

# Scaling Continuous Deployment @ Etsy

Avleen Vig

Staff Operations Engineer

@avleen

With much credit: Daniel Schauenberg: (@mrtazz)

# Statistics

— OVER —  
**25**  
— million —  
**MEMBERS**

**18M**  
**ITEMS**  
— listed —

**60**  
MILLION  
— MONTHLY —  
**VISITORS**

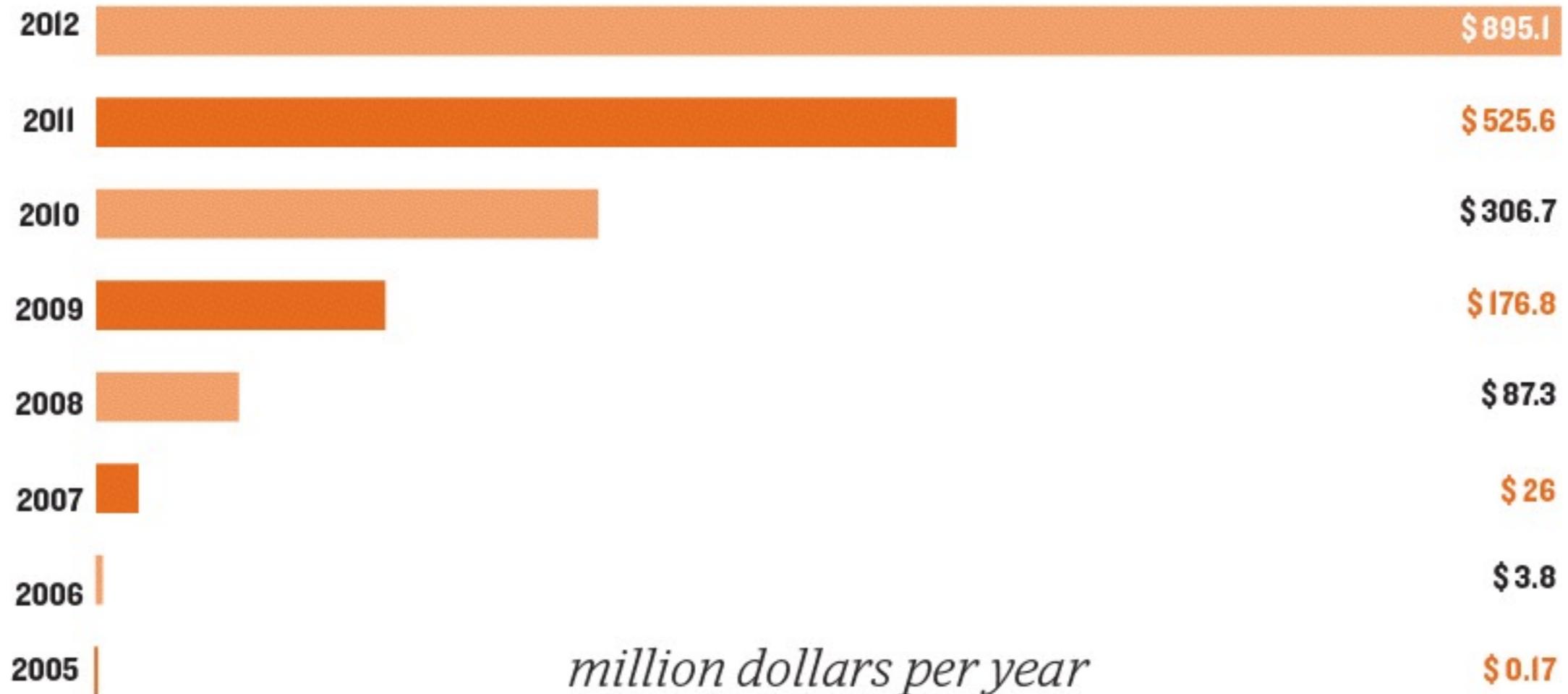
**1.5**  
BILLION  
**PAGEVIEWS**  
— per month —

