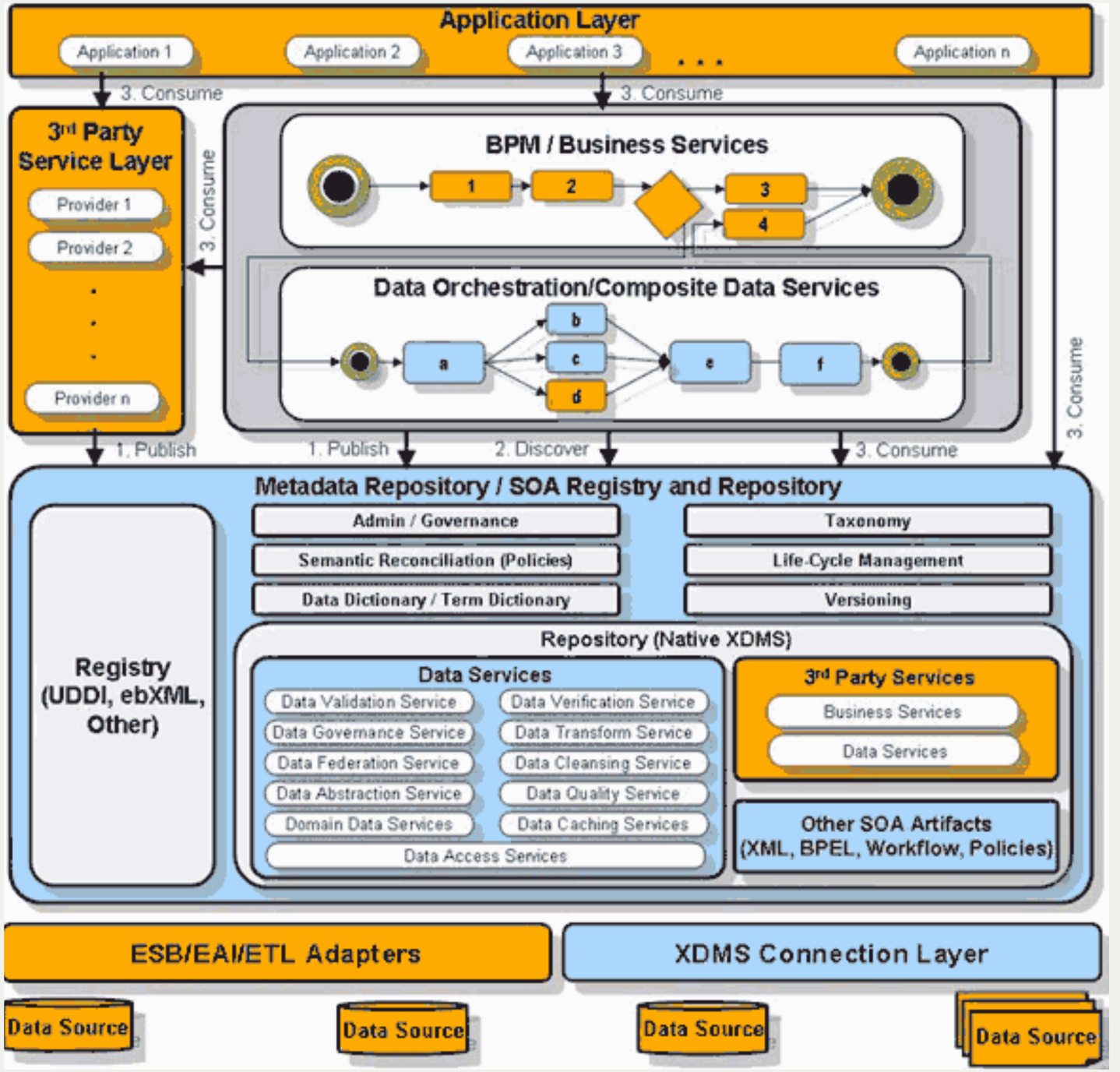


- Señor Engineer @ Get Satisfaction
- I <3 Ruby
- Mentor
- Entrepreneur wannabe

I take pictures and like good food

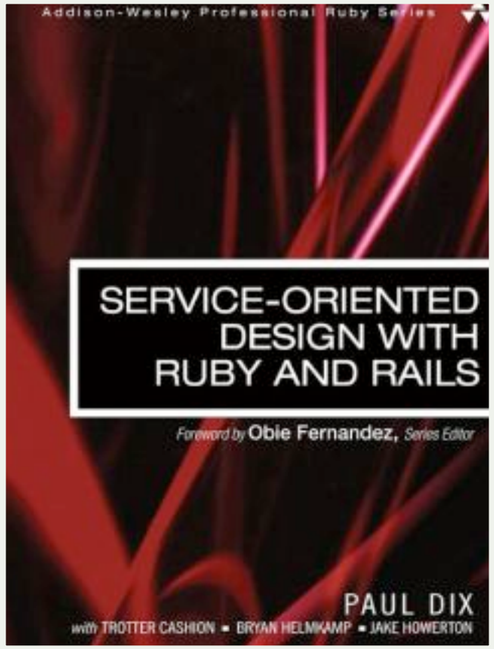
Some will say I am a hipster


SOA
WHAT'S THAT?
OUR NEW SOA
SOA
SOA
WHY IS IT SO BIG?
THAT'S THE MINIMUM SIZE ACCORDING TO OUR FORTUNE 500 RANKING
SOA
SOA
GREAT! LET'S TURN IT ON!!!
SOA
SORRY! I HAVEN'T FOUND THE POWER-ON SWITCH YET
geek and poke


Application Layer
Application 1
Application 2
Application 3
. . .
Application n
3. Consume
3. Consume
3rd Party Service Layer
Provider 1
Provider 2
Provider n
3. Consume
BPM / Business Services
1
2
3
4
Data Orchestration/Composite Data Services
a
b
c
d
e
f
3. Consume
1. Publish
1. Publish
2. Discover
3. Consume
Metadata Repository / SOA Registry and Repository
Admin / Governance
Semantic Reconciliation (Policies)
Data Dictionary / Term Dictionary
Taxonomy
Life-Cycle Management
Versioning
Registry (UDDI, ebXML, Other)
Repository (Native XDMS)
Data Services
Data Validation Service
Data Verification Service
Data Governance Service
Data Transform Service
Data Federation Service
Data Cleansing Service
