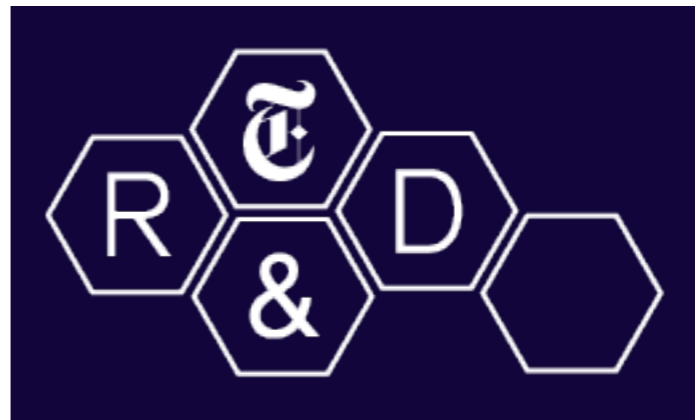


THE NEW YORK TIMES  
RESEARCH & DEVELOPMENT

**Jake Porway**  
**Data Scientist**  
**@jakeporway**



+

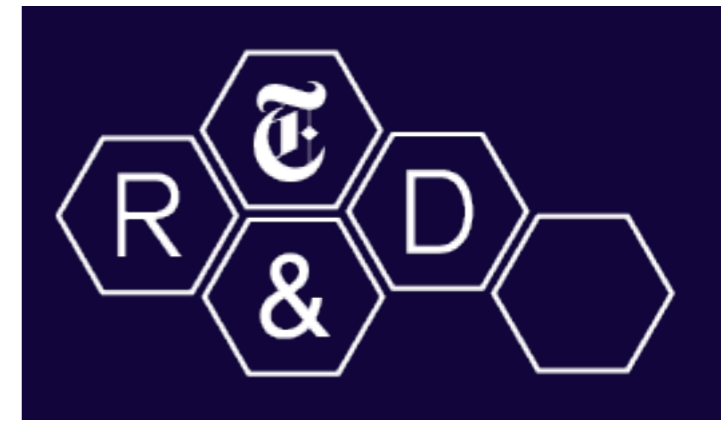


## Overview

- What is NYT R&D?
- Projects Using mongoDB
  - Project Cascade
    - Social sharing of news on Twitter.
  - [openpaths.cc](http://openpaths.cc)
    - Anonymous and private database of iPhone / iPad location data.



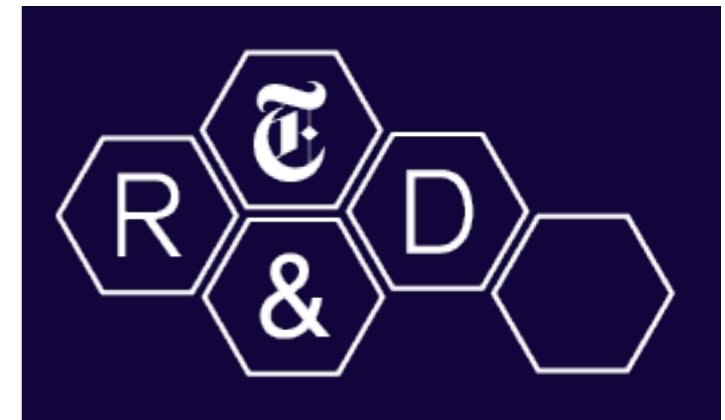
# What is NYT R&D?



- Mandate is to “look around corners.”
- We analyze, test, and prototype technologies that will be commonplace 2-5 years down the road.
- Large focus on “data” (big data, sharing of data, contextual or personalized experiences, privacy and anonymity, etc.)



# What is NYT R&D?



- Mandate is to “look around corners.”
- We analyze, test, and prototype technologies that will be commonplace 2-5 years down the road.
- Large focus on “data” (big data, sharing of data, contextual or personalized experiences, privacy and anonymity, etc.)
- How to flexibly deal with lots of data? `mongoDB!`



# Project Cascade: Social Sharing of NYT Content



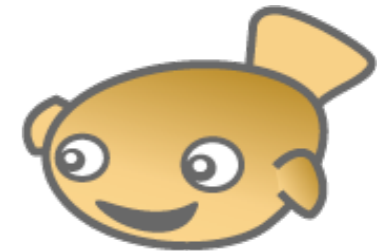
# Project Cascade: Social Sharing of NYT Content

- Discussions have always taken place around the news we publish and the stories we tell



# Project Cascade: Social Sharing of NYT Content

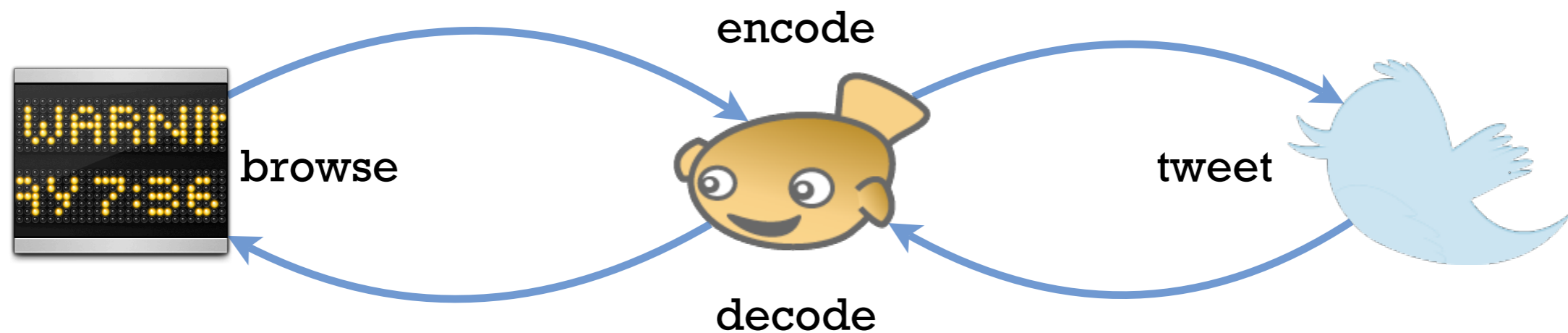
- Discussions have always taken place around the news we publish and the stories we tell



- Increasingly, those discussions are taking place within the realms of social media

