LeafyChat, DjangoDose, Hurricane, and PyCon, Lessons Learned with the Real-Time Web and Python

Alex Gaynor
PyCon 2010, Atlanta, GA

What is "the real time web"?

Sounds like marketing!

Ajax!

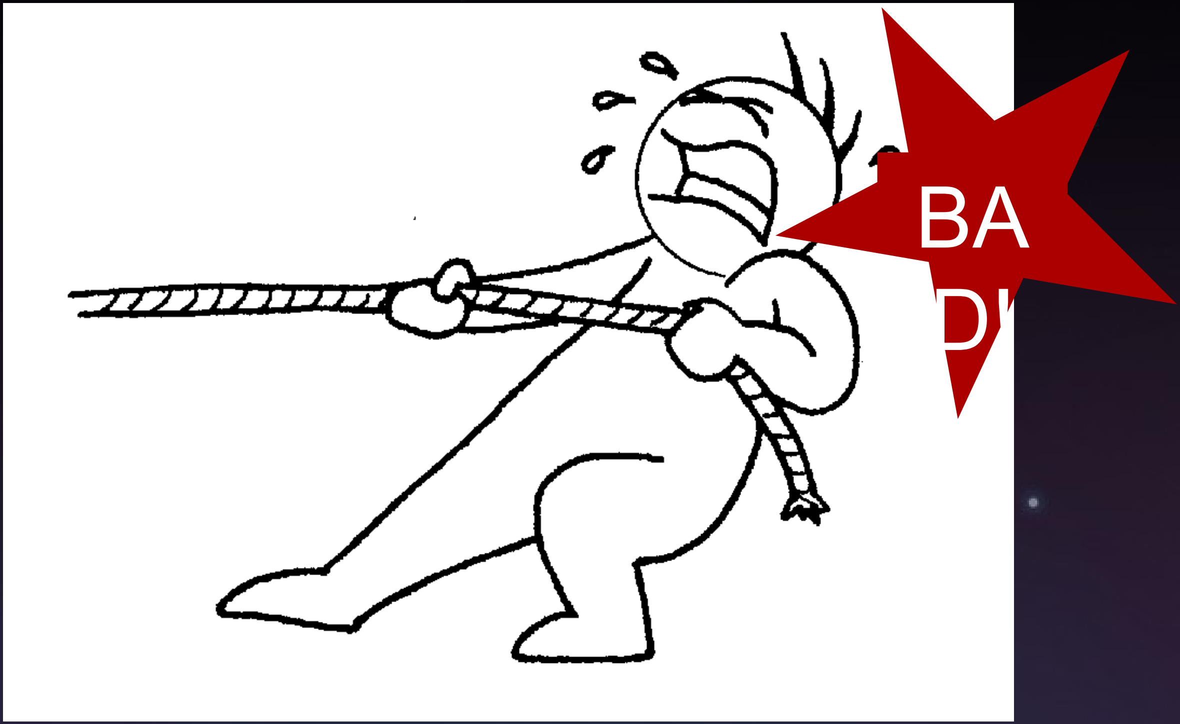
teed Clean!
alue!
AJAX
with BLEACH
CLEANSER
EASY RINSE FORMULA
NET WT 14 OZ (396g)

