

# Scaling Rails: a Case Study

Kennon Ballou  
BarCamp Leipzig 2008

# \*eezy.com

- Sites I developed on contract
- Brusheezy.com, Vecteezy.com, Fresheezy.com, Flasheezy.com
- Place for artists to find and submit photoshop brushes / vector files / themes / flash widgets
- Initial development time: ~20 hours (brusheezy)

# Plugins used

- `acts_as_authenticated`
- `acts_as_ferret`
- `attachment_fu` - s3 integration
- `paginating_find`
- `simple_captcha`
- `acts_as_ratable`

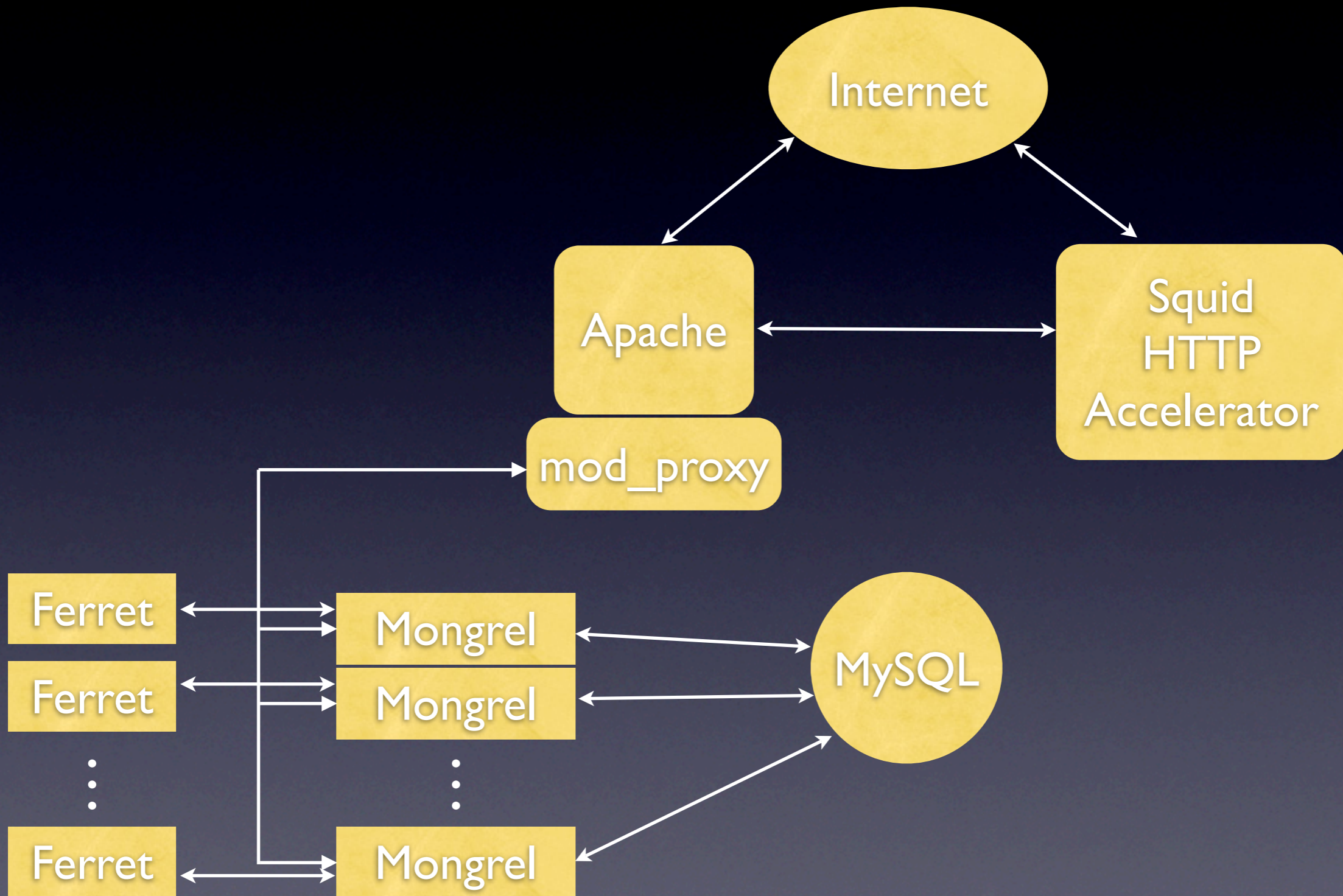
# Daily traffic

- Brusheezy: ~20k visitors, ~180k pageviews
- Vecteezy: ~18k visitors, ~144k pageviews
- Fresheezy: ~10k visitors, ~90k pageviews
- Flasheezy: ~8k visitors, ~80k pageviews

# Vecteezy digg hit

- On May 18th 2007, Vecteezy made it to the front page of Digg.com (3041 diggs)
- 7 hours of pure server hell (pre-optimized)
- Peak: 20+ new visitors/second (no clue how many pageviews, stopped counting)
- Sites slowed down, but never crashed

# Architecture



# Scaling Tips

- MySQL
  - Turning off binary logging
  - Changing cache sizes, etc (find a tutorial)
- Apache
  - MaxClients
  - Timeout settings
  - Watch server-status

# Scaling Tips cont.

- Squid HTTP Acceleration
  - == AWESOME for static content
  - ~5 minutes to set up
  - `config.action_controller.asset_host` in `config/environments`
  - for \*eezy, ~95% cache hit ratio
  - 2nd IP address on server

# Scaling Tips cont.

- Caching
  - Page cache front page
  - Fragment cache as much as possible
  - /Fairly/ easy to set up, annoying to update
- Watch logs
  - `tail -f log/production | grep Completed`

# Scaling Tips cont.

- Hide slow features behind ajax
- Compress your javascript/css
- \* add servers: horizontal scaling
- \* use memcache
- \* use nginx instead of apache

# What I hate about Rails

- Core community attitudes
- Convention over configuration ...
- Lack of authoritative 'meta' documentation
- Deployment not super easy
- Demanding server requirements (installed software - gems, etc)

# What I love about Rails

(and why I'll never go back to PHP)

- Sooo fast to develop in
- Ruby is **fun** - very powerful and elegant (never want to see a \$variable again)
- Rails is getting better all the time
- Many many wonderful plugins
- Does 90% of the typical junk for you
- Easy to scale once you know how
- Great ajax integration - super easy

# Other/future possibilities

- Ruby 1.9
- Amazon EC2
- Phusion Passenger (mod\_rails)

# Thank you!

- Questions/comments/suggestions?