Data Abstraction Service
Data Quality Service
Domain Data Services
Data Caching Services
Data Access Services
3rd Party Services
Business Services
Data Services
Other SOA Artifacts (XML, BPEL, Workflow, Policies)
ESB/EAI/ETL Adapters
XDMS Connection Layer
Data Source
Data Source
Data Source
Data Source

SOA Service Oriented Architectures

- Well defined
- Built as components
- Can be re-used
- Loosely coupled



Ruby is no longer a kid

- February 24, 1993
- December 21, 1995
- February 4, 2004
- 2.0
- 4.0

Rails is SOA ready

The Novella

63

days since last drama
@rubydramas #rubydrama
RSS Feed

Why Brit Ruby 2013 was cancelled and why this is not ok (38 days)
Implement Routing Concerns (88 days)
Why you shouldnt invite Yehuda Katz to your user group meeting (132 days)
Testing like the TSA (1 days)
Our Culture of Exclusion (9 days)
rails.app (5 days)
wow how come I commit in master? O_o (24 days)
Rails Went Off The Rails: Why I'm Rebuilding Archaeopteryx In CoffeeScript (10 days)
The Ruby Colored Box (23 days)
rbenv: A Simple, New Ruby Version Management Tool (167 days)
Fix an issue with the `rbenv` command (4 days)
Include CoffeeScript in Gemfile (121 days)