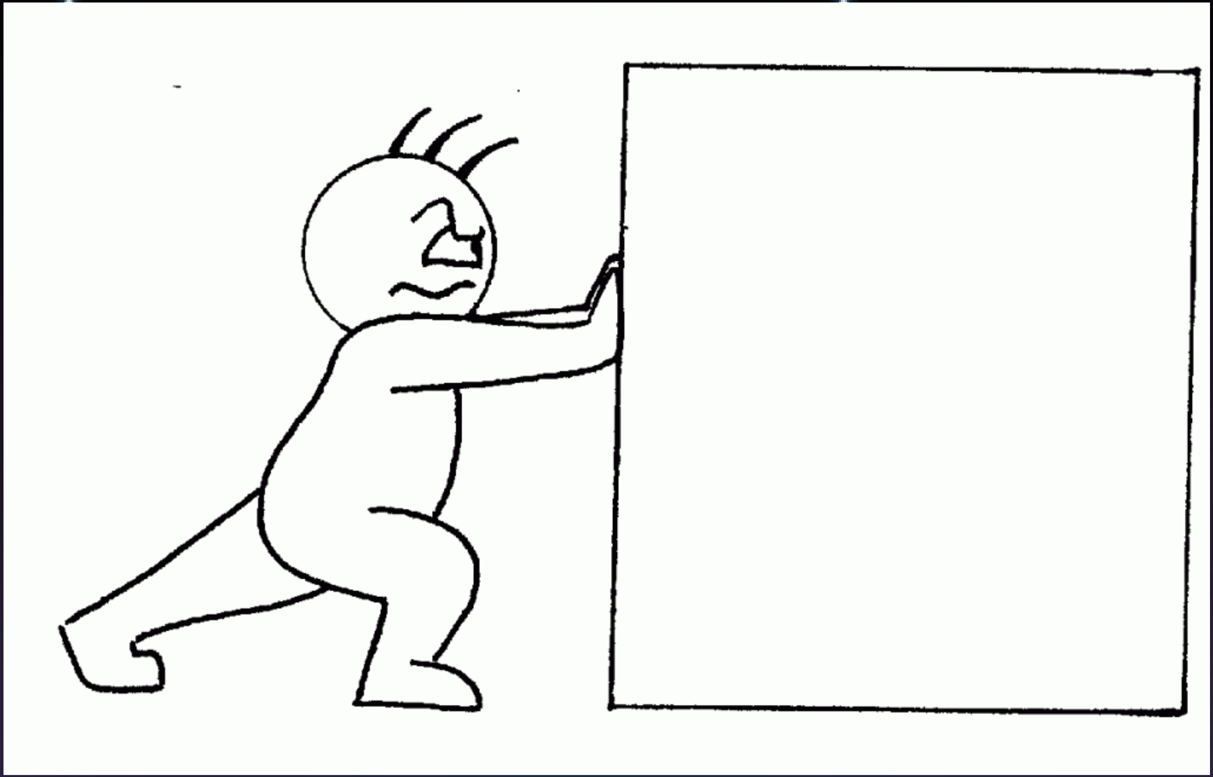
Comet!

Scratch Free
Comet
WITH Bleach
DISINFECTANT CLEANSER
Disinfects & Cleans Tough Stains
NET WT 21 OZ (1.31 LB) 595g
KEEP OUT OF REACH OF CHILDREN
CAUTION: CAUSES MODERATE EYE IRRITATION. READ PRECAUTIONS ON BACK.

Just like Ajax!

Pushing, not pulling





HTTP is stateless

- HTTP is basically designed for browsers pulling stuff from servers
- Normal socket code is a 2 way connection
- Comet is all about making the browser a bit more like normal socket code

Technologies

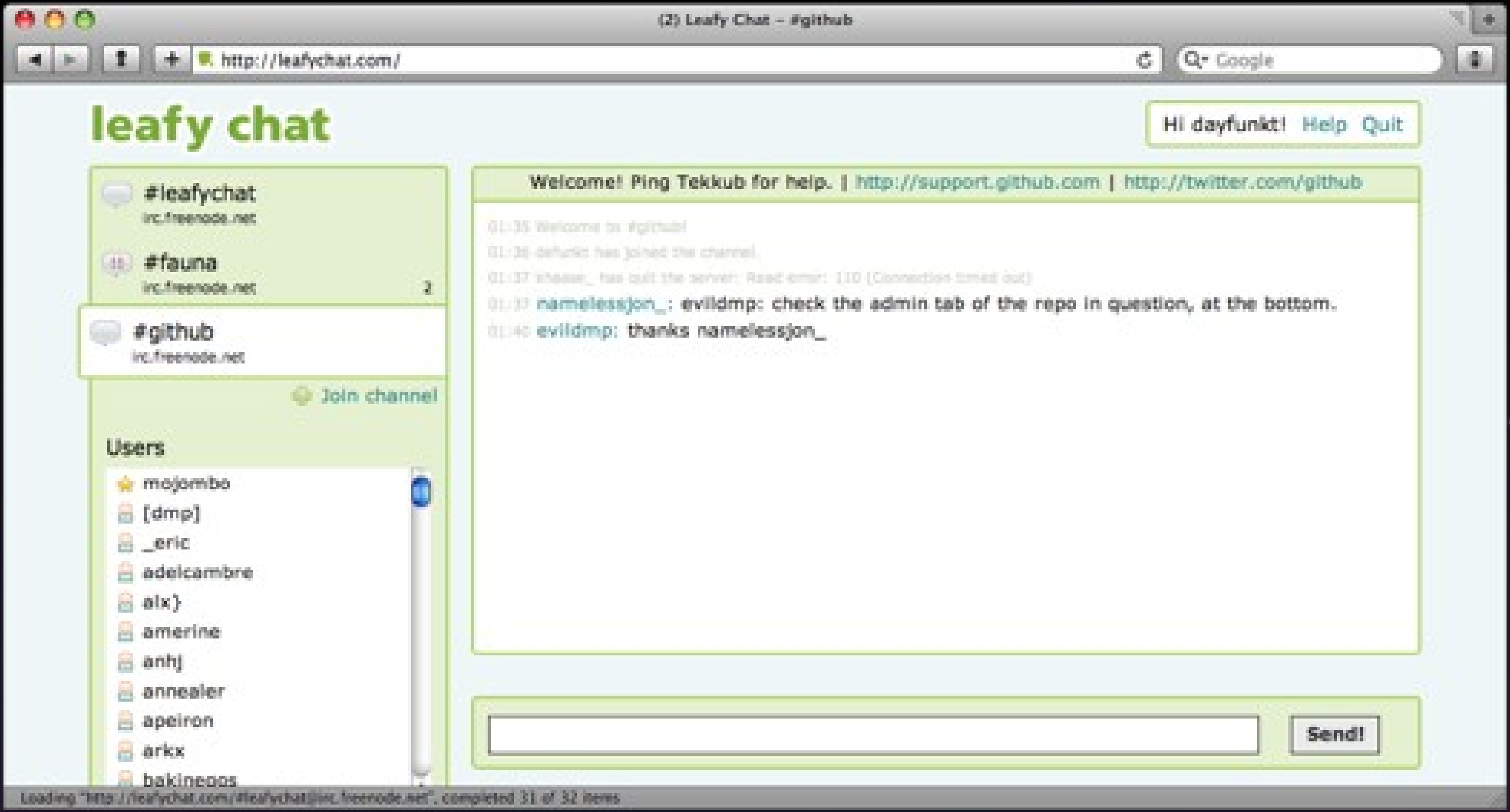
There are a ton of ways to do HTTP-push. And at least as many libraries to help you. Not enough time in this talk!