**2M**  
VISITORS  
TO THE  
**BLOG**  
— per month —

**1 300 000**  
Facebook fans  
  
  
**1 800 000**  
Twitter followers

# Statistics

## SALES GROWTH



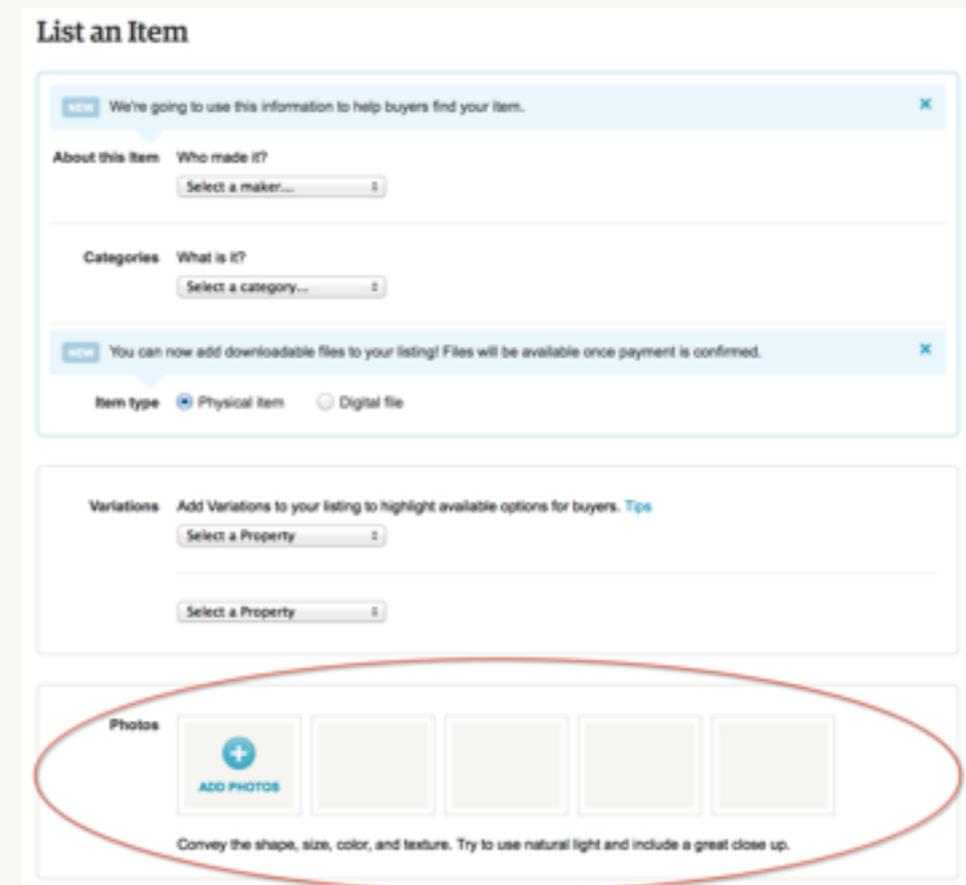
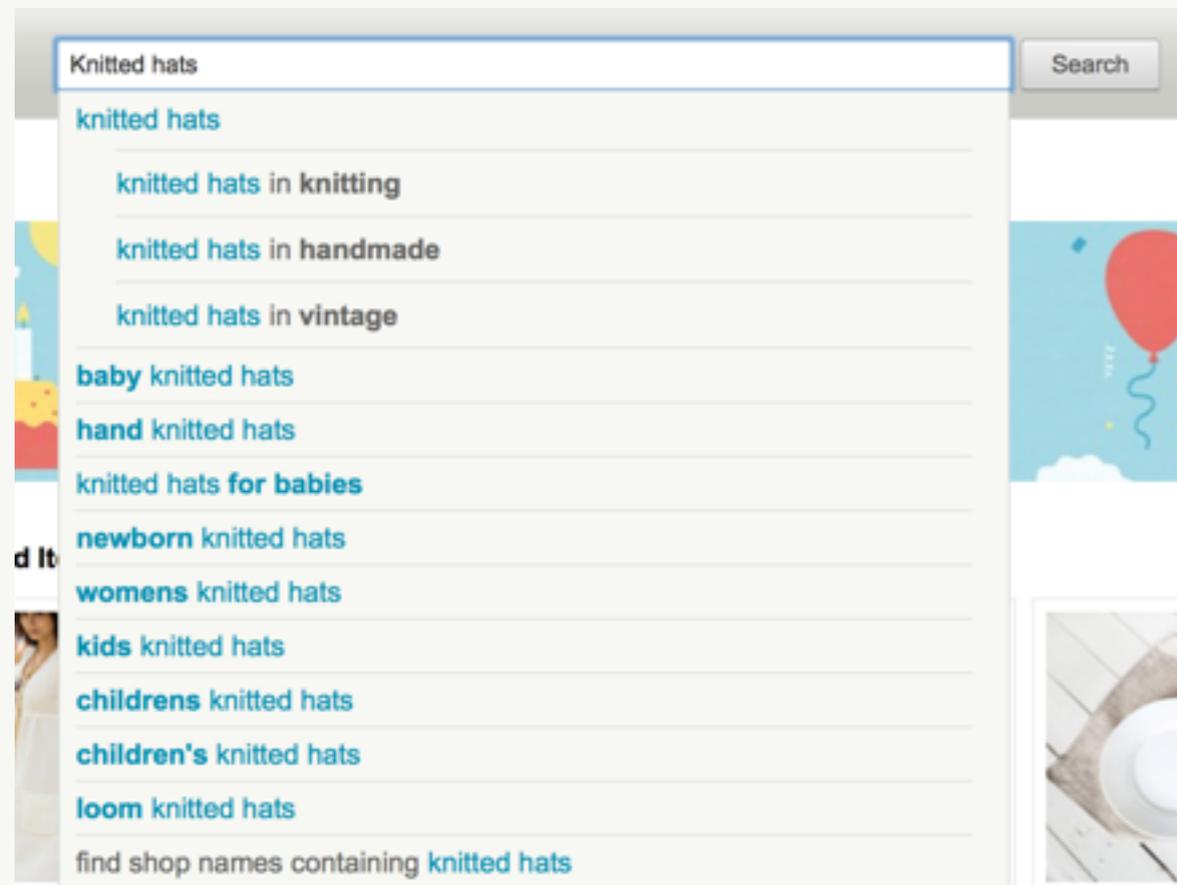
# Our application

