

GATHERING AND VISUALIZING METRICS WITH ZEROMQ, REDIS & GRAPHITE

Brian Dorsey
brian@dorseys.org
<http://briandorsey.info>

CONTEXT

about 50 instances
of a server which does
interesting work...

... at least I *think* so.

I need to see some graphs.

look into hosted services...

**mostly charge by data volume
quickly becomes expensive**

I want:

- **inexpensive!**
- **don't write much custom code**
- **easy to extend (more measurements)**
- **flexible**
- **play with new things. :)**

Thank you Etsy!

Code as Craft

Etsy

Measure Anything, Measure Everything

Posted by Ian Malpass | Filed under [data](#), [engineering](#), [infrastructure](#)

If Engineering at Etsy has a religion, it's the Church of Graphs. If it moves, we track it. Sometimes we'll draw a graph of something that isn't moving yet, just in case it decides to make a run for it. In general, we tend to measure at three levels: network, machine, and application. (You can read more about our graphs in Mike's [Tracking Every Release](#) post.)

Application metrics are usually the hardest, yet most important, of the three. They're very specific to your business, and they change as your applications change (and Etsy changes a lot). Instead of trying to plan out everything we wanted to measure and putting it in a classical [configuration management system](#), we decided to make it ridiculously simple for any engineer to get anything they can count or time into a

Search

Search

Subscribe

 Posts RSS
 Comments RSS

Archives

» August 2012
» July 2012

awesome, but I didn't want to deploy another service

statsd

with apologies to Philip Greenspun: <http://c2.com/cgi/wiki?GreenspunsTenthRuleOfProgramming>

this talk describes
an ad-hoc,
informally-specified,
bug-ridden, slow
implementation of half of
their metrics system

the plan for this talk:

- Graphite
- Redis
- ZeroMQ
- glue (mostly Python scripts)
- alternatives and improvements
- questions

GRAPHITE

“SCALABLE REALTIME GRAPHING”

Graphite - Scalable Realtime

graphite.wikidot.com/start

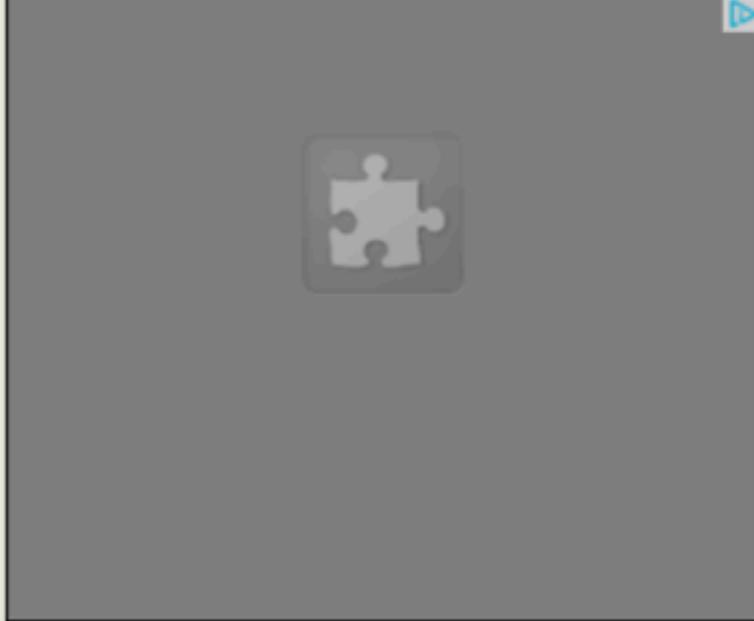
wiki dot site-name .wikidot.com Share on [Twitter](#) [Facebook](#) [Digg](#) [StumbleUpon](#) [Reddit](#) [Link](#) [Join this site](#) [Edit](#) [History](#) [Tags](#) [Source](#) [Explore »](#) [Create account or Sign in](#)

Search this site [Search](#)

Graphite

Home
Downloads
Screen Shots
Documentation
FAQ
Ask a question
Report a bug
How to Contribute
Join This Wiki
Invite Others To This Wiki
Site Manager

Graphite - Scalable Realtime Graphing



New Release - 0.9.10 - 5/31/12

Graphite 0.9.10 has been released and is now available. The packages for Whisper, Carbon, and Graphite-web are available via several sources:

- PyPi (and by extension, pip)
 - <http://pypi.python.org/pypi/graphite-web/>
 - <http://pypi.python.org/pypi/carbon/>
 - <http://pypi.python.org/pypi/whisper/>
- Github
 - <https://github.com/graphite-project/graphite-web/downloads>
 - <https://github.com/graphite-project/carbon/downloads>
 - <https://github.com/graphite-project/whisper/downloads>
- Launchpad
 - <https://launchpad.net/graphite/0.9/0.9.10>

Graphite:

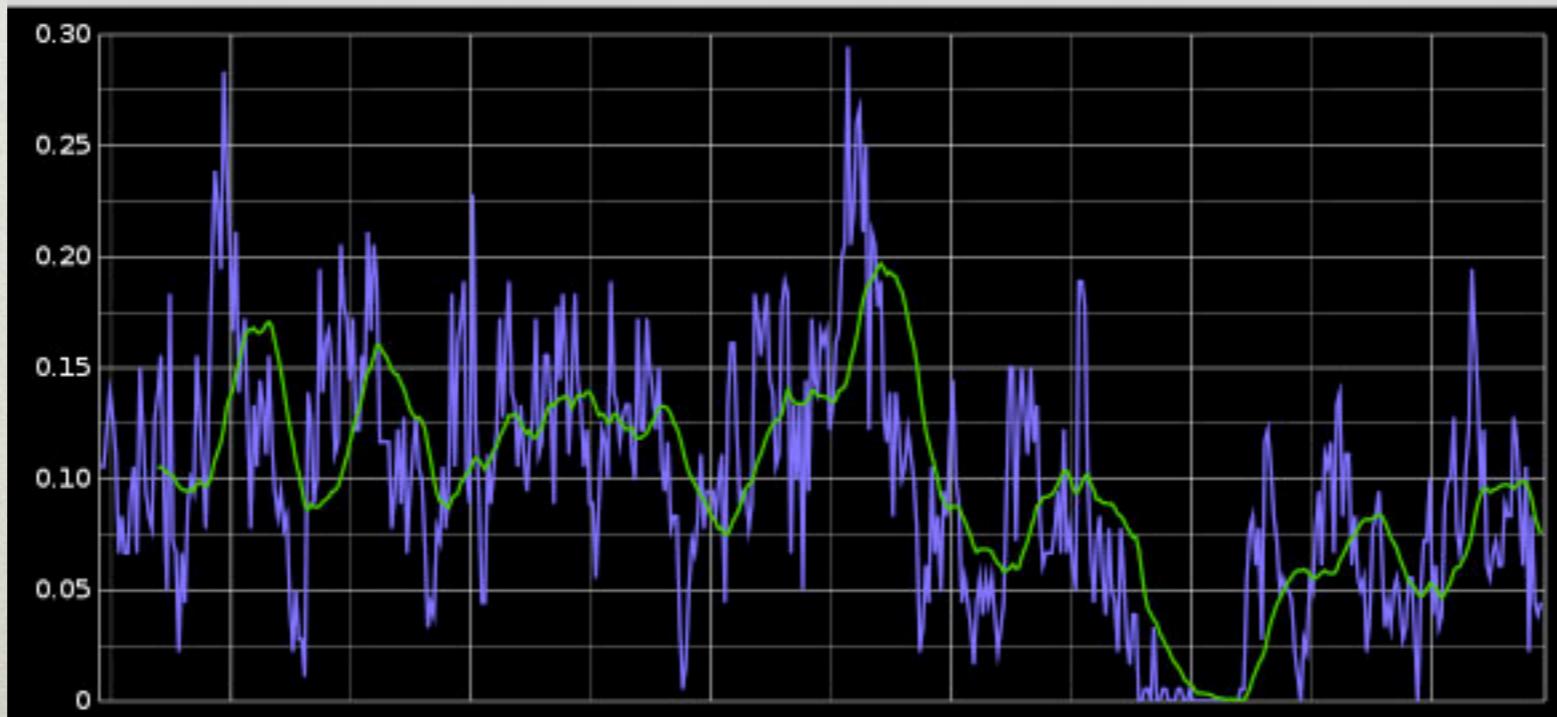
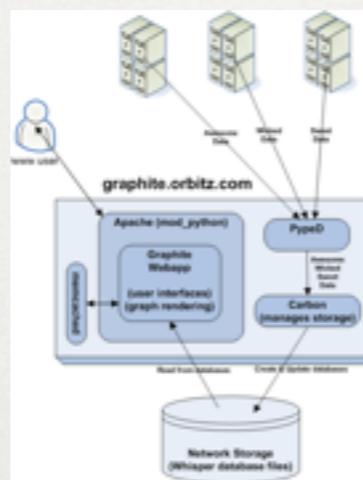
takes numbers
remembers them
makes graphs