A tale of 4 websites

LeafyChat

- 2009 Django Dash
- 2nd place!
- Real time IRC, in your browser

bling!
bling!
Coke



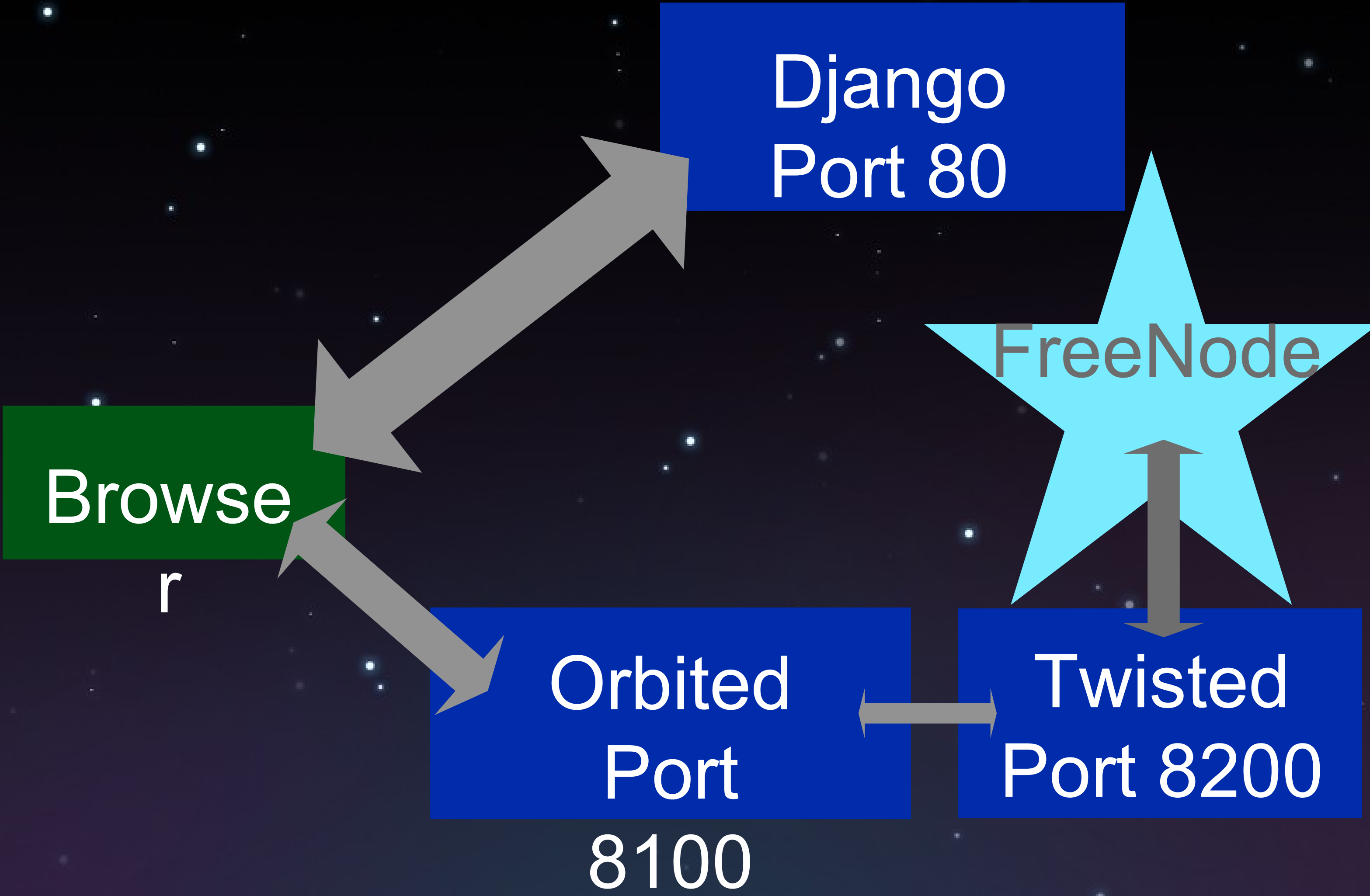
Welcome! Ping Tekkub for help. | http://support.github.com | http://twitter.com/github

```
01:35 Welcome to #github!
01:36 defunkt has joined the channel.
01:37 kheaser_ has quit the server: Read error: 110 (Connection timed out)
01:37 namelessjon_: evildmp: check the admin tab of the repo in question, at the bottom.
01:40 evildmp: thanks namelessjon_
```