Mostly monolithic



# Our application

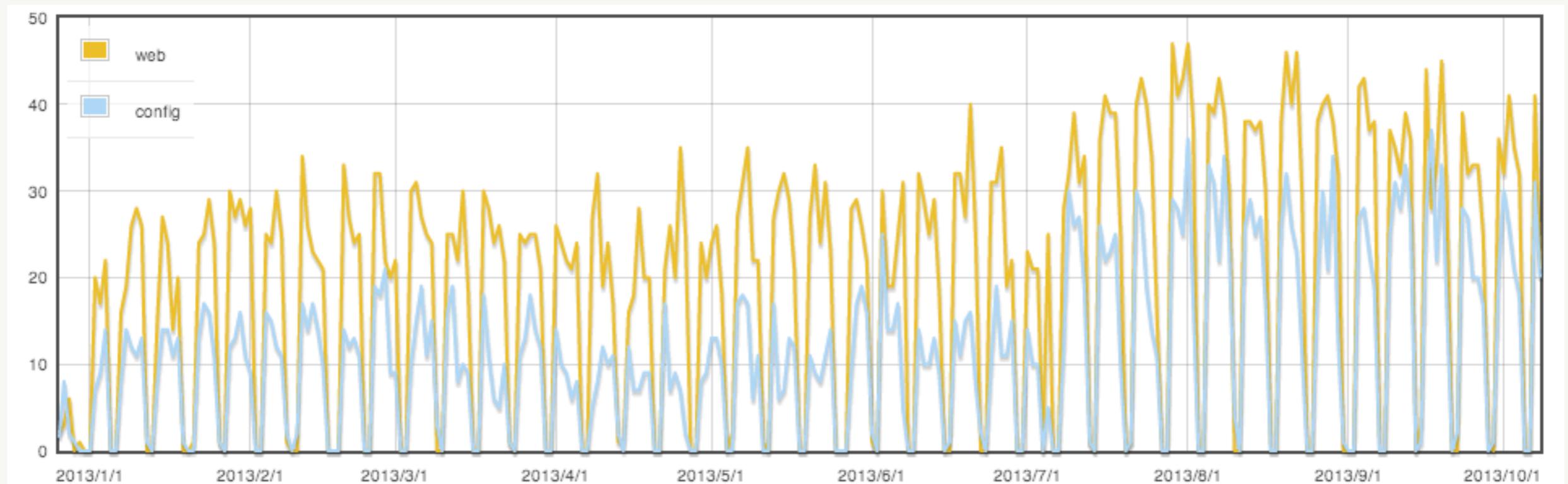
A few services too



 Gearman

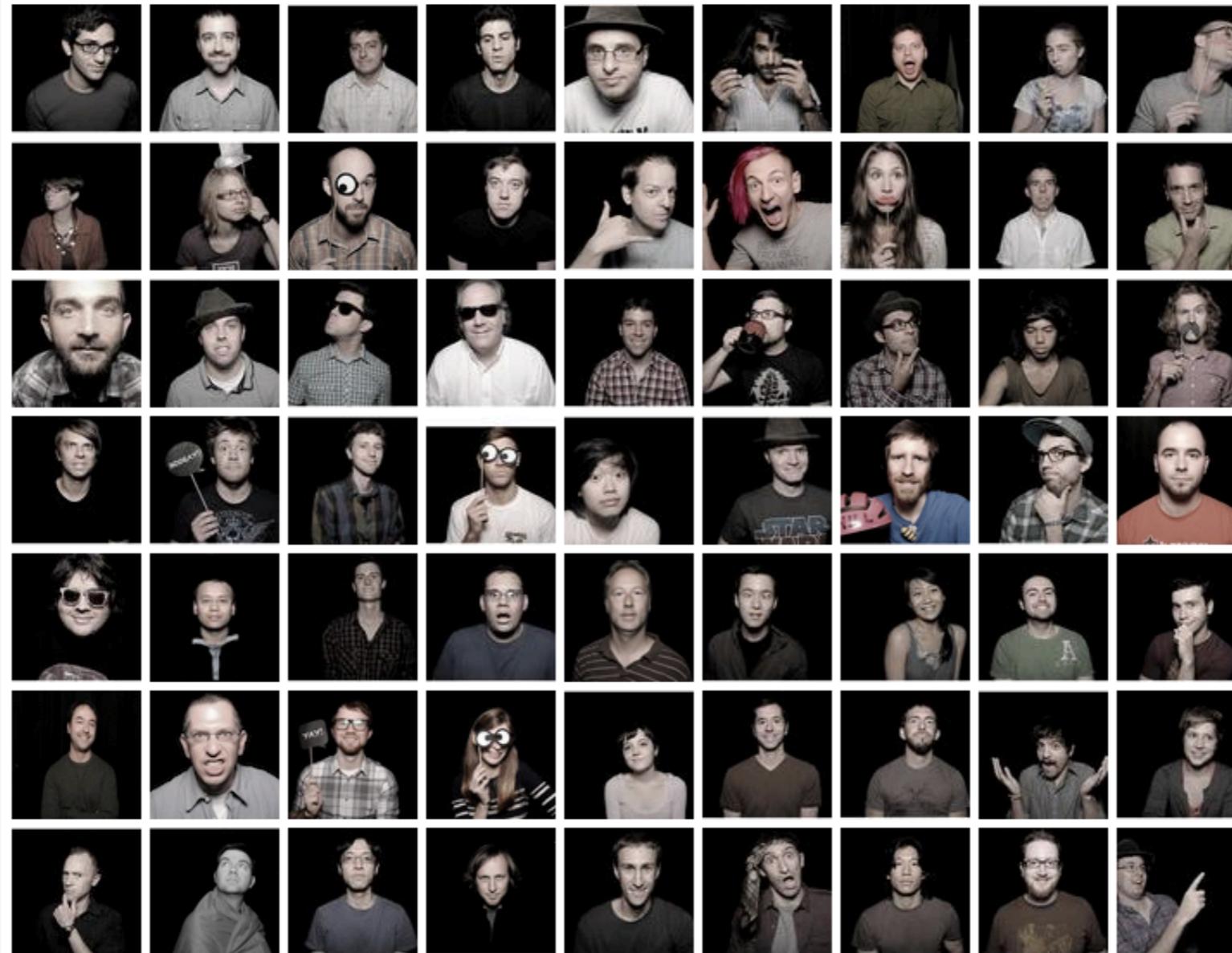