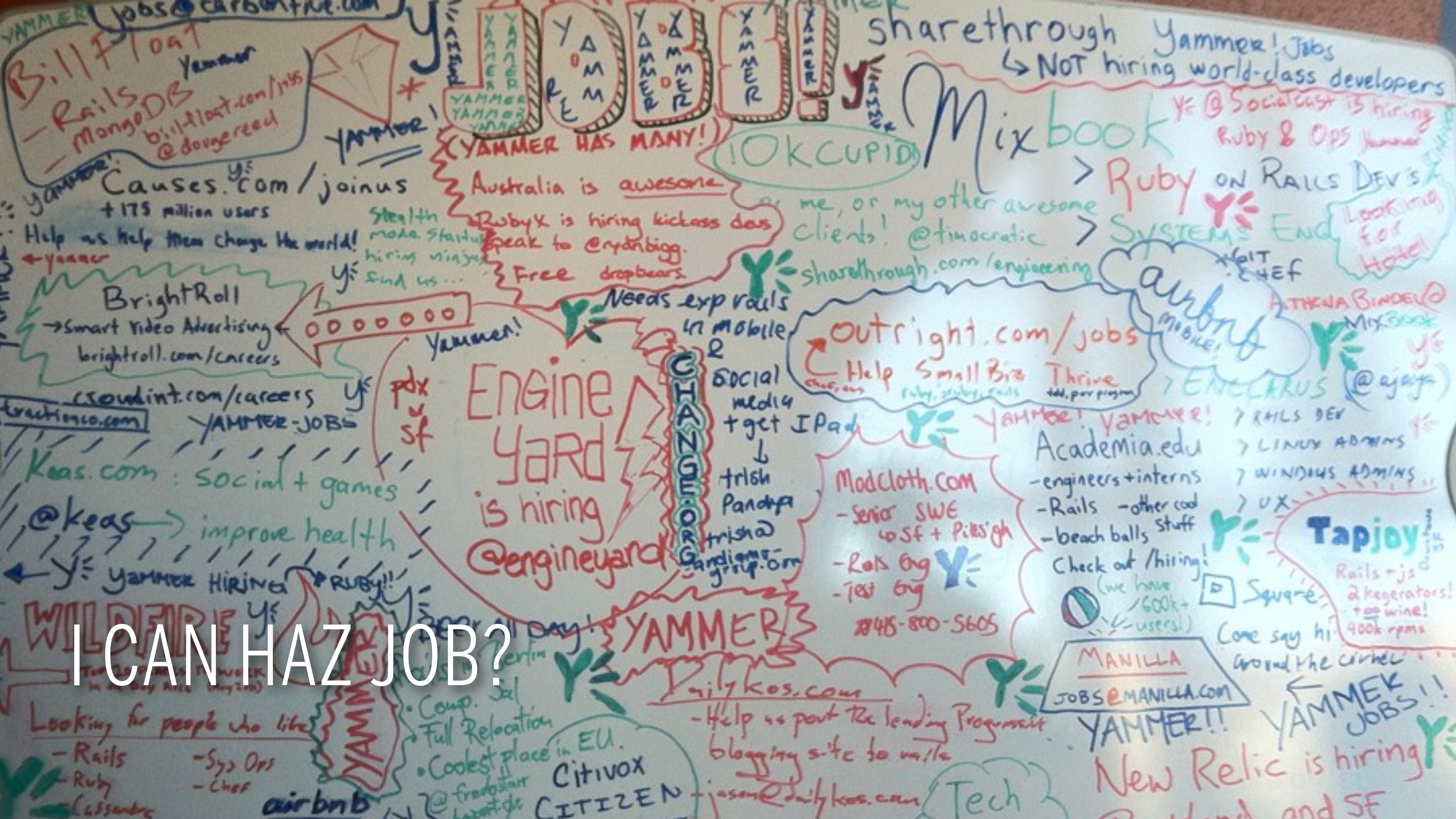
Can Rails scale?

NO

http://canrailsscale.com/

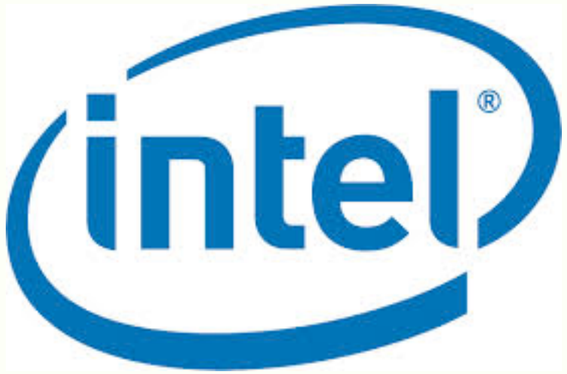
“Ruby is simple in appearance, but is very complex inside, just like our human body.”