mojombo
[dmp]
_eric
adelcambre
alx)
amerine
anhj
annealer
apeiron
arkx
bakineggs

Loading "http://leafychat.com/#leafychat@irc.freenode.net", completed 31 of 32 items

- Built in 48 hours
- Django "frontend" (it was the DjangoDash...)
- Orbited, proxying to a twisted daemon that handed browser connections and IRC connections


Django
Port 80
FreeNode
Browse
r
Orbited
Port
8100
Twisted
Port 8200

JSON

- We pass around JSON packets that look like:

```
{
    "type": "message",
    "from": "Alex_Gaynor",
    "to": "ericflo",
    "body": {
        "msg": "Hey, what's up",
        "timestamp": "something",
    }
}
```

JSON

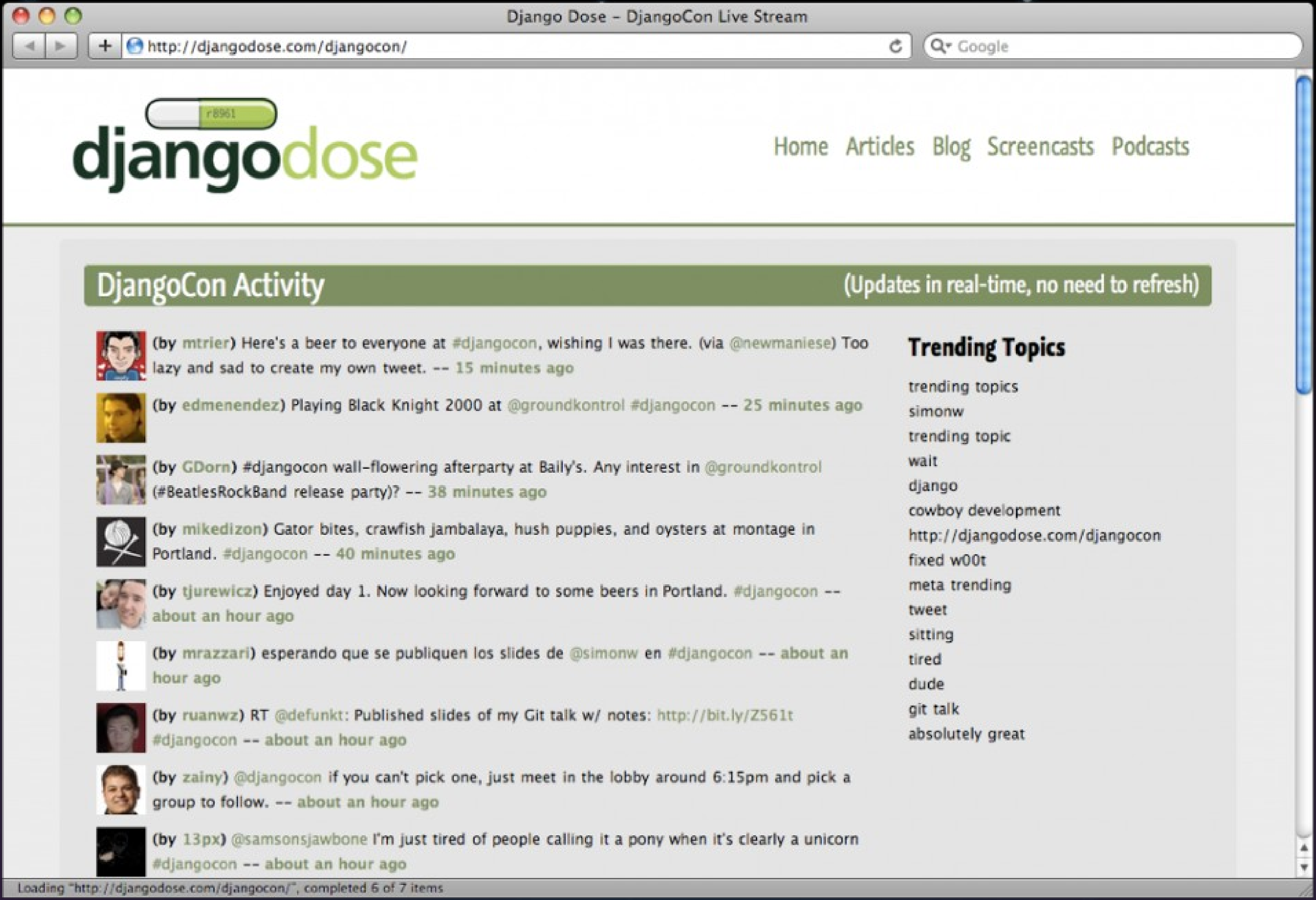
- Both the client and server deal in this JSON
- Client sends it to the server
- Server sends it to clients
- Client takes messages and turns them into calls to our UI lib (built on jQuery) which update the browser