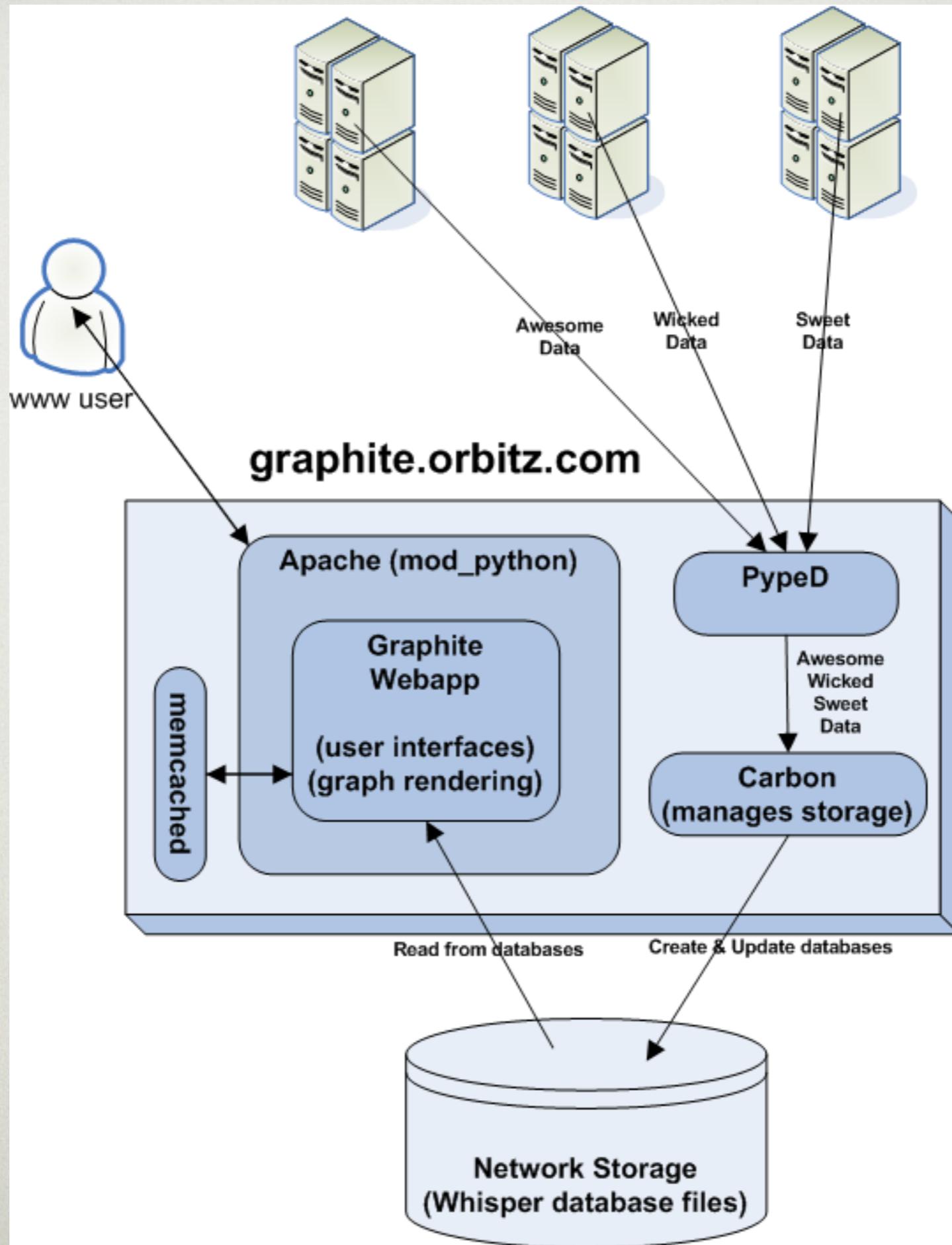
(examples.posts, 12, 1347406123)

(examples.posts, 33, 1347406957)

(examples.posts, 42, 1347407021)

...





- Carbon → "Backend data caching and persistence daemon"
<http://graphite.wikidot.com/carbon>
- Whisper →
"Fixed size round-robin style database"
subset of RRDtool features, plus a bit
<http://graphite.wikidot.com/whisper>
- Graphite-Web →
"Enterprise scalable realtime graphing"

(examples.posts, 12, 1347406123)

↑

↑

↑

dotted.stat.name value unix epoch

graphite

[Login](#)[Documentation](#)User Interface: [Dashboard](#) | [flot \(experimental\)](#) | [events \(experimental\)](#) |

http://en.wikipedia.org/wiki/Round-Robin_Database

Round-Robin Database?

Swiss Original. Mirrors: [UK](#) [TW](#) [DE](#) [US](#)

About RRDtool

What RRDtool does

RRDtool is the OpenSource industry standard, high performance data logging and graphing system for time series data. RRDtool can be easily integrated in shell scripts, perl, python, ruby, lua or tcl applications.

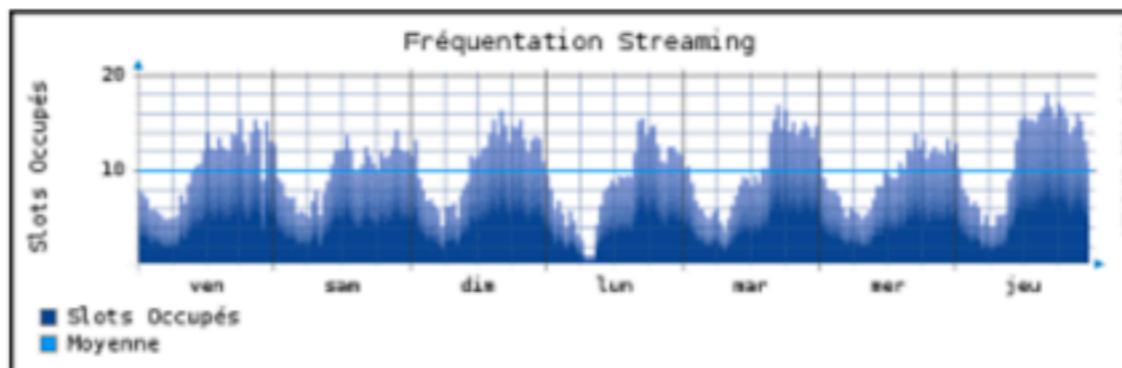


40 people like this. [Sign Up](#) to see what your friends like.

PRTG - Free Download

Monitor network, website, servers and more. Windows software.
www.Paessler.com/download

AdChoices ▾



Download

RRDtool is available for [download](#) from this site. It compiles on a number of different Linux and Unix platforms as well as on Microsoft Windows.