# Our application

## Deploy frequency



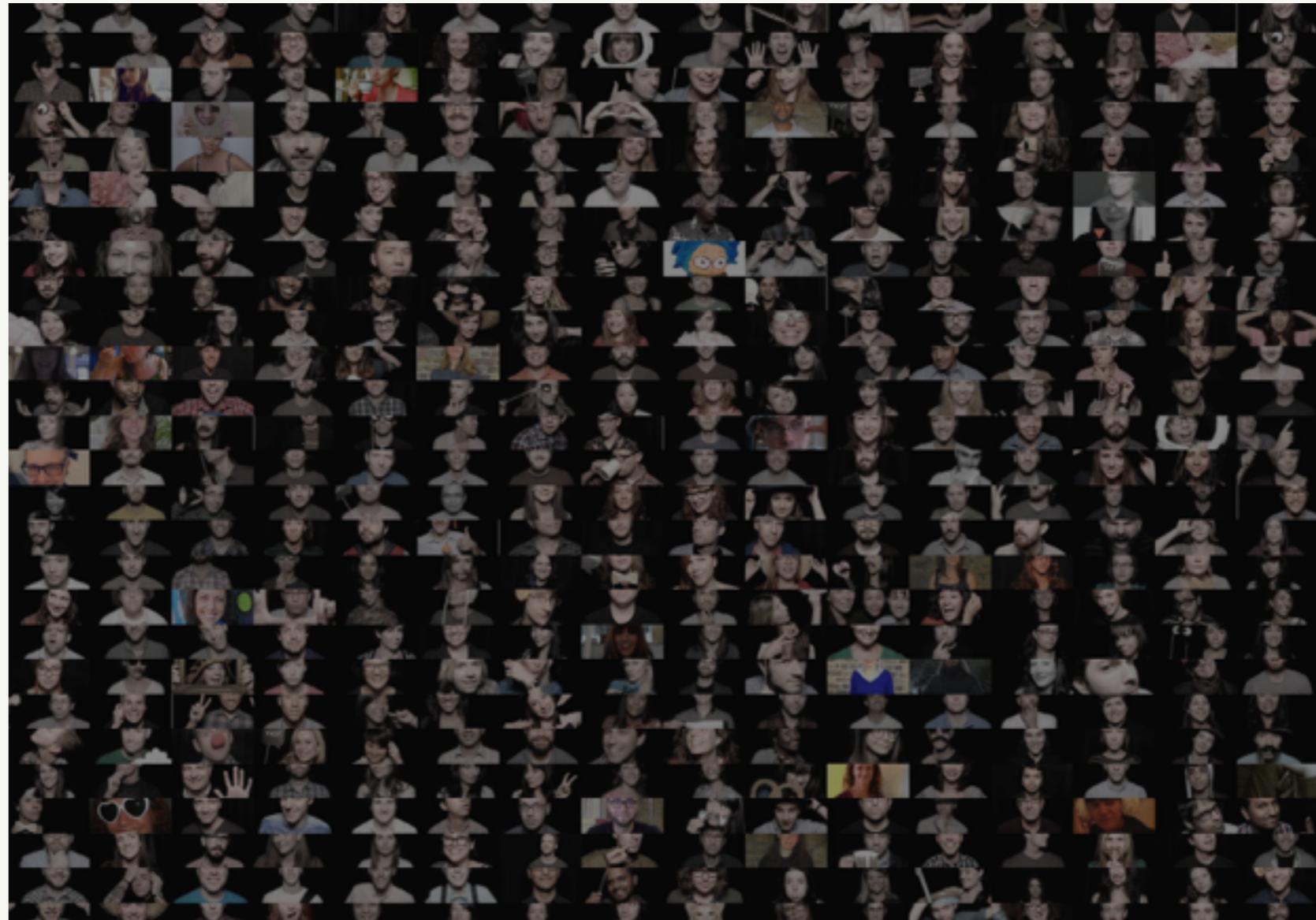
# Our team

Before..



# Our team

Today . .



. . and that's just a fraction!