Conclusions

- It all sort of works (we got 2nd place!)
- It's messy
  - IRC in the same process as comment, so it doesn't scale (and isn't isolated)
  - When you want to add new functionality it goes on the server, the client, and the UI lib

DjangoDose

- Built in a week before DjangoCon (this is less time than LeafyChat, since we weren't working 24/7)
- Built on twisted, orbited, Django (it was DjangoCon), and StompMQ


Django Dose - DjangoCon Live Stream
http://djangodose.com/djangocon/
Google
r8961
djangodose
Home Articles Blog Screencasts Podcasts

DjangoCon Activity (Updates in real-time, no need to refresh)

(by mtrier) Here's a beer to everyone at #djangocon, wishing I was there. (via @newmaniese) Too lazy and sad to create my own tweet. -- 15 minutes ago

(by edmenendez) Playing Black Knight 2000 at @groundkontrol #djangocon -- 25 minutes ago

(by GDorn) #djangocon wall-flowering afterparty at Baily's. Any interest in @groundkontrol (#BeatlesRockBand release party)? -- 38 minutes ago

(by mikedizon) Gator bites, crawfish jambalaya, hush puppies, and oysters at montage in Portland. #djangocon -- 40 minutes ago

(by tjurewicz) Enjoyed day 1. Now looking forward to some beers in Portland. #djangocon -- about an hour ago

(by mrazzari) esperando que se publiquen los slides de @simonw en #djangocon -- about an hour ago

(by ruanwz) RT @defunkt: Published slides of my Git talk w/ notes: http://bit.ly/Z561t #djangocon -- about an hour ago

(by zainy) @djangocon if you can't pick one, just meet in the lobby around 6:15pm and pick a group to follow. -- about an hour ago

(by 13px) @samsonsjawbone I'm just tired of people calling it a pony when it's clearly a unicorn #djangocon -- about an hour ago

Trending Topics

trending topics
simonw
trending topic
wait
django
cowboy development
http://djangodose.com/djangocon
fixed w00t
meta trending
tweet
sitting
tired
dude
git talk
absolutely great

Loading "http://djangodose.com/djangocon/", completed 6 of 7 items

- A live feed of everything that was going on on Twitter about DjangoCon (also upcoming talks)
- Also has it's own localized trending topics (with our crappy home grown algorithm)

How it worked

- Client joins a MorbidMQ, on initial connect it joins an initial channel, and a feed channel.
- The initial channel spits out the most recent items when someone joins (so you always have some data)
- The feed channel spits out JSON for tweets when they come in

- Instead of individual communications (like LeafyChat), just one channel on a message queue (plus the initial channel)
- Works really well when all clients are receiving the same channel.
- But not perfect

- When a new user connected to the stream they'd actually get the initial items 3 or 4 times, because the server didn't know when they'd recieved it (no individualized connections)
- Plus they probably got the data multiple times if there were multiple simultaneous connections

- Also, it had the same problems LeafyChat (no isolation between Comet and Twitter connections)

Hurricane

- An attempt to take the lessons learned from DjangoDose (and LeafyChat) and build a framework around them
- Based on the idea of things being producers and consumers

Producer and Consumers

One Producer/Consumer to Rule Them All

It's all about Comet!