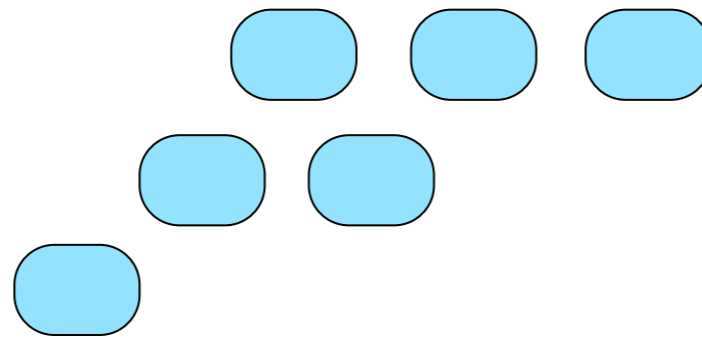
# Overview of R&D Analysis

- With the help of Mark Hansen, Professor of Statistics at UCLA, and Jer Thorp, Data Artist in Residence, we have developed a series of methods for extracting, summarizing, and linking the events of the sharing process
- We combine data from Twitter / Backtweets and from bit.ly to plot whole conversation from nytimes.com to Twitter and back again.



# High-Level Overview

Backtweets API gives  
tweets with  
“nytimes.com” links



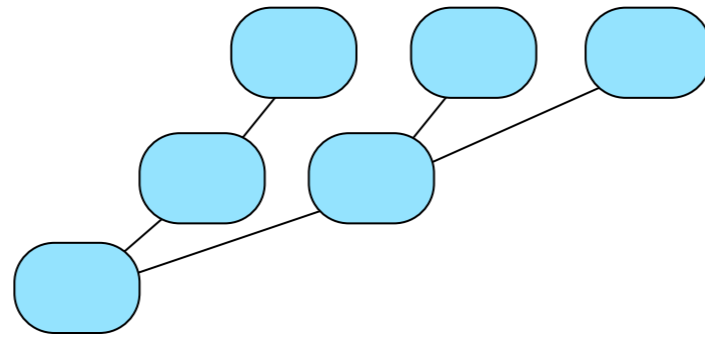
**Twitter**

**NYT**



# High-Level Overview

Backtweets API gives tweets with “nytimes.com” links



We form tweet / retweet connections

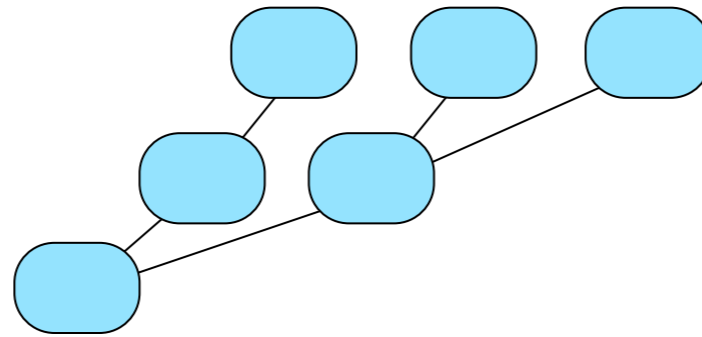
**Twitter**

**NYT**



# High-Level Overview

Backtweets API gives tweets with “nytimes.com” links



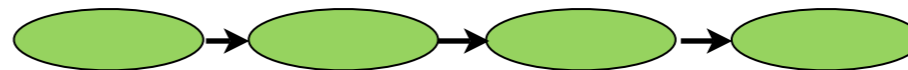
We form tweet / retweet connections

**Twitter**

**NYT**



Pages visited are combined into user sessions



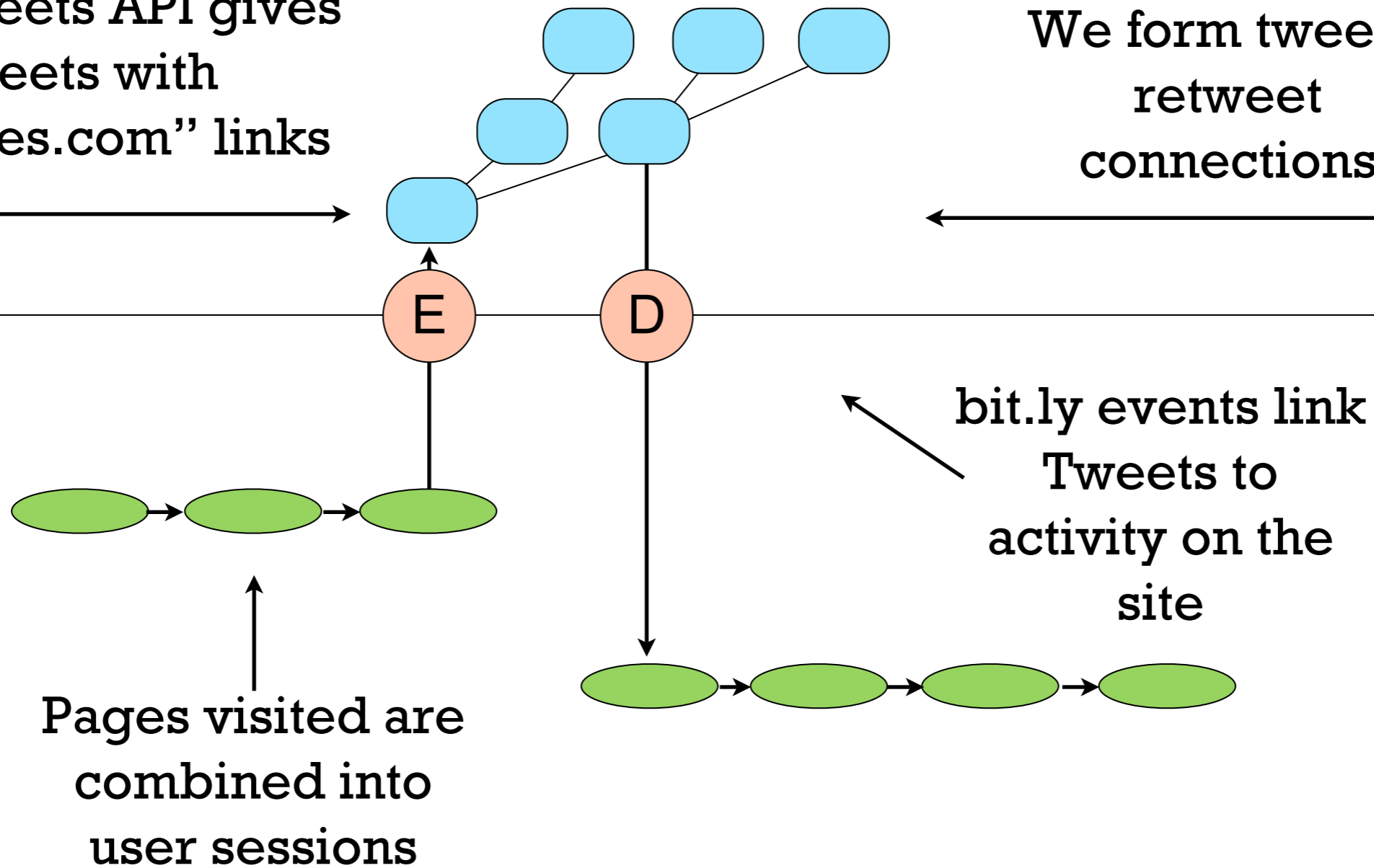
# High-Level Overview

Backtweets API gives tweets with "nytimes.com" links

We form tweet / retweet connections

**Twitter**

**NYT**



# Why Did We Choose MongoDB?

- Needed to pull data from a lot of different APIs (Backtweets, Twitter, bit.ly) and store easily.
- Needed a scalable option.
- Map/Reduce framework conducive to running batch processing.
- Research platform - need schemaless NoSQL option.
- Everything's already JSON / Python dictionary



# ~~Big~~ Data

- NYT publishes ~600 pieces of content a day.
- On average we see about 25K encodes/day, 250K decodes/day, and 25K tweets/day that include links to [nytimes.com](http://nytimes.com)
- With extra metadata resulting from building cascades, ends up at ~100 GB per month
- Could store about a year's worth of cascades on 1.2TB. We backup to S3.
- NOTE: In the video I think I misspoke and said we get 10GB of logs/hour. It's closer to 2-5GB