-Matz


JOBS!
sharethrough
Yammer! Jobs
NOT hiring world-class developers
Mixbook
@Socialcast is hiring Ruby & Ops
Ruby on Rails Dev's
Systems Eng
BillFloat
Rails
MongoDB
billfloat.com/jobs
@dougereed
(YAMMER HAS MANY!)
Australia is awesome
RubyX is hiring kickass devs
Speak to @ryanbigg
Free dropbears
!OKCUPID
or me, or my other awesome clients! @timocratic
sharethrough.com/engineering
Causes.com/joinus
+175 million users
Help us help them change the world!
Stealth mode Startup hiring ninjas
BrightRoll
Smart Video Advertising
brightroll.com/careers
Needs exp rails in mobile
social media
get IPad
Engine Yard is hiring
@engineyard
CHANGE.ORG
outright.com/jobs
Help Small Biz Thrive
airbnb
Looking for Hotel
Mixbook
cloudint.com/careers
YAMMER-JOBS
Keas.com : social + games
@keas improve health
Academia.edu
-engineers + interns
-Rails -other cool stuff
-beach balls
Check out /hiring!
Modcloth.com
-Senior SWE SF + Pitts'gh
-Rails Eng
-Test Eng
Rails Dev
Linux Admins
Windows Admins
UX
Tapjoy
Rails + js
2 kegerators!
400k rpms
Square
Come say hi around the corner
yammer hiring ruby!!
WILDFIRE
YAMMER
MANILLA
JOBS@MANILLA.COM
YAMMER!!
YAMMER JOBS!!
Dailykos.com
-Help us power the leading Progressive blogging site to vaile
Looking for people who like
-Rails
-Ruby
-Sys Ops
-Chef
Full Relocation
Coolest place in EU.
Citivox
CITIZEN
New Relic is hiring
Tech
airbnb

I CAN HAZ JOB?

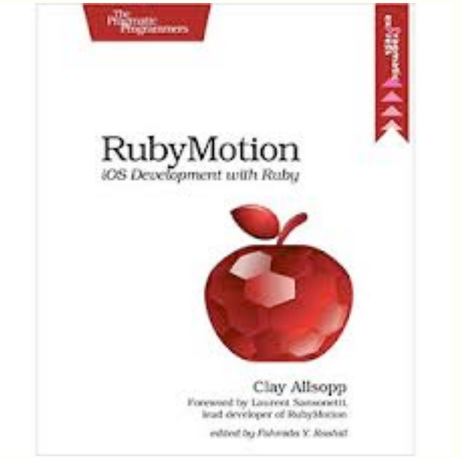
Ruby is in big companies



at&t

Ruby is in the cloud

Ruby is everywhere



MRuby

Ruby is above and beyond

Rails is ...

Rails is omakase

By David Heinemeier Hansson on Dec 27, 2012

Video
http://www.youtube.com/v/E99FnoYqoII?end=129&version=3

Rails is ola ke ase?


CHALUPA
Default
Stack


My
Stack


The SF Burrito


SOA Taquiza!