Welcome

Download

Documentation

Tutorials

Programming

Bugs, Wiki

Support

Gallery

Projects using RRDtool

Sponsoring RRDtool

License

Cast

Web oss.oetiker.ch

[Google Search](#)

Gold Sponsors

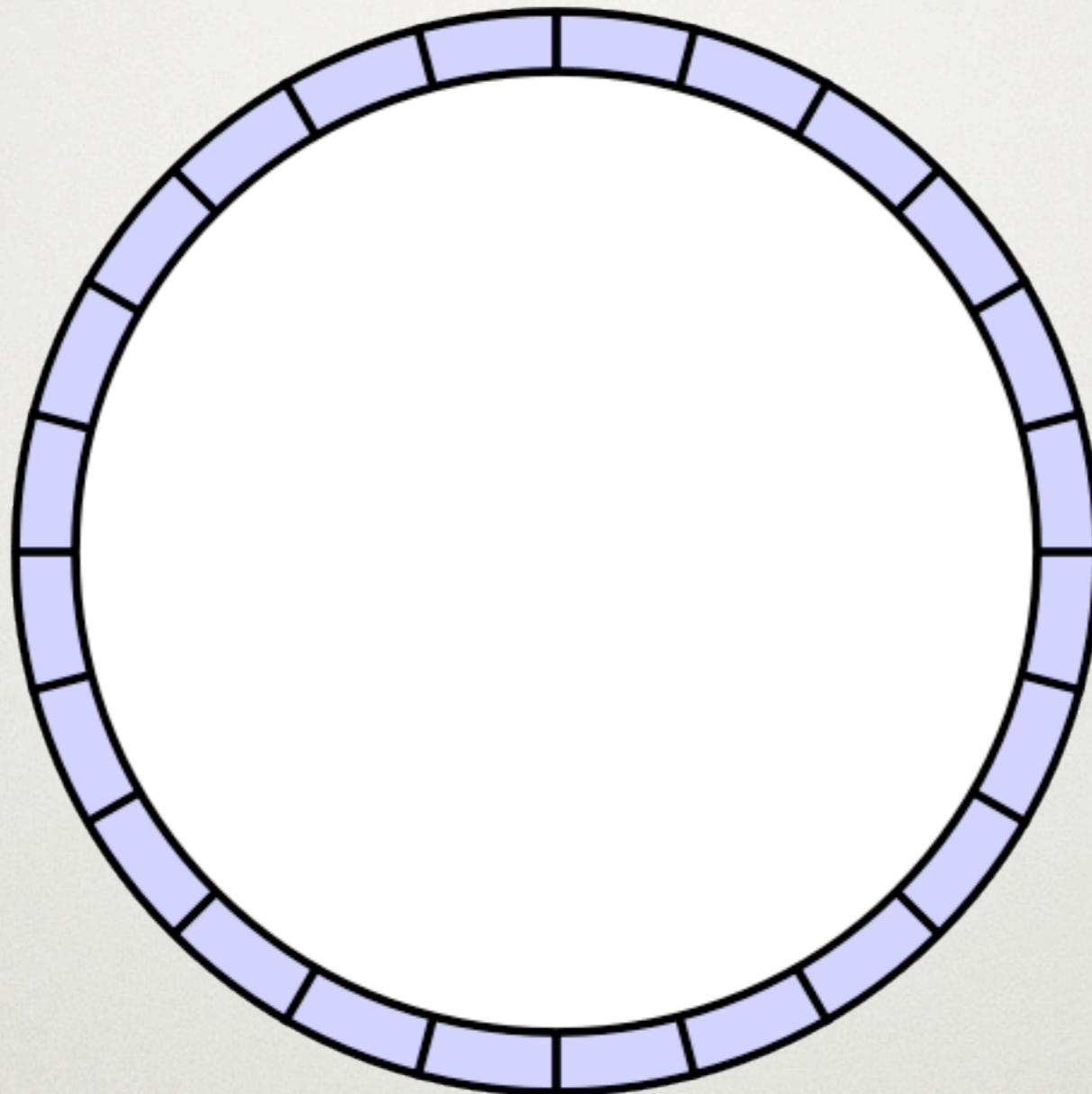


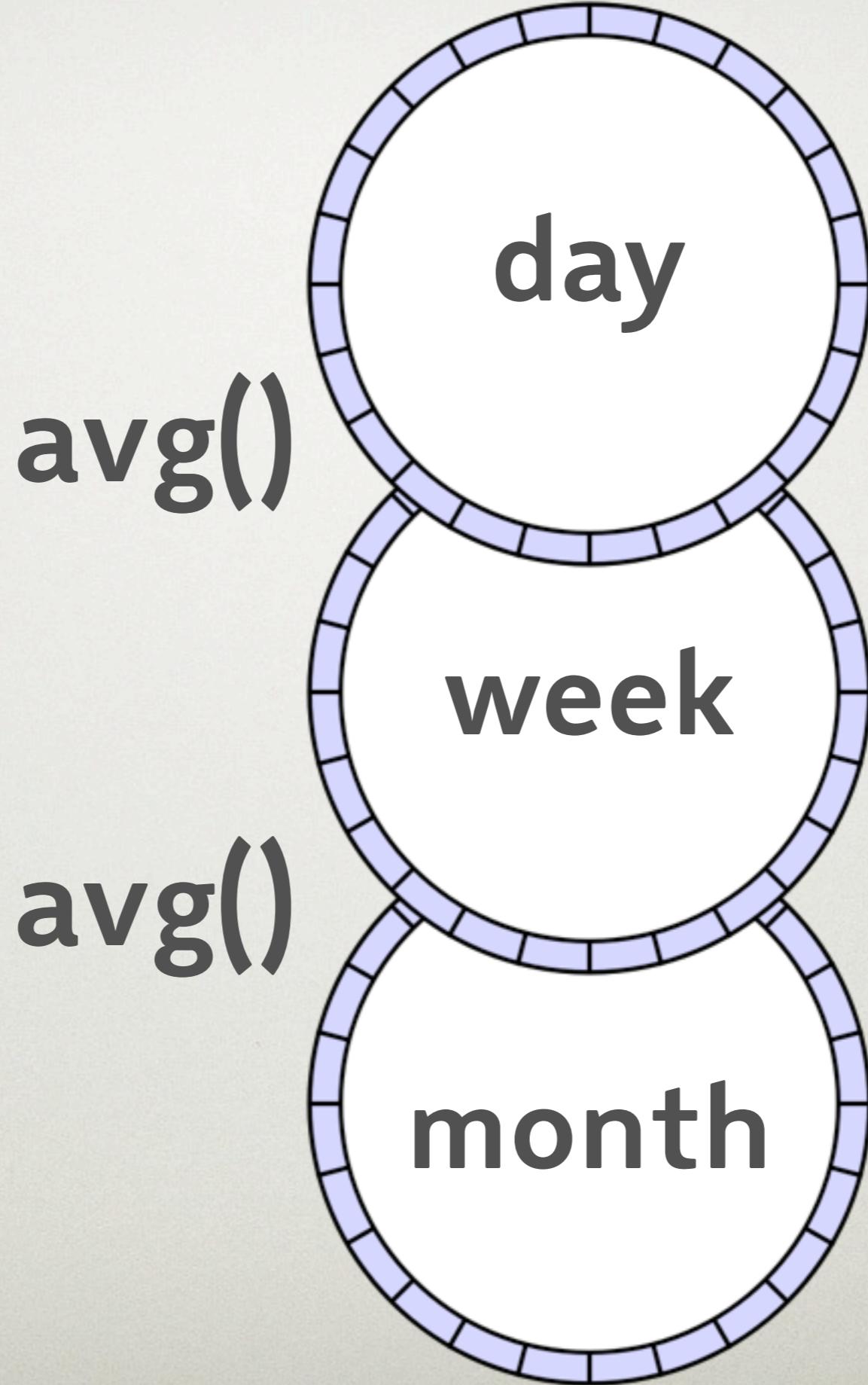
Environmental sensors
for server rooms
PoE - SNMP - Web



SERVER ROOM MONITORING
SERVERSCHECK

Your logo could
be here! If you
want to support
this project, just
contact us.





graphite

[Login](#)[Documentation](#)User Interface: [Dashboard](#) | [flot \(experimental\)](#) | [events \(experimental\)](#) |

```

```

graphs are **very
customizable**

**colors
lines, stacking, grouping
data functions**

REDIS

“REDIS IS AN OPEN SOURCE,
ADVANCED ... DATA STRUCTURE SERVER.”

The screenshot shows the Redis website (redis.io) in a web browser. The page features a dark header with the Redis logo and navigation links for Commands, Clients, Documentation, Community, Download, and Issues. The main content area includes a brief introduction to Redis, a 'Try it' section with an interactive tutorial link, and a 'Download it' section with a link to the latest stable version (Redis 2.4.17). To the right, there's a sidebar titled 'What people are saying' displaying several tweets from users like 'Playin game...smh yu need to stop for real', 'After switch to Couch, looking like need to use Redis + something else.', and 'Build a Real Time Activity Stream on Cloud Foundry with #Node.js, #Redis and #MongoDB 2.0'. A 'More...' link is at the bottom of the sidebar.

