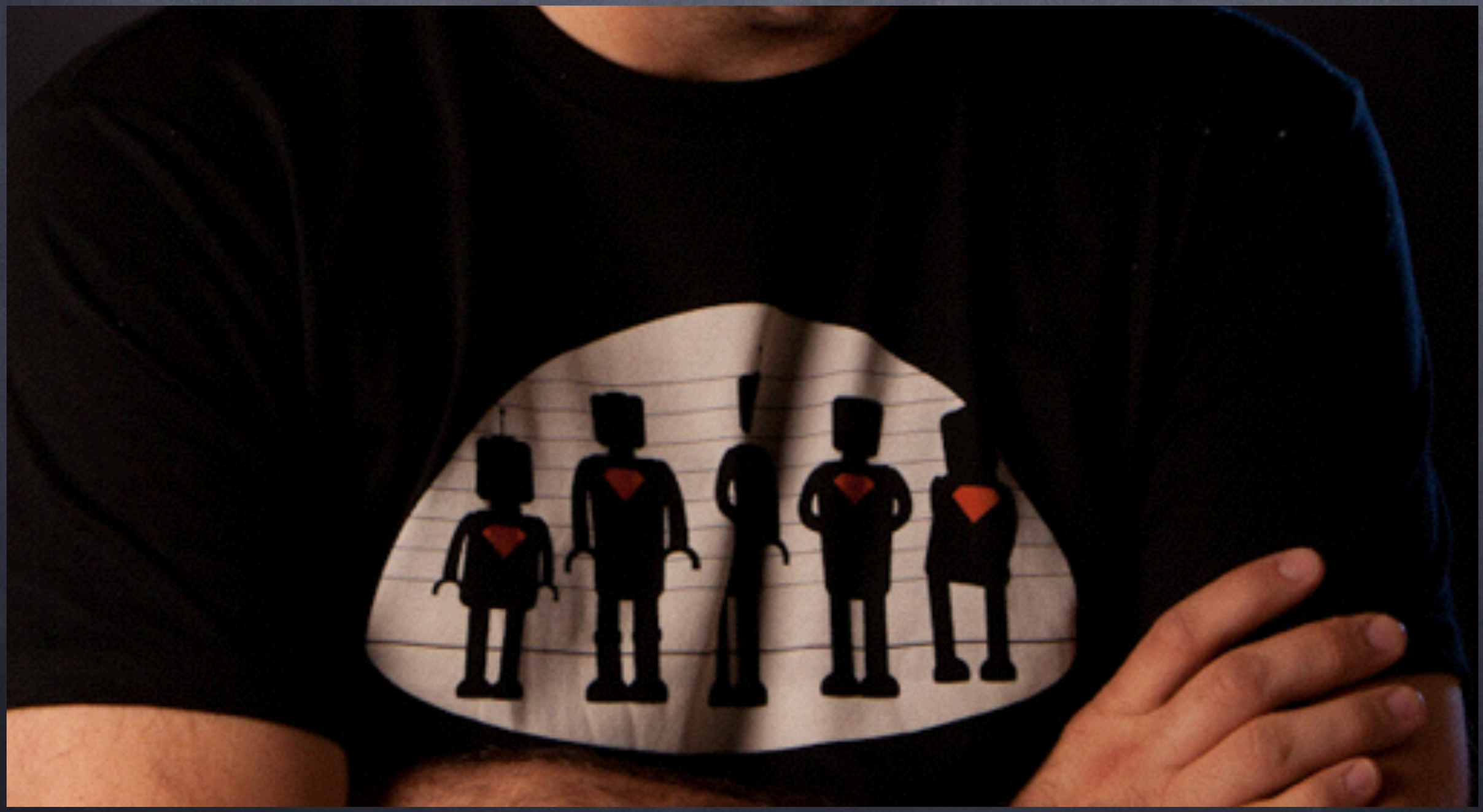
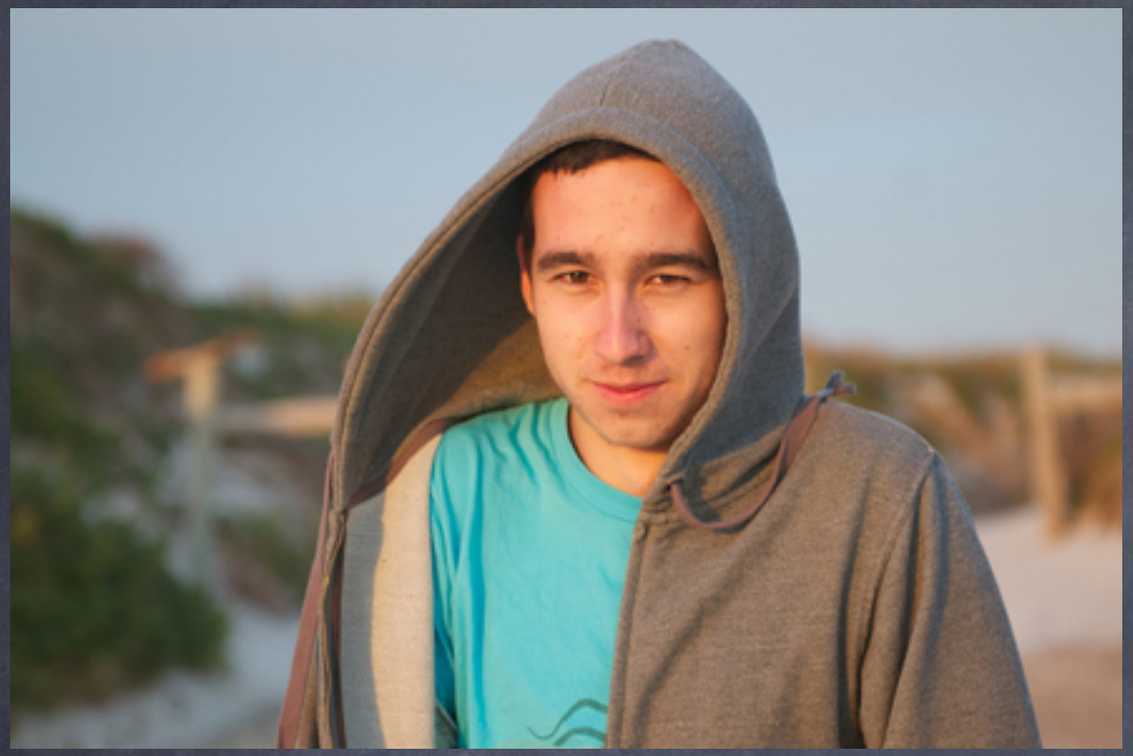
