Main sponsor

From RoR & the Ruby VM to the JVM

Raffi Krikorian / @raffi

33rd Degree Conference
Conference for Java Masters
19-21 March 2012
Kraków, Poland


@DanaDanger
Dana Contreras
Tweet!
1 min ago via Twitter for iPhone
Favorite
Retweet
Reply
I ♥ your mom

> 2.5E9 “deliveries” a day

Ruby on Rails

Sustainable productivity for web-application development

HAS...

- huge number of concurrent connections
- lots of I/O
- few persistent objects

Gosai Kiji

Phasianus versicolor also known as Japanese Pheasant is a bird of the lowlands. Closely related to the Common Pheasant, the cock is distinguished by dark green plumage on breast and mantle. The male has an iridescent violet neck, red bare facial skin and purplish green tail. The female is smaller than male, and has a dull brown plumage with dark spots.


Twitter Engineering: Building a Faster
http://engineering.twitter.com/2011/03/building-faster-ruby
Share
Report Abuse
Next Blog»
Twitter Engineering
FRIDAY, MARCH 4, 2011
Building a Faster Ruby Garbage Collector
Since late 2009, much of www.twitter.com has run on Ruby Enterprise Editi
modified version of the standard MRI 1.8.7 Ruby interpreter. At the time, we
the REE team to integrate some third-party patches that allowed us to tune
collector for long-lived workloads. We knew this was not a perfect choice, b
a new runtime (even MRI 1.9x) would introduce compatibility problems, and
indicated that alternative runtimes are not necessarily faster for our workloa
Nevertheless, the CPU cost of REE remained too high.
To address this problem, we decided to explore options for optimizing the R
We called this effort Project Kiji, after the Japanese bird.
INEFFICIENT GARBAGE COLLECTION
Our performance measurements revealed that even after our patches, the R
uses a significant fraction of the CPU for running the garbage collector on tw
This is largely because MRI's garbage collector uses a single heap:
The garbage collector's naive stop-the-world mark-and-sweep process
accesses the entire memory set several times. It first marks all object
"root-set" level as "in-use" and then reexamines all the objects to relea
memory of those not in use. Additionally, the collector suspends the s
during every sweep, thereby periodically "freezing" some of the progra
The collection process is not generational. That is, the collector does
move objects between heaps; they all stay at the same address for th
lifetime. The resulting fragmented memory extracts a penalty in bookk
cost because it can neither be consolidated nor discarded.
We needed to make the garbage collector more efficient but had limited opti

NEEDS...

- ability to handle server workloads
- an efficient language

NEEDS...

- ability to handle server workloads
- an efficient language

FIRST CHOSE  Scala

- fast
- functional and expressive
- statically typed
- concurrent
- beautiful

NEEDS...

- ability to handle server workloads
- flexibility in language

NEEDS...

- ability to handle server workloads
- flexibility in language

NEEDS...

- ability to handle server workloads
- flexibility in language
- a real concurrency model

IS AN EVENT DRIVEN &
REAL-TIME PROBLEM

NEEDS...

finagle |fəˈnāgəl|
verb [trans.] informal
obtain (something) by devious or dishonest means : *Ted attended all the football games he could finagle tickets for.*
• [intrans.] act in a devious or dishonest manner : *they wrangled and finagled over the fine points.*

- ability to handle server
- n
- om Old

scratch

```
abstract class Service[-Req, +Rep]
extends (Req => Future[Rep]) {
  ...

  /**
   * This is the method to override/implement
   * to create your own Service.
   */
  def apply(request: Req): Future[Rep]

  ...
}

val response = service(request)
```

--:**- *scratch*   All (1,0)   (Scala)----12:16AM 1.56----------

scratch

```
service(request).onSuccess { response =>
  // compute result from response
  println("got response! " + response)
}
```

--:**- *scratch*   All (1,0)   (Scala)----12:16AM 1.56----------

HAS A HOME TIMELINE

- figure out which tweets to show you
- get those tweets
- get the users that authored those tweets

scratch

```
timelineService(userId).andThen { tweetIds =>
  tweetService(tweetIds).andThen { tweets =>
    val userIds = getUserIds(tweets)
    userService(userIds).onSuccess { users =>
      // use users + tweets to produce JSON
    }
  }
}
```