Redis is an open source, advanced **key-value store**. It is often referred to as a **data structure server** since keys can contain strings, hashes, lists, sets and sorted sets.

[Learn more →](#)

Try it

Ready for a test drive? Check this [interactive tutorial](#) that will walk you through the most important features of Redis.

Download it

Redis 2.4.17 is the latest stable version. Interested in release candidates or unstable versions? [Check the downloads page](#).

What people are saying

Playin game...smh yu need to stop for real

After switch to Couch, looking like need to use Redis + something else. 700,000 inserts a day + equal deletions, 210M+ records at any time.

Build a Real Time Activity Stream on Cloud Foundry with #Node.js, #Redis and #MongoDB 2.0
<http://t.co/fyBgW83u> probamos?

Friends! New #Job! We're looking for a #Database #Administrator: #redis #mysql #linux #vertica #hive #Hbase #dba See
<http://t.co/iR2TkZ43>!

@HectorLuisIII @e_estrada10
 @redis_bo ya vas faltando a clases -

[More...](#)



This website is open source software developed by Citrusbyte.
The Redis logo was designed by Carlos Prioglio. See more credits.

Sponsored by
vmware

Redis is
worth looking into.

If “Redis” is the only thing you remember from this talk, I’m OK with that.

Seriously.

**in-memory key-value
store...**

**... where the values are
data structures:
strings, hashes, lists,
sets & sorted sets**

cache
shared data structures
queueing primitives
pub-sub messaging
master-slave replication
lua scripting

**BLPOP** key [key ...] timeout

Remove and get the first element in a list, or block until one is available

LLEN key

Get the length of a list

LREM key count value

Remove elements from a list

RPUSH key value [value ...]

Append one or multiple values to a list

BRPOP key [key ...] timeout

Remove and get the last element in a list, or block until one is available

LPOP key

Remove and get the first element in a list

LSET key index value

Set the value of an element in a list by its index

RPUSHX key value

Append a value to a list, only if the list exists

BRPOPLPUSH source destination ...

Pop a value from a list, push it to another list and return it; or block until one is available

LPUSH key value [value ...]

Prepend one or multiple values to a list

LTRIM key start stop

Trim a list to the specified range

LINDEX key index

Get an element from a list by its index

LPUSHX key value

Prepend a value to a list, only if the list exists

RPOP key

Remove and get the last element in a list

LINSERT key BEFORE|AFTER pivot ...

Insert an element before or after another element in a list

LRANGE key start stop

Get a range of elements from a list

RPOPLPUSH source destination

Remove the last element in a list, append it to another list and return it



This website is open source software developed by Citrusbyte.

The Redis logo was designed by Carlos Prioglio. See more credits.

Sponsored by
vmware

lpush

rpop

```
redis> lpush "key" "a"
(integer) 1
redis> lpush "key" "b"
(integer) 2
redis> lpush "key" "c"
(integer) 3
redis> llen "key"
(integer) 3
redis> lrange "key" 0 -1
1) "c"
2) "b"
3) "a"
redis> rpop "key"
"a"
redis> rpop "key"
"b"
```



brpop

ZEROMQ (ØMQ)

“THE INTELLIGENT TRANSPORT LAYER”

The screenshot shows a web browser window with the URL www.zeromq.org in the address bar. The page content is as follows:

ØMQ

The Intelligent Transport Layer

ØMQ \zeromq\:

- Ø The socket library that acts as a concurrency framework.
- Ø Faster than TCP, for clustered products and supercomputing.
- Ø Carries messages across inproc, IPC, TCP, and multicast.
- Ø Connect N-to-N via fanout, pubsub, pipeline, request-reply.
- Ø Asynch I/O for scalable multicore message-passing apps.
- Ø Large and active open source community.
- Ø 30+ languages including C, C++, Java, .NET, Python.
- Ø Most OSes including Linux, Windows, OS X.
- Ø LGPL free software with full commercial support from iMatix.

Learn
the Basics

Get
the Software

Solve
a Problem

Join
the Community

Buy
Support

Post
a Job Offer

Web site design and content is copyright (c) 2007-2012 iMatix Corporation. Contact us for professional support. Site

ZeroMQ isn't a
Message Queue

it is a toolbox for building
messaging systems

... but I used it because
it guarantees atomic
message delivery

ZeroMQ can do ***many*** things, for details see: <http://www.zeromq.org/>

REQ, REP
DEALER, ROUTER
PUB, SUB
PUSH, PULL
PAIR

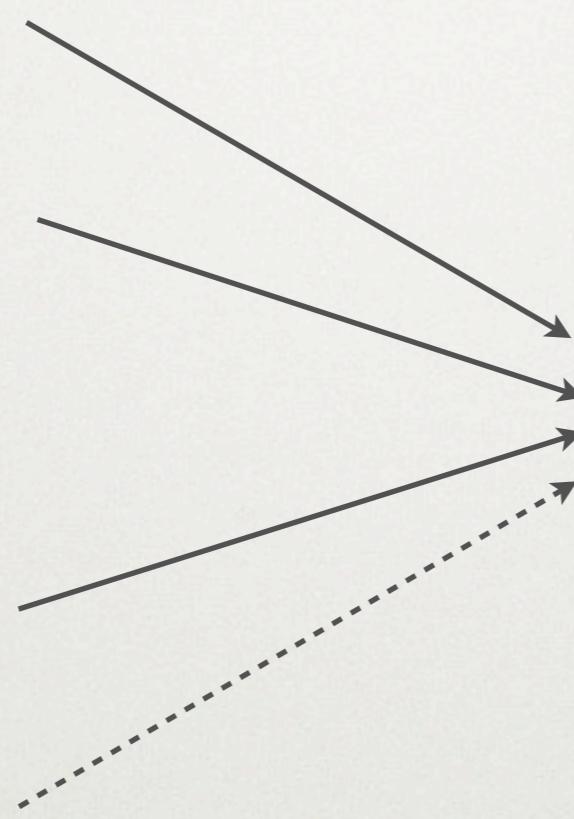
publisher

publisher

...