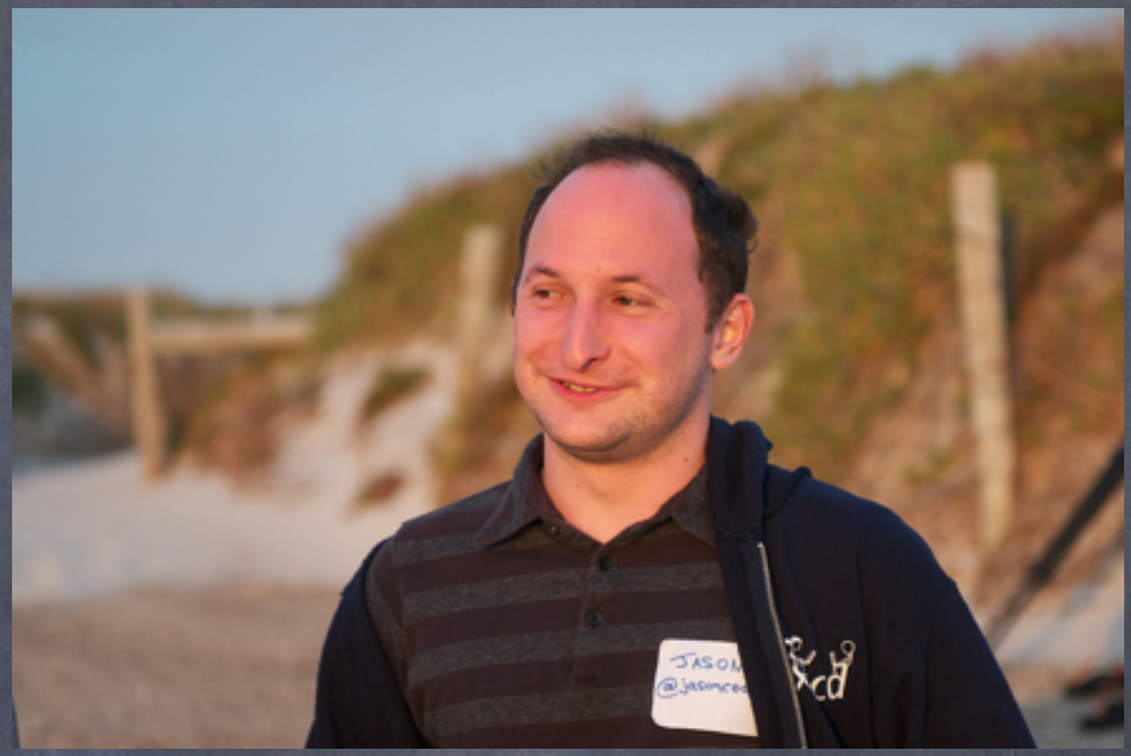
Dragonfly

