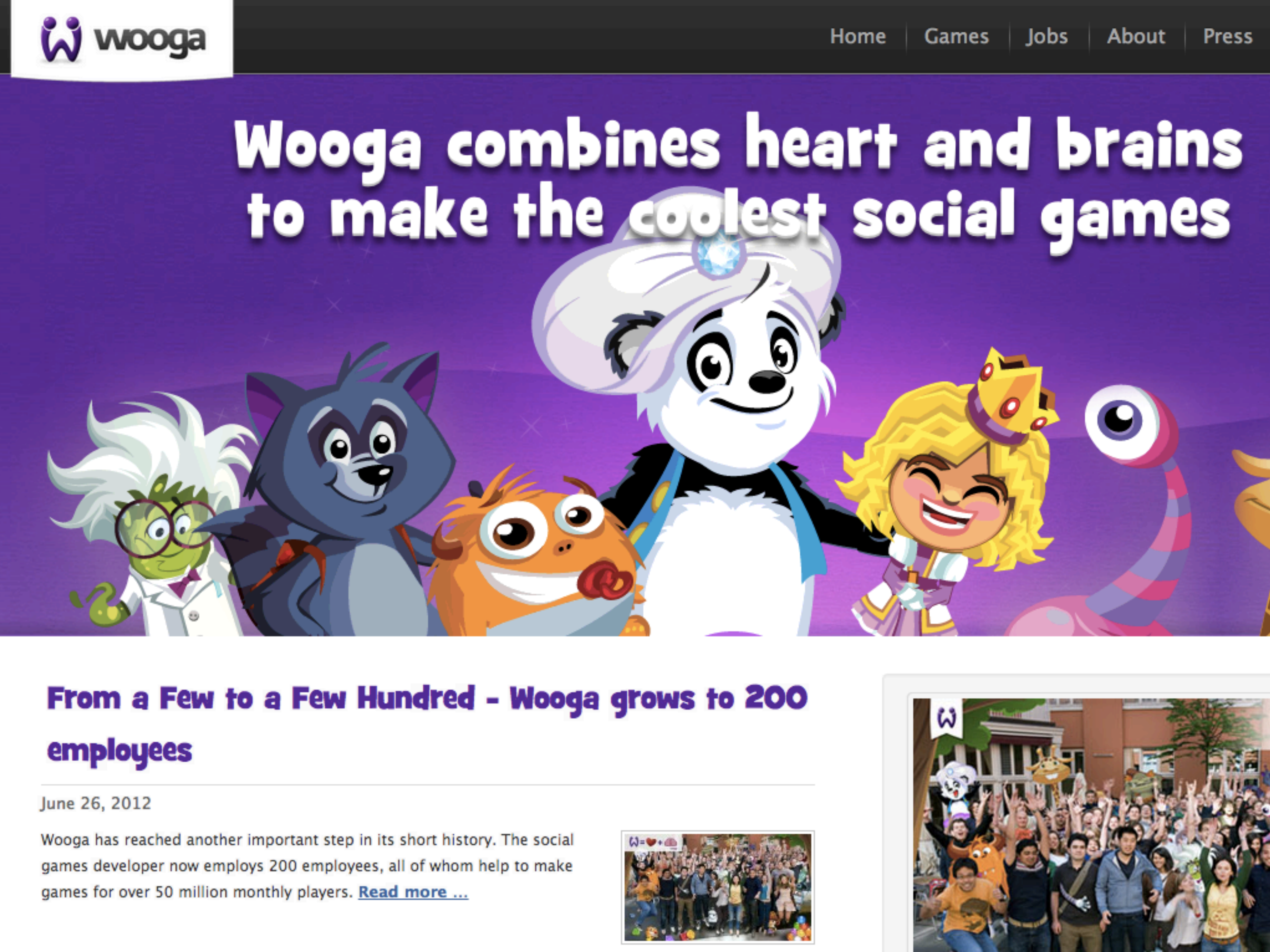
Dashboard Mania

Tim Lossen · @tlossen · Wooga

flickr.com/photos/jbgeronimi/6363087361/

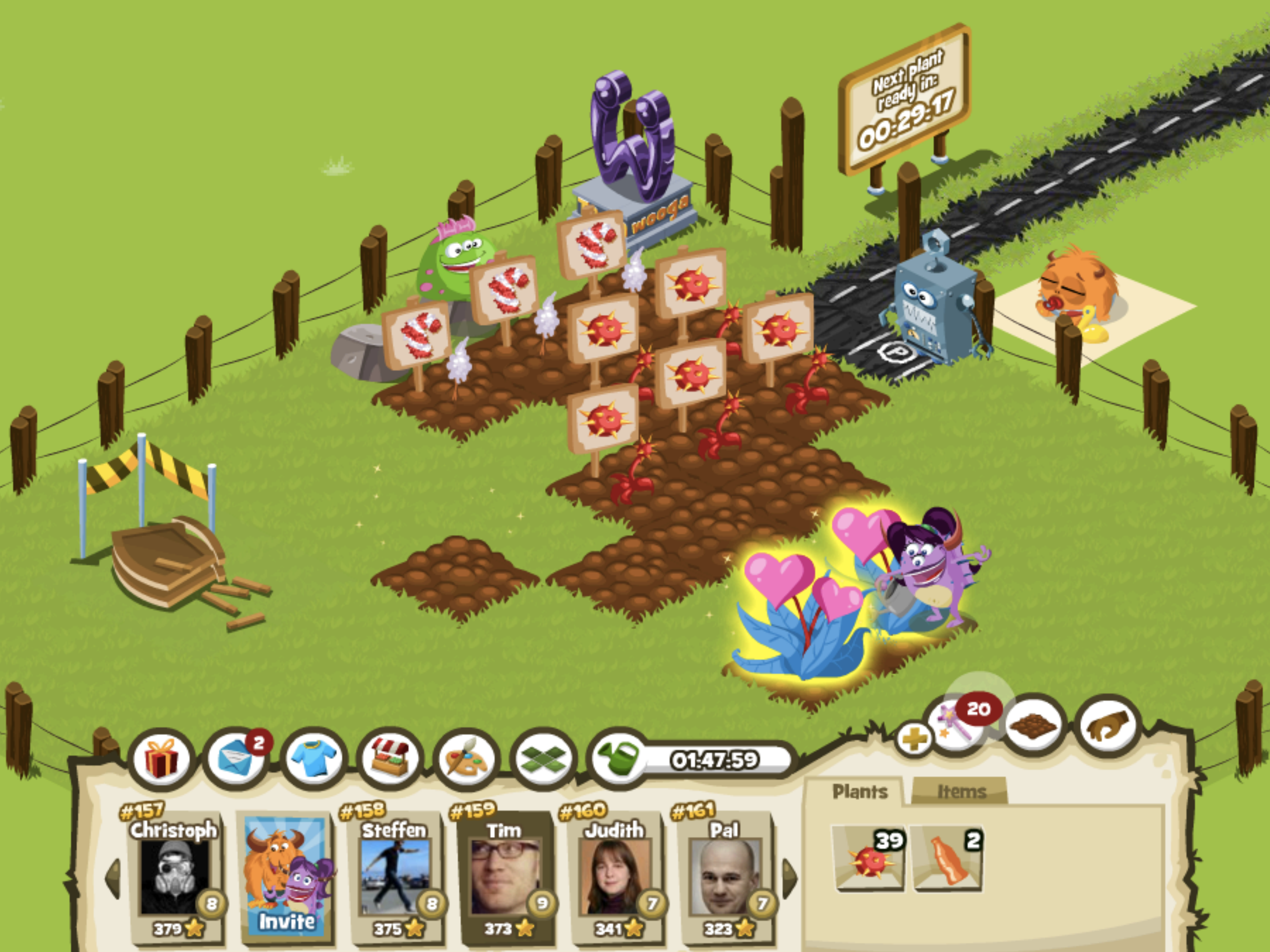


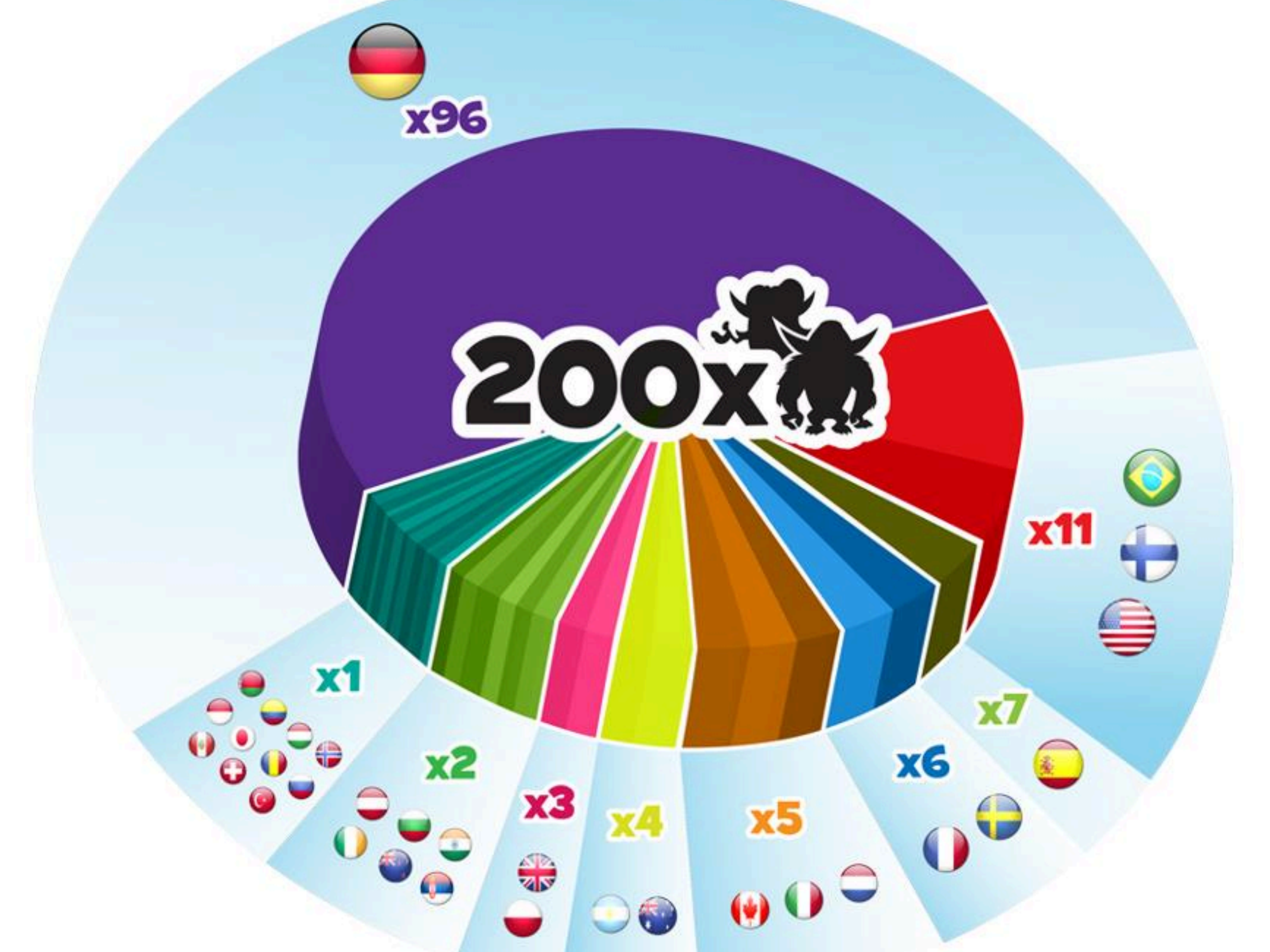
Wooga combines heart and brains to make the coolest social games

From a Few to a Few Hundred - Wooga grows to 200 employees

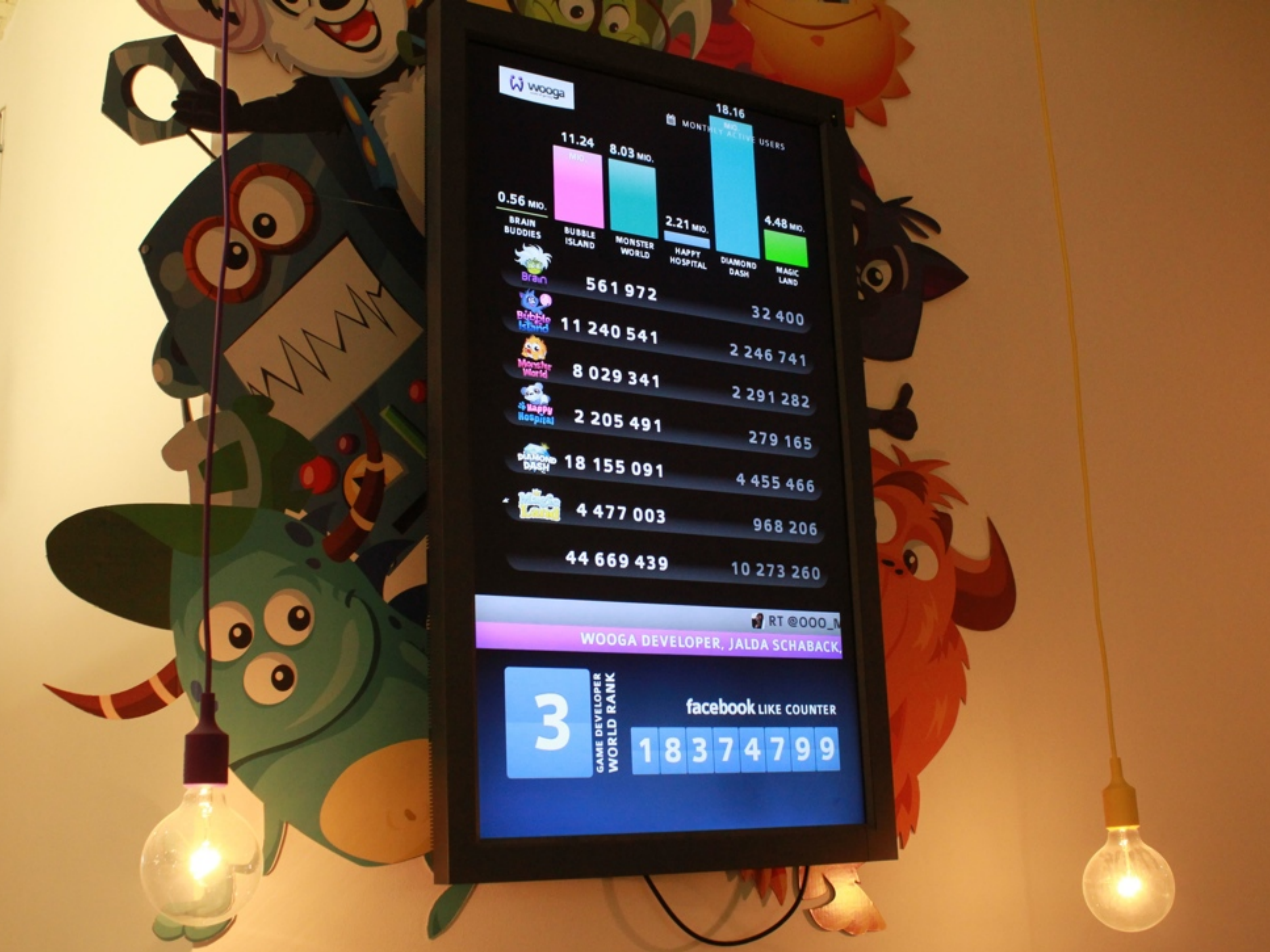
June 26, 2012

Wooga has reached another important step in its short history. The social games developer now employs 200 employees, all of whom help to make games for over 50 million monthly players. Read more ...