Technologies

- Long polling with jQuery
- Tornado Server
- Multiprocessing to have producers and consumers run concurrently
- And a queue abstraction over multiprocessing.Queue, AMQP, and anything else you could throw at it!

We got it working

- We had a twitter feed example
- and a chat room

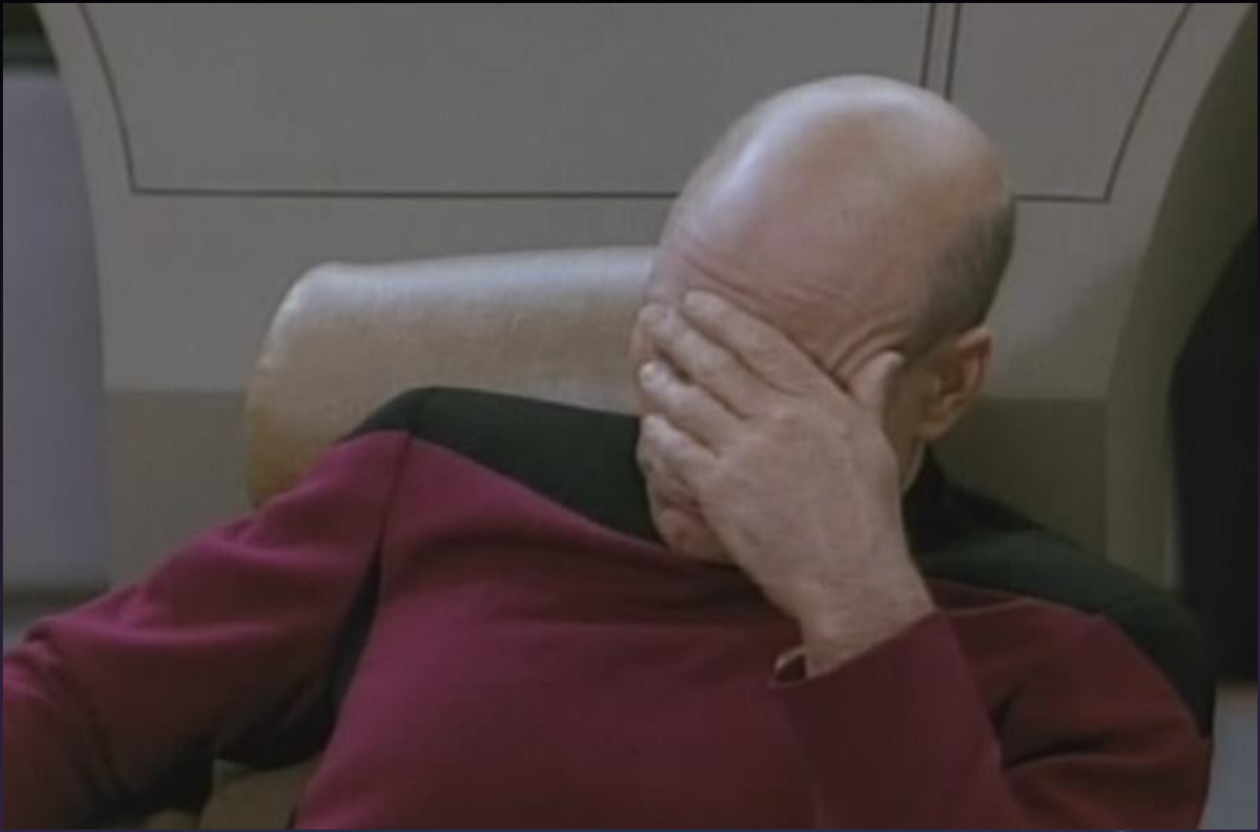
The Problems

- Entire application state was in memory
- Producer and consumers were in different processes (multiprocessing), but not isolated
- We couldn't get the abstractions right for the comet producer