--:**- *scratch*   All (1,0)   (Scala)----12:16AM 1.56----------

 'S FINAGLE = SUBSTRATE

- connection management
- protocol codecs
- transient error handling
- distributed tracing
- service discovery
- observability

scratch

```
ServerBuilder()
  .name("ServiceName")
  .reportTo(statsReceiver)
  .tracer(traceReceiver)
  .codec(Http())
  .maxConcurrentRequests(1000)
  .requestTimeout(500.milliseconds)
  .build(Service[Request, Response])
```

--:**- *scratch* All (1,0) (Scala)----12:16AM 1.56----------

scratch

```
ClientBuilder()
  .cluster(TimelineServiceCluster)
  .hostConnectionCoresize(5)
  .hostConnectionLimit(10)
  .hostConnectionIdleTime(5.seconds)
  .retries(3)
  .timeout(500.milliseconds)
```

--:**- *scratch*   All (1,0)   (Scala)----12:16AM 1.56----------


Timeline Storage
Tweet Storage
Social Graph
User Storage
Storage / Retrieval

Composition

**Models /
Biz Objects**

Timeline Storage

Tweet Storage

Social Graph

User Storage

**Storage /
Retrieval**

Composition

Twitter App

Models / Biz Objects

Timeline Storage

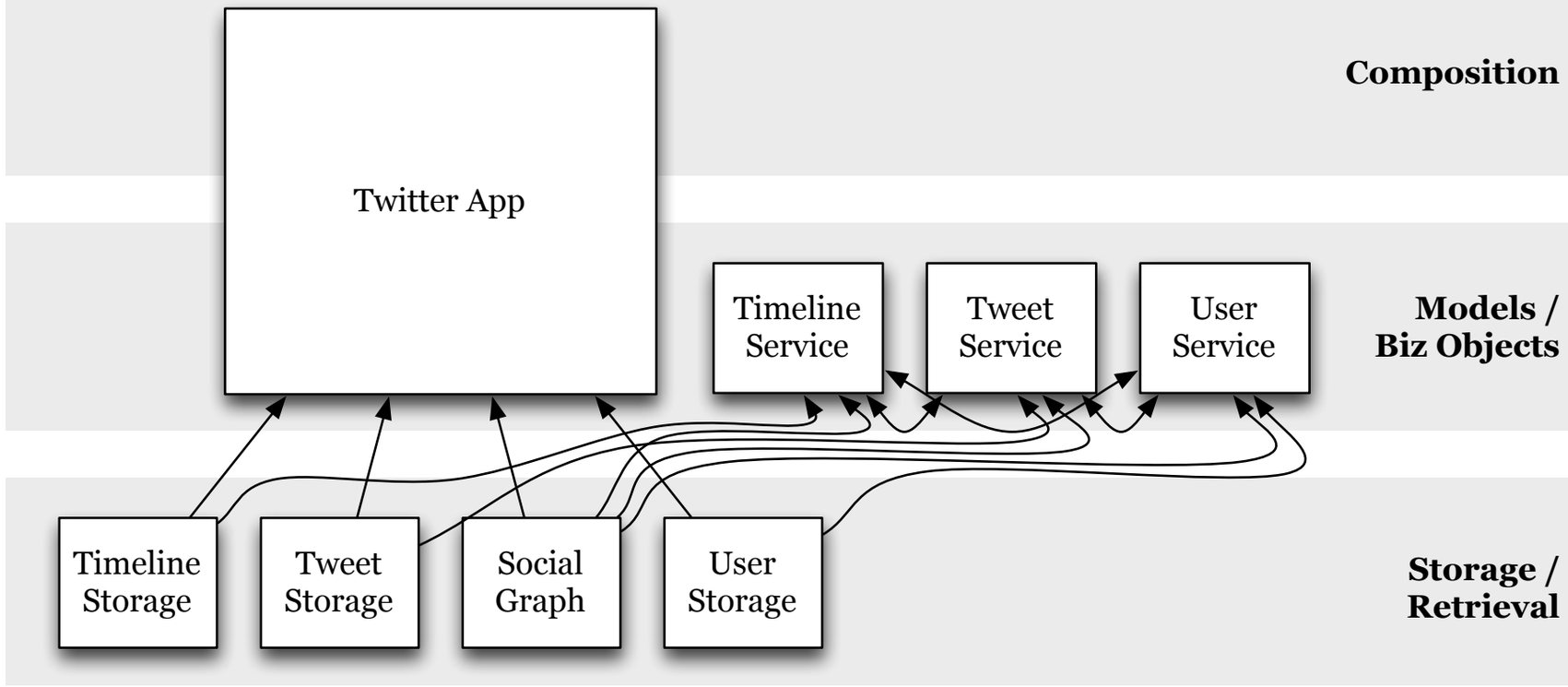
Tweet Storage

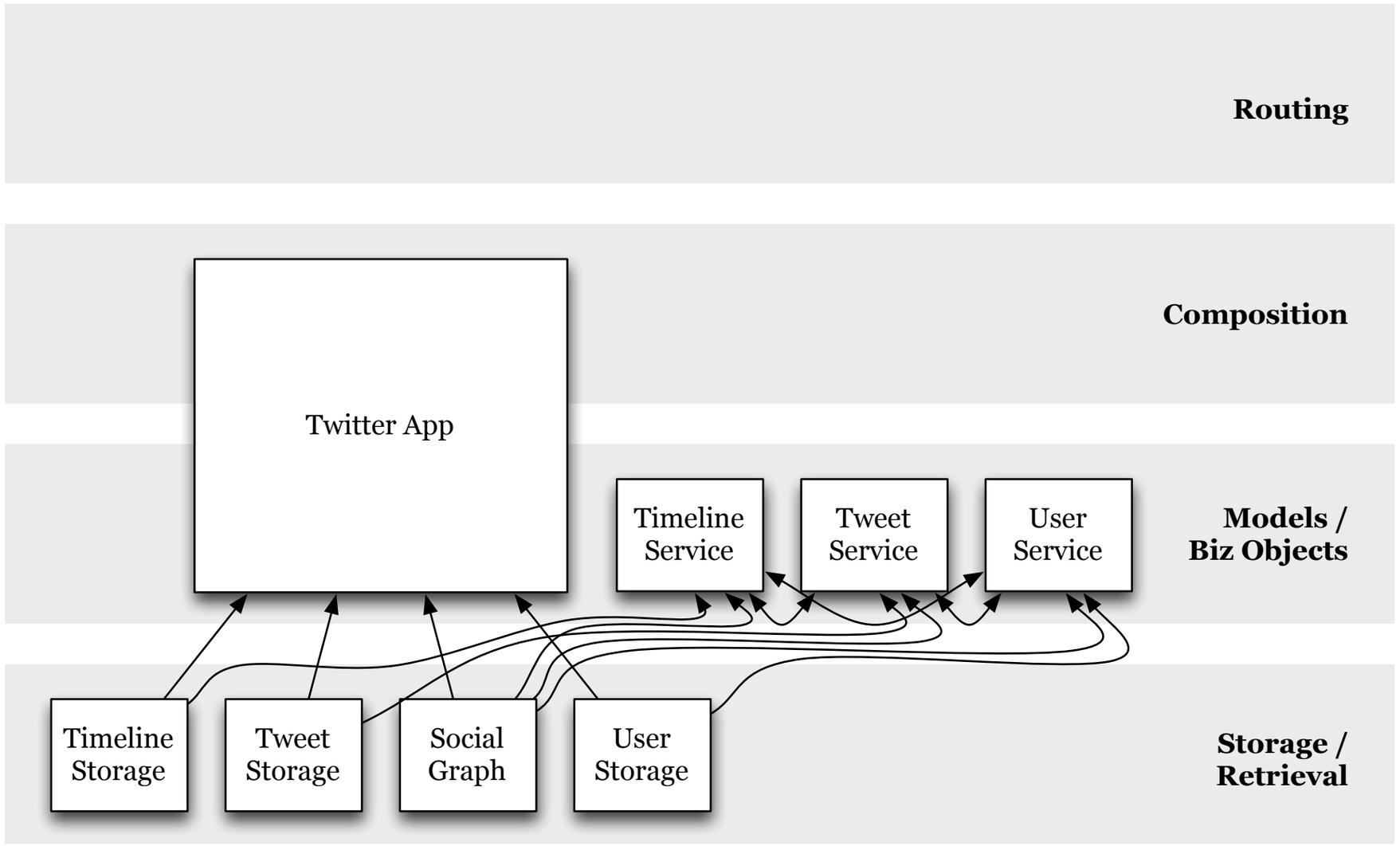
Social Graph

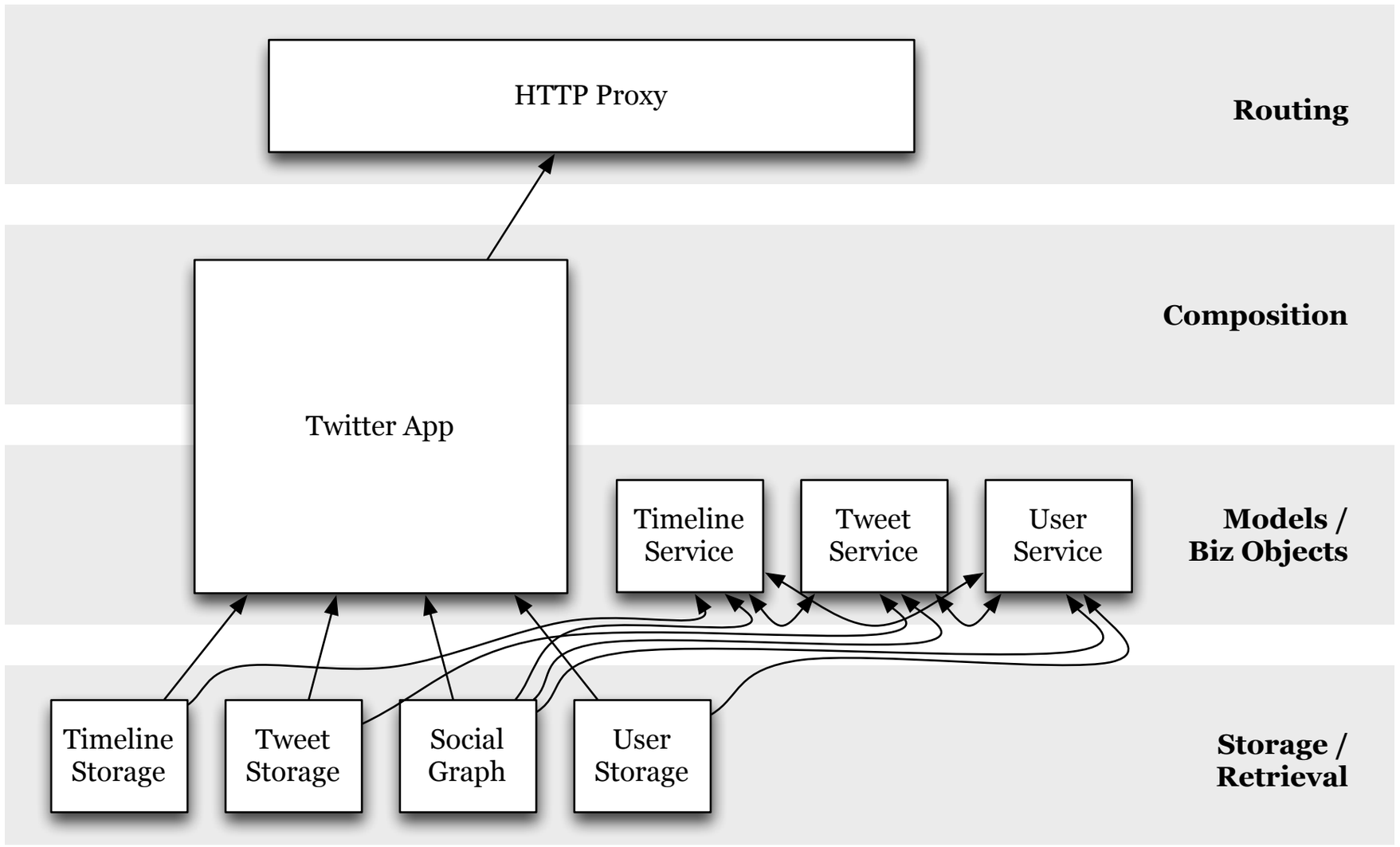
User Storage

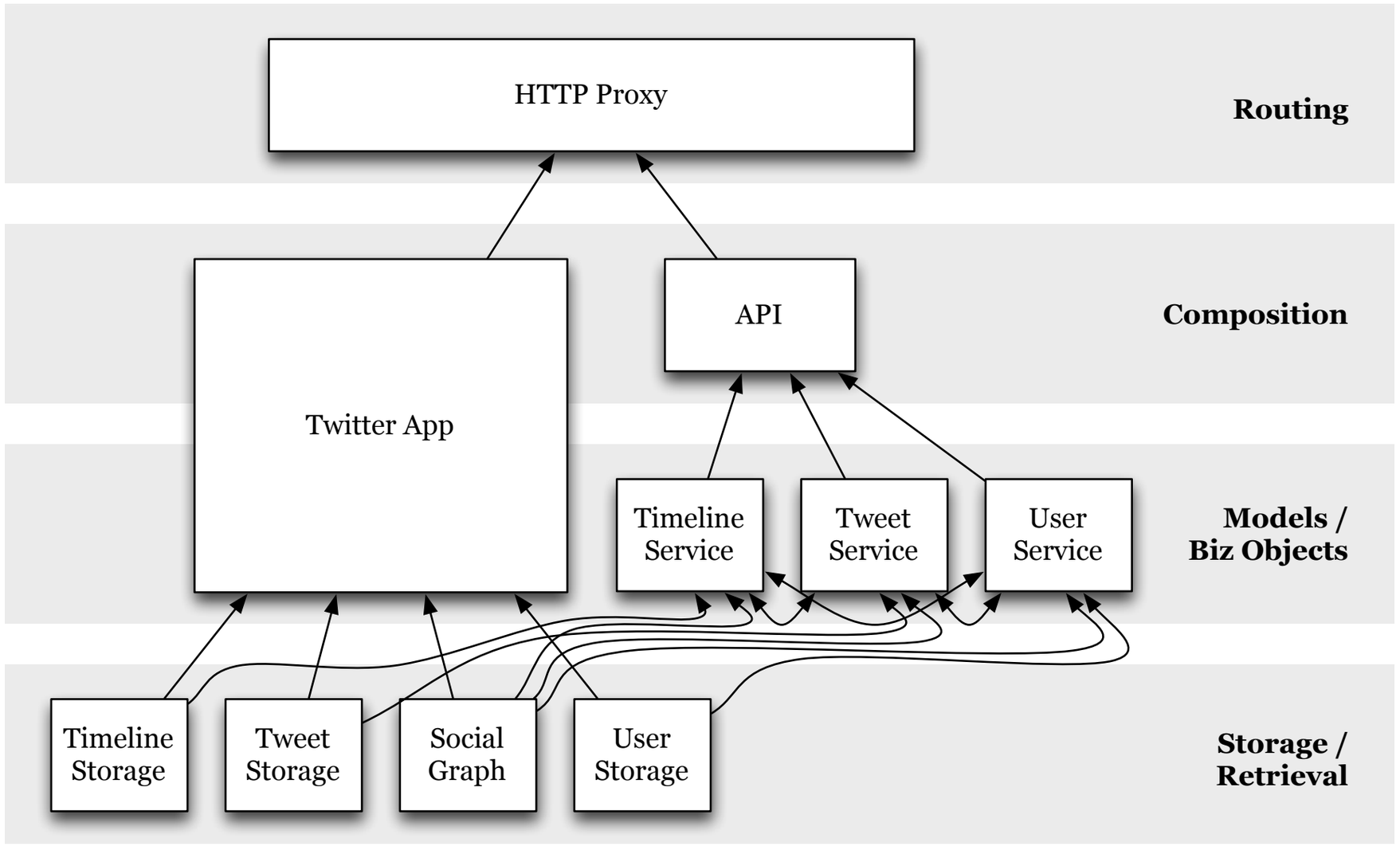
Storage / Retrieval