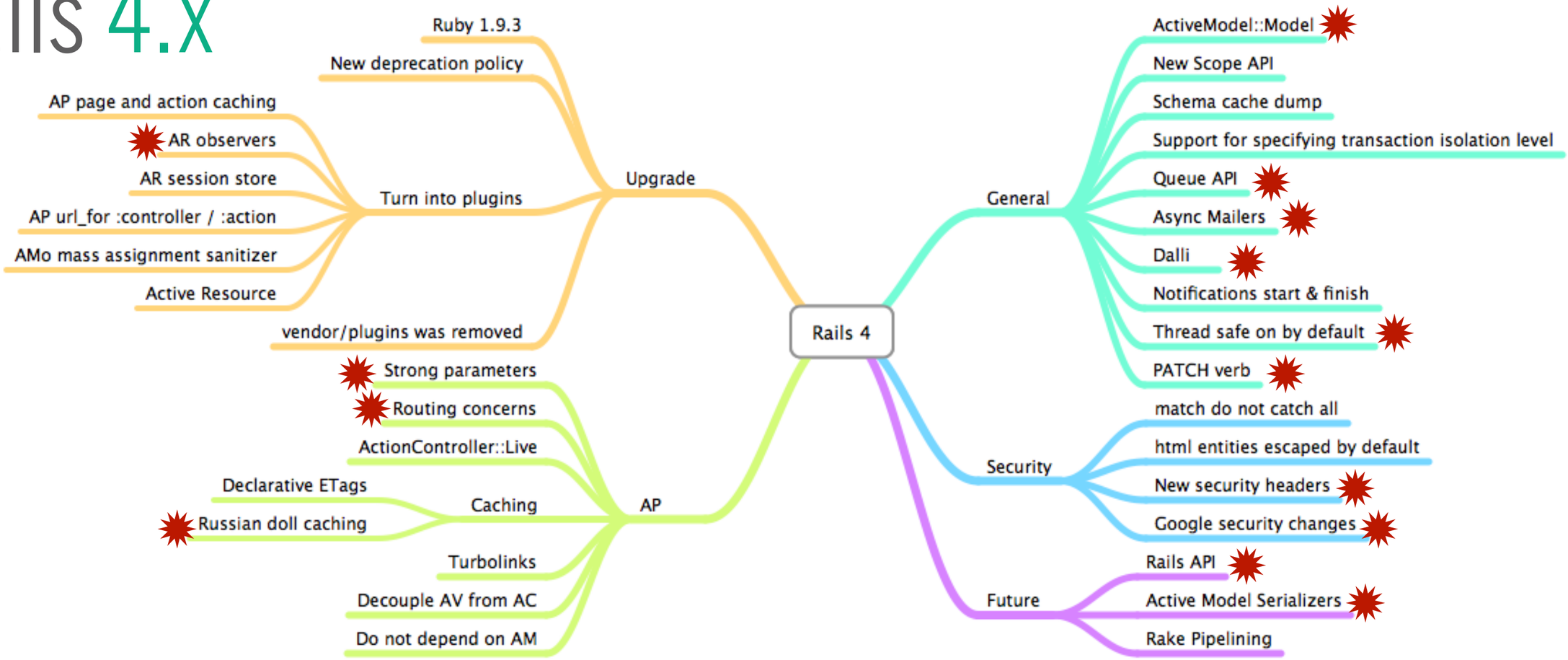

Now lets get
a closer look

Ruby 2.0

- Its all about scale
- A brand new GC
- Great performance increase
- Some new syntax sparkles
- API compatible with 1.9

```ruby
def some_method(x: 10, y: 20, z: 30)
  p x, y, z
end

some_method x: 1, y: 2, z: 3
```

Rails 4.x


Rails 4
Upgrade
Ruby 1.9.3
New deprecation policy
Turn into plugins
AP page and action caching
AR observers
AR session store
AP url_for :controller / :action
AMo mass assignment sanitizer
Active Resource
vendor/plugins was removed
AP
Strong parameters
Routing concerns
ActionController::Live
Caching
Declarative ETags
Russian doll caching
Turbolinks
Decouple AV from AC
Do not depend on AM
General
ActiveModel::Model
New Scope API
Schema cache dump
Support for specifying transaction isolation level
Queue API
Async Mailers
Dalli
Notifications start & finish
Thread safe on by default
PATCH verb
Security
match do not catch all
html entities escaped by default
New security headers
Google security changes
Future
Rails API
Active Model Serializers
Rake Pipelining

SOA

```ruby
class PeopleController < ActionController::Base
  # This will raise an ActiveModel::ForbiddenAttributes exception because it's using mass assignment
  # without an explicit permit step.
  def create
    Person.create(params[:person])
  end

  # This will pass with flying colors as long as there's a person key in the parameters, otherwise
  # it'll raise a ActionController::MissingParameter exception, which will get caught by
  # ActionController::Base and turned into that 400 Bad Request reply.
  def update
    person = current_account.people.find(params[:id])
    person.update_attributes!(person_params)
    redirect_to person
  end

  private
    # Using a private method to encapsulate the permissible parameters is just a good pattern
    # since you'll be able to reuse the same permit list between create and update. Also, you
    # can specialize this method with per-user checking of permissible attributes.
    def person_params
      params.require(:person).permit(:name, :age)
    end
end
```