Next plant ready in:
00:29:17
wooga
P
2
01:47:59
20
Plants
Items
39
2
#157
Christoph
8
379
Invite
#158
Steffen
8
375
#159
Tim
9
373
#160
Judith
7
341
#161
Pal
7
323


x96
200x
x11
x7
x6
x5
x4
x3
x2
x1

Examples



Brain	561 972	32 400
Bubble Island	11 240 541	2 246 741
Monster World	8 029 341	2 291 282
Happy Hospital	2 205 491	279 165
Diamond Dash	18 155 091	4 455 466
Magic Land	4 477 003	968 206
	44 669 439	10 273 260

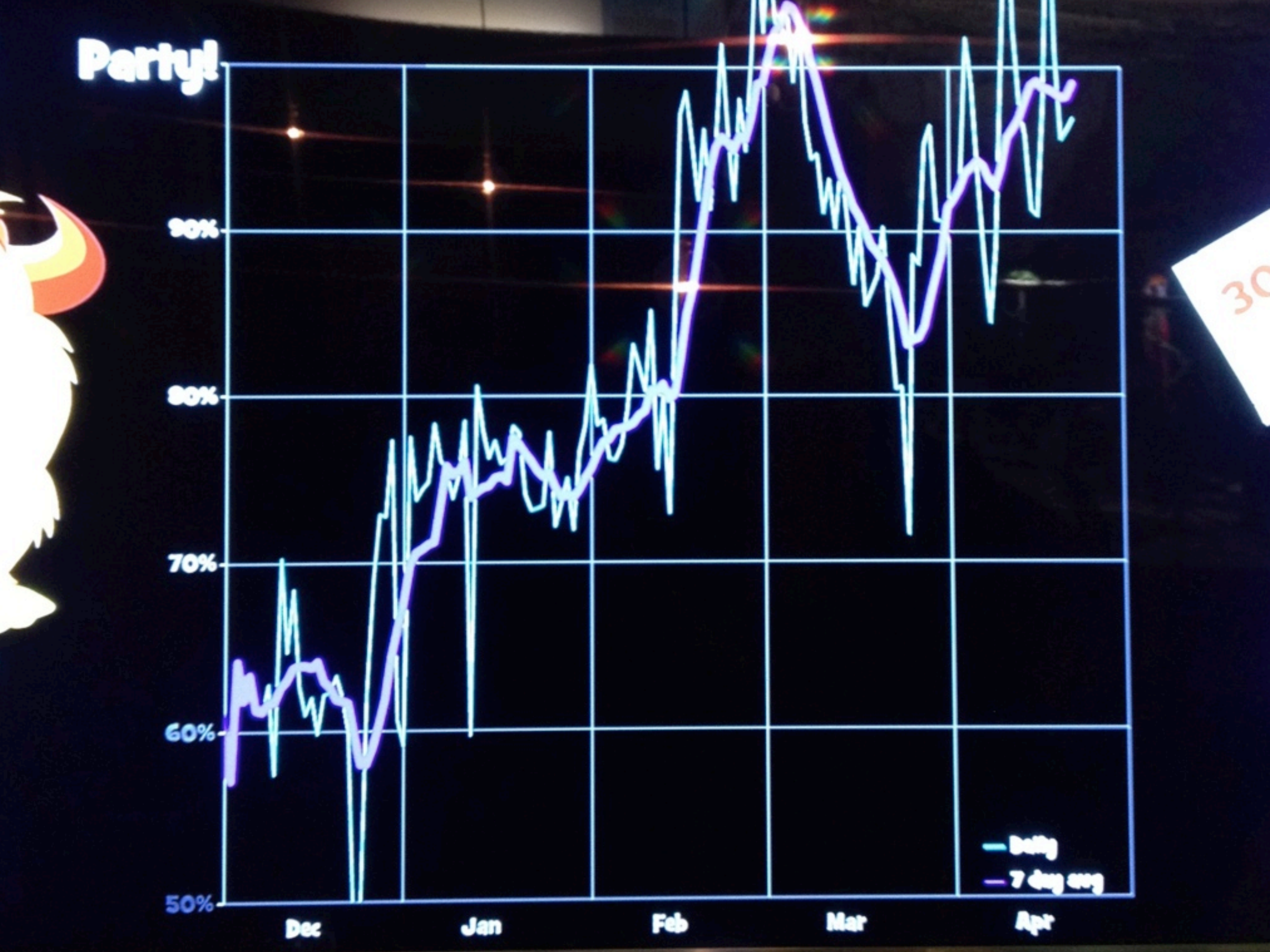
RT @OOO_M

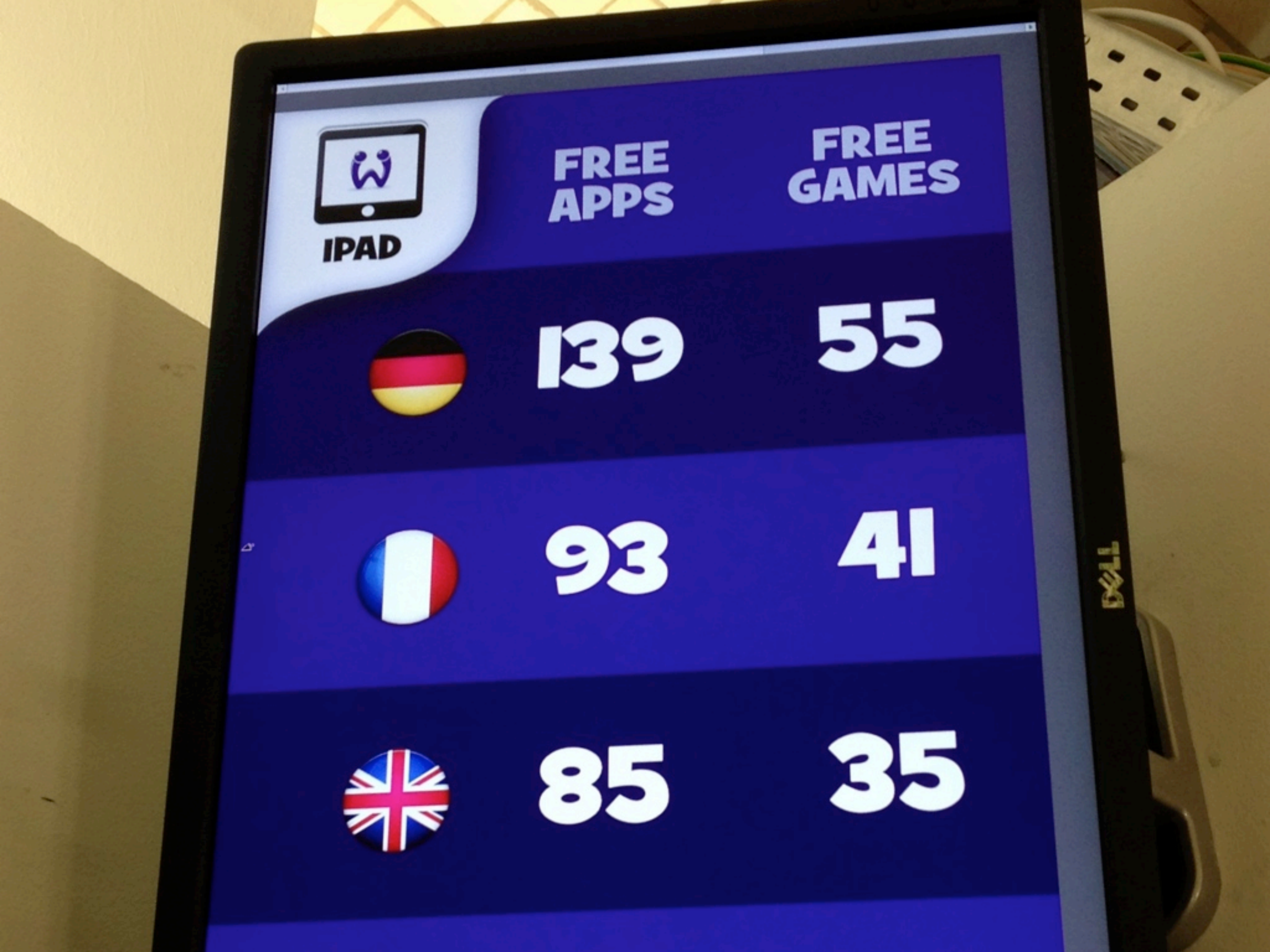
WOOGA DEVELOPER, JALDA SCHABACK,

3 GAME DEVELOPER WORLD RANK

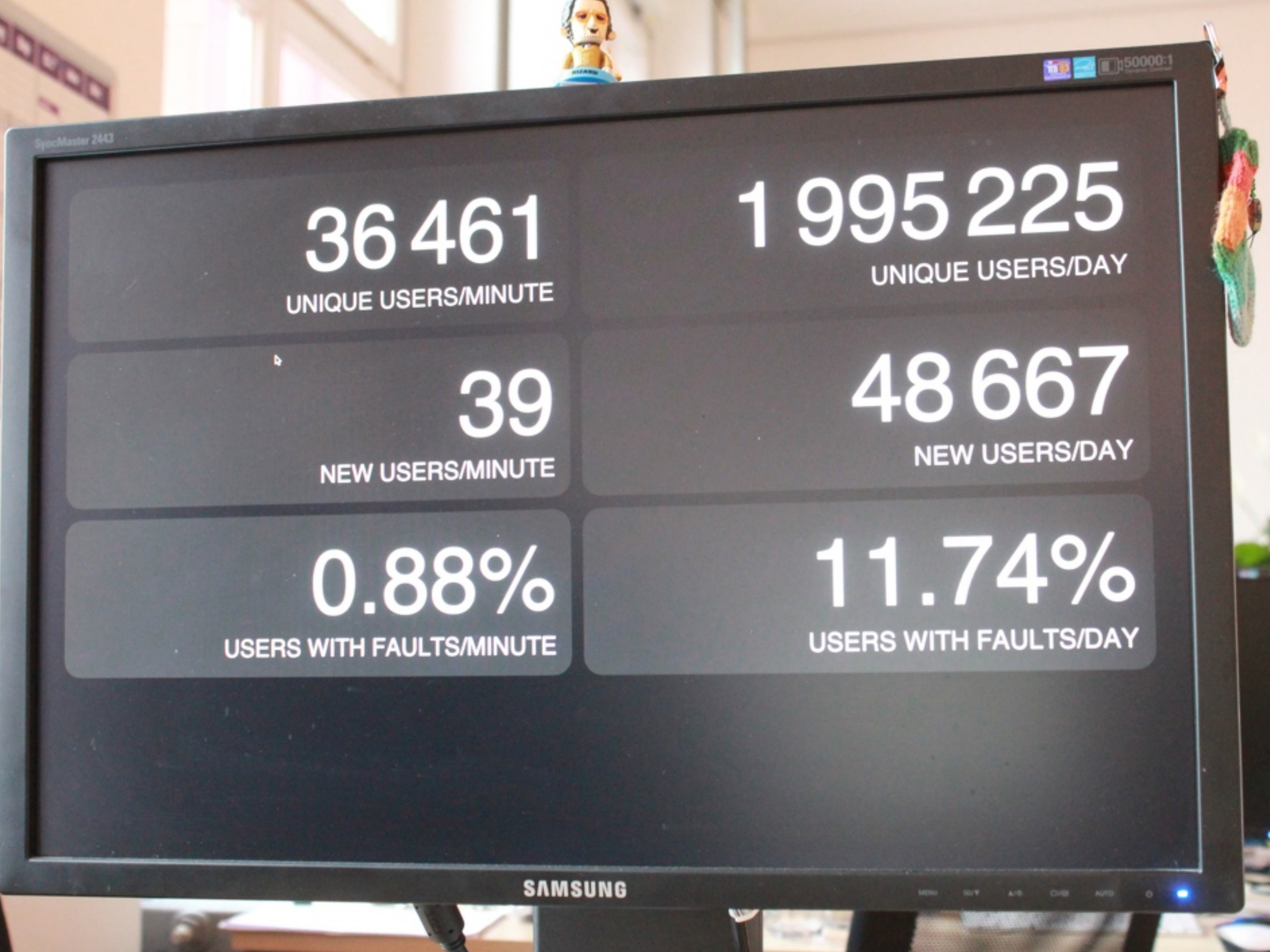
facebook LIKE COUNTER

18374799


Party!
90%
80%
70%
60%
50%
Dec
Jan
Feb
Mar
Apr
Daily
7 day avg
30



IPAD	FREE APPS	FREE GAMES
🇩🇪	139	55
🇫🇷	93	41
🇬🇧	85	35



36 461

UNIQUE USERS/MINUTE

1 995 225

UNIQUE USERS/DAY

39

NEW USERS/MINUTE

48 667

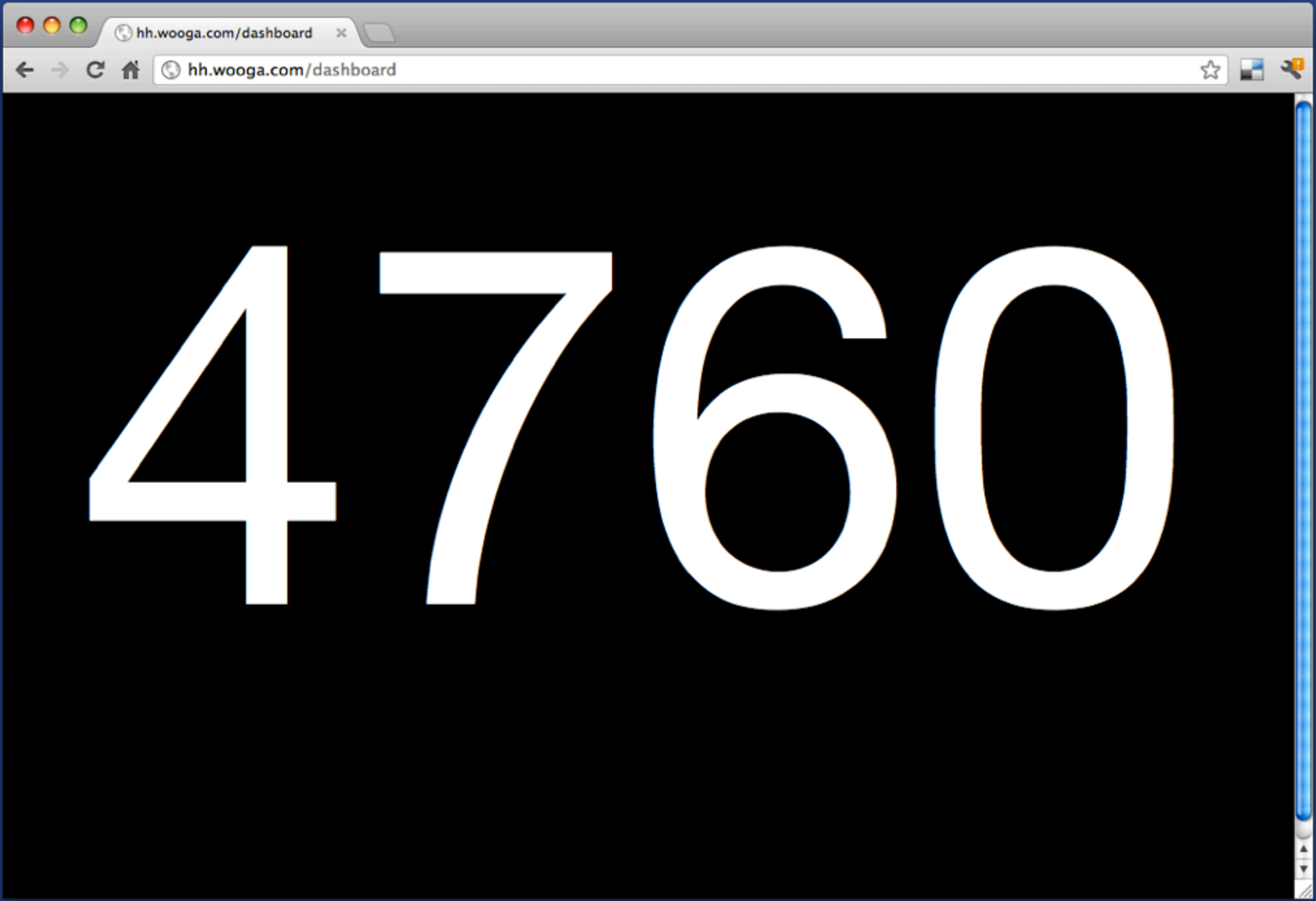
NEW USERS/DAY

0.88%

USERS WITH FAULTS/MINUTE

11.74%

USERS WITH FAULTS/DAY



4760

*“A good **information radiator***

- *is large and easily visible*
- *is understood at a glance*
- *changes periodically”*

— Alistair Cockburn

alistair.cockburn.us/Information+radiator

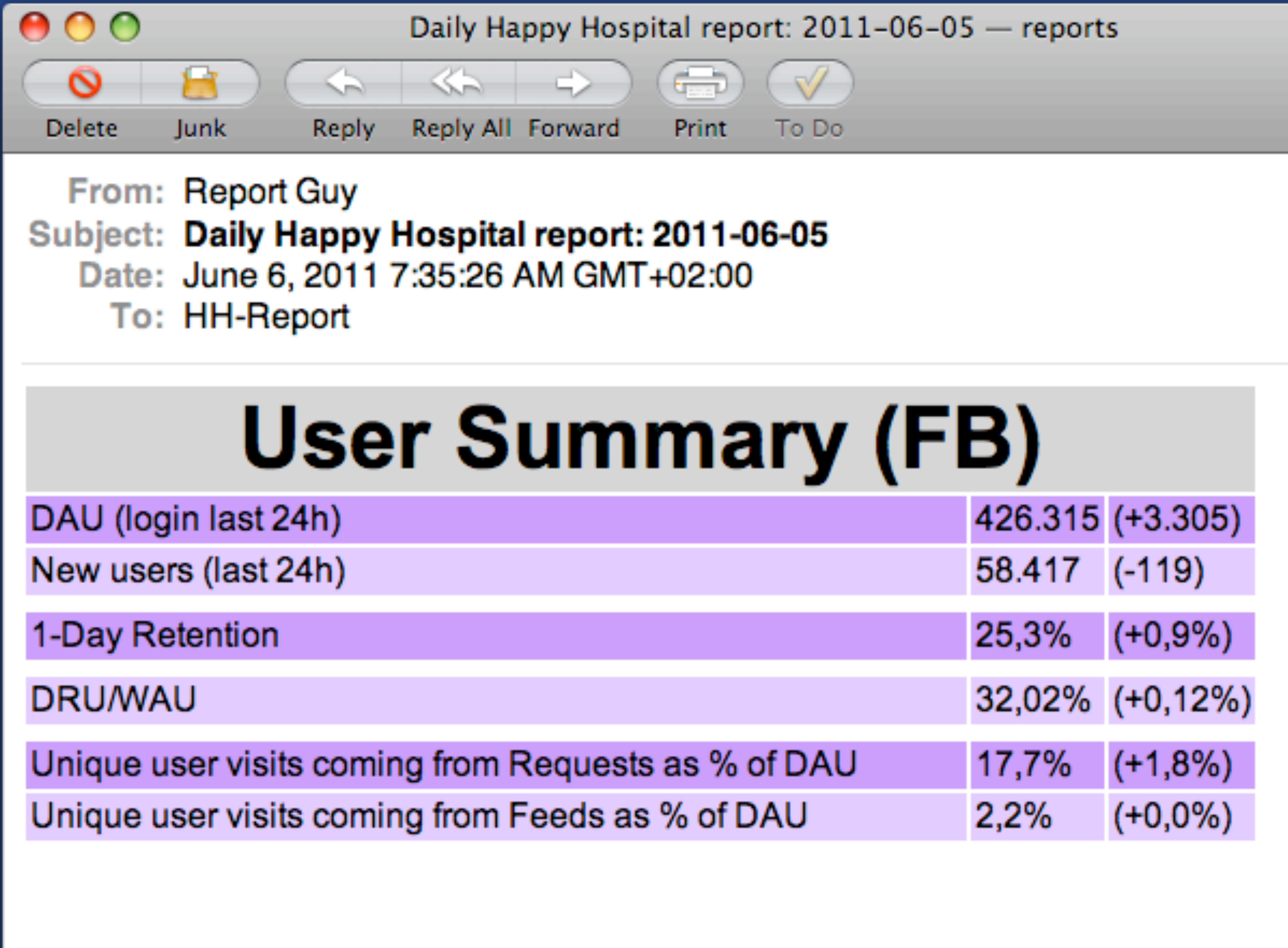
“A good ***information radiator***

- *is large and easily visible*
- *is* ***understood at a glance***
- *changes periodically”*