publisher

publisher



subscriber

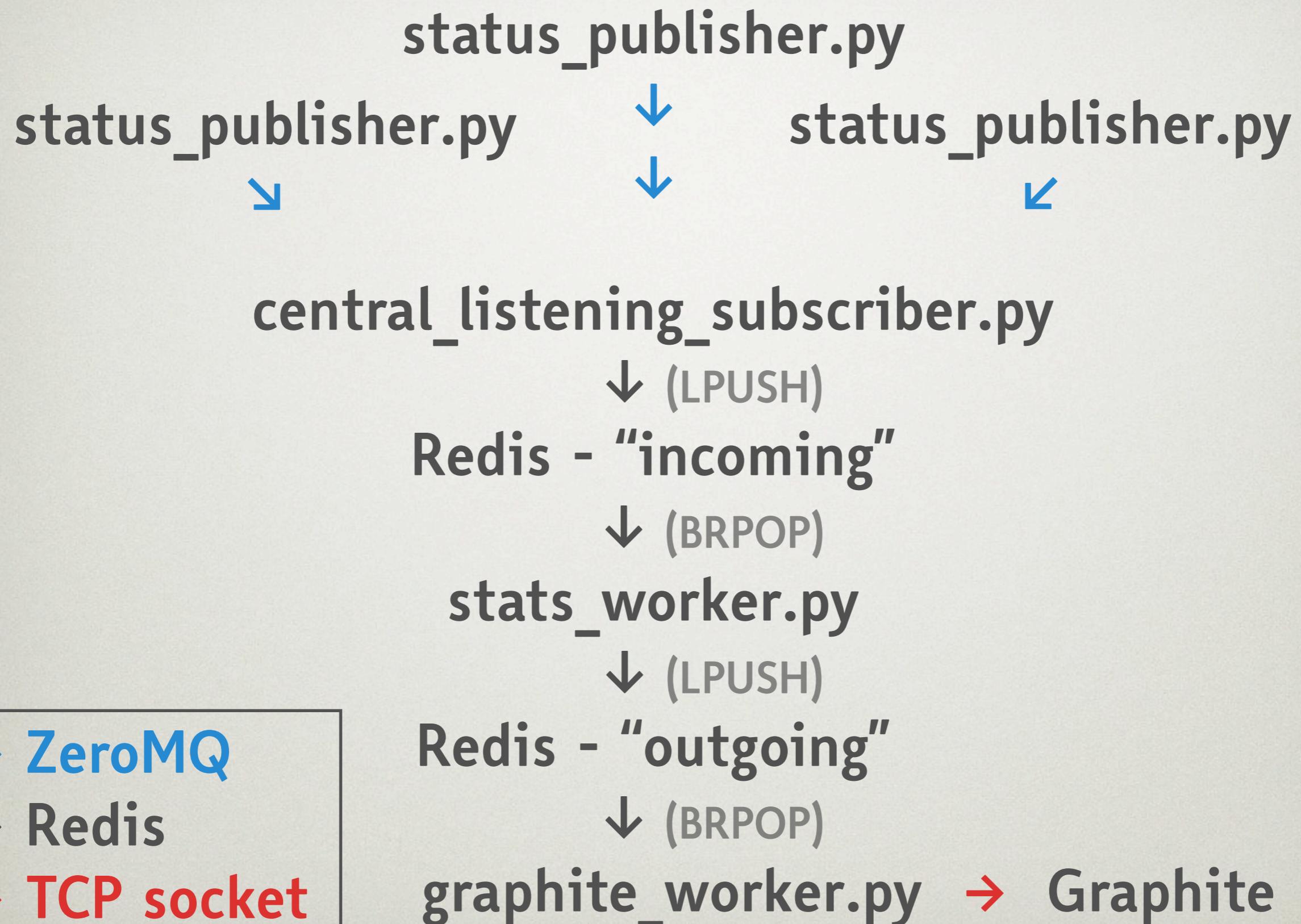
could have just
used UDP sockets

probably don't *need*
ZeroMQ for this

very convenient to never
work about packet splits

THE GLUE

SUPERVISORD, PYTHON SCRIPTS AND A BIT OF HTML



↓ ZeroMQ
↓ Redis
→ TCP socket

status publisher

(embedded in your
server-which-does-
interesting-work)

status_publisher.py

```
import json
import time
import zlib
import zmq

def get_stats():
    return [ {'value': time.time() % 100,
              'name': 'interesting_metric',
              'time': time.time()}]

c = zmq.Context()
s = c.socket(zmq.PUB)
s.connect('tcp://127.0.0.1:5555')
s.setsockopt(zmq.HWM, 1)

while True:
    data = get_stats()
    data = json.dumps(data)
    data = zlib.compress(data)
    s.send(data)
    time.sleep(5)
```

**central listening
subscriber**

central_listening_subscriber.py

```
import redis
import zmq

c = zmq.Context()
s = c.socket(zmq.SUB)
s.bind('tcp://127.0.0.1:5555')
s.setsockopt(zmq.SUBSCRIBE, '')

r = redis.Redis() # defaults to localhost

while True:
    data = s.recv()
    r.lpush('incoming', data)
```

**worker to process stats
JSON into
graphite format**

stats_worker.py

```
import json
import zlib
import redis

r = redis.Redis() # defaults to localhost

while True:
    _, data = r.brpop('incoming')
    data = zlib.decompress(data)
    data = json.loads(data)
    for stat in data:
        metric = 'demo.%s %s %s' % stat
        r.lpush('outgoing', metric)
```

more: map to dotted name, filter characters in names, etc

push stats to graphite

graphite_worker.py

```
import socket
import redis

r = redis.Redis() # defaults to localhost
sock = socket.socket()
sock.connect( ('127.0.0.1',2003) )

while True:
    _, message = r.brpop('outgoing')
    message = message + '\n' # all lines must end in
a newline
    print message,
    sock.sendall(message)
```

web page to
view the graphs

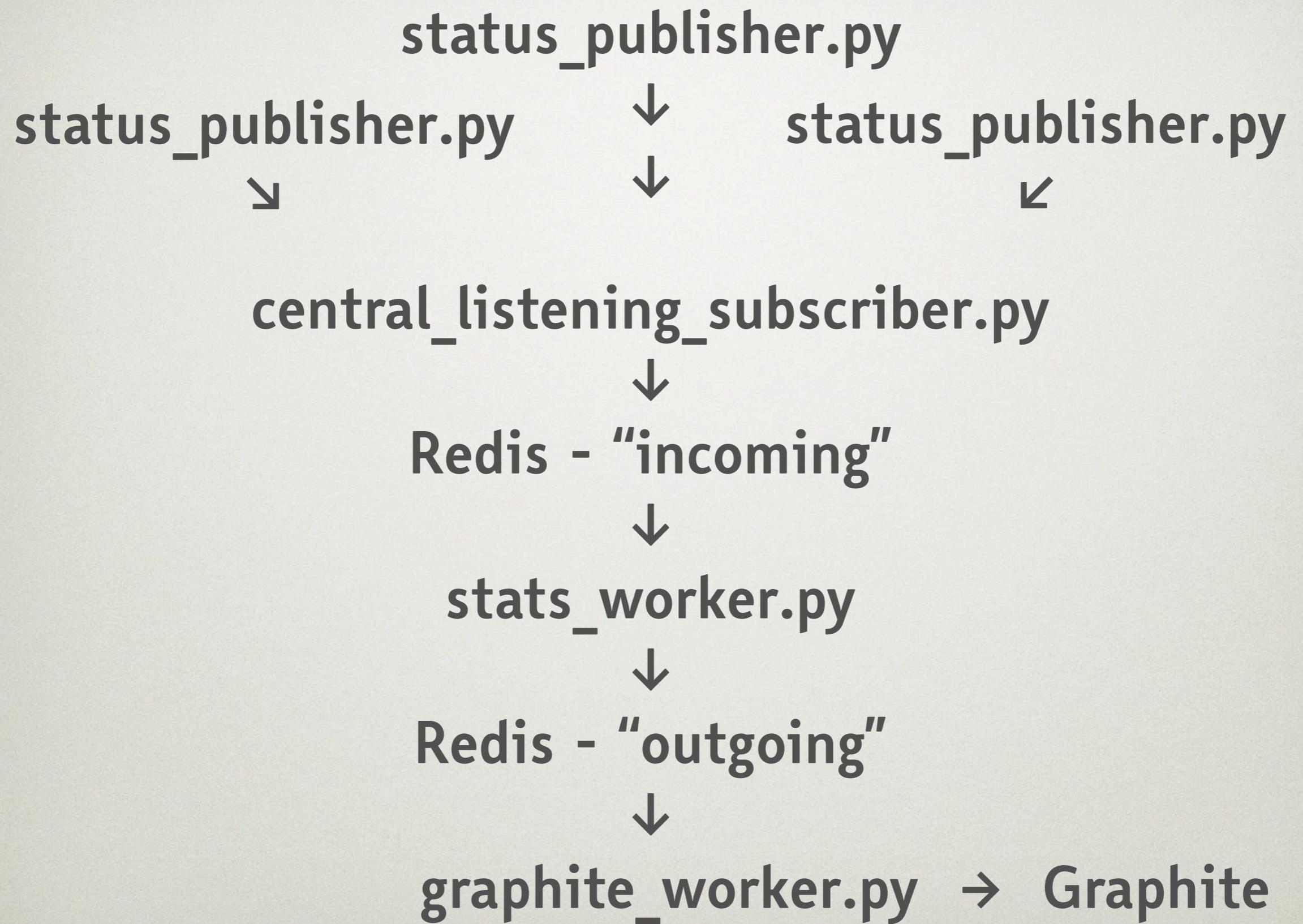
graphs.html

```
<html>
  <head>
    <title>Interesting Metrics</title>
  </head>
  <body>
    
  </body>
</html>
```

`apt-get install supervisor`

these scripts are all
hosted by supervisord

simple config
just works



BTW:

Graphite and Redis are only supported on unixy OSes.

If you're in a windows shop, setup an Ubuntu VM.

ALL TOGETHER

**~50 servers
status every 5 seconds
(~10/sec)**

~20 metrics per status

**~200 data points / sec
~518,400,000 / month**

prototyped

...