Strong Parameters

Routing Concerns

```
BCX::Application.routes.draw do
  concern :commentable do
    resources :comments
  end

  resources :messages, :forwards, :uploads, :documents, :todos, concerns: :commentable
end
```


Tapa de telera, 110 kcal
Tamal de Rojo, 210 kcal
Contratapa de telera, 110 kcal

Russian Doll (Torta de tamal) catching

```ruby
class Team < ActiveRecord::Base
  has_many :members
end

class Member < ActiveRecord::Base
  belongs_to :team, touch: true
end
```

```ruby
# app/views/v1/teams/show.rabl
cache @team
attributes :name
child(:members => :categories) do
  extends "v1/categories/product"
end

# app/views/v1/members/member.rabl
cache member
attributes :name, :bio
```

SOA also applies to code

- PORO
- Classes are your friends
- Single responsibility principle
- Resilient to change
- Easy to test

PORO

```ruby
class ApplicationUserCreator
  def initialize(welcome_mailer=nil)
    @welcome_mailer = welcome_mailer
  end

  def create_new_user(params)
    User.create(params).tap { |new_user|
      if new_user.valid?
        self.welcome_mailer.deliver_welcome_email(new_user)
      end
    }
  end

private

  def welcome_mailer
    @welcome_mailer ||= UserMailer
  end
end
```

Single responsibility principle

```
class AdminUsersController < ApplicationController
  def create
    @user = ApplicationUserCreator.new(AdminUserMailer).create_new_user(params[:user])
  end
end
```

Skinny Models

- Keep callback logic separate (decouple)
- Extract business logic to classes
- Don't mix Authorization with validations

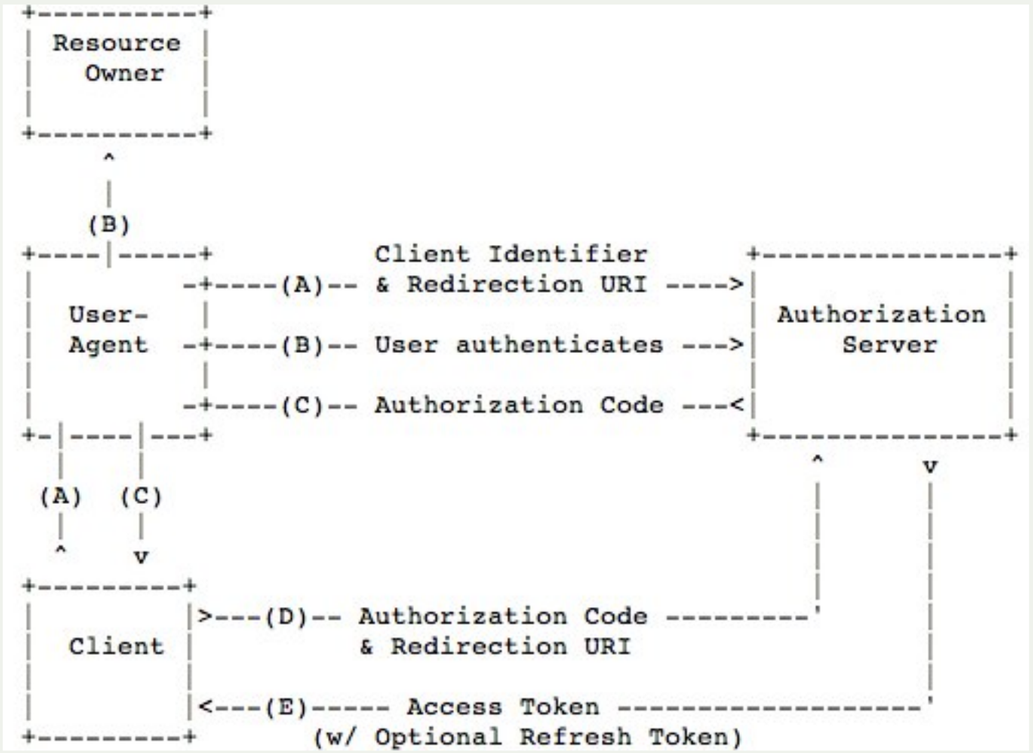
http://blog.codeclimate.com/blog/2012/10/17/7-ways-to-decompose-fat-activerecord-models/

```ruby
class Post < ActiveRecord::Base
  after_save PostCallbacks.new
end

class PostCallbacks
  def after_save(post)
    Rails.queue.push(PostNotification.new(post_id,'save'))
  end
end

class PostNotification
  def initialize(post_id, action)
     @post   = Post.find(post_id)
     @action = action
  end

  def run
    case @action
    when 'create'
      ...
    when 'save'
      ...
    end
  end
end
```