— Alistair Cockburn

alistair.cockburn.us/Information+radiator

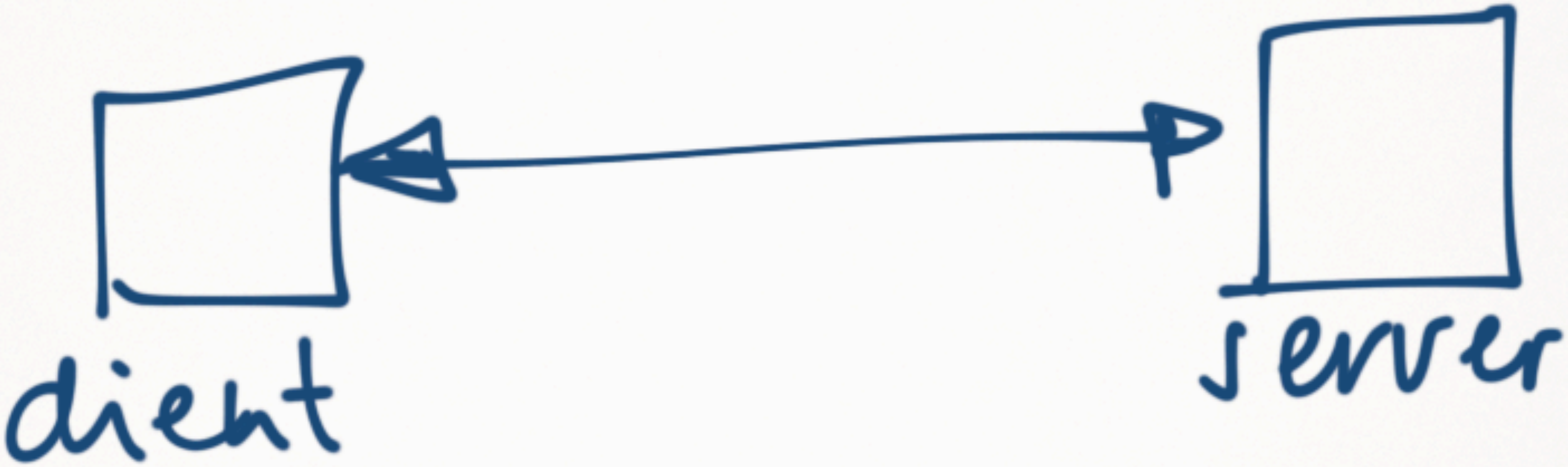
Tracking

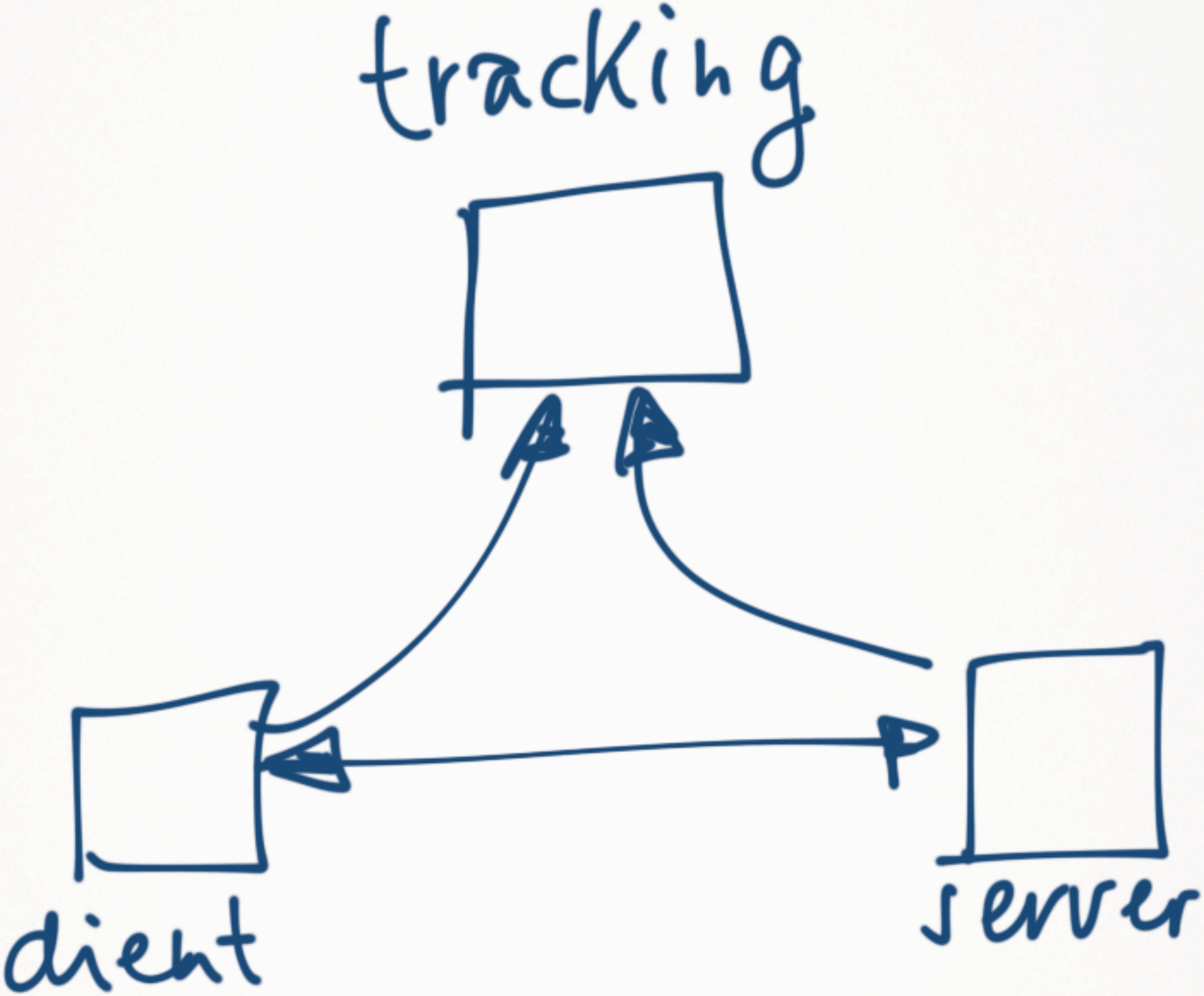


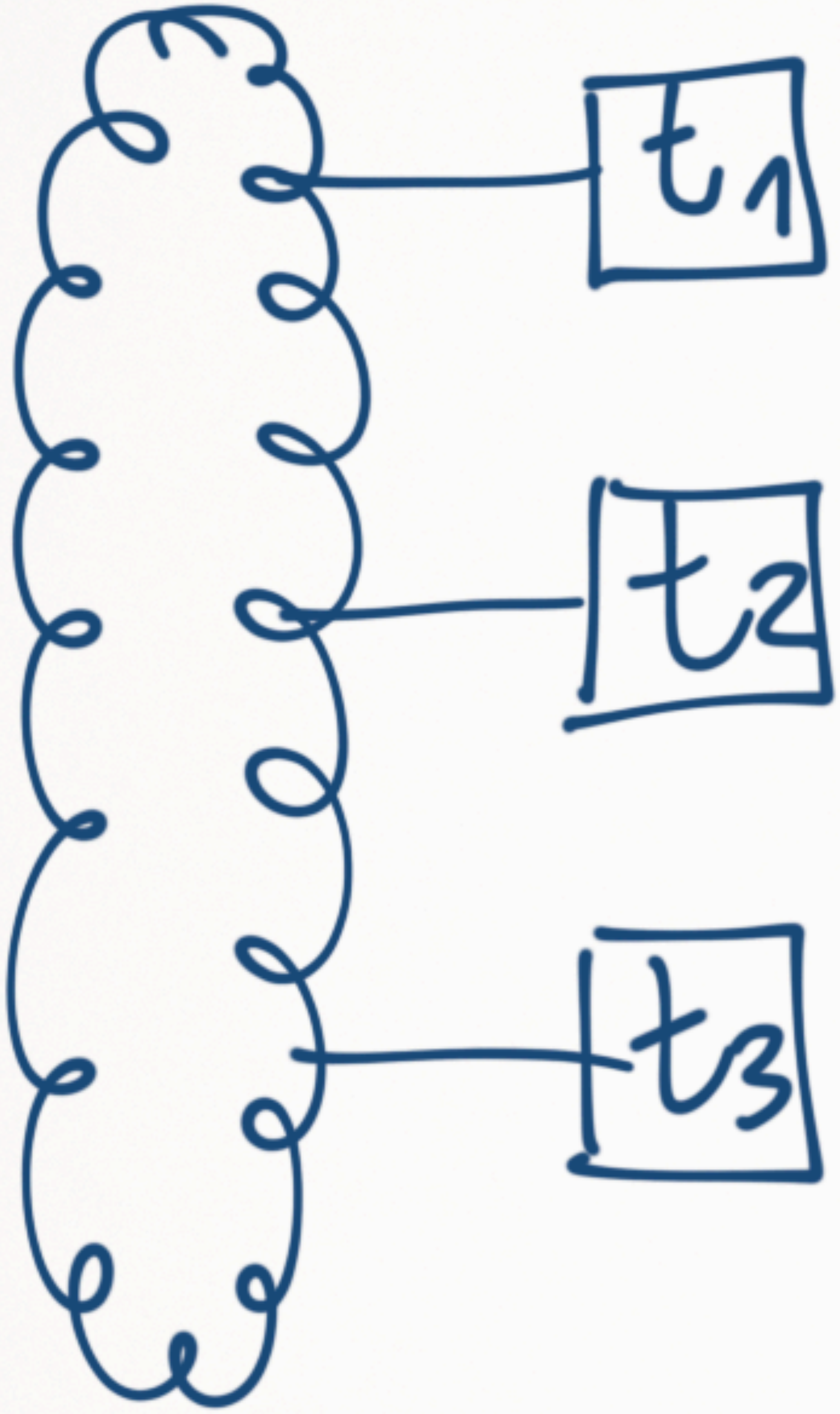
From: Report Guy
Subject: **Daily Happy Hospital report: 2011-06-05**
Date: June 6, 2011 7:35:26 AM GMT+02:00
To: HH-Report

User Summary (FB)

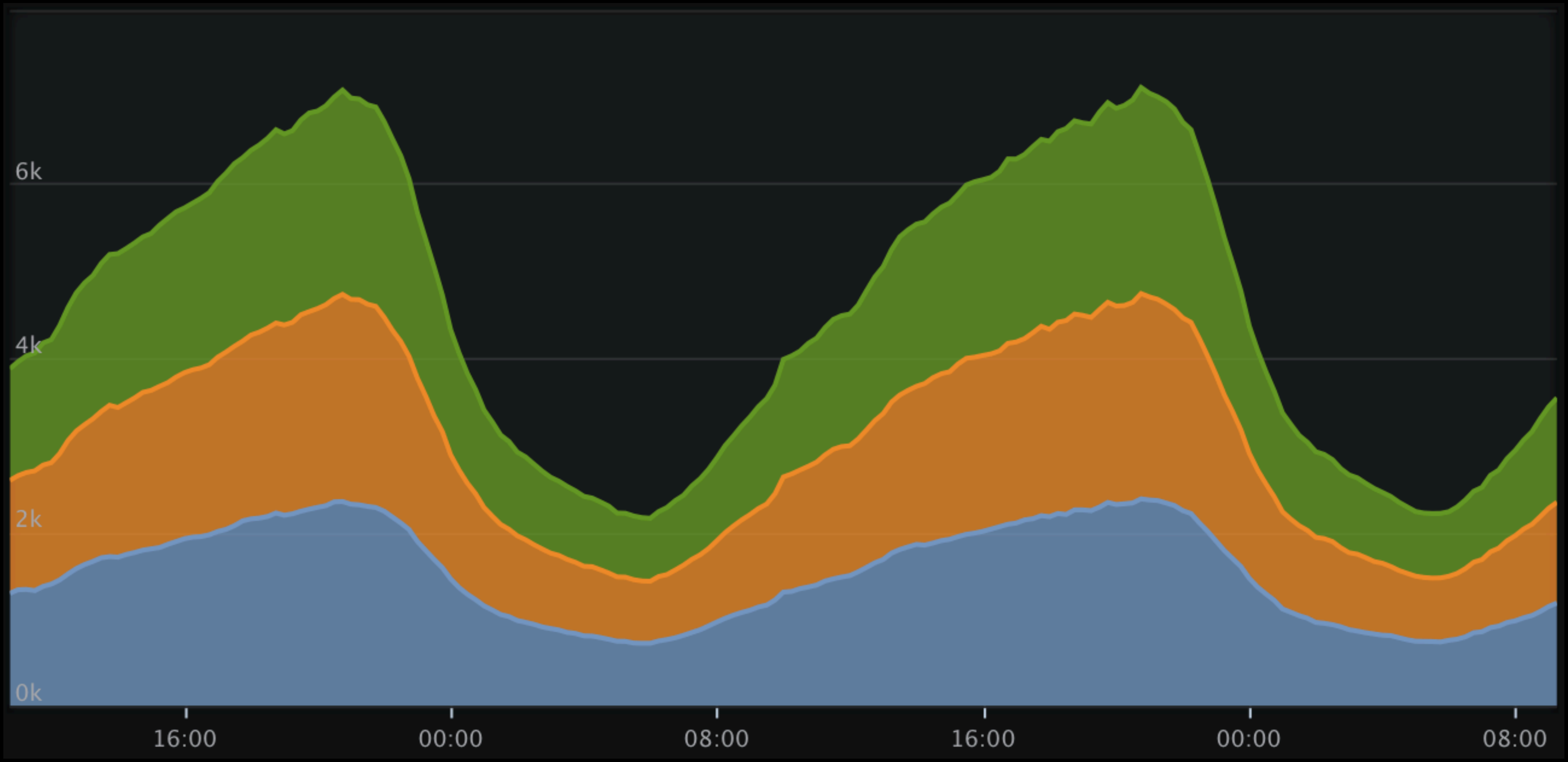
DAU (login last 24h)	426.315	(+3.305)
New users (last 24h)	58.417	(-119)
1-Day Retention	25,3%	(+0,9%)
DRU/WAU	32,02%	(+0,12%)
Unique user visits coming from Requests as % of DAU	17,7%	(+1,8%)
Unique user visits coming from Feeds as % of DAU	2,2%	(+0,0%)