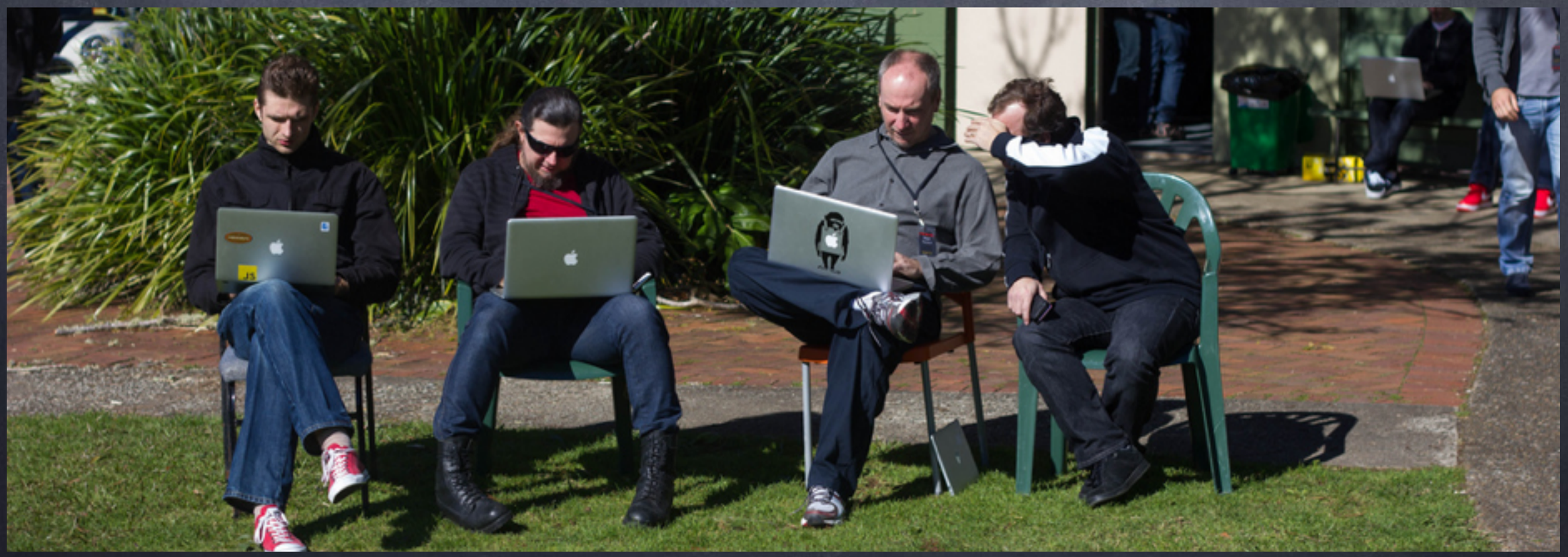
June 2012

IndieCrews