Authentication as a middleware

- Rollout your own
- Warden rocks!
- Disable session based auth
- Share across your services
- Oauth 2.0

https://github.com/kidpollo/warden-oauth2



```ruby
require 'grape'
require 'warden-oauth2'

class MyAPI < Grape::API
  use Warden::Manager do |config|
    strategies.add :bearer, Warden::OAuth2::Strategies::Bearer
    strategies.add :client, Warden::OAuth2::Strategies::Client
    strategies.add :public, Warden::OAuth2::Strategies::Public

    config.default_strategies :bearer, :client, :public
    config.failure_app Warden::OAuth2::FailureApp
  end

  helpers do
    def warden; env['warden'] end
  end

  resources :hamburgers do
    before do
      warden.authenticate! scope: :hamburgers
    end
  end
end
```

Authorization that does not suck!

- Rollout your own
- Use plain classes
- Pundit rocks!
- Don't mix with validations
- Share across services

https://github.com/elabs/pundit

```
class PostPolicy
  attr_reader :user, :post

  def initialize(user, post)
    @user = user
    @post = post
  end

  def create?
    user.admin? or not post.published?
  end
end
```

Whats wrong?

```
class Post
  validate :is_admin

  def is_admin
    errors.add(:base, 'cant post') if user.admin?
  end
end
```

Versioning from the start

- Catching
- Documentation
- Lifecycle
- Dependencies

```
Service::Application.routes.draw do
  namespace :v1 do
    resources :posts, only: [:index, :show, :update]
  end
end
```