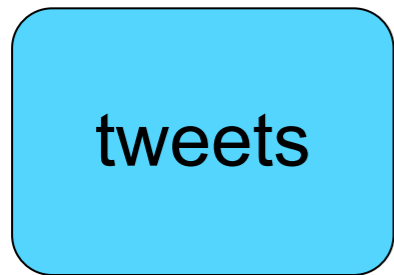
# mongoDB and Cascade Architecture



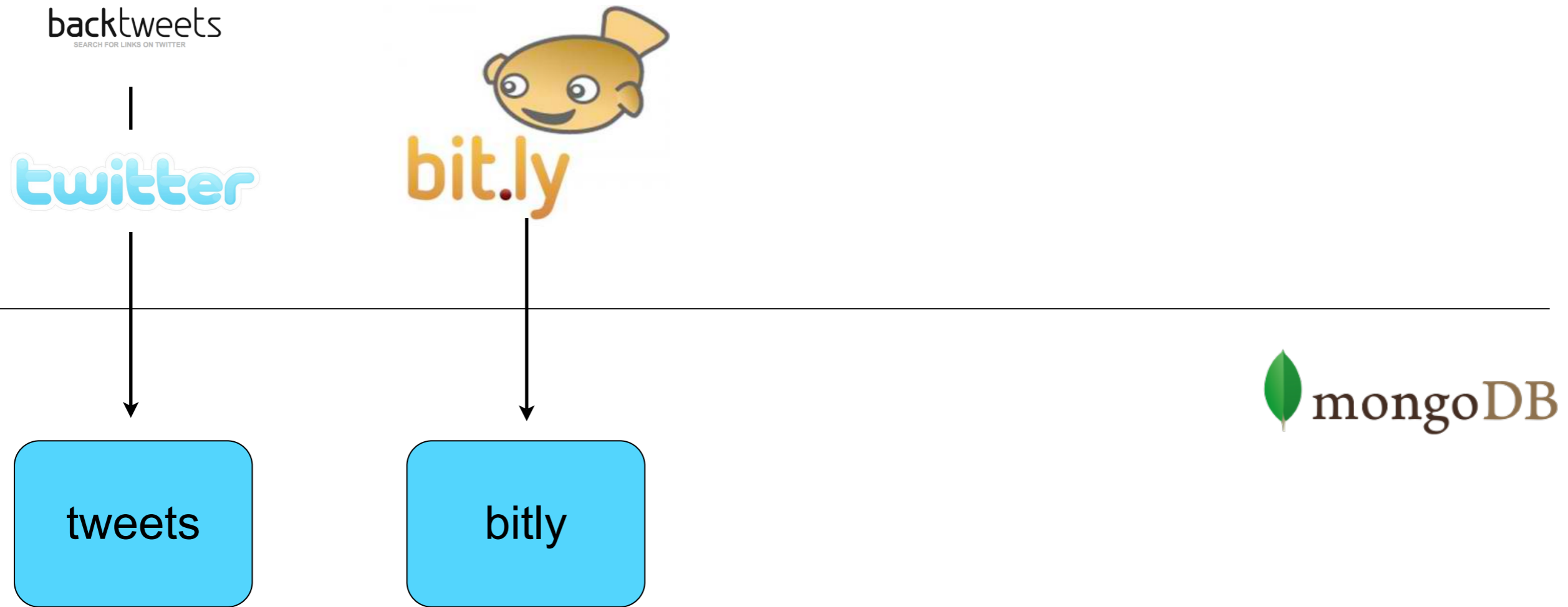
# mongoDB and Cascade Architecture

backtweets  
SEARCH FOR LINKS ON TWITTER

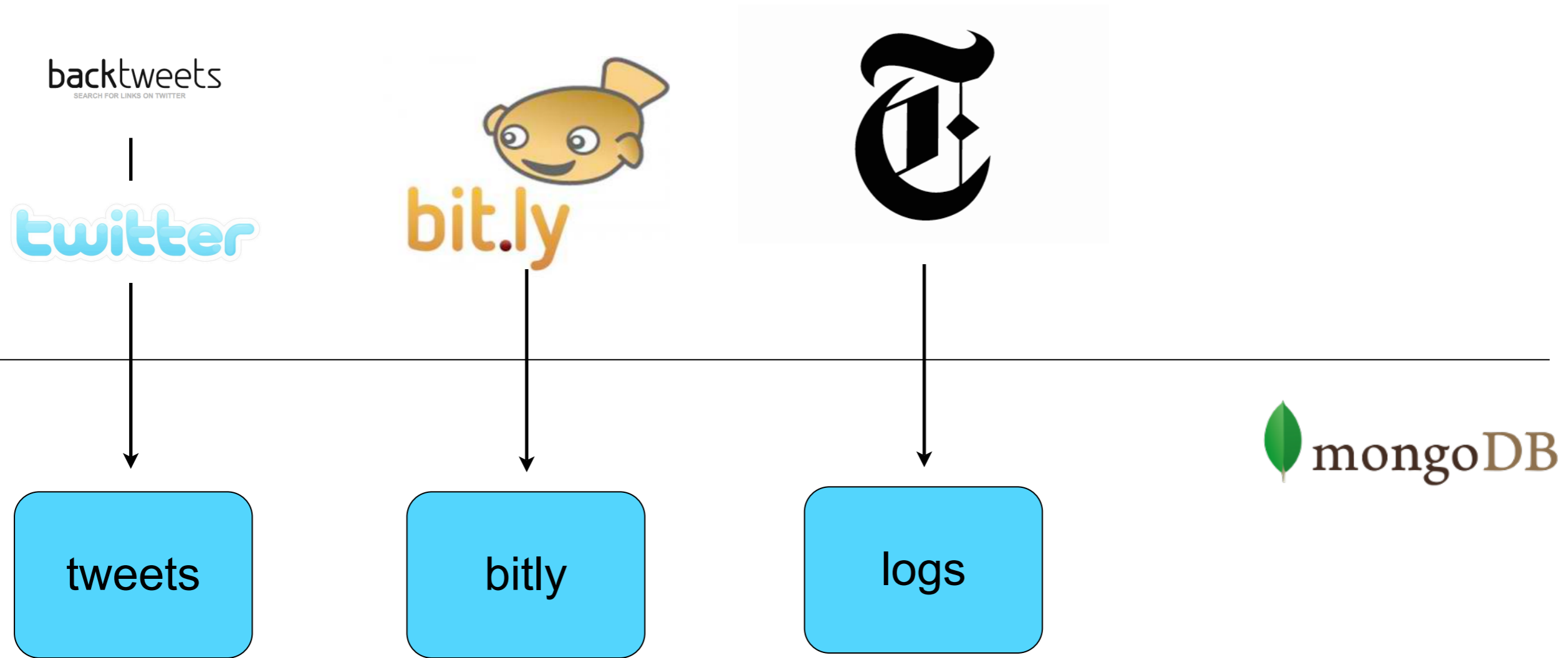
twitter