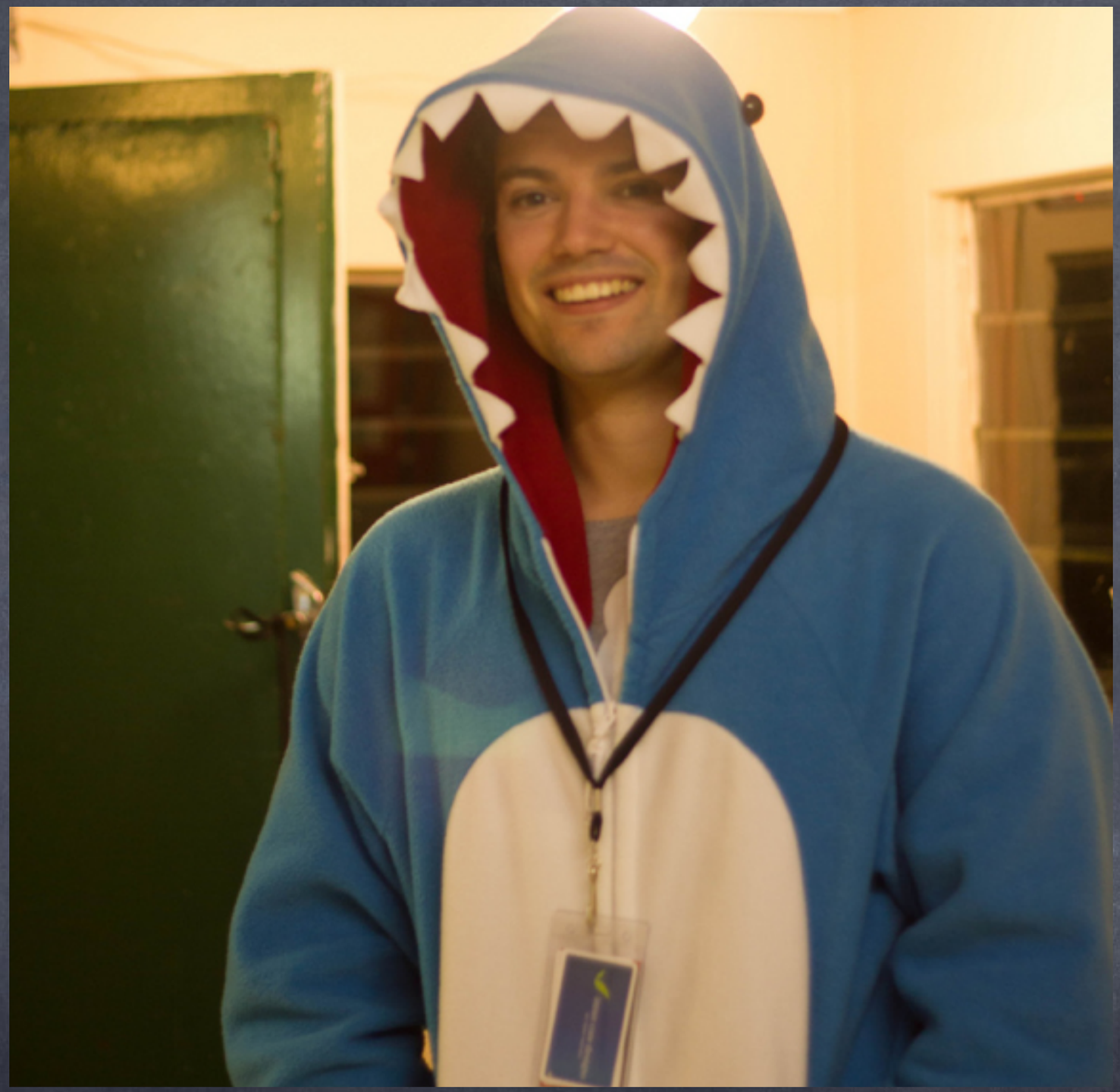



JASON
@jasoncu



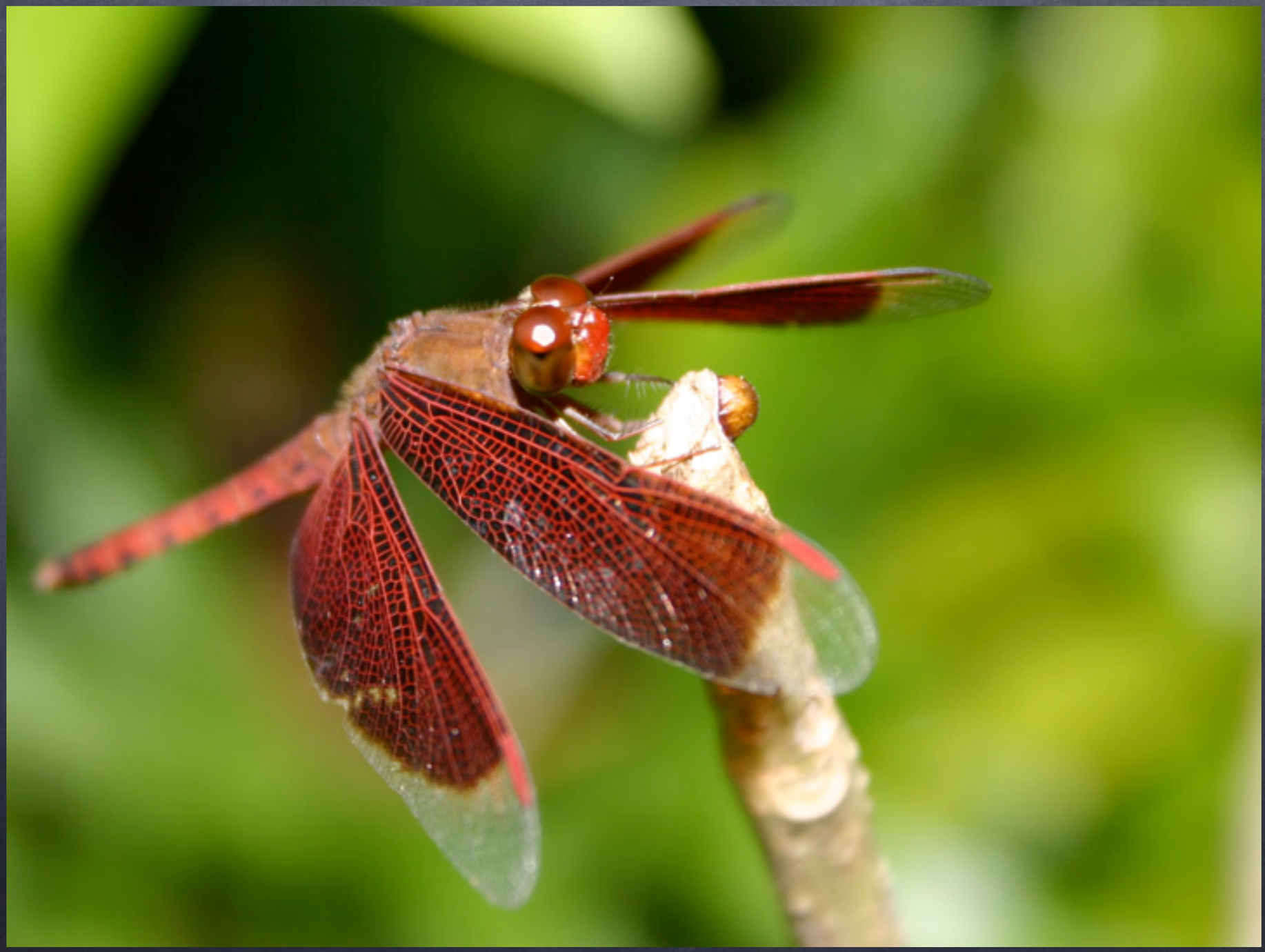
Paperclip

- https://github.com/thoughtbot/paperclip



Dinosaur / Dragon ?