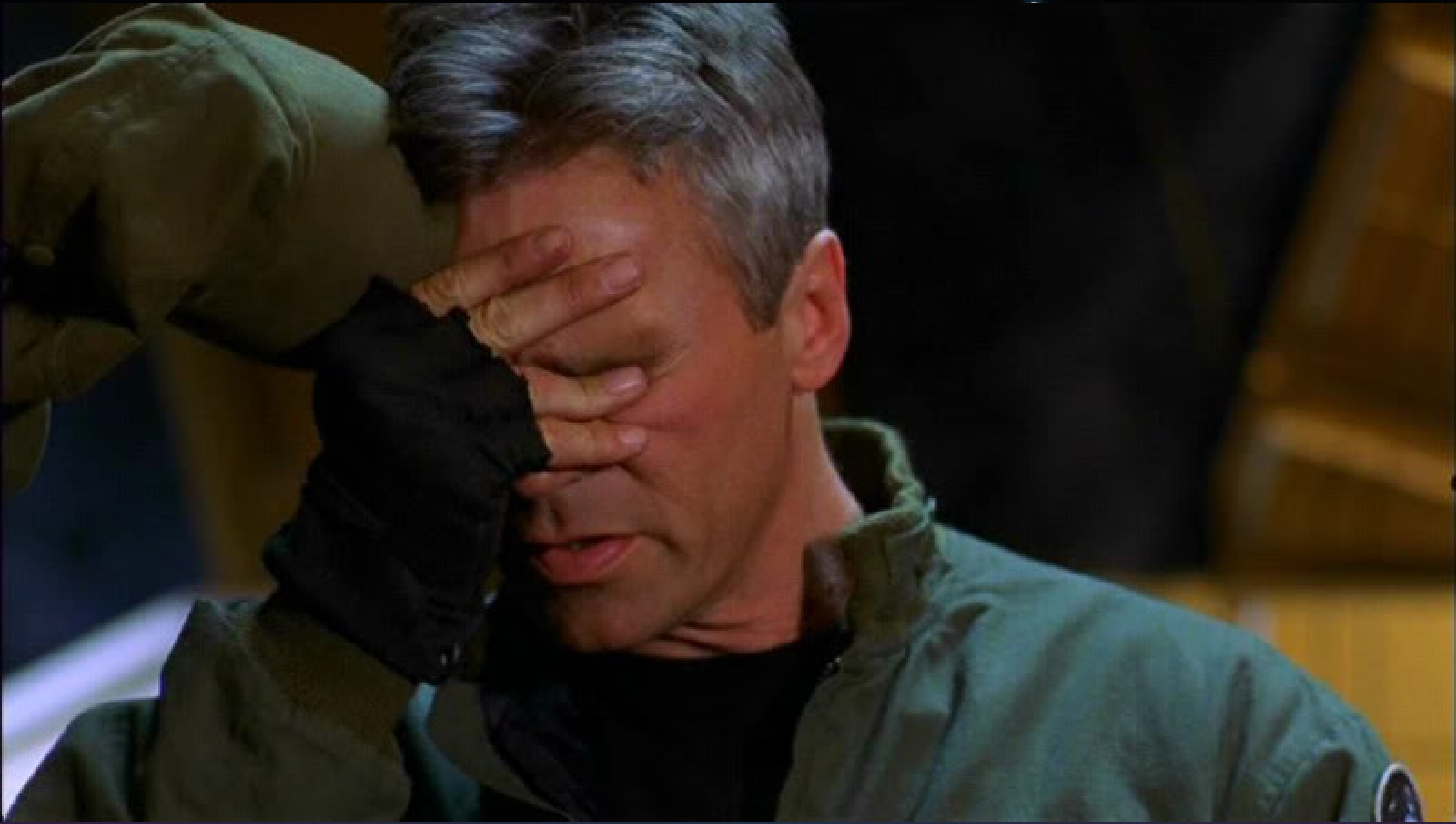
Abstractions

- How do you know which user is which?
- How do you know what messages go to what users?

Frameworks



We ended up rewriting the entire Comet Producer/Consumer when we wanted to do anything fancy



PyCon 2010

- Do the same thing as DjangoCon, but better.
- A chance to learn our lessons!
- More than just twitter: add flickr, and github, and bitbucket, and more!