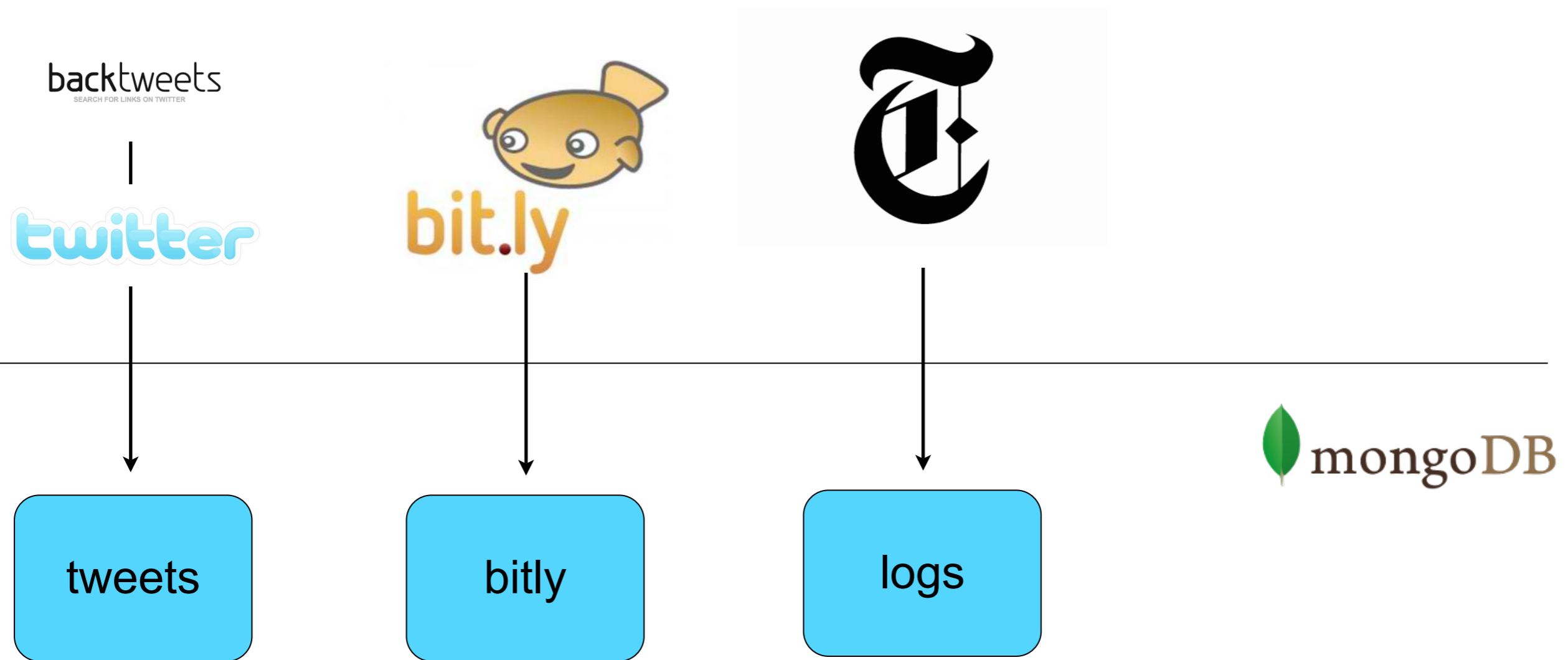
# mongoDB and Cascade Architecture



# mongoDB and Cascade Architecture



# mongoDB and Cascade Architecture



Why we love mongoDB: It eats this data up! We just dump data from our Python scripts or APIs into mongo and we're good to go.