Dragonfly



CarrierWave

Anyone?

GitHub stats

- CarrierWave - 3046 watchers, 390 forks
- Paperclip - 4294 watchers, 877 forks
- Dragonfly - 896 watchers, 99 forks

Dragonfly is

- a Rack framework for on-the-fly image handling.
- ideal for using with Ruby on Rails (2 & 3), Sinatra etc
- not just for Rails and not just for images

Prons

- Native mongomapper
- Generates thumbs on the fly
- Only one file stored
- Good documentation

Cons

- Uses a lot of CPU if you don't have cache enabled

Caching

- Pretty much required (Rack::Cache by default)
- Sends Cache-Control and Etag
- Use any proxy: Varnish, Squid, Rack::Cache, etc
- Control cache

```
app.cache_duration = 3600*24*365*3  # time in seconds
```

Chaining & Lazy

- Chain methods
- Generating urls without creating image

```
image = app.fetch('some_uid').process(:greyscale).process(:thumb, '40x20#').encode(:gif)

image.apply                  # actually 'does' the processing and returns self

app.fetch('some_uid').process(:thumb, '40x20#').encode(:gif).url
```

ImageMagic

```
image.thumb('40x30')                    # same as image.process(:thumb, '40x30')
image.jpg                               # same as image.encode(:jpg)
image.png                               # same as image.encode(:png)
image.gif                               # same as image.encode(:gif)
image.strip                             # same as image.process(:strip)
image.convert('-scale 30x30')           # same as image.process(:convert, '-scale 30x30')
```

```
image.process(:crop, :width => 40, :height => 50, :x => 20, :y => 30)
image.process(:crop, :width => 40, :height => 50, :gravity => 'ne')

image.process(:flip)                            # flips it vertically
image.process(:flop)                            # flips it horizontally

image.process(:greyscale, :depth => 128)        # default depth 256

image.process(:resize, '40x40')
image.process(:resize_and_crop, :width => 40, :height=> 50, :gravity => 'ne')

image.process(:rotate, 45, :background_colour => 'transparent')   # default bg
black
```

Up-front thumbnailing

```
class Person
  image_accessor :mugshot do
    copy_to(:smaller_mugshot){|a| a.thumb('200x200#') }
  end
  image_accessor :smaller_mugshot
end

person.mugshot = Pathname.new('some/400x300/image.png')

person.mugshot              # ---> 400x300 image
person.smaller_mugshot      # ---> 200x200 image
```

Paperclip alternative

```
class User < ActiveRecord::Base
  has_attached_file :avatar, :styles => { :medium => "300x300>",
                             :thumb => "100x100>" }
end
```

```
<%= image_tag @user.avatar.url %>
<%= image_tag @user.avatar.url(:medium) %>
<%= image_tag @user.avatar.url(:thumb) %>
```

Validation

```
class Album

  validates_presence_of :cover_image
  validates_size_of :cover_image, :maximum => 500.kilobytes

  validates_property :format, :of => :cover_image, :in => [:jpeg, :png, :gif]
  # ..or..
  validates_property :mime_type, :of => :cover_image,
                                 :as => 'image/jpeg', :case_sensitive => false

  validates_property :width, :of => :cover_image, :in => (0..400),
                             :message => "é demais cara!"

  # ...
end
```

Meta data

```ruby
@album.cover_image = File.new('path/to/my_image.png')
@album.cover_image.meta = {:taken => Date.yesterday}
@album.save!

@album.cover_image.meta       # => {:model_class=>"Album",
                              #     :model_attachment=>:cover_image,
                              #     :taken=>Sat, 11 Sep 2010}
```

Serving Processed Content "on-the-fly"