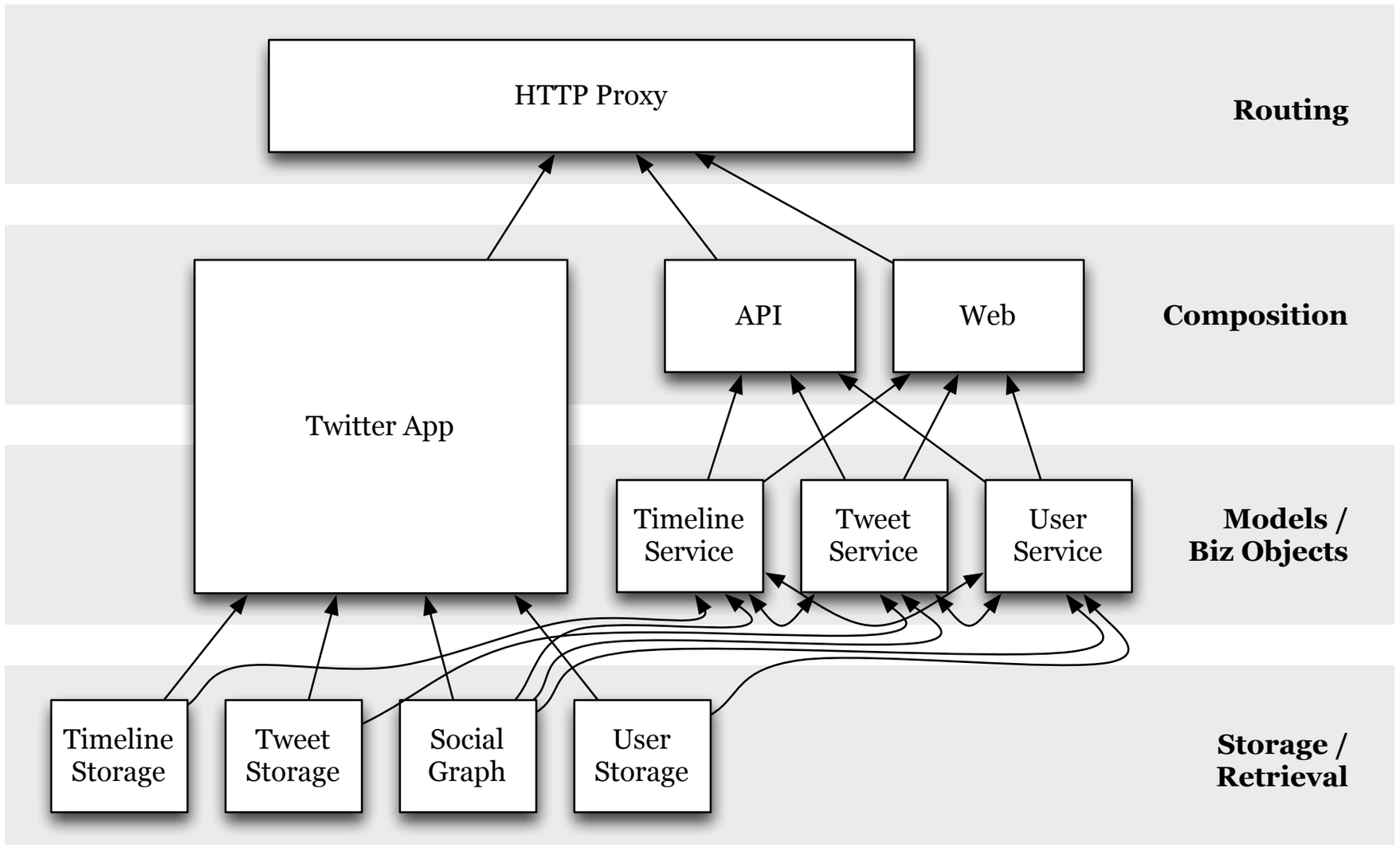
Composition

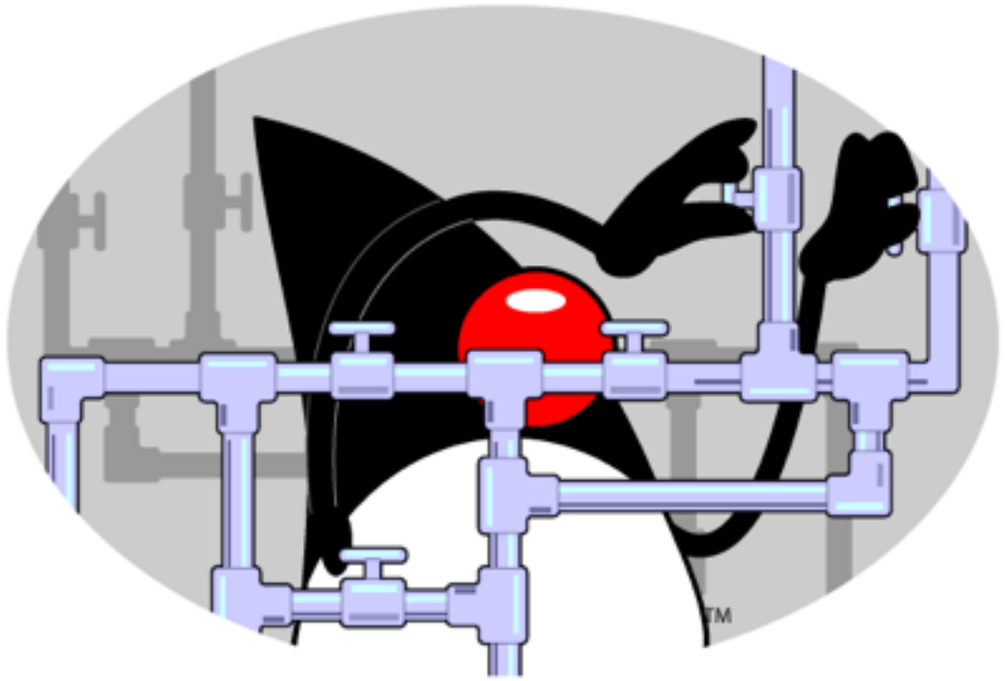



Routing
Composition
Twitter App
Timeline Service
Tweet Service
User Service
Models / Biz Objects
Timeline Storage
Tweet Storage
Social Graph
User Storage
Storage / Retrieval


HTTP Proxy
Routing
Twitter App
Composition
Timeline Service
Tweet Service
User Service
Models / Biz Objects
Timeline Storage
Tweet Storage
Social Graph
User Storage
Storage / Retrieval


HTTP Proxy
Routing
API
Composition
Twitter App
Timeline Service
Tweet Service
User Service
Models / Biz Objects
Timeline Storage
Tweet Storage
Social Graph
User Storage
Storage / Retrieval


HTTP Proxy
Routing
Twitter App
API
Web
Composition
Timeline Service
Tweet Service
User Service
Models / Biz Objects
Timeline Storage
Tweet Storage
Social Graph
User Storage
Storage / Retrieval



SWITCHING TO  DOESN'T IMPLY THAT IS A MISTAKE

Twitter
@twitter

Following

In the final three minutes of the Super Bowl tonight, there were an average of 10,000 Tweets per second.

50+ RETWEETS	50+ FAVORITES

7:43 PM - 5 Feb 12 via Twitter for iPhone · Embed this Tweet

Reply Retweet Favorite

Reply to @twitter

Twitter
@twitter

Following

Madonna's performance during the Super Bowl's halftime show saw an average of 8,000 Tweets per second for five minutes.