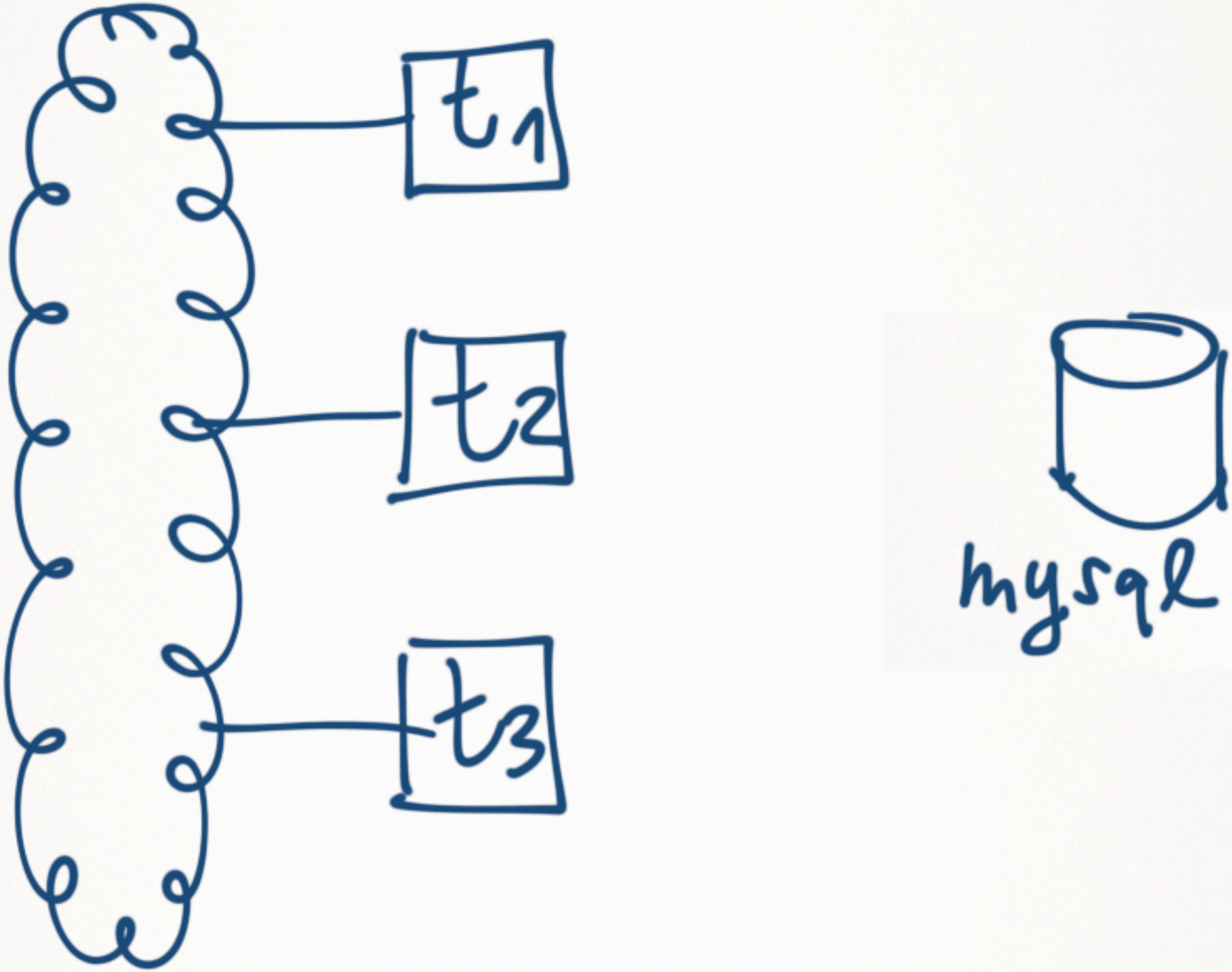

client
server

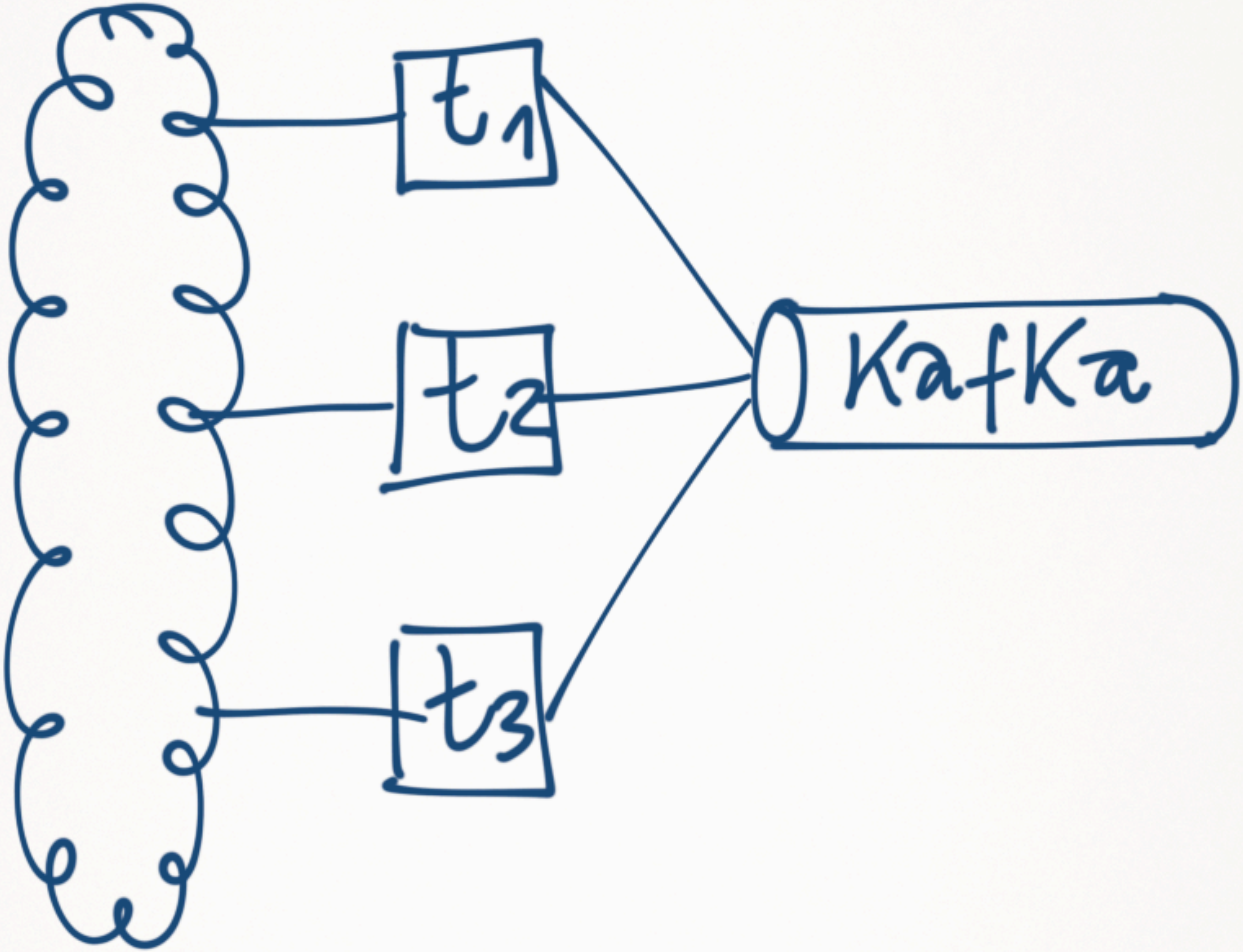

tracking
client
server

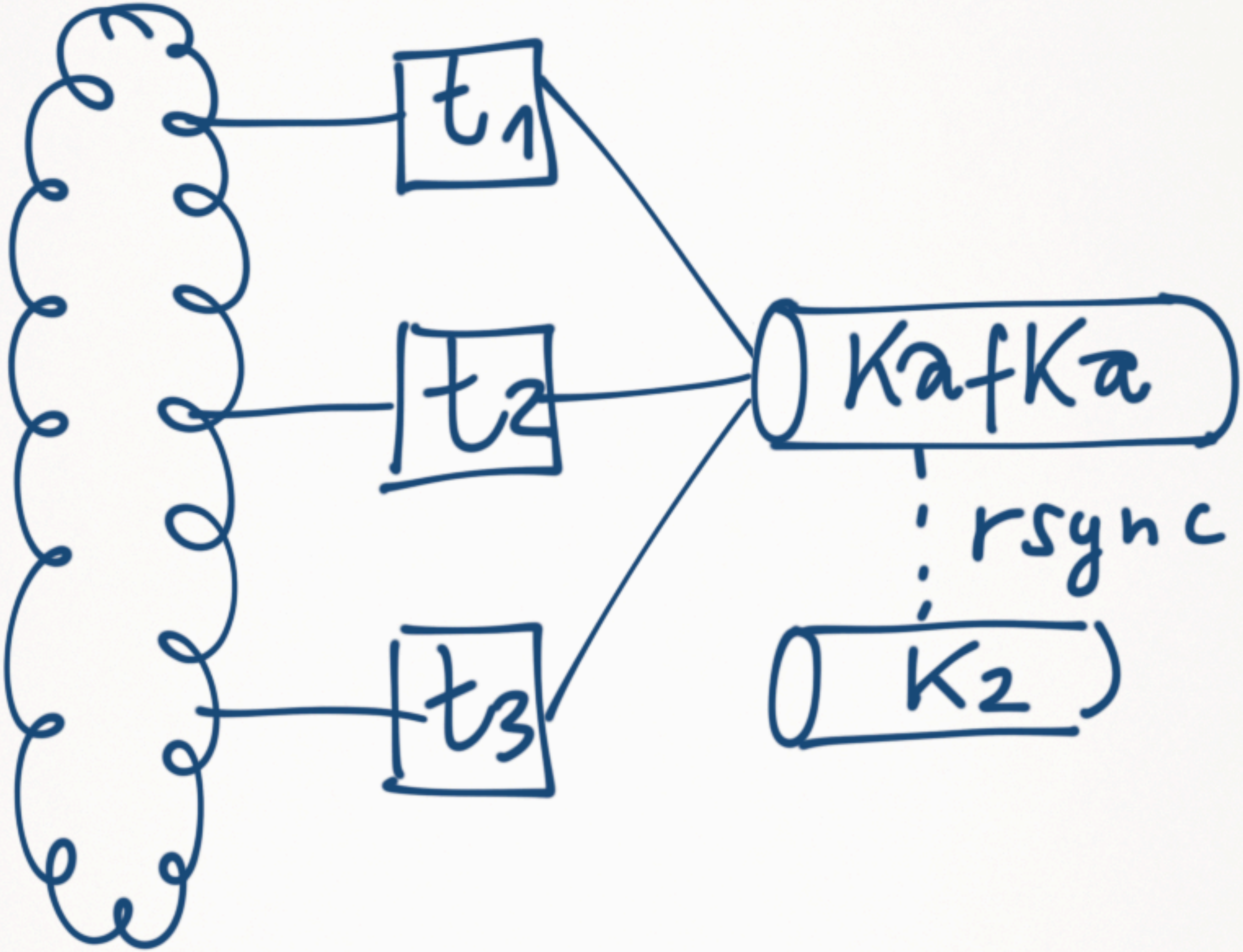

t_1
t_2
t_3

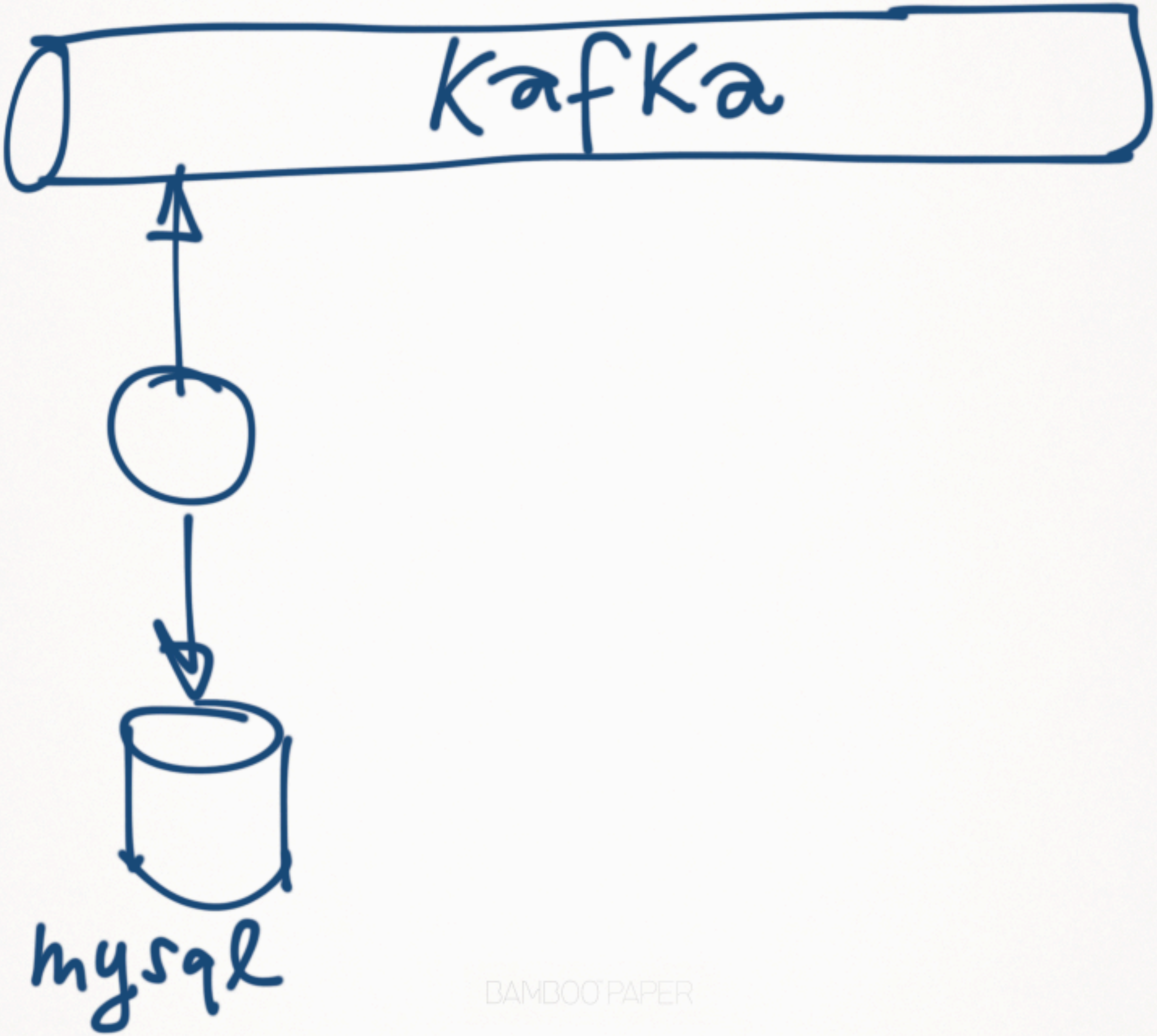
requests per second

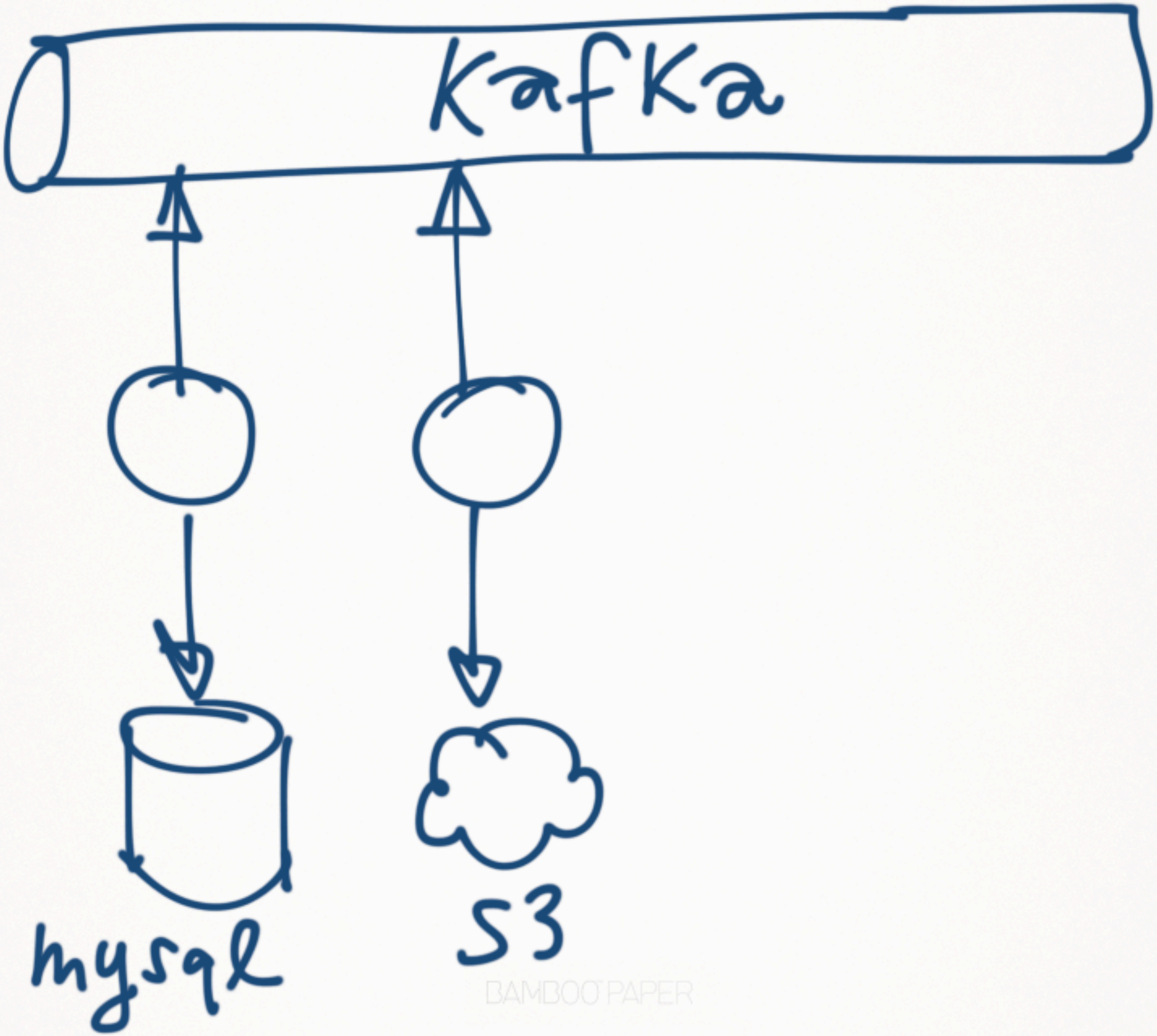


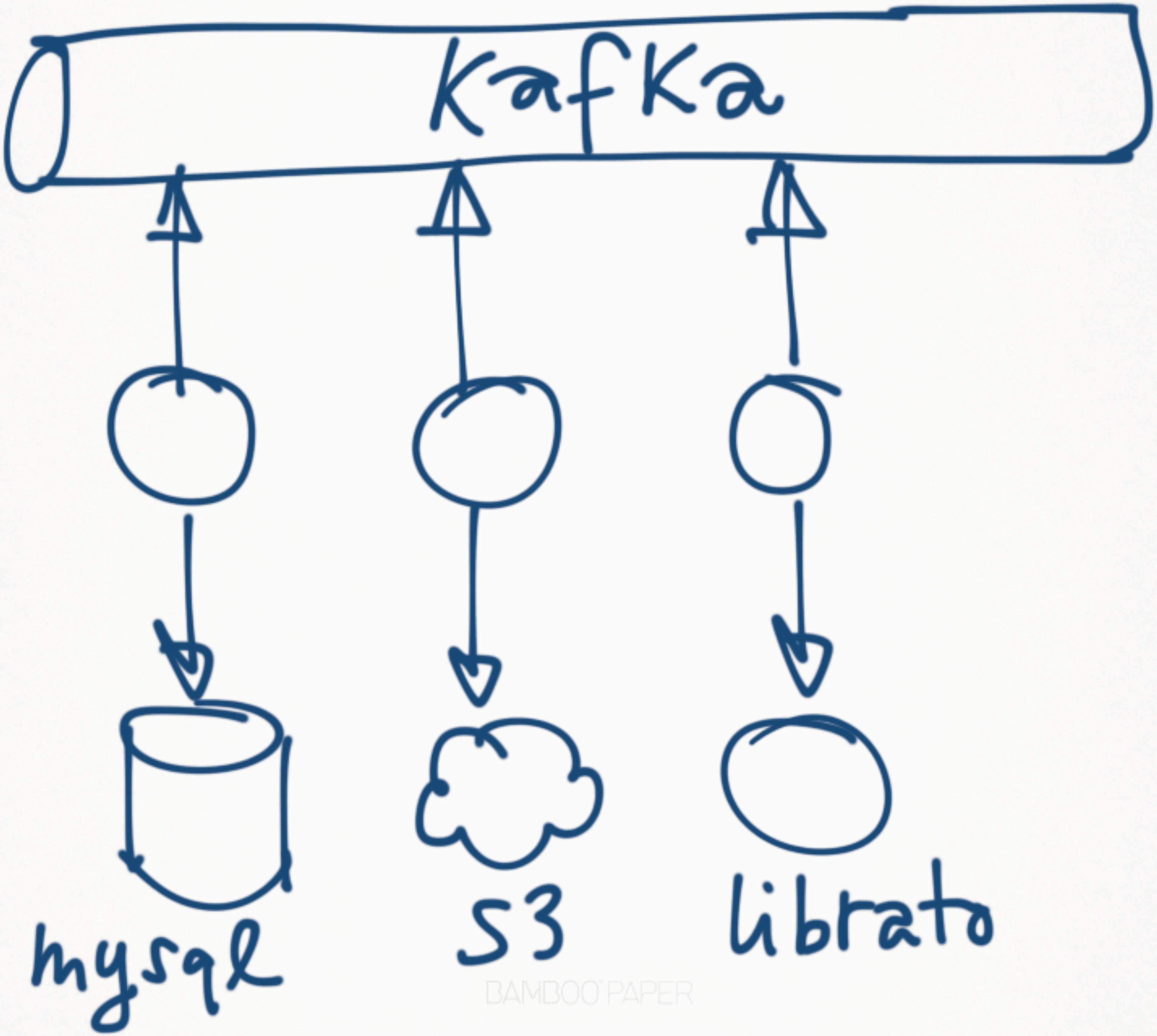

t_1
t_2
t_3
mysql


t_1
t_2
t_3
Kafka

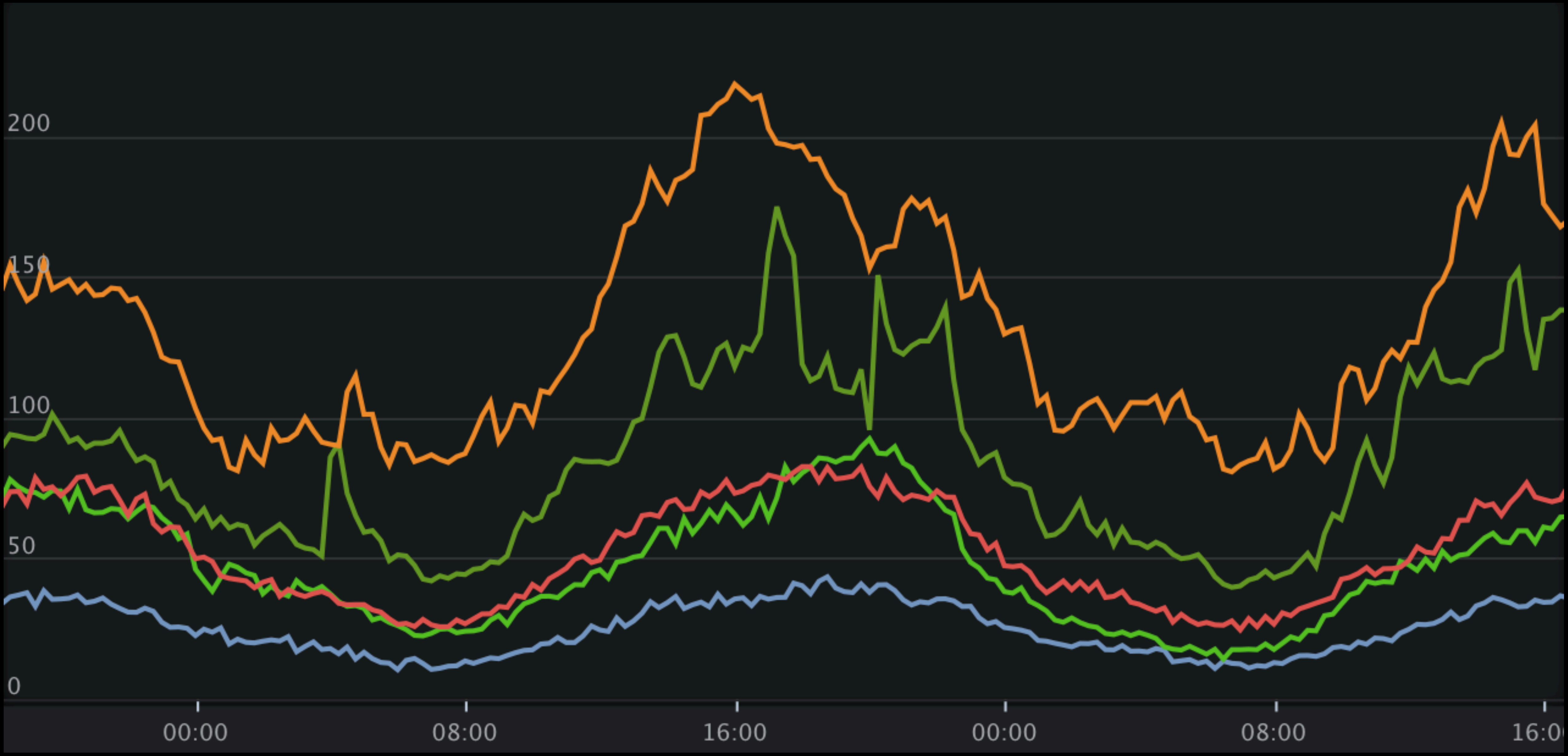

t_1
t_2
t_3
Kafka
rsync
K_2

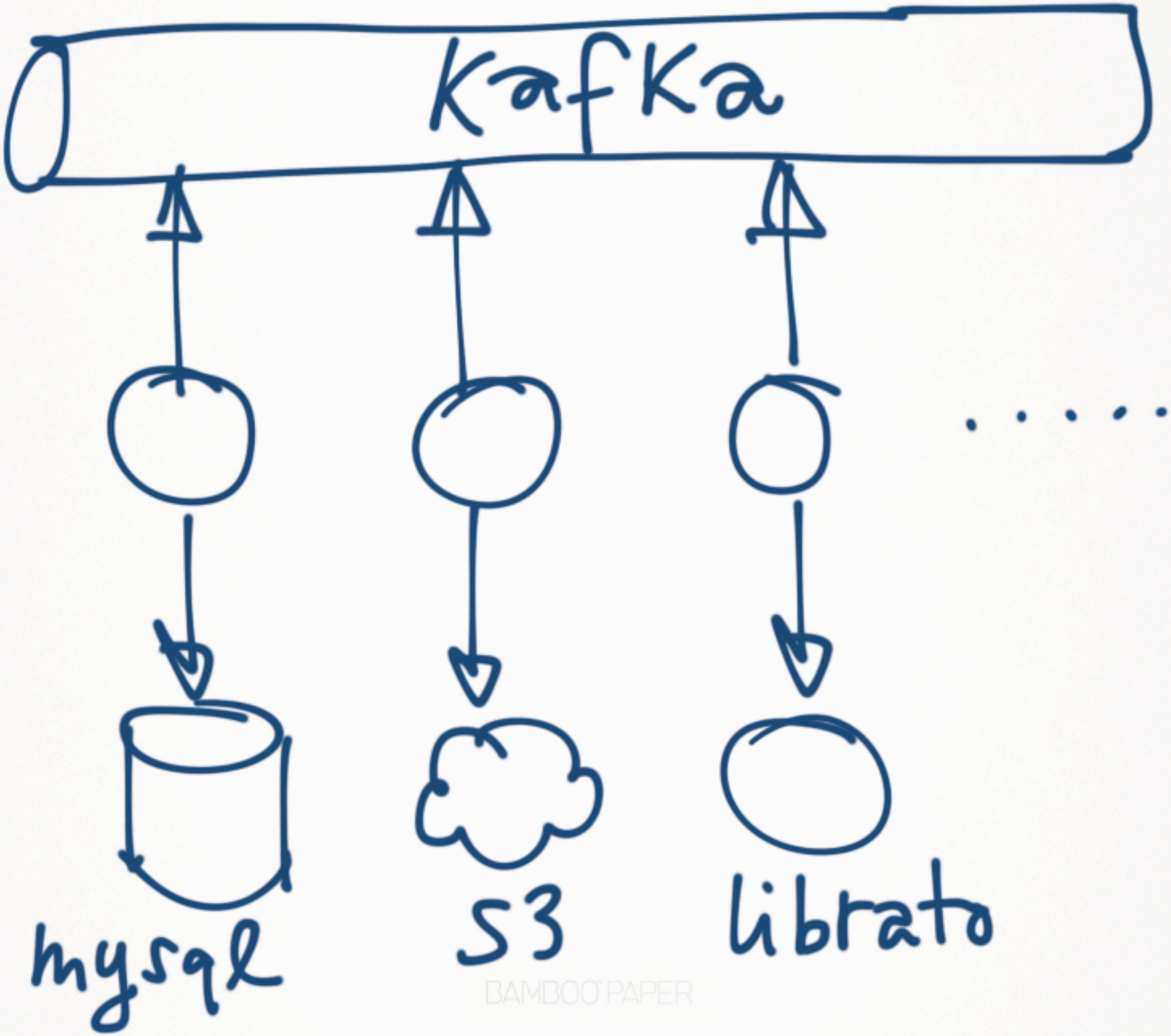

kafka
mysql


kafka
mysql
S3

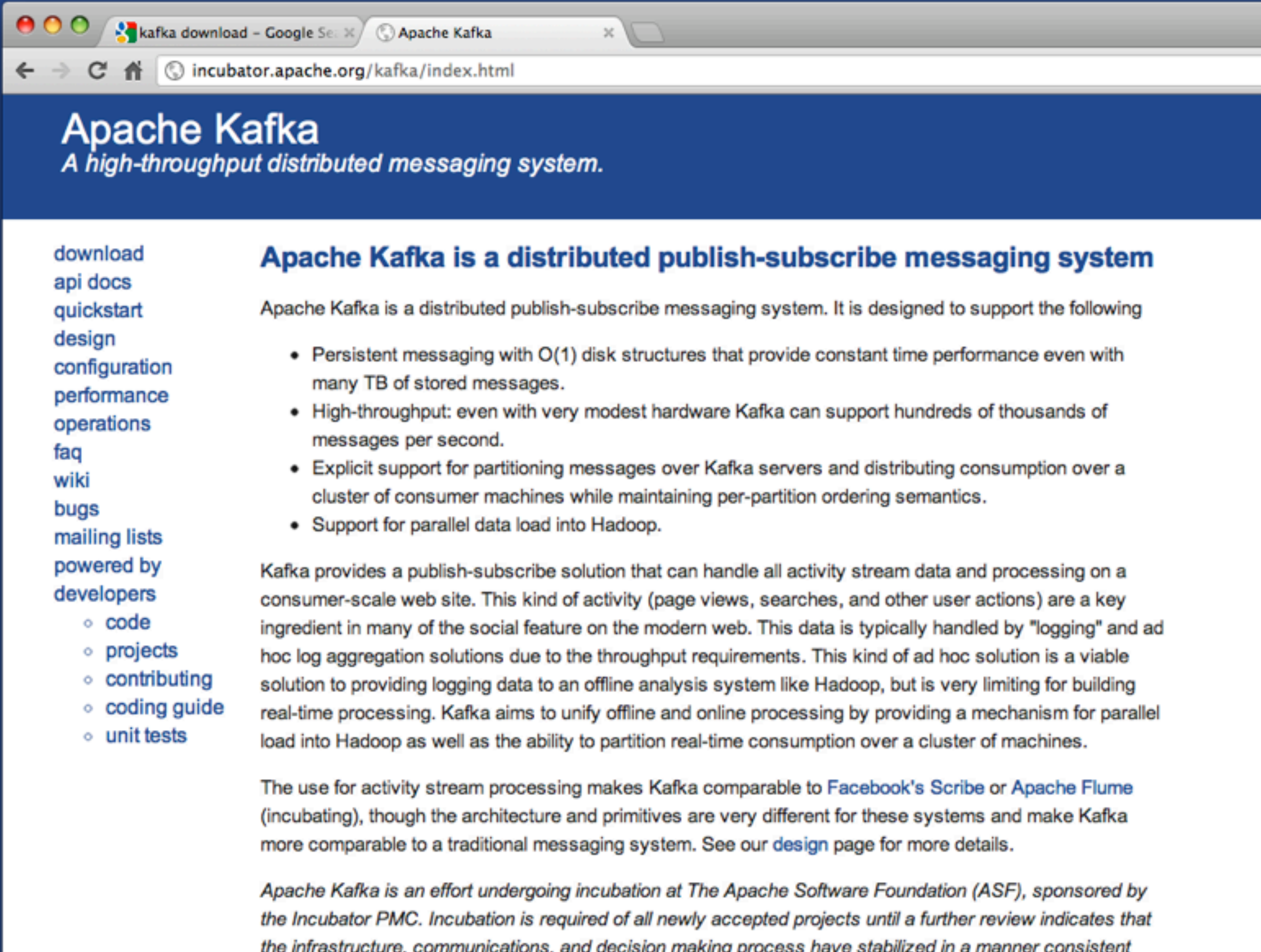

kafka
mysql
S3
librato

installs per minute




kafka
. . . . .
mysql
S3
librato

Kafka


kafka download - Google Se
Apache Kafka
incubator.apache.org/kafka/index.html

Apache Kafka

A high-throughput distributed messaging system.

- download
- api docs
- quickstart
- design
- configuration
- performance
- operations
- faq
- wiki
- bugs
- mailing lists
- powered by
- developers
  - code
  - projects
  - contributing
  - coding guide
  - unit tests

Apache Kafka is a distributed publish-subscribe messaging system

Apache Kafka is a distributed publish-subscribe messaging system. It is designed to support the following

- Persistent messaging with O(1) disk structures that provide constant time performance even with many TB of stored messages.
- High-throughput: even with very modest hardware Kafka can support hundreds of thousands of messages per second.
- Explicit support for partitioning messages over Kafka servers and distributing consumption over a cluster of consumer machines while maintaining per-partition ordering semantics.
- Support for parallel data load into Hadoop.