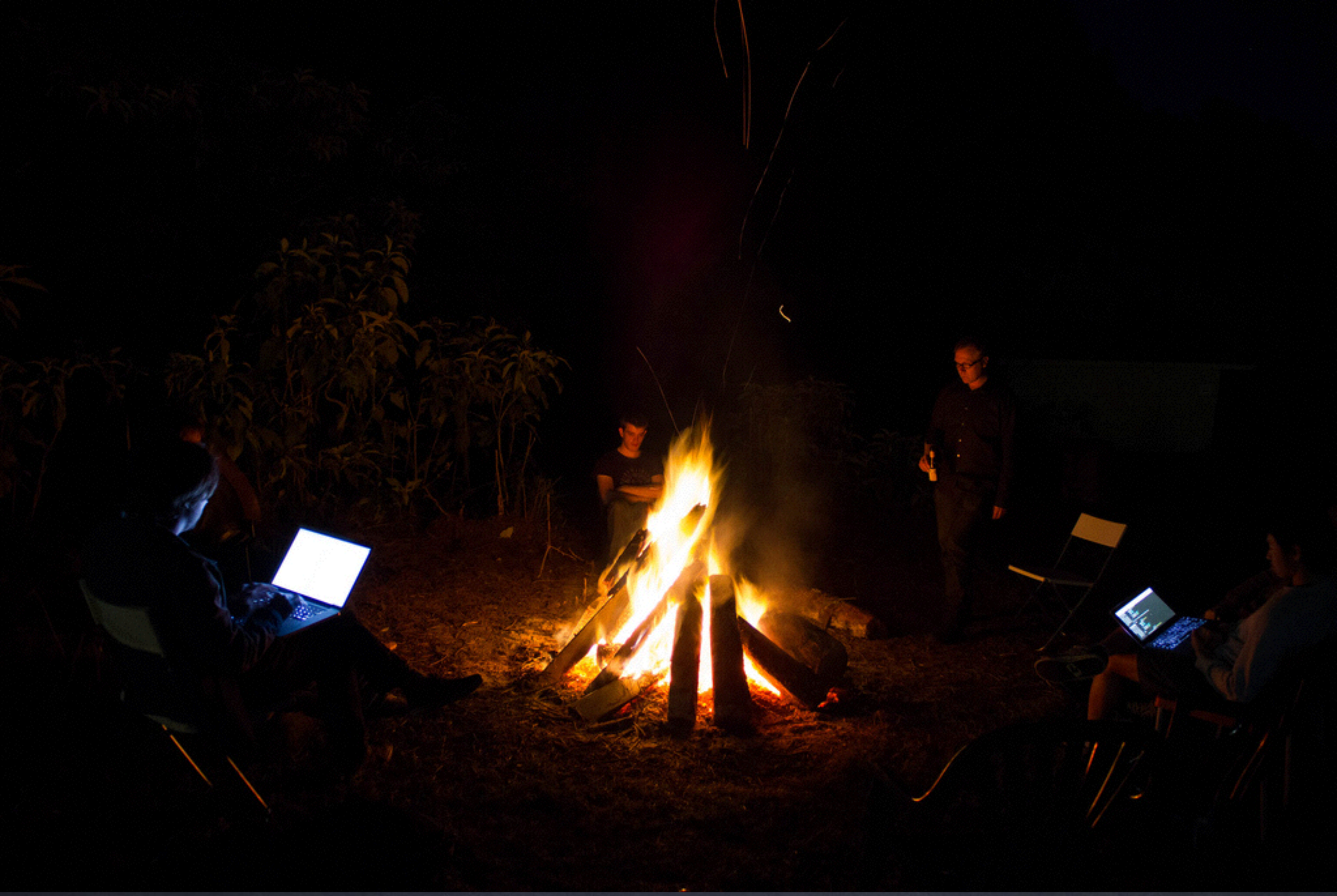
- It is even possible to store processed versions of content remotely
- Example: store every served image on Amazon S3 and serve it from there instead

http://markevans.github.com/dragonfly/file.ServingRemotely.html

Links

- https://github.com/thoughtbot/paperclip
- https://github.com/markevans/dragonfly
- https://github.com/ jnicklas/carrierwave

Conclusion



RailsCamp is awesome