# Schema For Data

## bit.ly

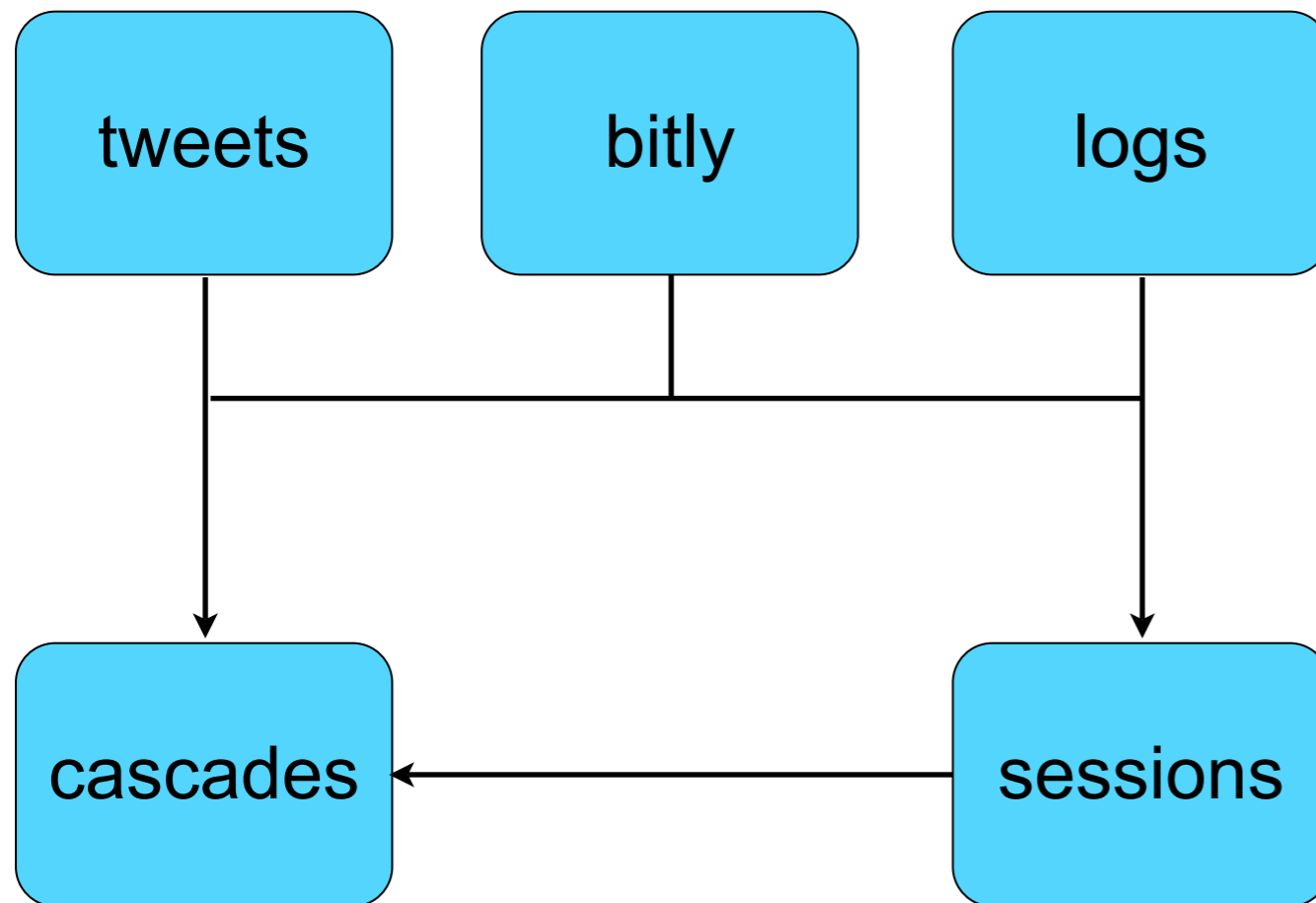
```
{
  "_id" : ObjectId
("4de69d6233f3a54a45000082"),
  "a" : "Mozilla/5.0",
  "c" : "US",
  "tz" : "America/New_York",
  "gr" : "MA",
  "g" : "mfjXRi",
  "what" : "decode",
  "i" : "7xce234jd0c01eb4d0978a13739ab1cf",
  "h" : "jgsA7j",
  "cy" : "New York",
  "l" : "nytimesapi",
  "al" : "en-US,en;q=0.8",
  "hh" : "nyti.ms",
  "r" : "awesome.com",
  "u" : "awesome.com/rad_page.html",
  "t" : 1306959202,
  "hc" : 1306892836,
  "story" : "awesome.com/rad_page.html",
  "ll" : [42.809898,-71.123201]
}
```

## tweets

```
{ "_id" : ObjectId("4de7e81e33f3a57424000001"),
  "contributors" : null,
  "truncated" : false,
  "text" : "Loose yourself...\nhttp://www.nytimes.com/2011/05/31/opinion/31brooks.html?src=ISMR_AP_LO_MST_FB",
  "previous_statuses" : [
    "Loose yourself...\nhttp://www.nytimes.com/2011/05/31/opinion/31brooks.html?src=ISMR_AP_LO_MST_FB",
    "NutriSmart: Edible RFID Tags That Track Food Down The Supply Chain @psfk http://t.co/0yFSc84 via @AddThis",
    "Coconut water & wine my fav things:\nhttp://lnkd.in/y5tG67",
    ...],
  "in_reply_to_status_id" : null,
  "id" : NumberLong("76373135196299264"),
  "story" : "http://www.nytimes.com/2011/05/31/opinion/31brooks.html",
  "retweeted" : false,
  "created_at_seconds" : 1307058149,
  "id_str" : "76373135196299264",
  "user" : {
    "follow_request_sent" : false,
    "profile_use_background_image" : true,
    "id" : 19117015,
    "profile_sidebar_fill_color" : "efefef",
    "screen_name" : "kiehner",
    ...
  },
  "geo" : null,
  "in_reply_to_user_id_str" : null,
  "created_at" : "Thu Jun 02 19:42:29 +0000 2011",
  "liberal_retweet" : false,
  "twitter_id" : 19117015,
  "local_hash" : [ ],
}
```