Kafka provides a publish-subscribe solution that can handle all activity stream data and processing on a consumer-scale web site. This kind of activity (page views, searches, and other user actions) are a key ingredient in many of the social feature on the modern web. This data is typically handled by "logging" and ad hoc log aggregation solutions due to the throughput requirements. This kind of ad hoc solution is a viable solution to providing logging data to an offline analysis system like Hadoop, but is very limiting for building real-time processing. Kafka aims to unify offline and online processing by providing a mechanism for parallel load into Hadoop as well as the ability to partition real-time consumption over a cluster of machines.

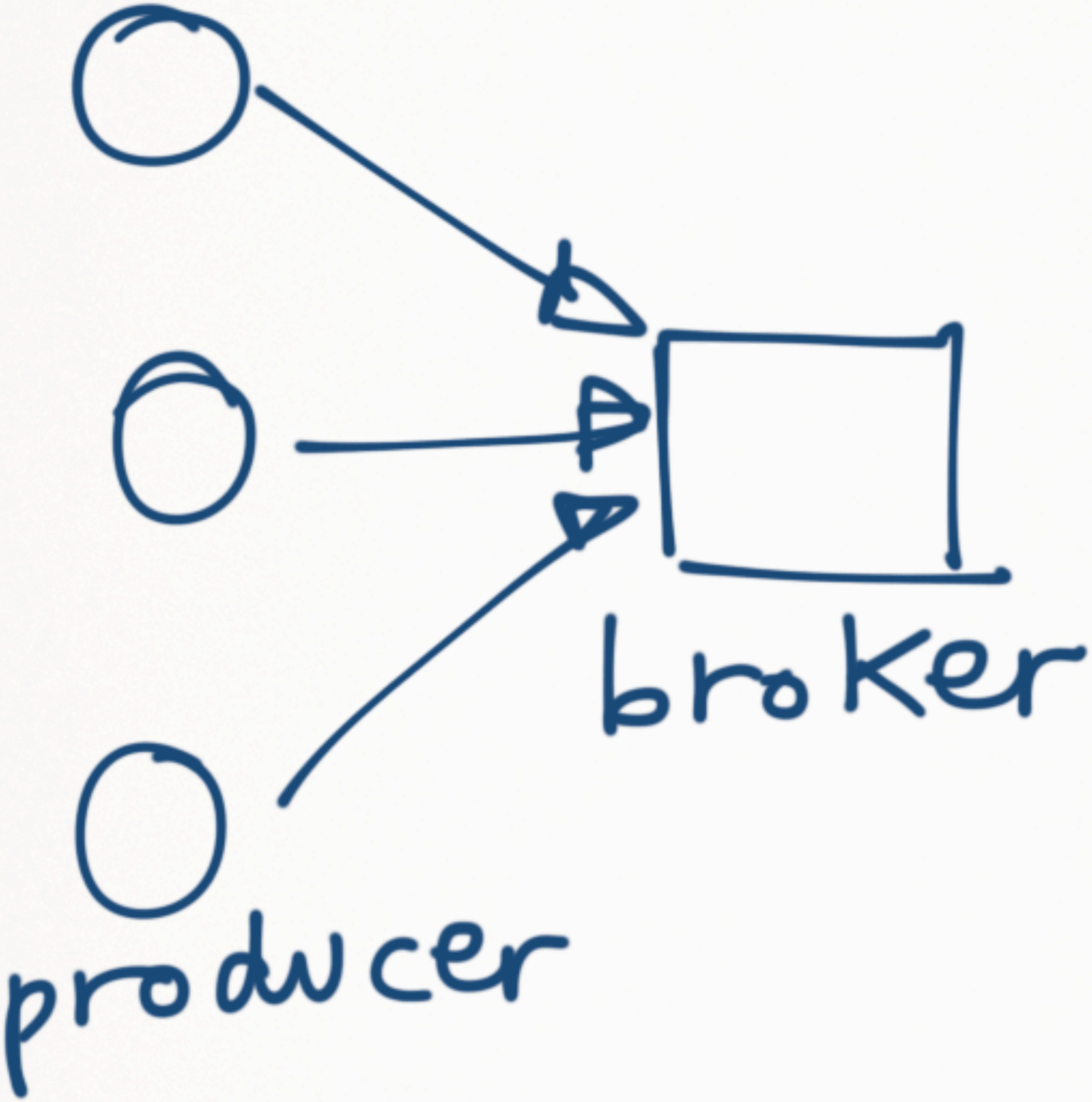
The use for activity stream processing makes Kafka comparable to Facebook's Scribe or Apache Flume (incubating), though the architecture and primitives are very different for these systems and make Kafka more comparable to a traditional messaging system. See our design page for more details.

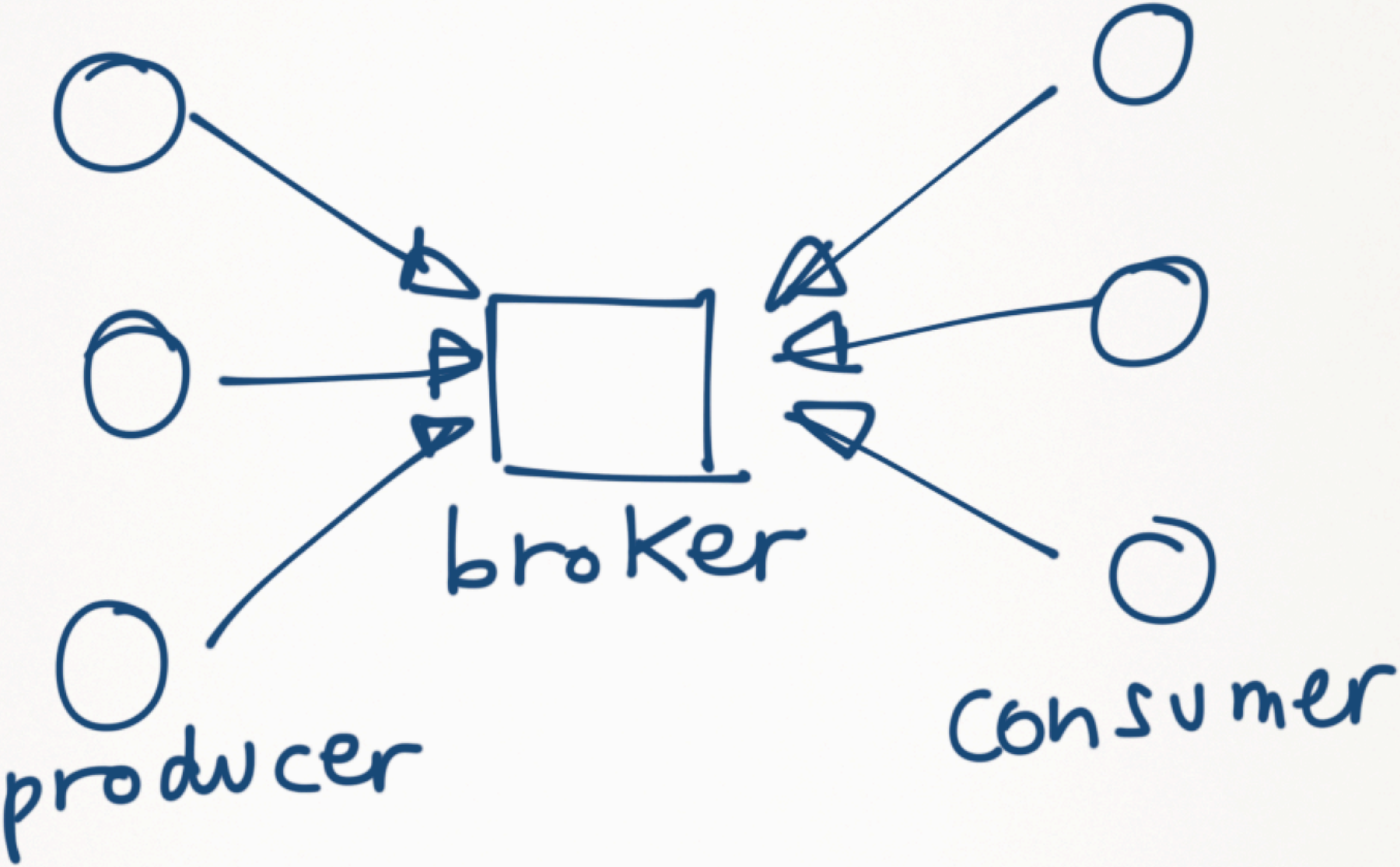
Apache Kafka is an effort undergoing incubation at The Apache Software Foundation (ASF), sponsored by the Incubator PMC. Incubation is required of all newly accepted projects until a further review indicates that the infrastructure, communications, and decision making process have stabilized in a manner consistent

high-throughput
persistent
publish-subscribe
messaging

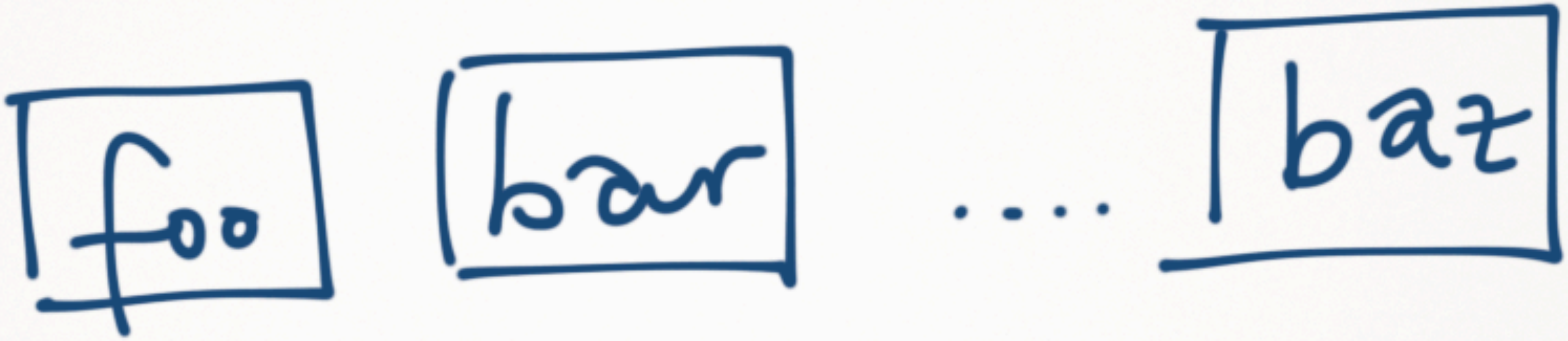
high-throughput
persistent
publish-subscribe
messaging

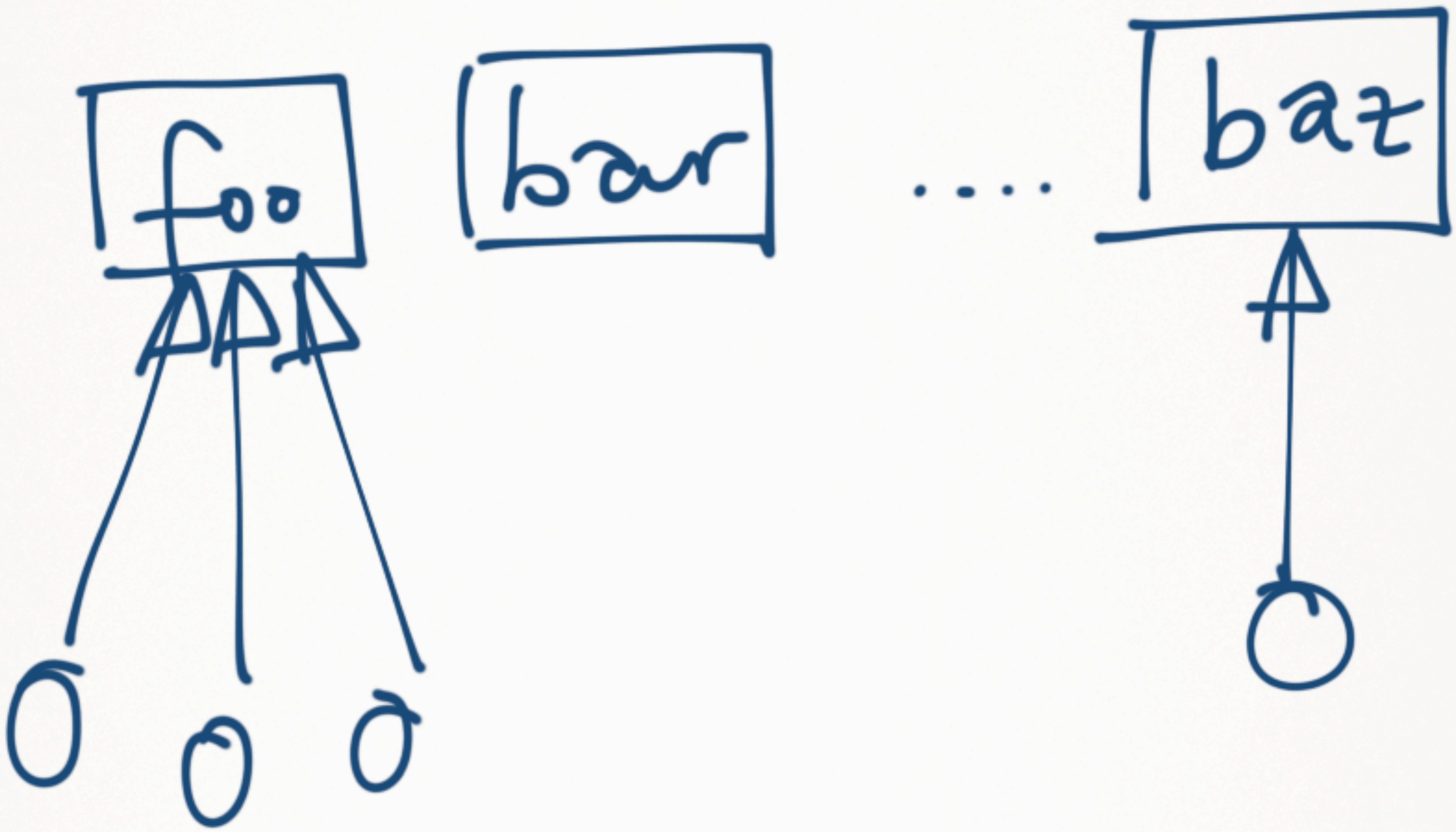
broker


broker
producer

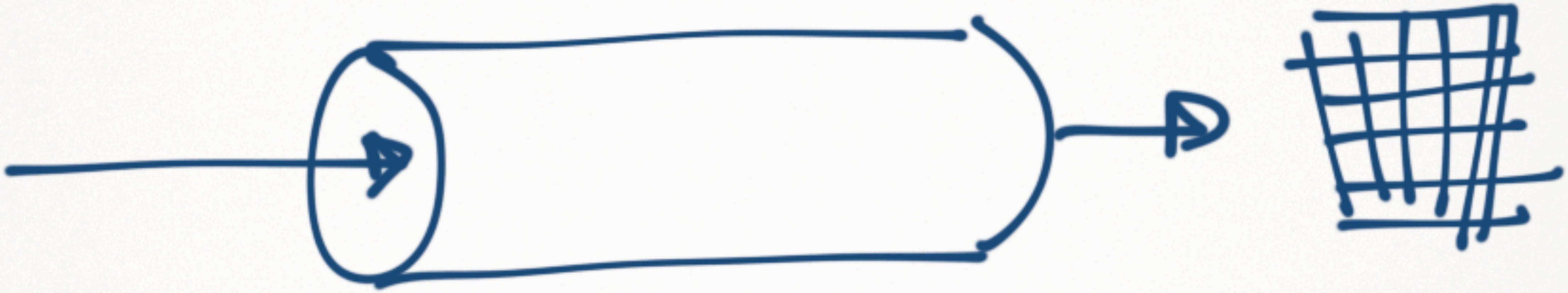

broker
producer
consumer

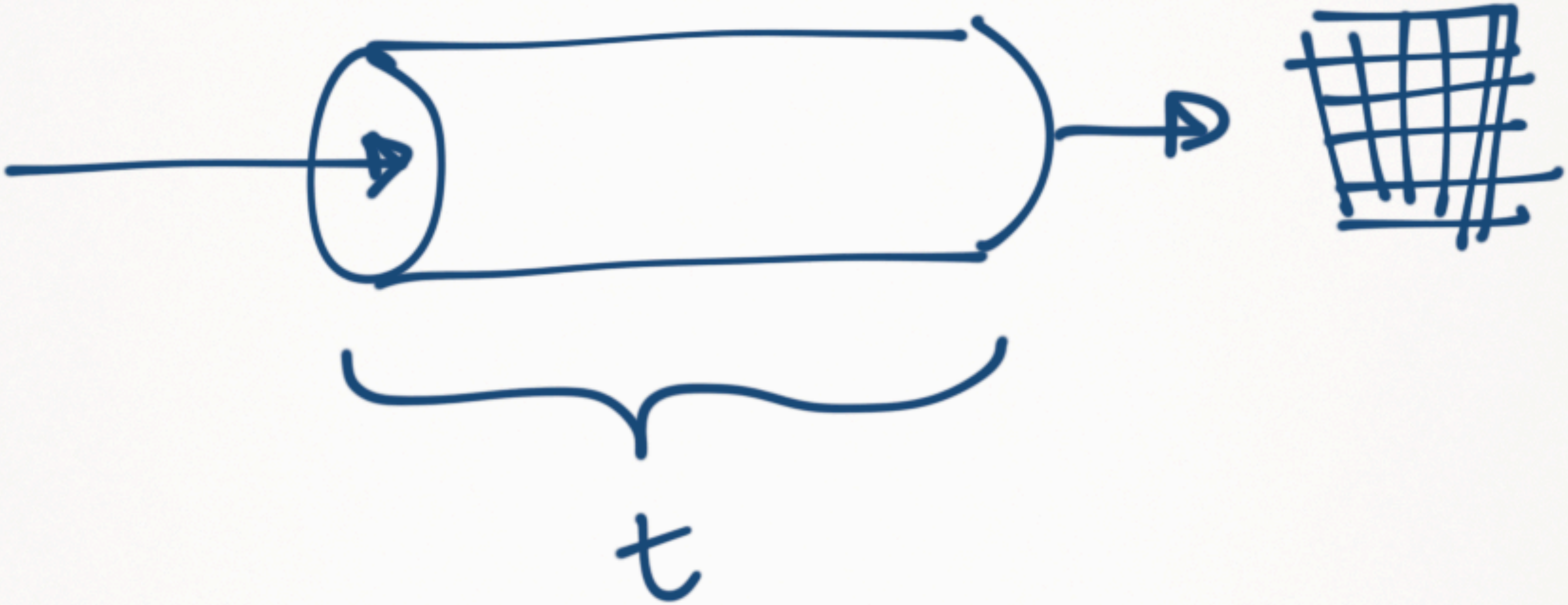
high-throughput
persistent
publish-subscribe
messaging