# Deploying code

The push train



Item by decomodwalls

# Deploying code

#push

- IRC channel to organize push trains
- Join a train if you want to deploy changes
- Schedule is planned via the channel topic
- First in the train is the driver

# Deploying code

#push

<prod> kseever\* + jameslee | jpaul | avleen (c)

# Deploying code

#push

```
<prod> bateman* + krunal* + enorris* |  
tristan (c) + jameslee (c) + jlaster (c) | dawa +  
corey + sandosh + jklein + magera +  
seth_home + mpascual + nathan | bateman |  
russp (c)
```

# Deploying code

## Deployinator

#Push topic: Off-Hours!! | <prod> premsree

**ppillai**  
on prod  
6h since 7:59

**01. Save the Princess with Tests**  
17965d4 → 77ea0b9 

**Save the Princess with Tests**

Princess Log Watcher (supergrep)

**02. Deploy to Production**  
17965d4 → 17965d4

**Deploy to Production**

What to watch after a push  
Log Watcher (supergrep)  
Forums

**Deploy Logs**

Write an arbitrary log message

web\_config – 2013-04-13 23:59:49 | CONFIG | ppillai | CONFIG PRODUCTION Deploy

web\_config – 2013-04-13 23:56:19 | CONFIG | ppillai | CONFIG Princess Deploy: old e

web\_config – 2013-04-13 21:06:51 | CONFIG | cjee | CONFIG PRODUCTION Deploy: c

web\_config – 2013-04-13 21:02:10 | CONFIG | cjee | CONFIG Princess Deploy: old e0:

web – 2013-04-13 19:55:01 | PRODUCTION | rposluszny | Production Deploy: old 824

web – 2013-04-13 19:48:55 | PRINCESS | rposluszny | Princess Deploy: old: 8246182

web – 2013-04-12 23:02:30 | PRODUCTION | jlee | Production Deploy: old 71e6d6f, ne

web – 2013-04-12 22:51:58 | PRINCESS | jlee | Princess Deploy: old: 71e6d6f, new: 8:

web – 2013-04-12 22:37:32 | PRODUCTION | jlee | Production Deploy: old 4b9dd78, n

[https://github.com/  
etsy/deployinator](https://github.com/etsy/deployinator)

# Deploying code

So what's the problem?

# Deploying code

So what's the problem?

- Deploy-time requests are not atomic
- Weird limbo while syncing in-place
- Limits on pushes-per-day
- Long wait times

# Deploying code

Um, limits per day?

- $\frac{(\text{push\_queue\_hours} * 60)}{\text{minutes to deploy}}$
- At 15 mins/deploy, we get ~32 deploys per day – not enough!

# How can we scale it?

Our options:

- More code in each deploy
- Allow concurrent deploys
- Reduce deploy times
- Make deploys atomic
- Fork more concurrent rsyncs

# How can we scale it?

More code in each deploy:

- Also has limits
- How many people can be in each push?
- We found  $\sim 8$  to be our limit for reducing wait times

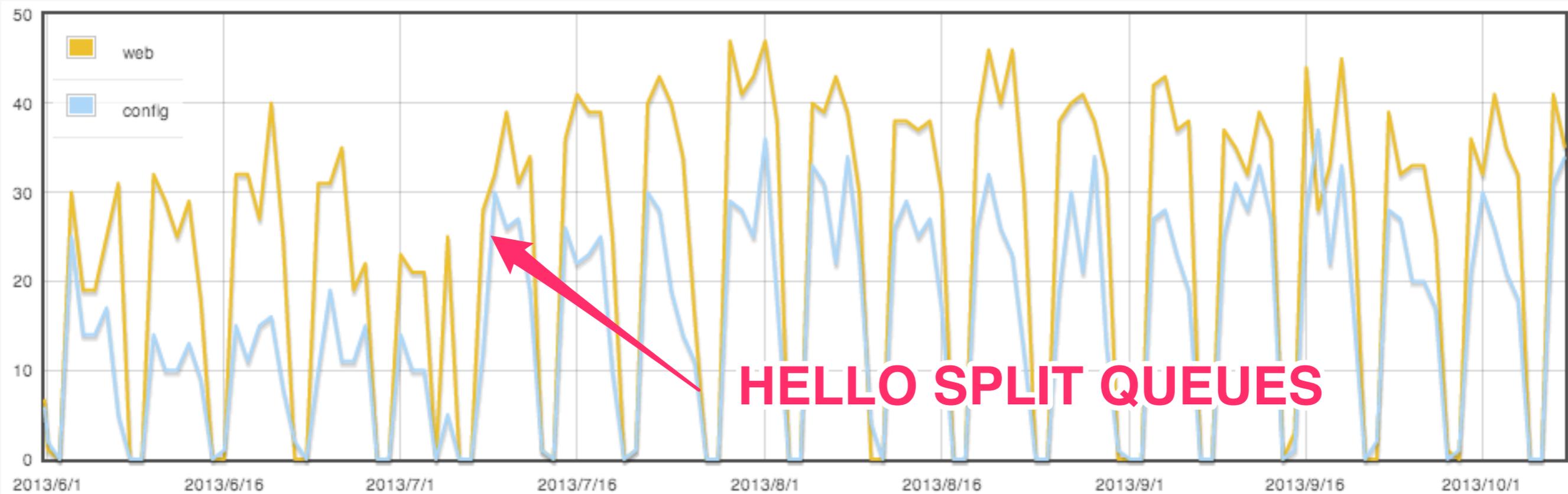
# How can we scale it?