A lot more data

A lot more users

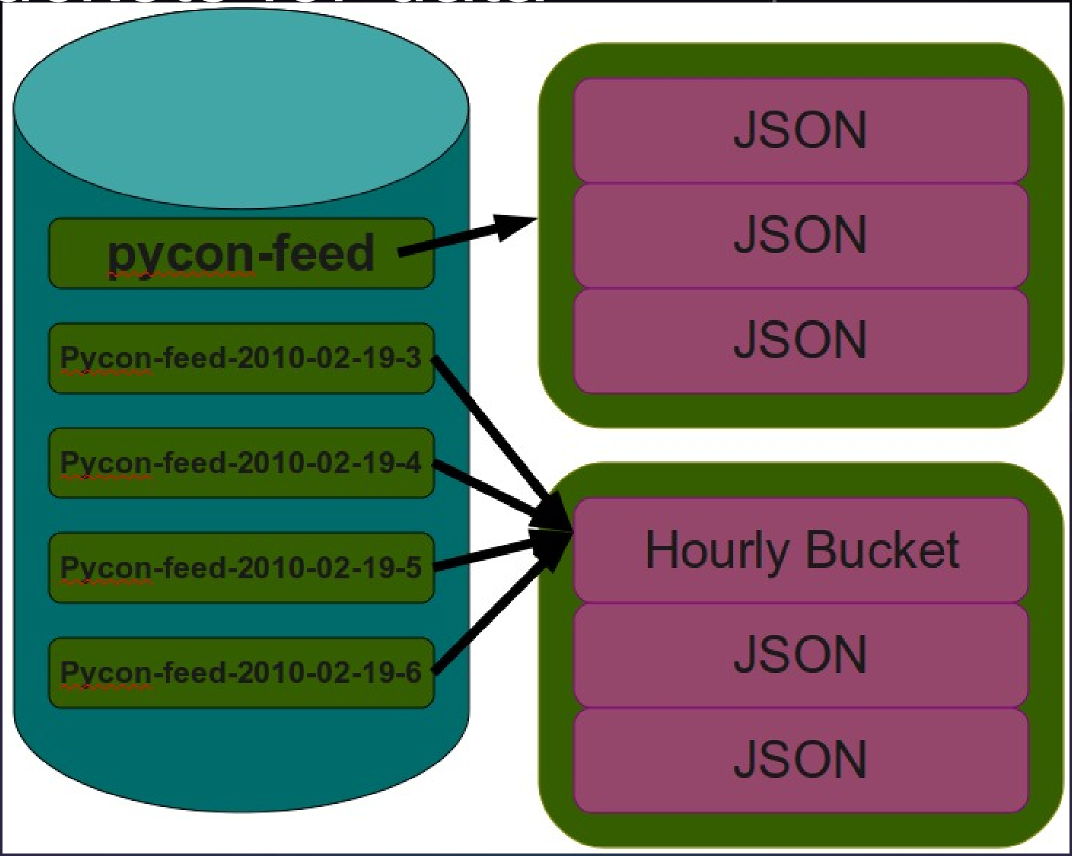
REDiS

What is Redis?

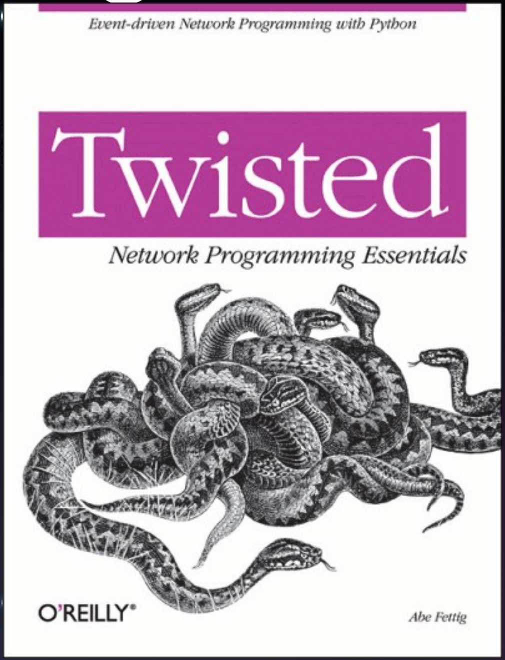
- A key-value store
- With real datastructures, list, sets, hashes.

How do we use it?

- Buckets for data



Other Technologies


Event-driven Network Programming with Python
Twisted
Network Programming Essentials
O'REILLY
Abe Fettig

- Orbited proxies to a twisted daemon
- Twisted daemon uses a blocking operation on redis to be notified when a new item comes in
- Backend processes feed items into redis and let listeners know when theres new items

Problems Solved

- One backend process per item (one for twitter, one for parsing RSS, etc.) -> better isolation
- Don't reinvent Orbited. It's really good at what it does.

The Take Away

Isolate Your Processes

Use Orbited, It's an Awesome Abstraction

Asynchronous Programming is Still Hard

No, You Can't Have a Pony
Not Yours

Generator Based Asynchronous Programming Rocks

@alex_gaynor on twitter

http://alexgaynor.net

slideshare.com for the slides