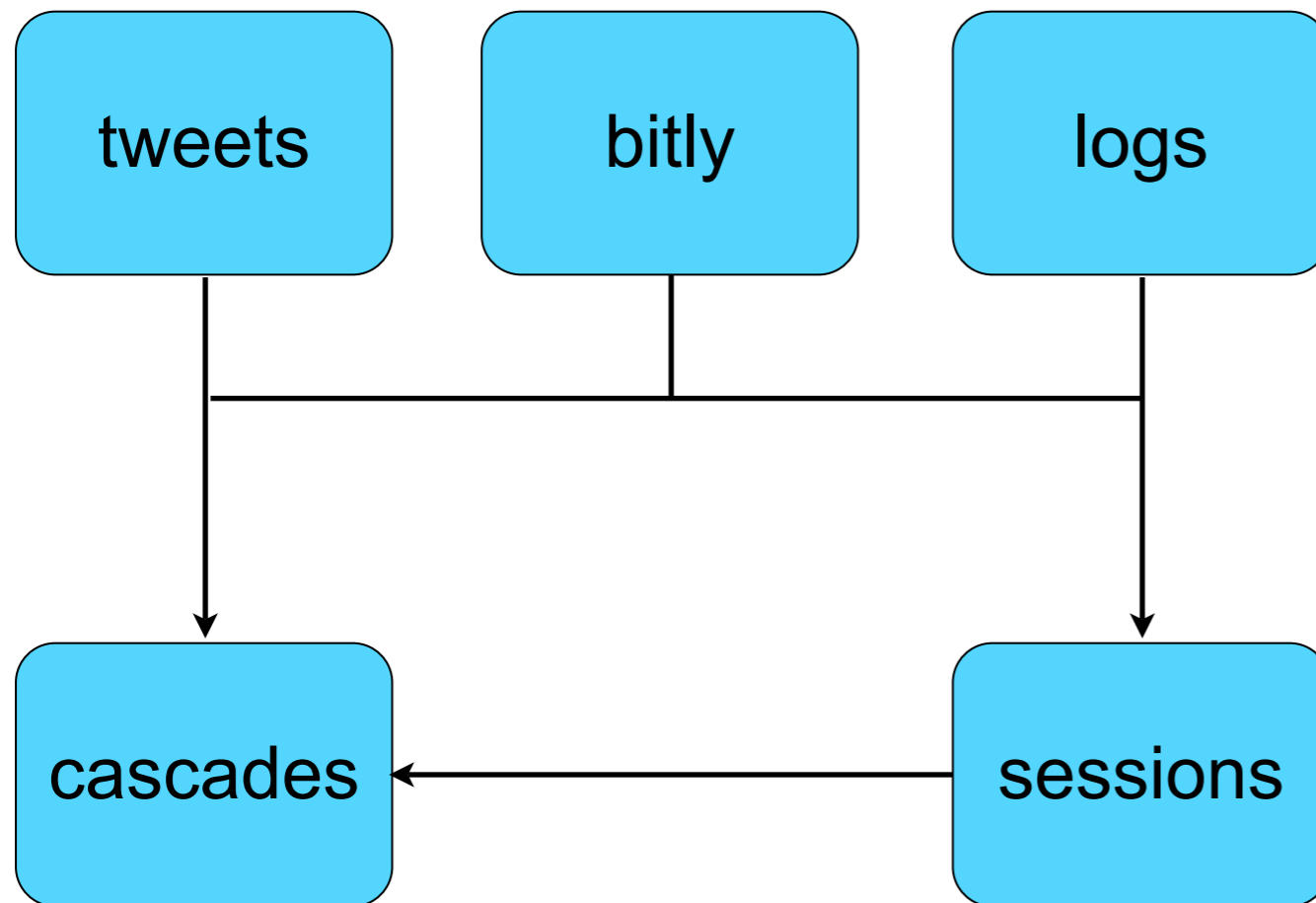
# mongoDB and Cascade Architecture



Raw tweets / bitly / logs are converted into site sessions and cascades.



# mongoDB and Cascade Architecture



Raw tweets / bitly / logs are converted into site sessions and cascades.

Why we love mongoDB: Cascades are trees. Ever try to manage hierarchical data in a relational DB? With mongoDB we can easily create Cascade objects as embedded documents.



# Sessions and Cascades Schema

## Sessions

```
{
  "_id" : ObjectId("4ded525433f3a5554d000035"),
  "end" : 1307396993,
  "session_id" : 54,
  "start" : 1307396983,
  "nevents" : 5,
  "key" : "b5563798c9c2a4ae05c723f125fc4dcf_3280aa5b3a394de2e2509325",
  "ip" : "67.86.68.138",
  "up_cookie" : "",
  "nytgr_cookie" : "",
  "user_agent" : "Mozilla/5.0 (Windows NT 6.1; rv:2.0.1) Gecko/20100101 Firefox/4.0.1"
  "events" : [
    {
      "date" : 1307396983,
      "id" : 0,
      "url" : "http://www.nytimes.com/interactive/2011/03/12/world/asia/what-happens-in-a-nuclear-meltdown.html",
      "section" : "interactive",
      "server_ip" : "f37f639f",
      "time" : 1307396983,
      "ref" : "",
      "referring_domain" : ""
    },
    {
      "date" : 1307396990,
      "id" : 1,
      "url" : "http://www.nytimes.com/interactive/2011/03/12/world/asia/what-happens-in-a-nuclear-meltdown.html",
      "section" : "interactive",
      "server_ip" : "cd7f63a1",
      "time" : 1307396990,
      "ref" : "",
      "referring_domain" : ""
    },
    ... ]
}
```



# Sessions and Cascades Schema

## Cascades

```
{
  "_id" : ObjectId("4debc5a63ab23f32d500001f"),
  "number_of_nodes" : 2,
  "root_node" : {
    "parent_tweet_id" : "",
    "user_id" : 205388264,
    "events" : [ ],
    "liberal_retweet" : false,
    "children" : {
      "76397011435388928" : {
        "parent_tweet_id" : "76397696990195712",
        "user_id" : 38215308,
        "liberal_retweet" : true,
        "number_of_followers" : 1539,
        "events" : [ ],
        "tweet_id" : "76397011435388928",
        ...,
      },
    },
  },
  "max_depth" : 0,
  "thumbnail_last_update" : 0,
  "top_influencers" : [ 20538823, 38215308 ],
  "updates_history" : [{ "number_of_nodes" : 2,
    "number_of_changes" : 2,
    "last_update" : 1307297189.702268
  } ],
  "local_hashes" : [ "kYMrzV" ],
  "story_url" : "http://www.nytimes.com/gwire/2011/06/02/02greenwire-chevron-looks-to-arbitrators-to-save-it-from-1-53361.html",
  "id" : 9
}
```