foo
bar
....
baz

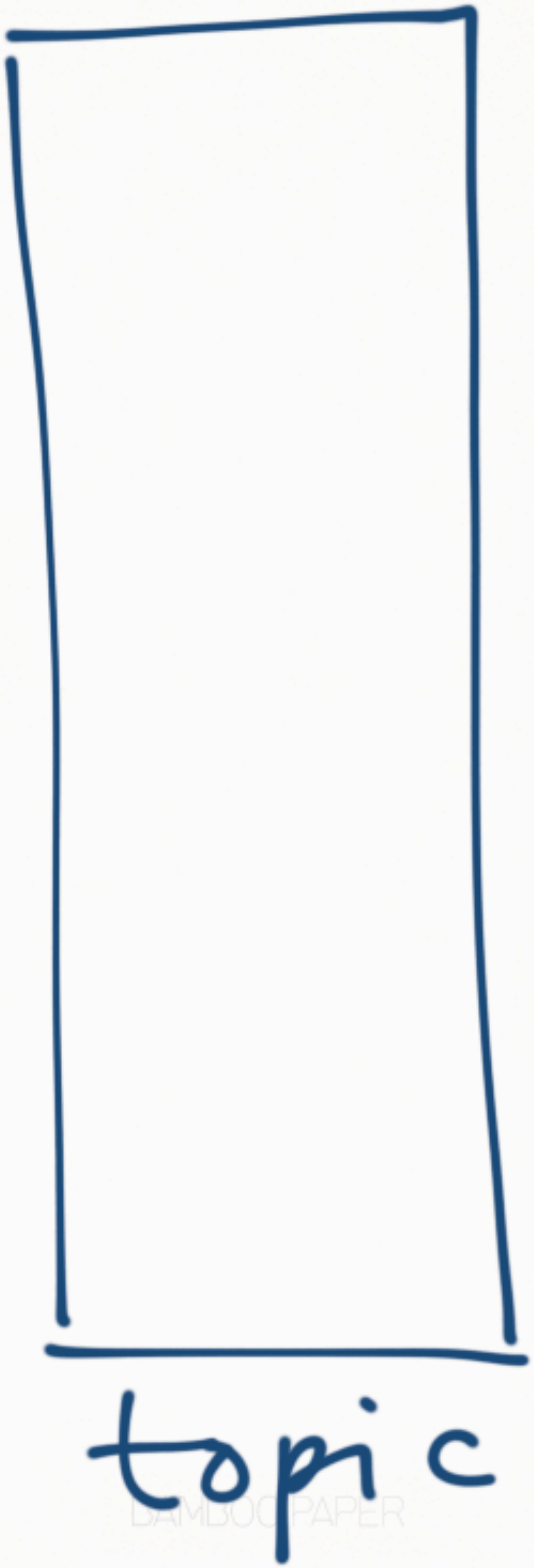

foo
bar
....
baz

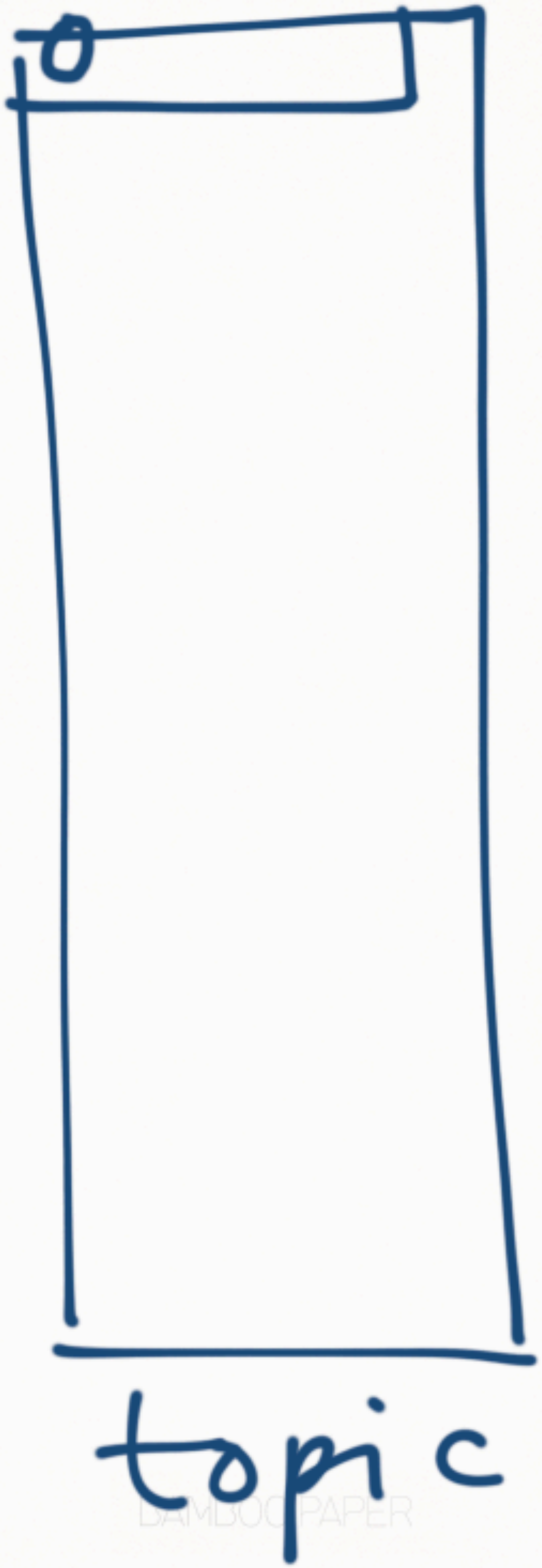
high-throughput
persistent
publish-subscribe
messaging

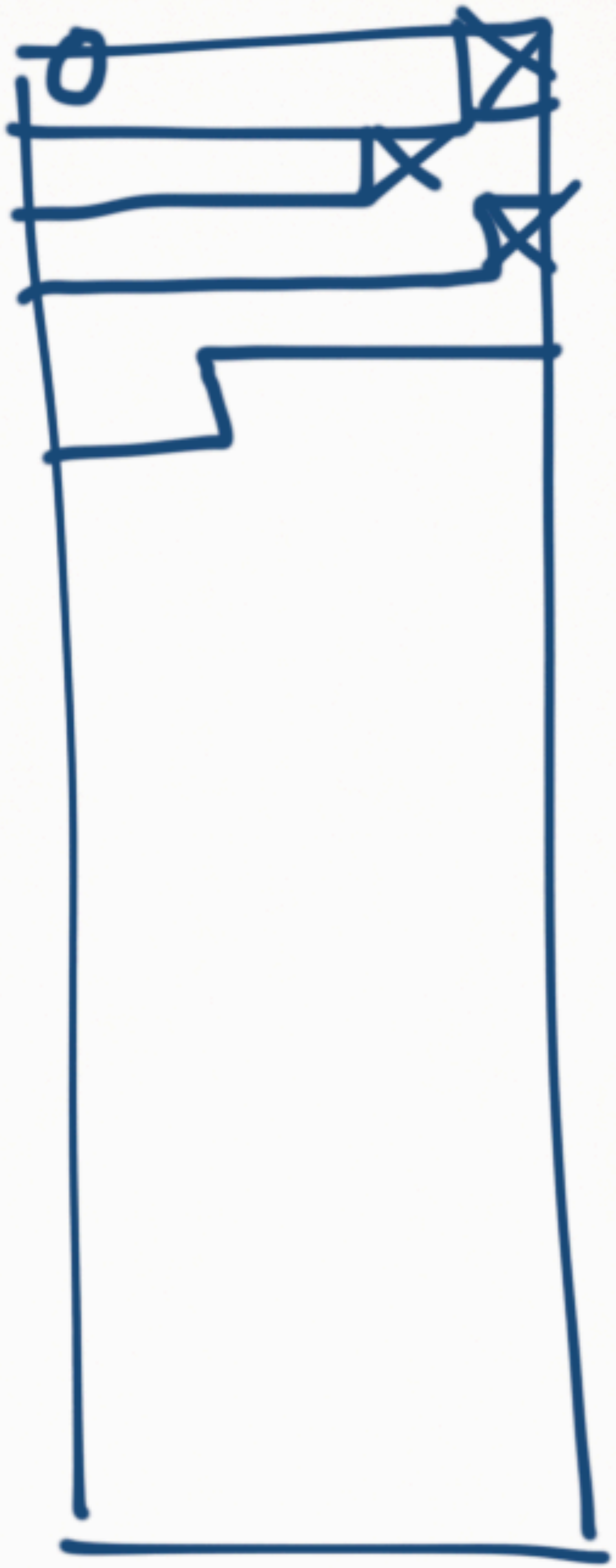



t

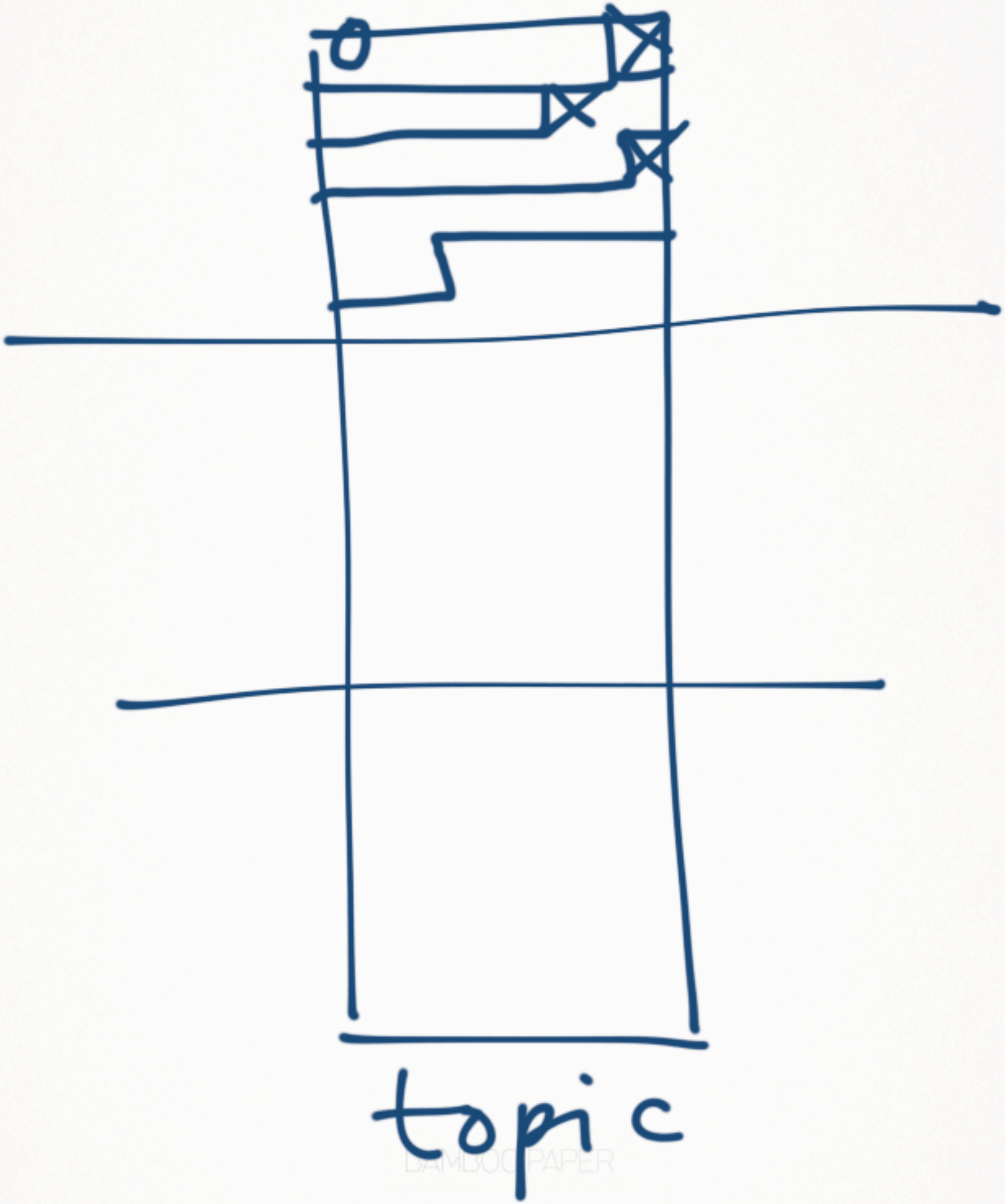
high-throughput
persistent
publish-subscribe
messaging