Allow concurrent deploys:

- For config changes
- Code on independent systems
- The few services we have

# How can we scale it?

Concurrent deploys:



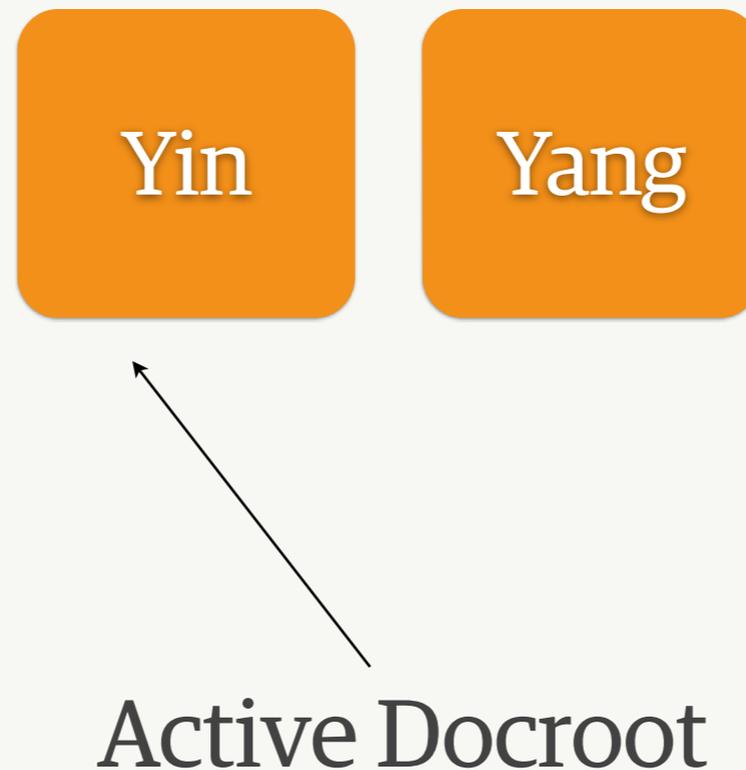
# How can we scale it?

Reduce deploy times:

- Tweaks around rsync
- Keep codebase in RAM (tmpfs)
- Increase rsync concurrency
- Reduce timeouts and retry intervals

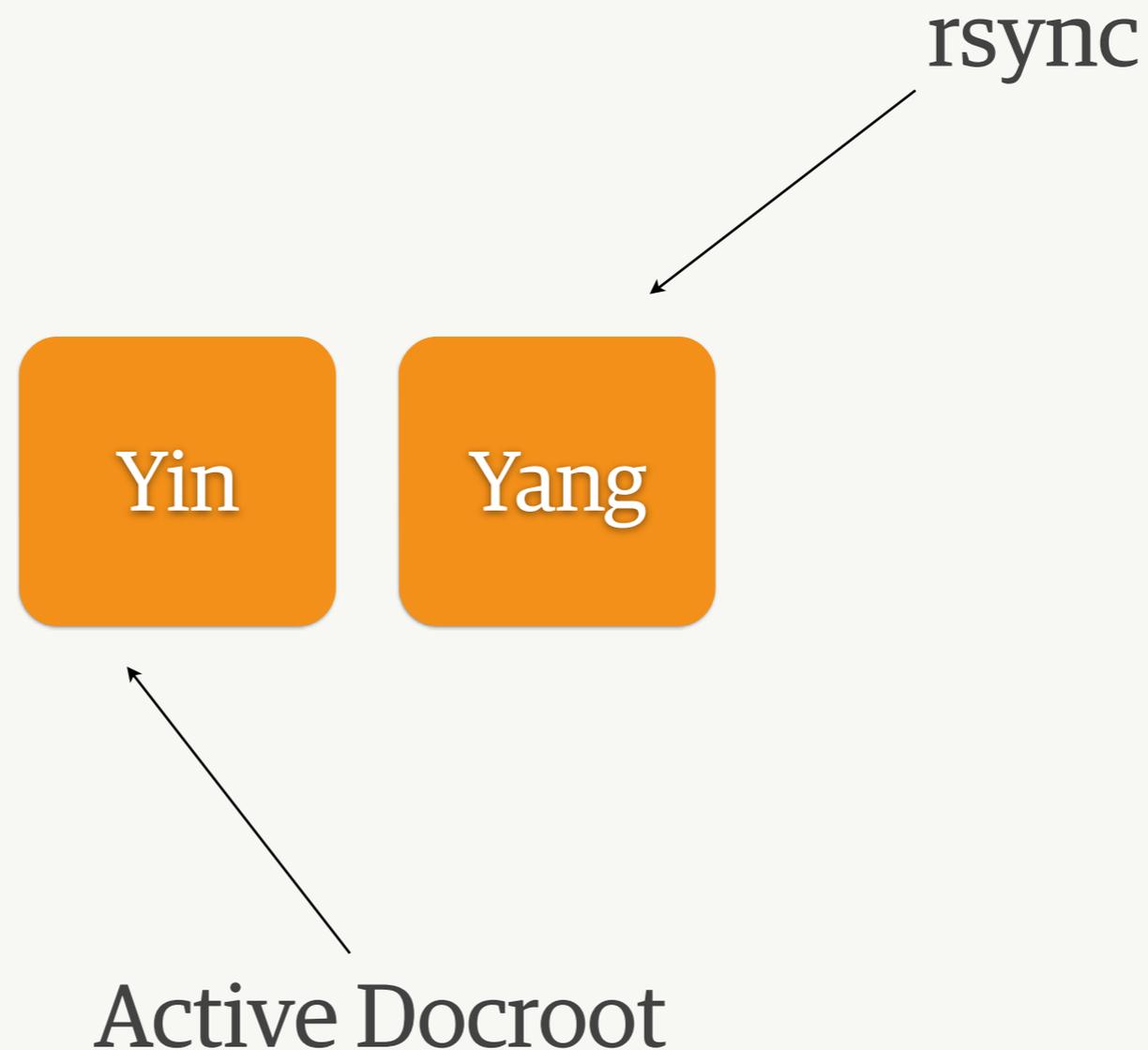
# How can we scale it?