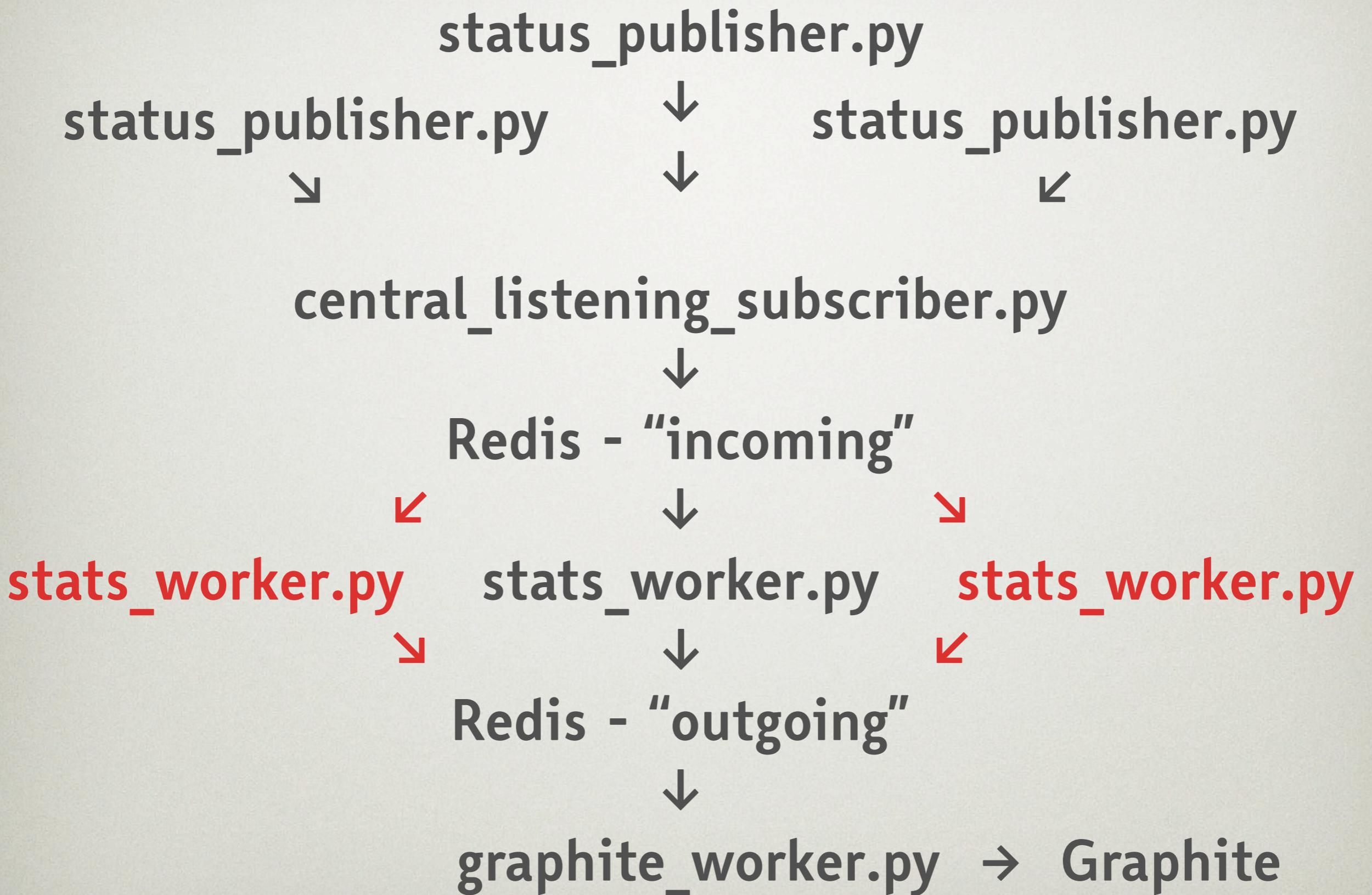
got busy with other
things

5 months later...

everything still worked
~4,000,000,000 Redis ops

ALTERNATIVES AND IMPROVEMENTS

**slowest code was
JSON parsing**



or get the central glue out
of the parsing business

probably wont get fragmented ;)

send smaller messages
over UDP?

batch and summarize
updates?

<http://joemiller.me/2011/09/21/list-of-statsd-server-implementations/>

wait!

isn't that just
re-inventing statsd?

Code as Craft

Etsy

Measure Anything, Measure Everything

Posted by Ian Malpass | Filed under [data](#), [engineering](#), [infrastructure](#)

If Engineering at Etsy has a religion, it's the Church of Graphs. If it moves, we track it. Sometimes we'll draw a graph of something that isn't moving yet, just in case it decides to make a run for it. In general, we tend to measure at three levels: network, machine, and application. (You can read more about our graphs in Mike's [Tracking Every Release](#) post.)

Application metrics are usually the hardest, yet most important, of the three. They're very specific to your business, and they change as your applications change (and Etsy changes a lot). Instead of trying to plan out everything we wanted to measure and putting it in a classical [configuration management system](#), we decided to make it ridiculously simple for any engineer to get anything they can count or time into a