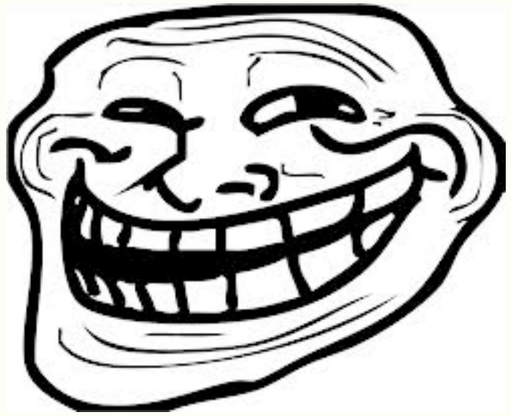
Measure everything

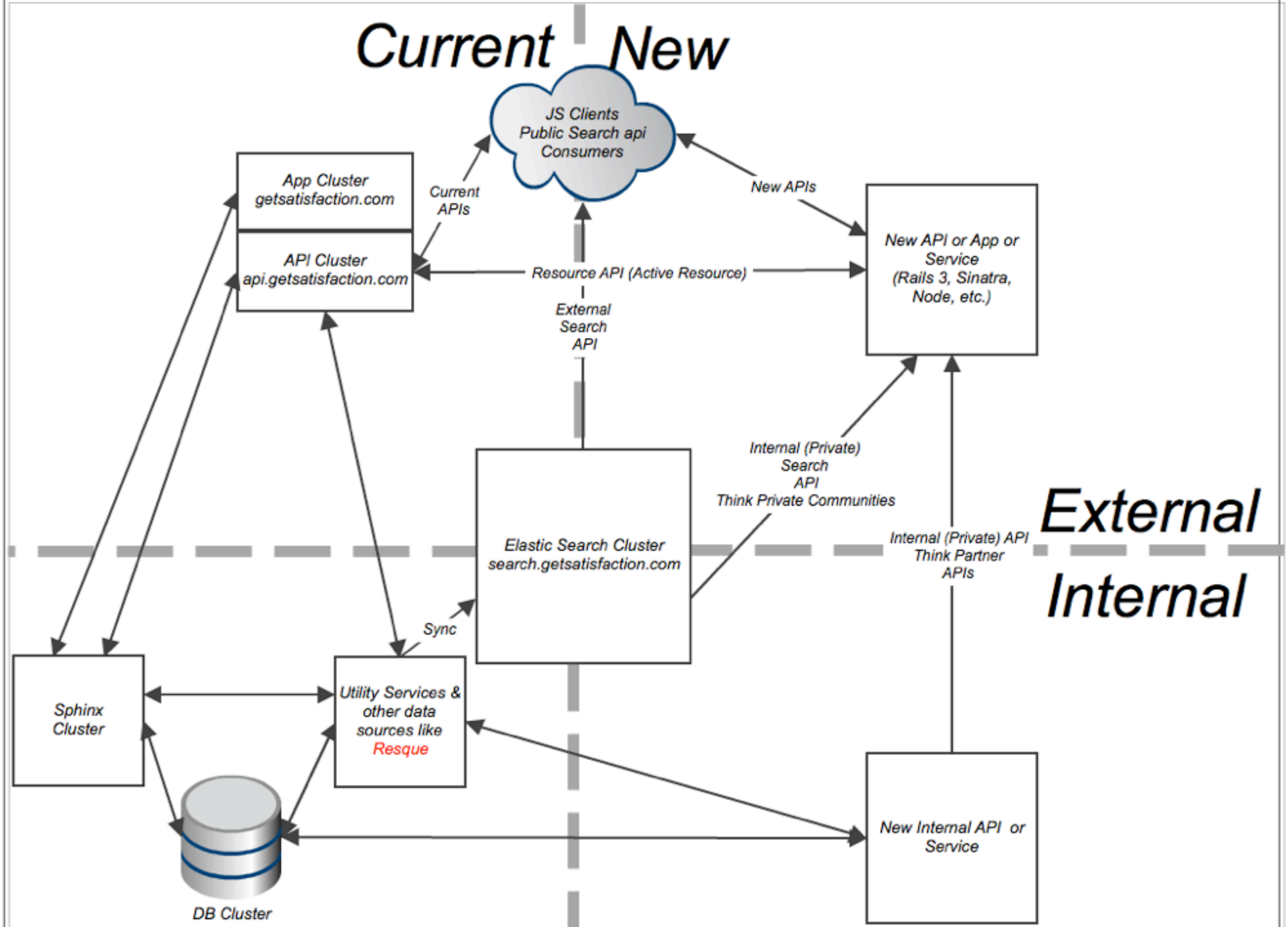
- Usage
- Performance
- Trace
- Logs

```
ActiveSupport::Notifications.subscribe do |name, start, finish, id, payload|
  Rails.logger.debug(["notification:", name, start, finish, id, payload].join(" "))
end
```

http://asciicasts.com/episodes/249-notifications-in-rails-3

Now make your own SOA diagram




Current
New
JS Clients
Public Search api
Consumers
App Cluster
getsatisfaction.com
API Cluster
api.getsatisfaction.com
Current
APIs
New APIs
Resource API (Active Resource)
External
Search
API
New API or App or
Service
(Rails 3, Sinatra,
Node, etc.)
Internal (Private)
Search
API
Think Private Communities
Internal (Private) API
Think Partner
APIs
External
Internal
Elastic Search Cluster
search.getsatisfaction.com
Sync
Sphinx
Cluster
Utility Services &
other data
sources like
Resque
DB Cluster
New Internal API or
Service

FIN

Thanks!

@kidpollo

http://www.linkedin.com/in/fviramontes