Make deploys atomic:



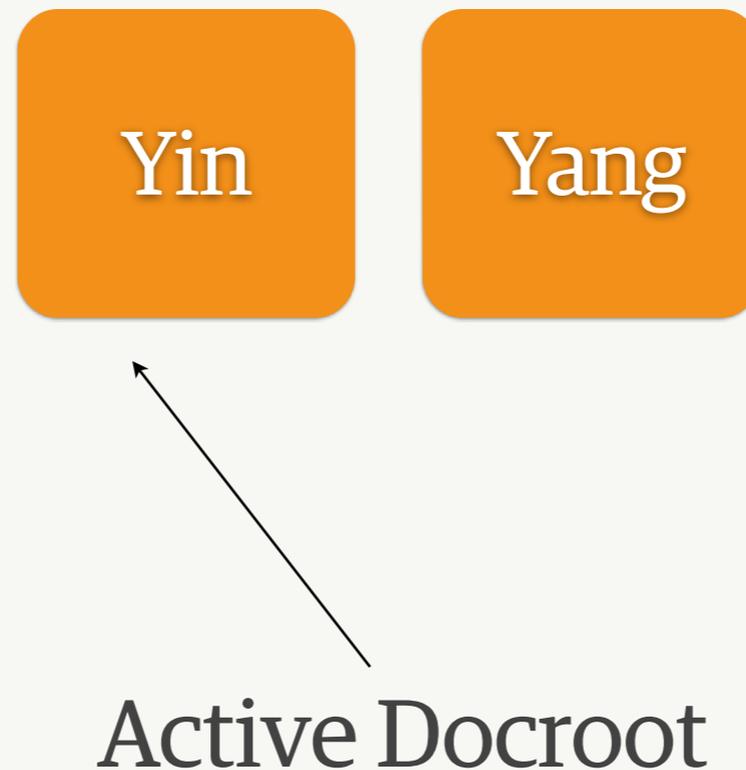
# How can we scale it?

Make deploys atomic:



# How can we scale it?

Make deploys atomic:



# How can we scale it?

Make deploys atomic:



Active Docroot

An arrow originates from the text 'Active Docroot' and points diagonally upwards and to the right, terminating at the bottom-left corner of the 'Yang' box in the diagram above.

# How can we scale it?

Make deploys atomic:

- Not so trivial
- PHP opcache problems
- `include_path` troubles
- Swapping symlinks mid-request

[http://github.com/  
etsy/mod\\_realdoc](http://github.com/etsy/mod_realdoc)

# How can we scale it?

Make deploys atomic, `mod_realdoc`:

- Apache `post_read_request` hook
- Whole request works on `realpath` of `docroot`
- Caches `realpath` for 2s

[http://github.com/  
etsy/incpath](http://github.com/etsy/incpath)

# How can we scale it?

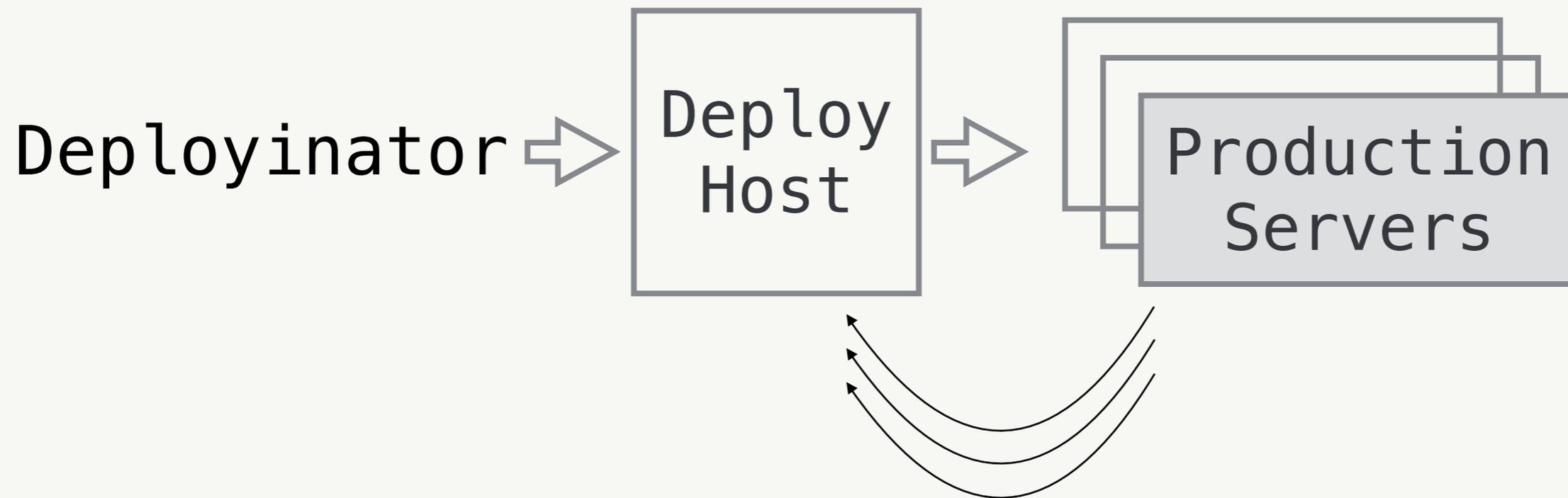
Make deploys atomic, incpath:

- PHP extension
- Updates a portion of `include_path`
- `$_SERVER["DOCUMENT_ROOT"]`

# Infrastructure

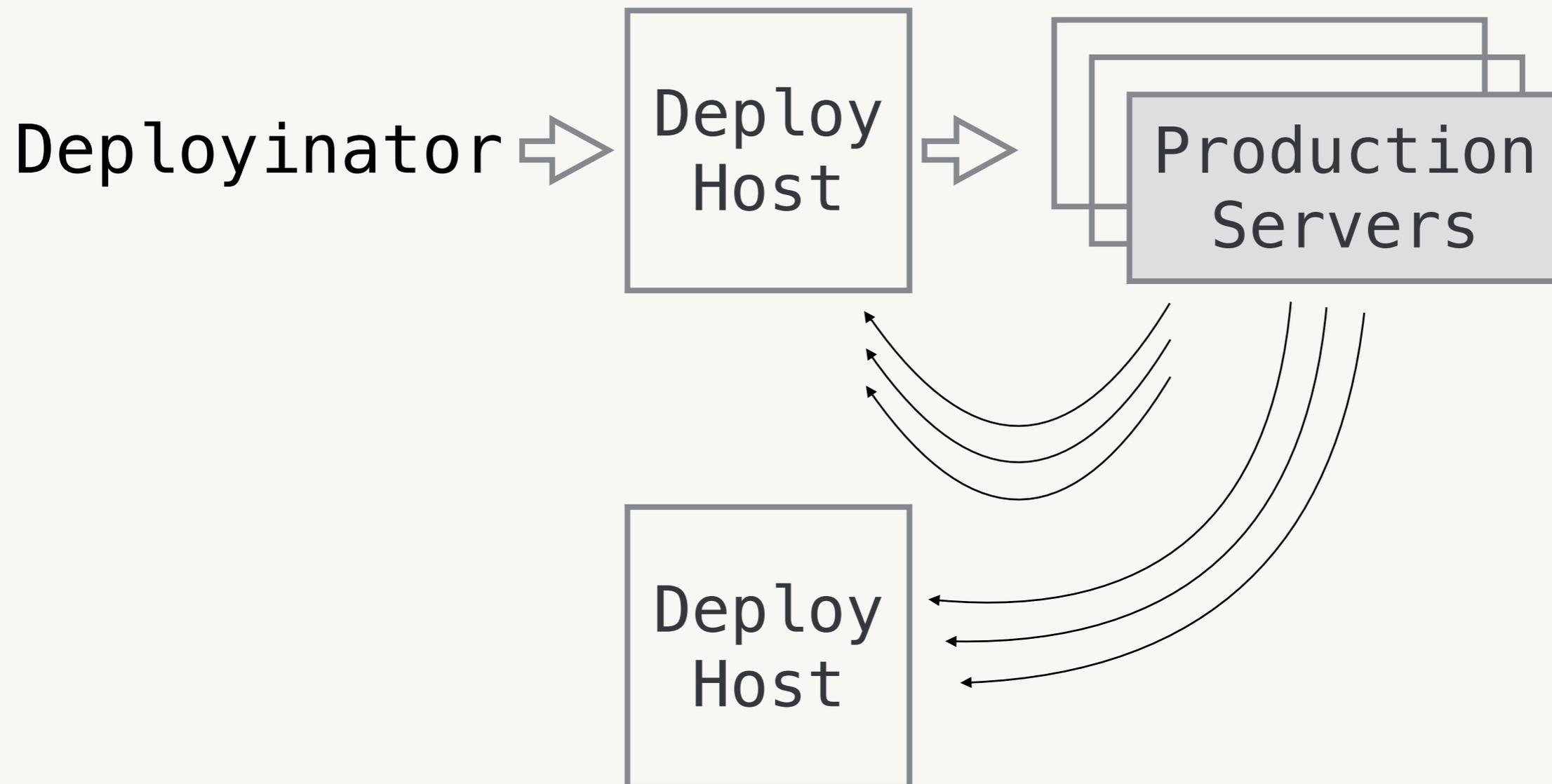
# Scaling infrastructure

Before:



# Scaling infrastructure

After:



# Results!

# Results!

What did we gain?

- No need to restart apache
- Entire deploy in one push
- Opcode cache stays warm!

# Results!

## Push frequency

- $\frac{(\text{push\_queue\_hours} * 60)}{\text{minutes to deploy}}$
- Still ~15mins/deploy:
  - Much more code going out
  - Tests still run fast
  - Less time waiting to deploy

# Q&A