Search

Search

Subscribe

 Posts RSS
 Comments RSS

Archives

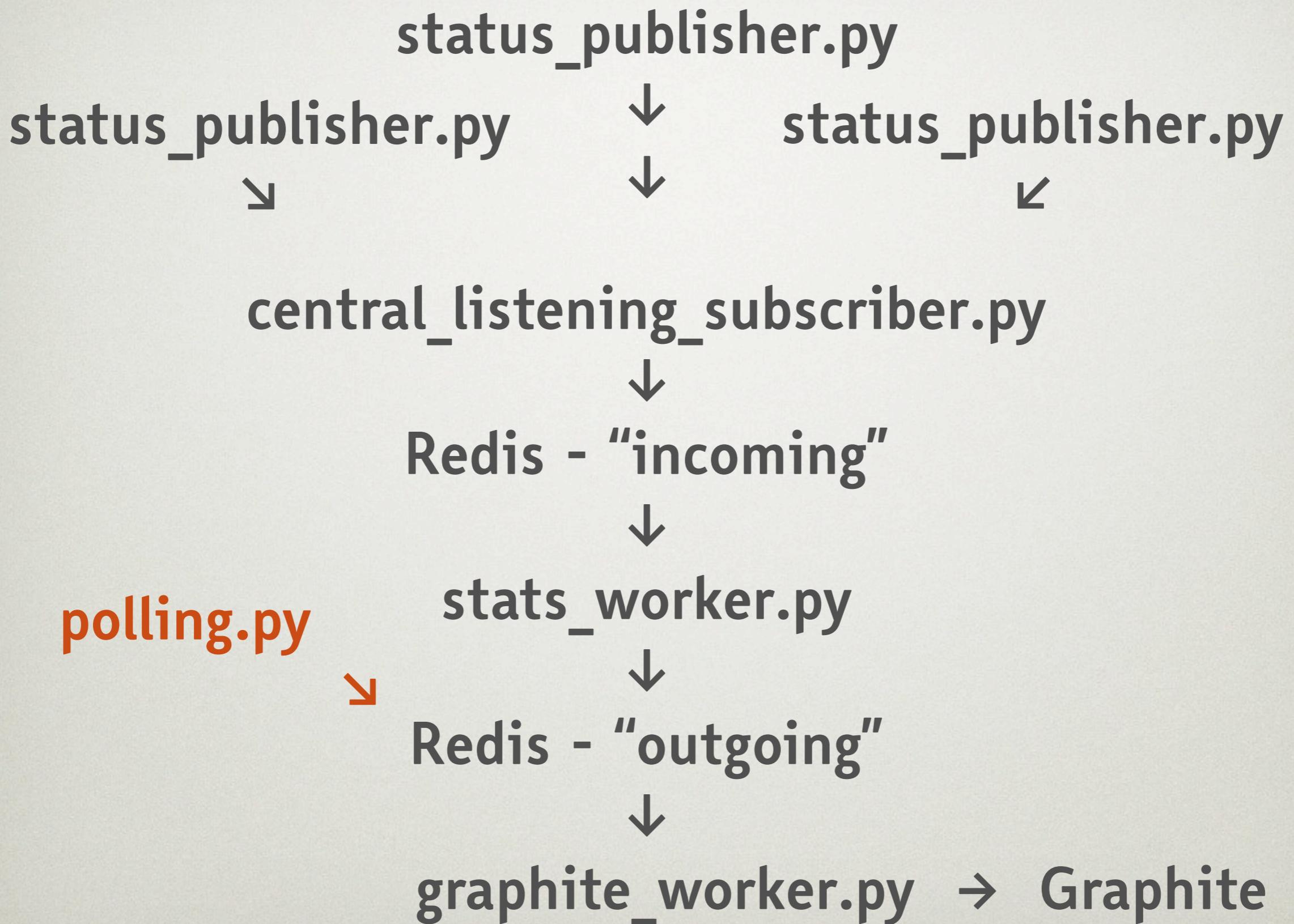
» August 2012
» July 2012

**what about other
data sources?**

**(don't want to modify
every source of stats
to publish)**

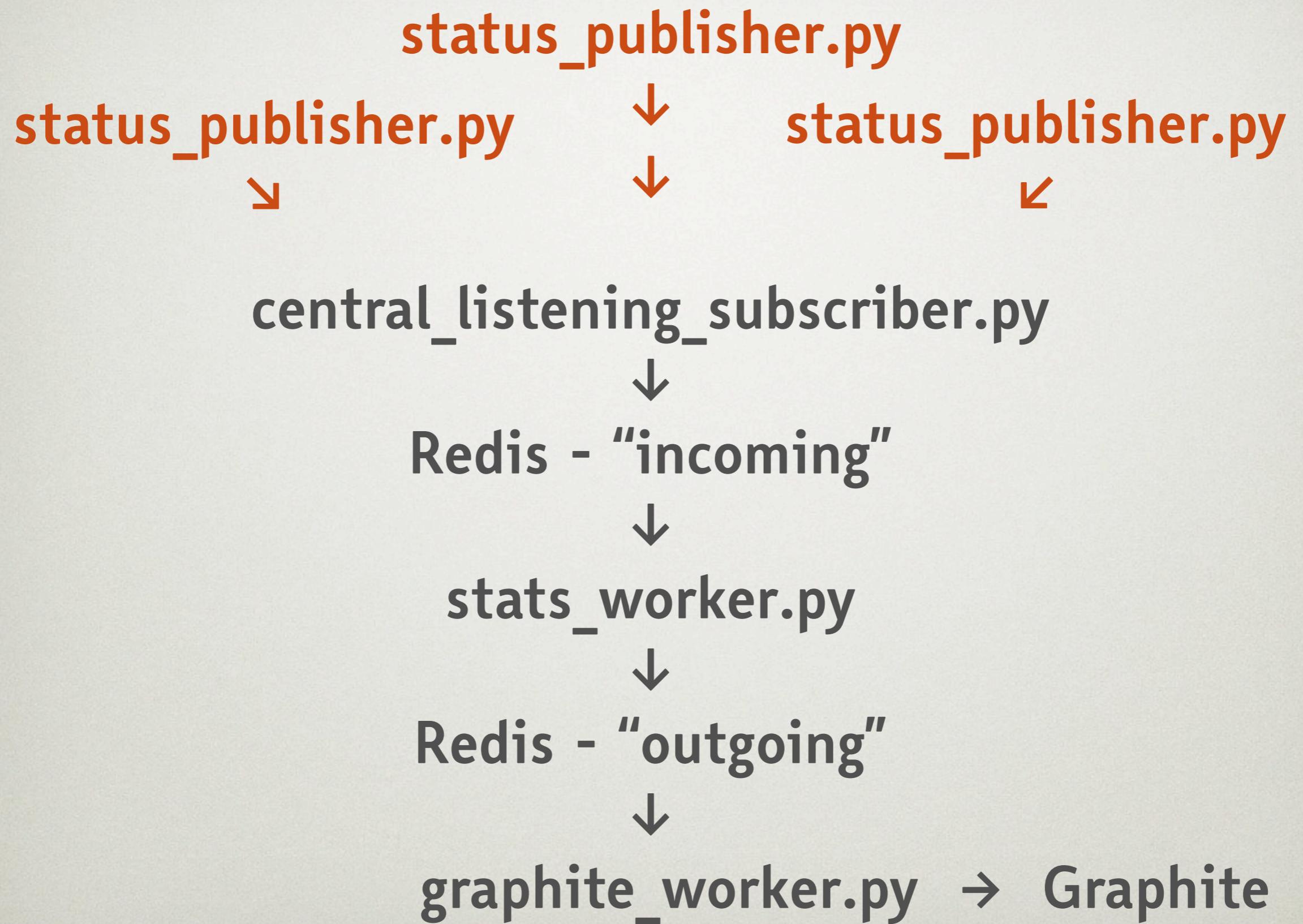
**do whatever works:
poll, parse logs, etc**

**as long as the metrics end
up in the redis list, they'll
get graphed
(or send to Graphite)**



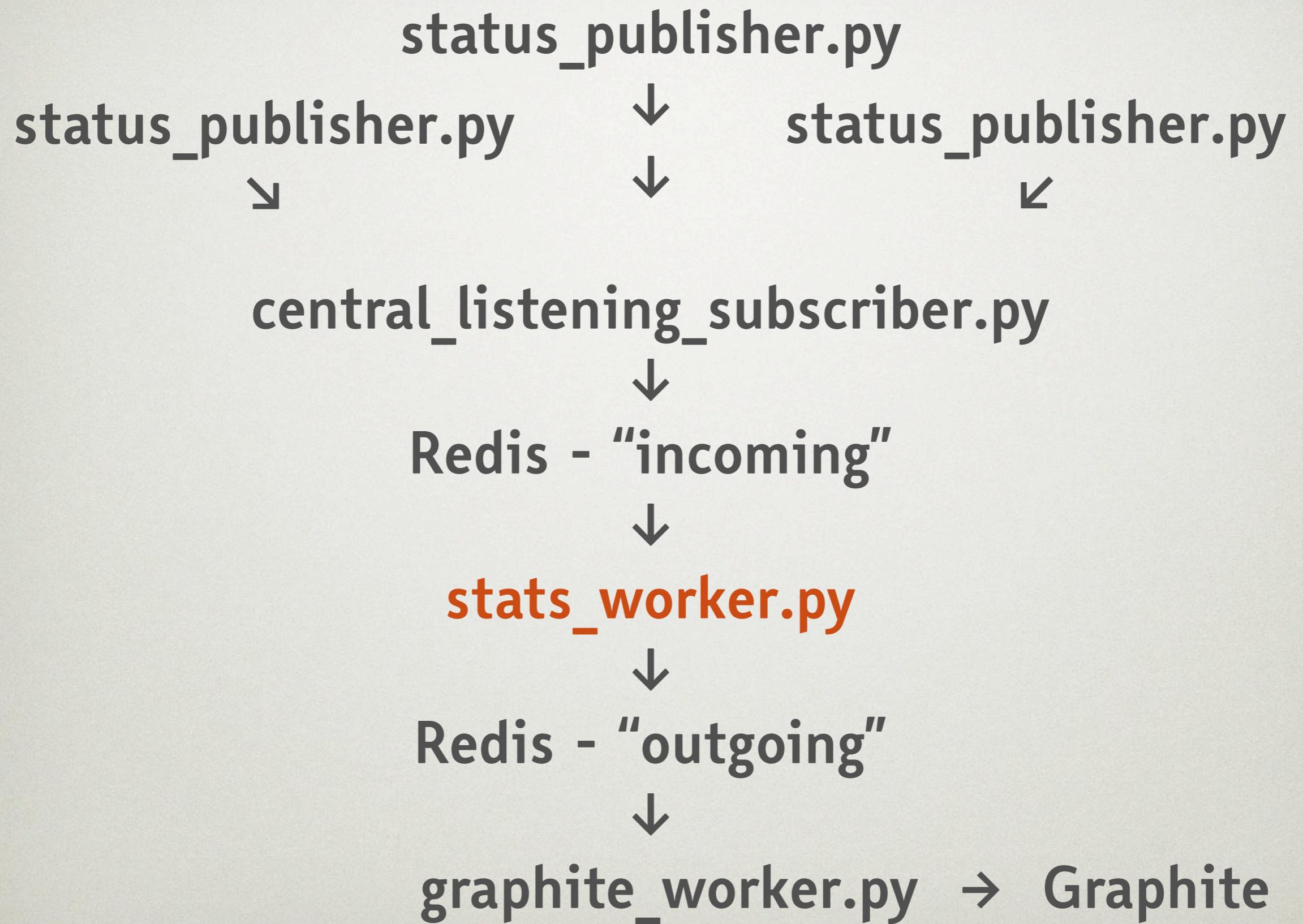
extending:

new metics



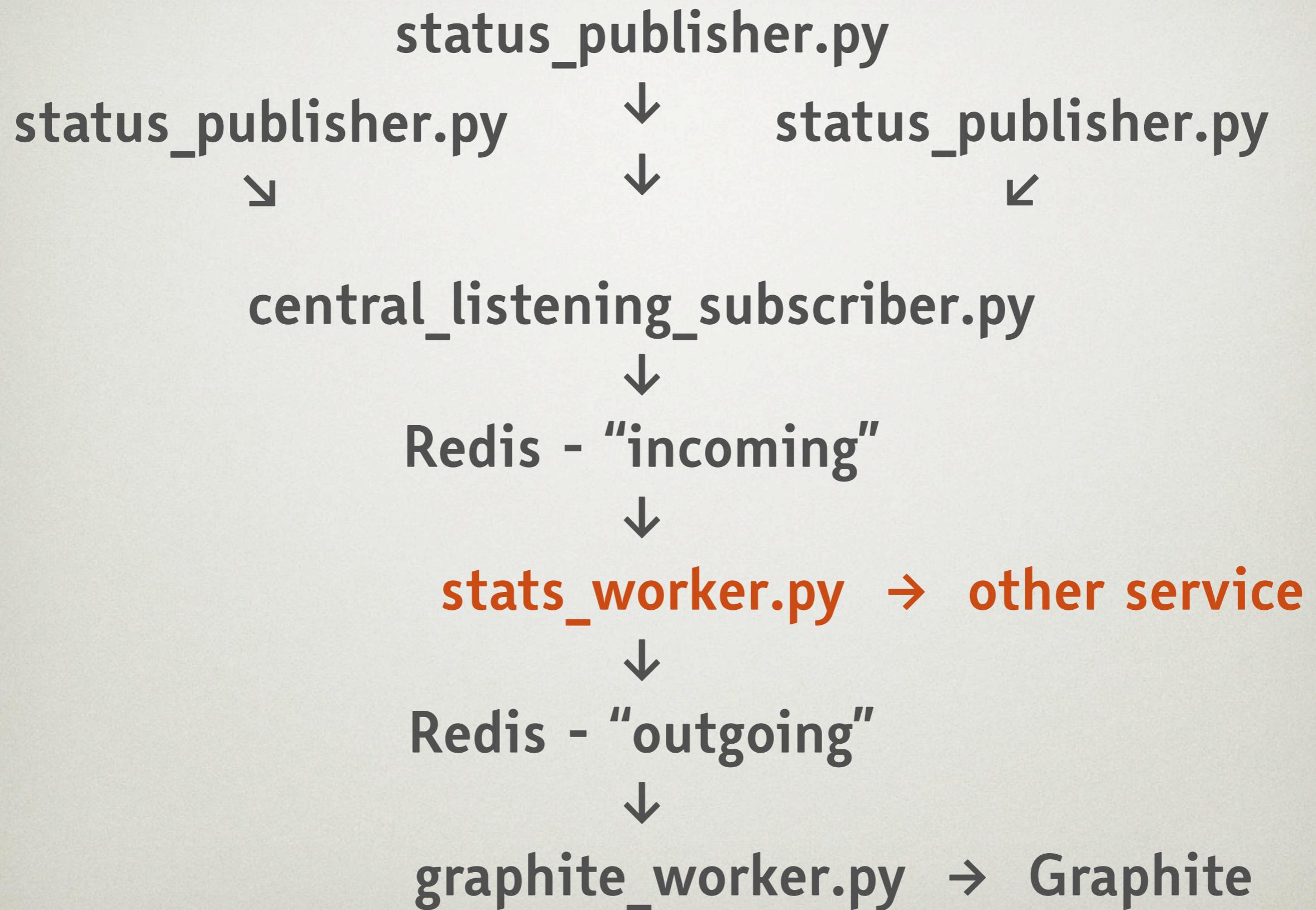
extending:

new actions on status



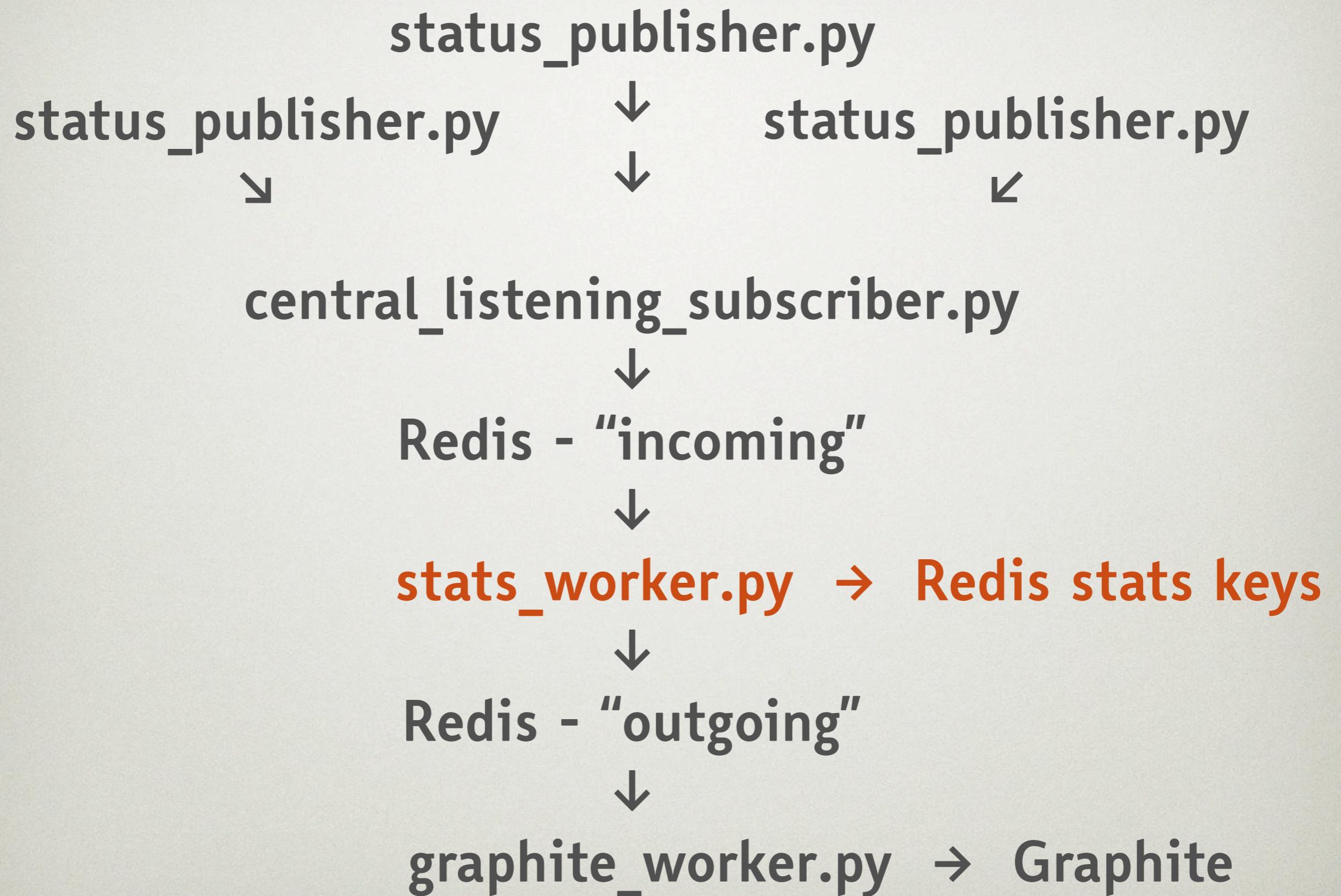
extending:

publish to other services



extending:

aggregate/precalculate
cache for other apps



Thank you!

Questions?

Code and slides:

https://github.com/briandorsey/metrics_talk