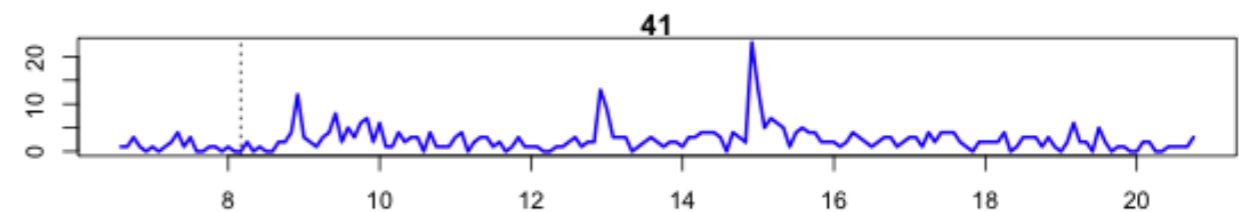
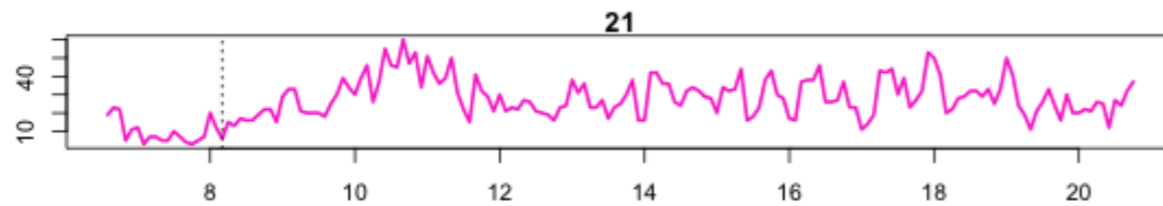


# Map/Reduce Jobs

- Need to run some batch operations:
  - Update thumbnails for cascades



- Rerank “interesting” cascades

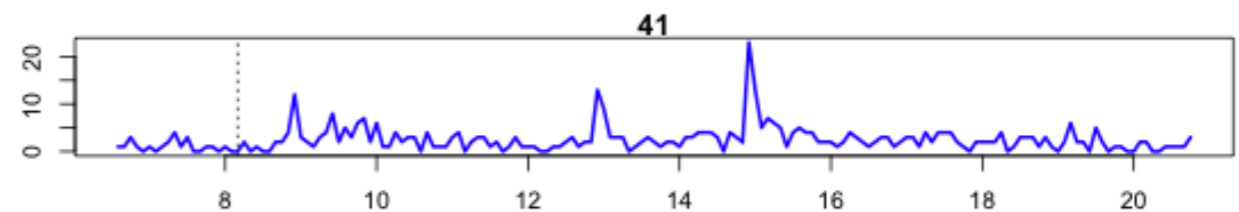
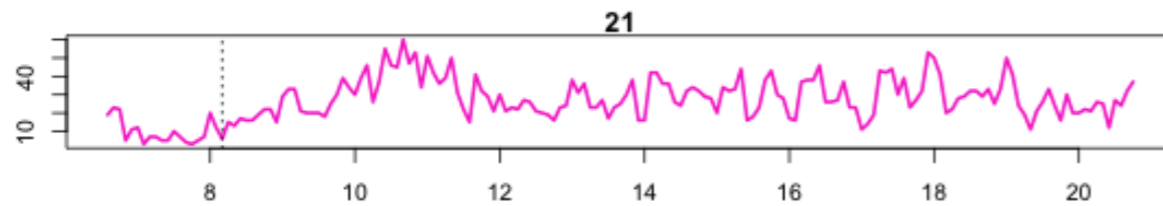


# Map/Reduce Jobs

- Need to run some batch operations:
  - Update thumbnails for cascades



- Rerank “interesting” cascades



Why we love mongoDB: Map/Reduce is perfect for running customized batch operations.

(What we don't love as much: This can be *\*slow\**! Sometimes faster to pull into Python, compute, and put back).



# Research! Science!

- Cascades is an exploratory tool.
- As such, we're *\*constantly\** adding new data to our DB, such as:
  - Different measures of “interestingness”
  - Measures of activity
  - Extra info in tweets / events for visualization



# Research! Science!

- Cascades is an exploratory tool.
- As such, we're *\*constantly\** adding new data to our DB, such as:
  - Different measures of “interestingness”
  - Measures of activity
  - Extra info in tweets / events for visualization

Why we love mongoDB: Schemaless! We can tack whatever we want on to the objects wherever we want. This is critical in a research environment.



# Graph Structures

- In addition to Cascades, we have lots of graph structures we're interested, notably activity/influence graphs.

User_ID	UserInfo	Activity_ID	ActivityInfo
1	...	1	...
2	...	2	...

User_ID	Activity_ID	User_ID	Influences
1	1	1	2
1	2	2	7



```
db.people.findOne()
{
  "_id" : ObjectId("4d87a44f92ed613e17000001"),
  "activity" : [
    {
      "index" : 2656260,
      "tweet_id" : NumberLong( 2.05768e+10 ),
      "forward_count" : 1,
      "tweet_proportion" : 0.947986577181208,
      "cascade" : 1056,
      "forward_weight" : 0.3333333333333333,
      "tweet_timeliness" : 0.999319683814954
    },
    ],
  "cached_image" : "21678699.jpg",
  "influenced_by" : [
    {
      "n" : 3,
      "followers" : 2611601,
      "influencer" : 807095
    }
  ],
}
```



# Project Cascade Summary

- **mongoDB really fit our needs for research / development:**
  - Takes in lots of data easily
  - Schemaless
  - Map/reduce is super useful
  - Complicated structures represented easily



**openpaths.cc**



openpaths.cc



# openpaths.cc

Wait! Shouldn't we have rights to our own data?

Moreover, could be a boon to urban planners, land use experts, transportation authorities, epidemiologists, etc.



Your device.  
Your data.

# openpaths.cc

Wait! Shouldn't we have rights to our own data?

Moreover, could be a boon to urban planners, land use experts, transportation authorities, epidemiologists, etc.

openpaths.cc provides a place for users to anonymously and privately upload their iPhone / iPad location data and visualize / download / remix / donate it as they like.



Your device.  
Your data.