50+ RETWEETS	50+ FAVORITES

7:43 PM - 5 Feb 12 via Twitter for iPhone · Embed this Tweet

Reply Retweet Favorite

Reply to @twitter

Twitter
@twitter

Following

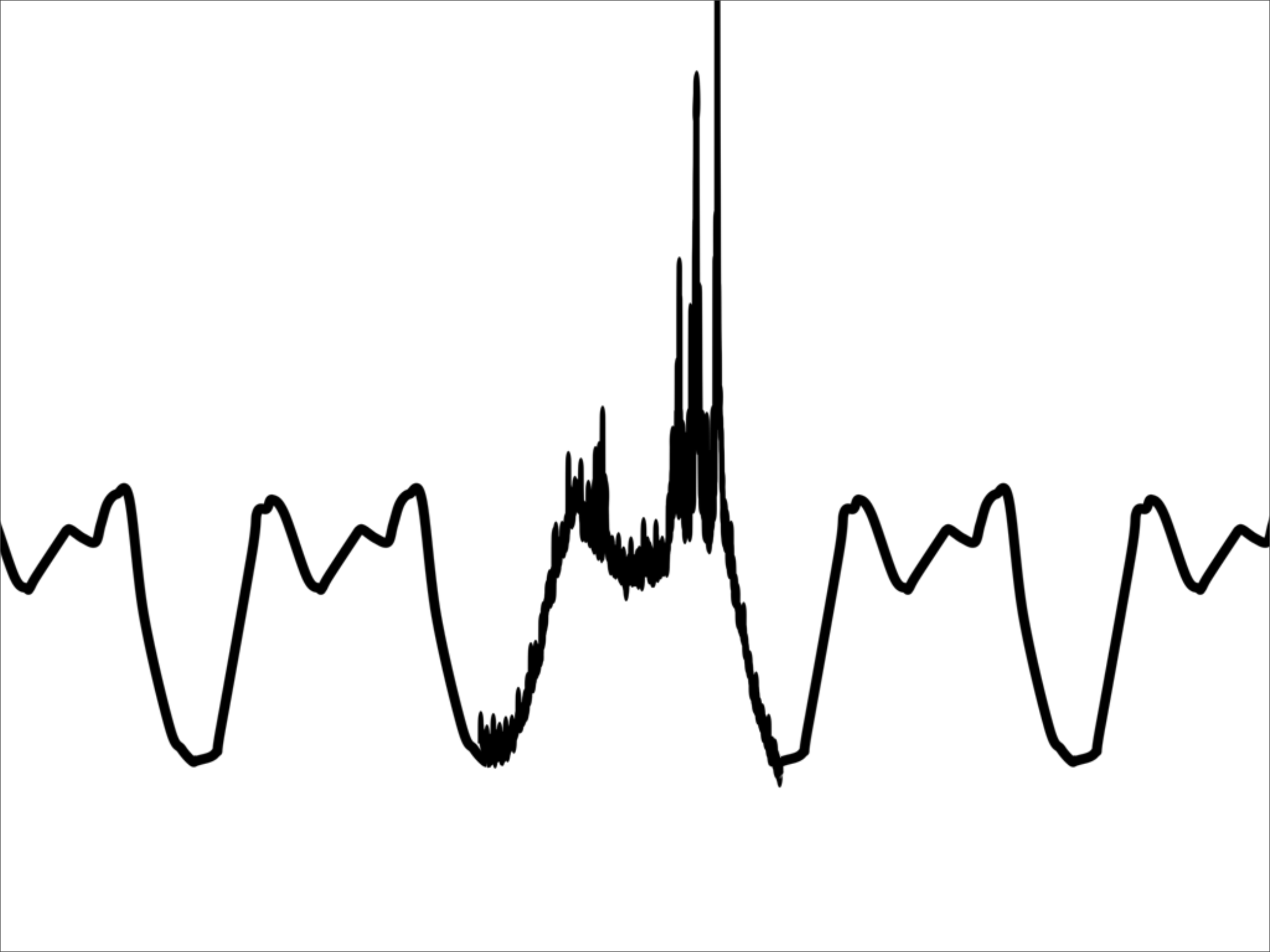
The highest Tweets per second #SuperBowl peak came at the end of the game: 12,233. 2nd highest was during Madonna's performance: 10,245.

50+ RETWEETS	50+ FAVORITES

8:31 PM - 5 Feb 12 by RachaelRad via web · Embed this Tweet

Reply Retweet Favorite

Reply to @twitter



’S HIRING!

@JOINTHEFLOCK

FOLLOW ME

@RAFFI