topic


0
topic

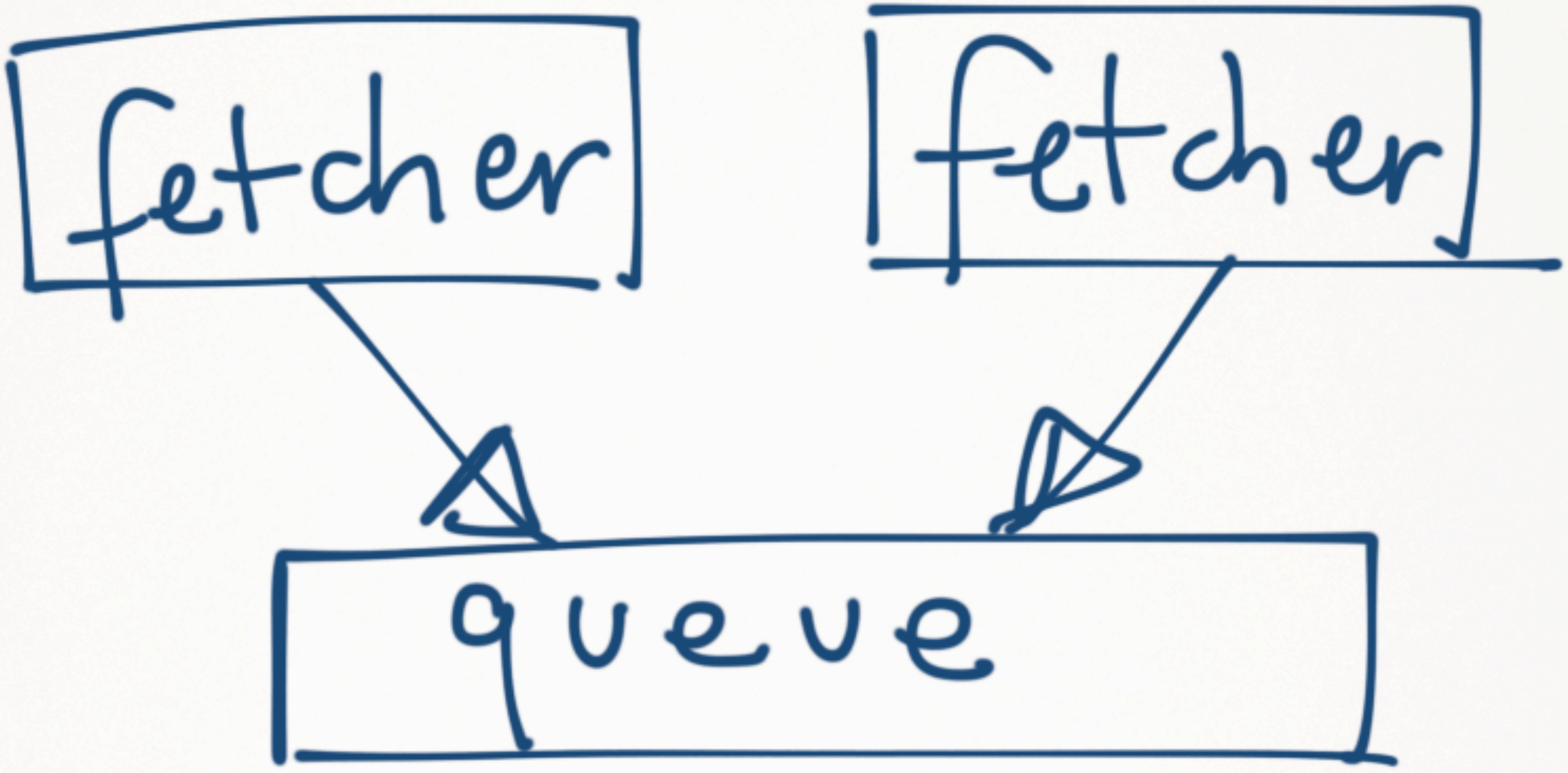


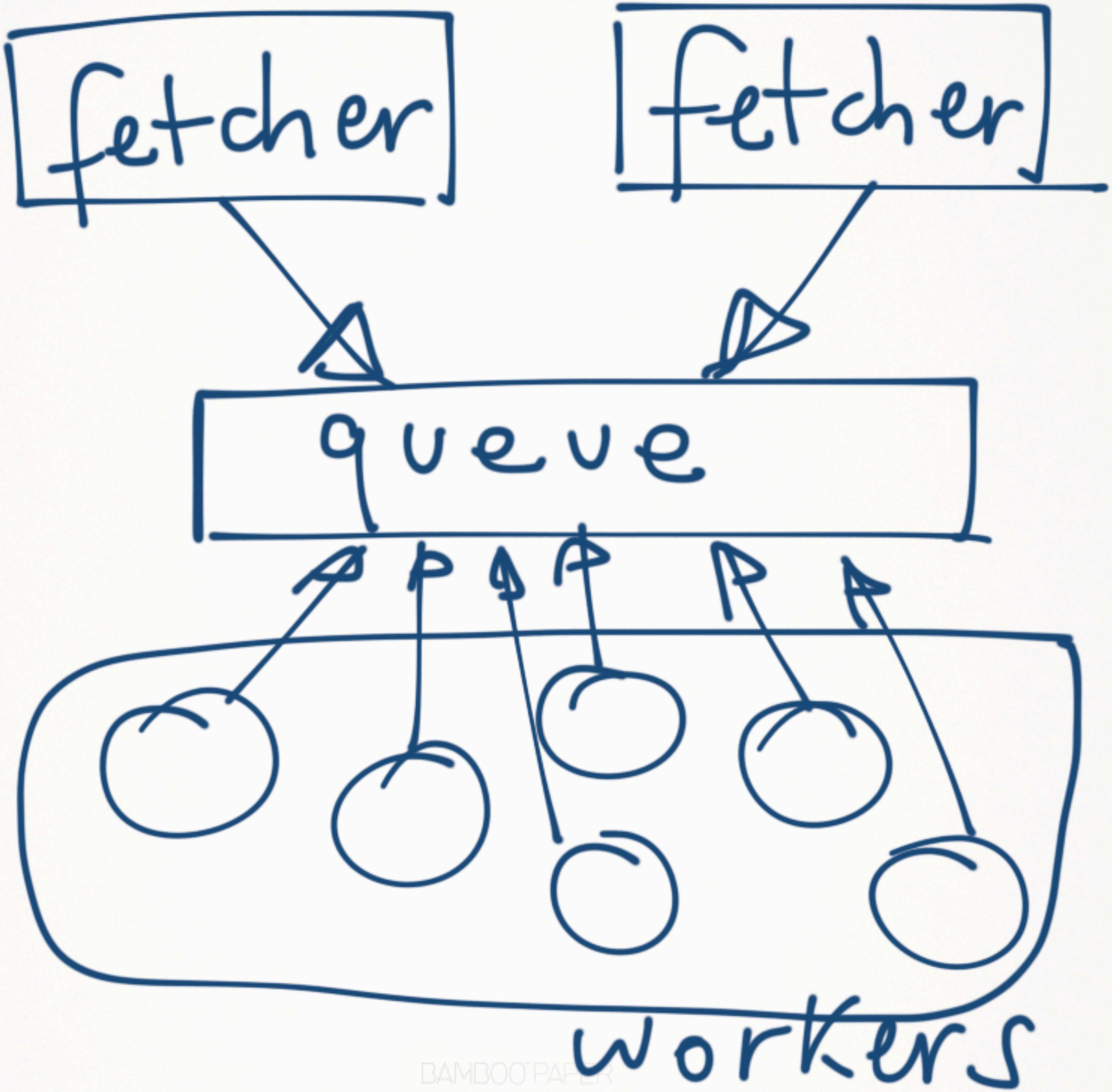
topic

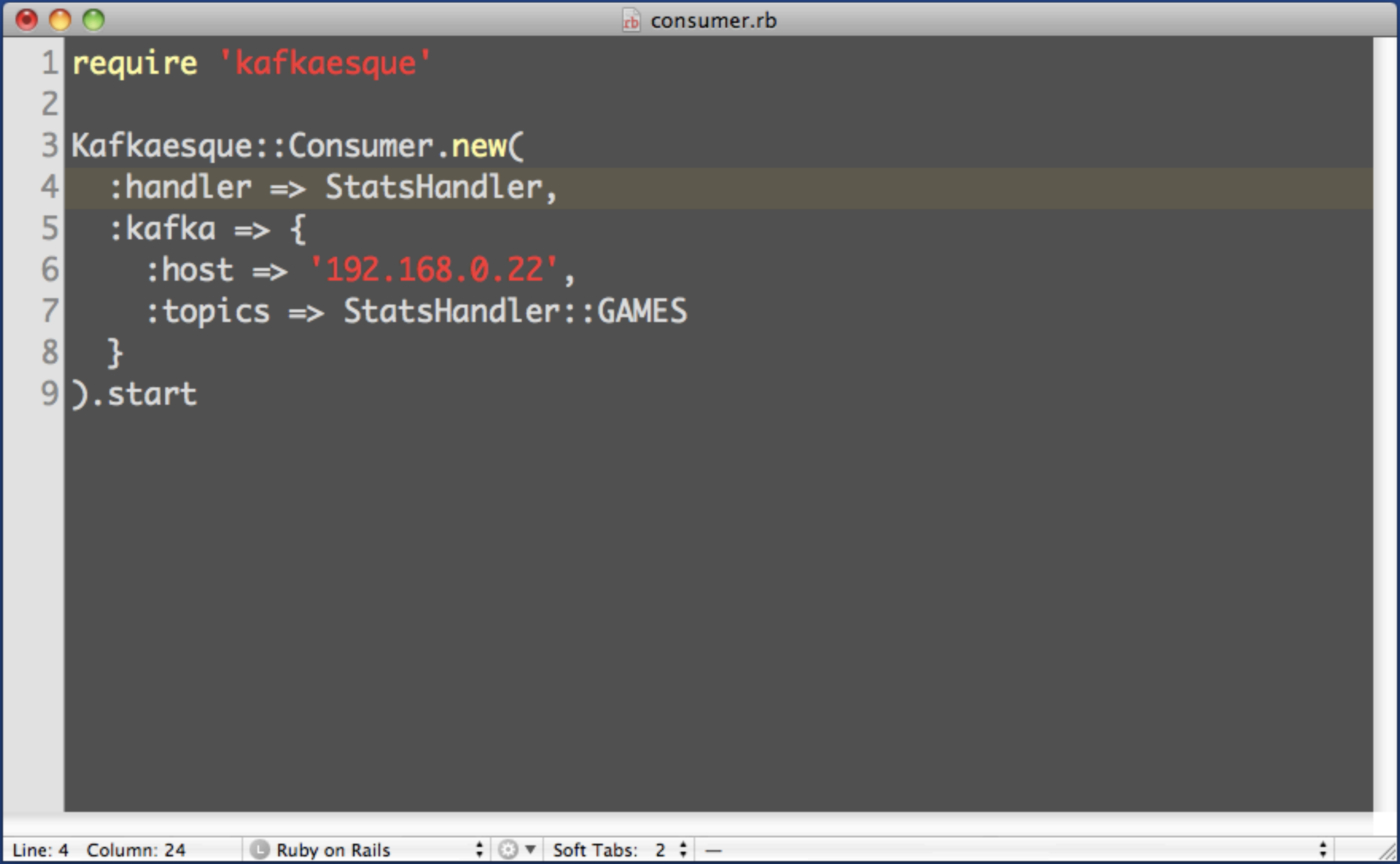

0
topic

Kafkaesque

queue

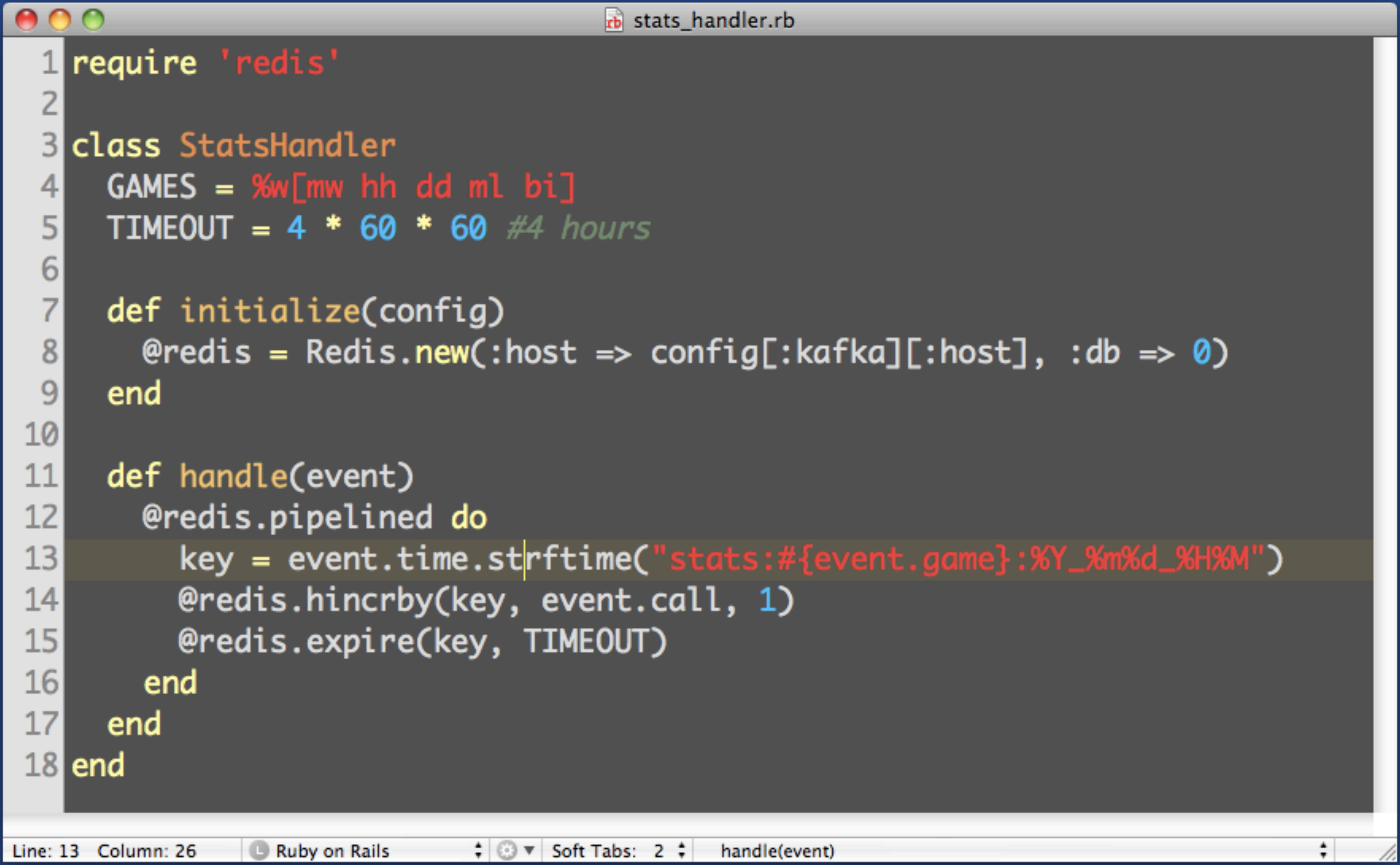

fetcher
fetcher
queue


fetcher
fetcher
queue
workers



```ruby
require 'kafkaesque'

Kafkaesque::Consumer.new(
  :handler => StatsHandler,
  :kafka => {
    :host => '192.168.0.22',
    :topics => StatsHandler::GAMES
  }
).start
```

Line: 4 Column: 24 Ruby on Rails Soft Tabs: 2

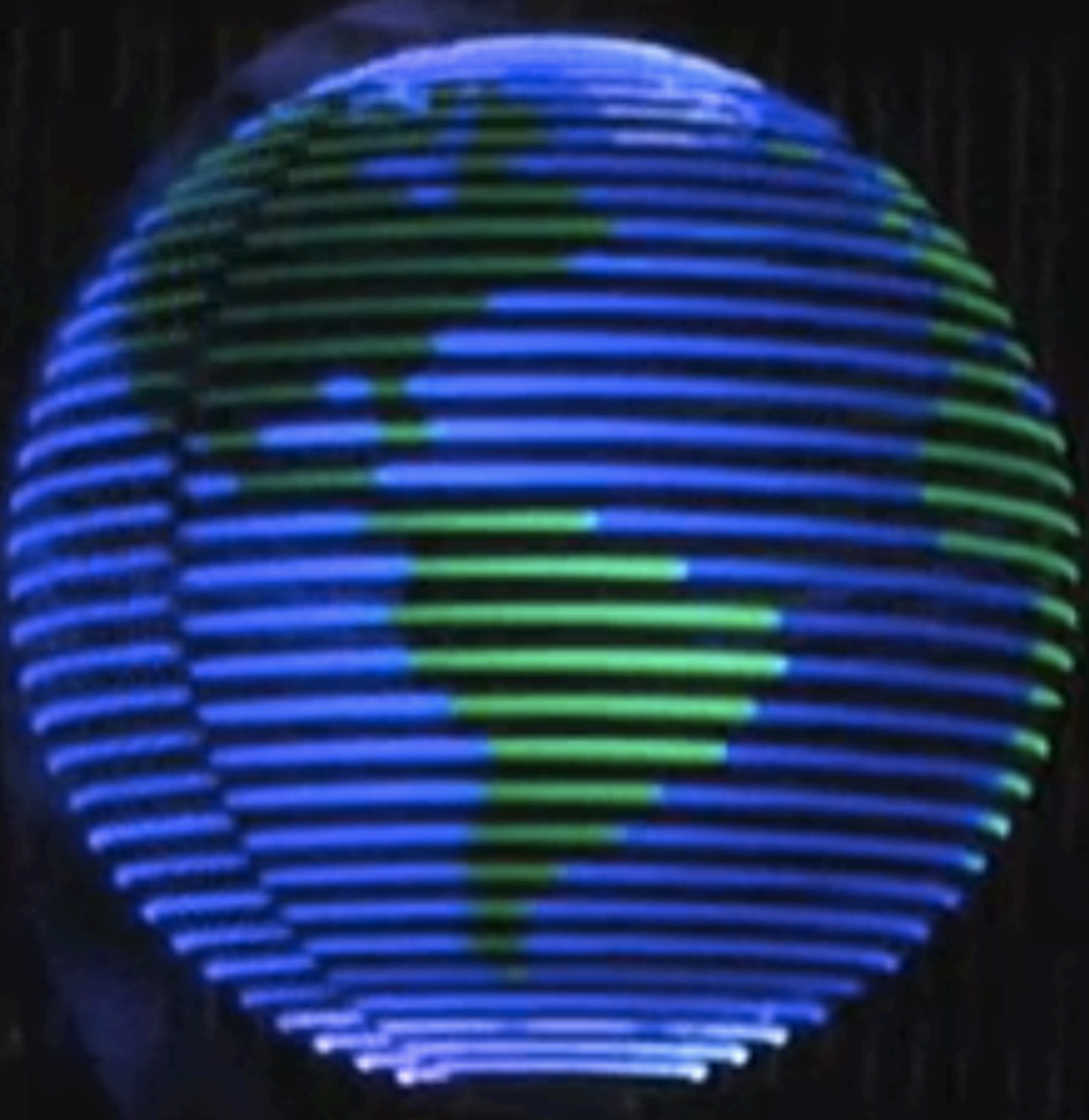


```ruby
require 'redis'

class StatsHandler
  GAMES = %w[mw hh dd ml bi]
  TIMEOUT = 4 * 60 * 60 #4 hours

  def initialize(config)
    @redis = Redis.new(:host => config[:kafka][:host], :db => 0)
  end

  def handle(event)
    @redis.pipelined do
      key = event.time.strftime("stats:#{event.game}:%Y_%m%d_%H%M")
      @redis.hincrby(key, event.call, 1)
      @redis.expire(key, TIMEOUT)
    end
  end
end
```

Line: 13 Column: 26 Ruby on Rails Soft Tabs: 2 handle(event)

Ideas

archdezart.com/2012/02/28/chronotronix-v400-nixie-tube-clock/



youtube.com/watch?v=xAYWvdeaznA

pixelinvaders.ch

Recap

- don’t fly blind

- don't fly blind
- check out kafka

- *don’t fly blind*
- *check out kafka*
- *be creative*

Links

incubator.apache.org/kafka/design

github.com/wooga/kafka-rb

github.com/wooga/kafkaesque

Questions?

flickr.com/photos/mabi/522864770/

[we are hiring]