# So Why mongoDB?

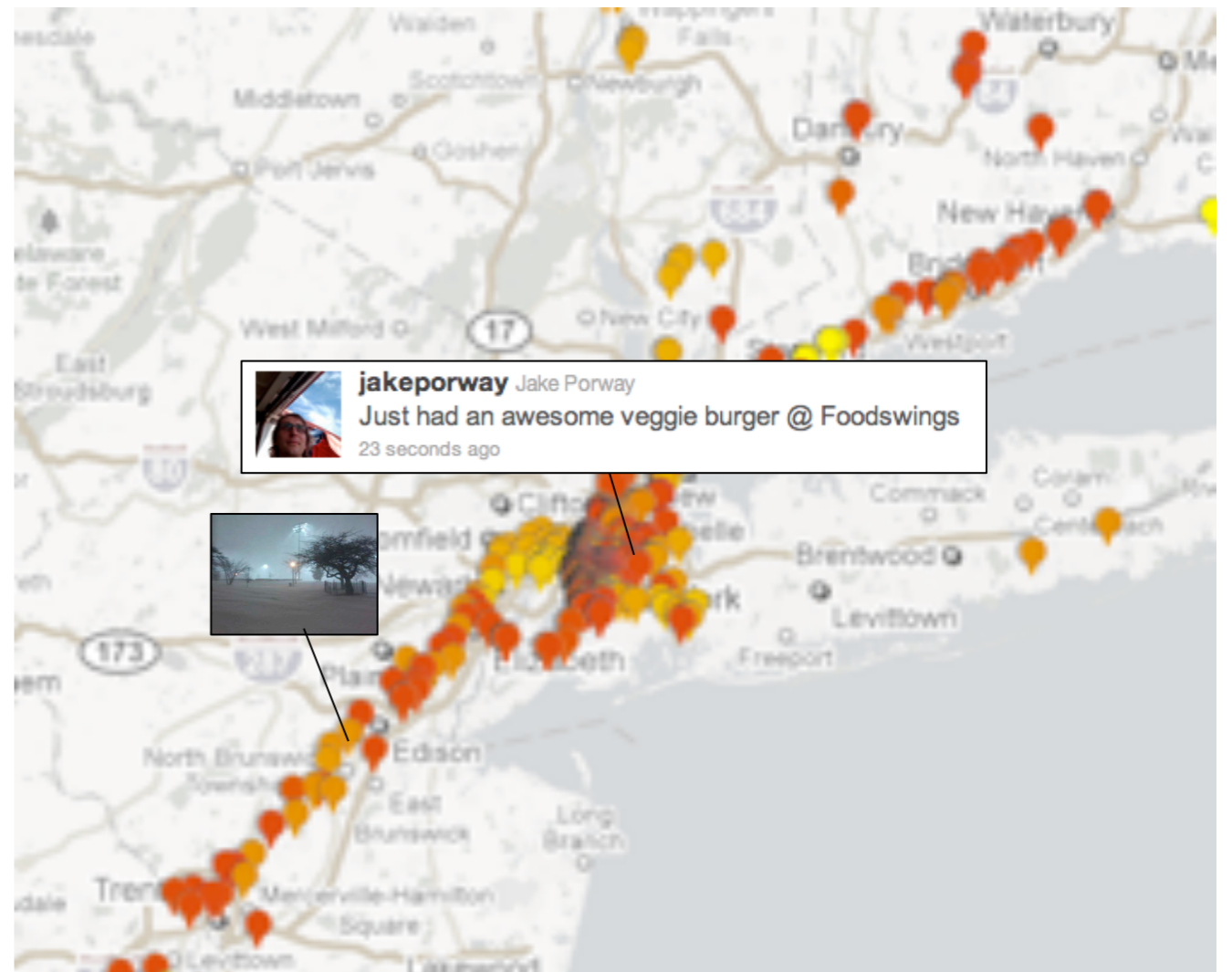
- iPhone data is actually already in a relational DB format. Why not just use it directly?
- Once again - research. Want to flexibly add things if needed.
- More importantly - Geo-indexing!

	MAC	Timestamp	Latitude	Longitude	HorizontalAccur	Altitude
1	0:22:b0:b5:9d:1	17915.230597	40.69091248	-73.95754283	89.0	0.0
2	0:23:97:80:a1:2	17915.230597	40.69097185	-73.95743179	60.0	0.0
3	0:23:97:31:e1:c	17915.230597	40.69104999	-73.95734107	61.0	0.0
4	0:26:5a:f5:df:d	17915.230597	40.69105523	-73.95720678	50.0	0.0
5	0:13:10:a0:bb:4	17915.230597	40.69094938	-73.95781272	92.0	0.0
6	0:23:97:87:56:3	17915.230597	40.69115346	-73.95743811	85.0	0.0
7	0:22:75:70:b3:8	17915.230597	40.69116401	-73.95742082	89.0	0.0
8	0:26:f2:bc:33:5	17915.230597	40.69102728	-73.95783603	67.0	0.0
9	0:18:3a:80:2a:2	17915.230597	40.69095414	-73.9568265	66.0	0.0
10	0:26:f2:d0:ec:1	17915.230597	40.69124788	-73.95726478	73.0	0.0
11	0:22:75:4a:f7:e	17915.230597	40.69128906	-73.95743376	99.0	0.0
12	0:f:66:38:3e:50	17915.230597	40.69117176	-73.95785462	77.0	0.0
13	94:44:52:6:cc:7	17915.230597	40.69129347	-73.95769411	69.0	0.0
14	0:22:b0:ba:b7:4	17915.230597	40.69134491	-73.9574269	64.0	0.0
15	94:44:52:9a:85	17915.230597	40.69105315	-73.95670694	50.0	0.0
16	f8:1e:df:fc:98:7	17915.230597	40.69125354	-73.95785462	50.0	0.0
17	90:84:d:ea:72:2	17915.230597	40.69111615	-73.9580602	84.0	0.0
18	0:25:4b:a:c3:5f	17915.230597	40.69124507	-73.95789247	92.0	0.0
19	0:1f:f3:f7:c0:2b	17915.230597	40.69124829	-73.95793062	50.0	0.0
20	0:11:24:eb:c:81	17915.230597	40.69125109	-73.9579274	50.0	0.0
21	0:16:b6:4b:78:9	17915.230597	40.69123297	-73.95797699	74.0	0.0
22	0:1b:2f:f2:ff:4	17915.230597	40.69123095	-73.95801317	50.0	0.0
23	0:1f:1f:3e:49:9	17915.230597	40.69122356	-73.95672059	97.0	0.0
24	0:f:66:d5:e4:5a	17915.230597	40.69137632	-73.9578039	59.0	0.0
25	0:22:3f:90:6a:6	17915.230597	40.69091731	-73.9583128	89.0	0.0
26	0:1c:10:bc:96:3	17915.230597	40.69115889	-73.95818656	50.0	0.0
27	0:1c:df:fc:99:de	17915.230597	40.69120323	-73.9581598	67.0	0.0



# Geo-Indexing with mongoDB

- We see people using this tool as a locative scrapbook.
- Want to be able to merge Twitter/Flickr/etc. with user's geo info, which mongoDB is great for.
- Also want to aggregate across users by position for researchers.
- (Stay tuned!)

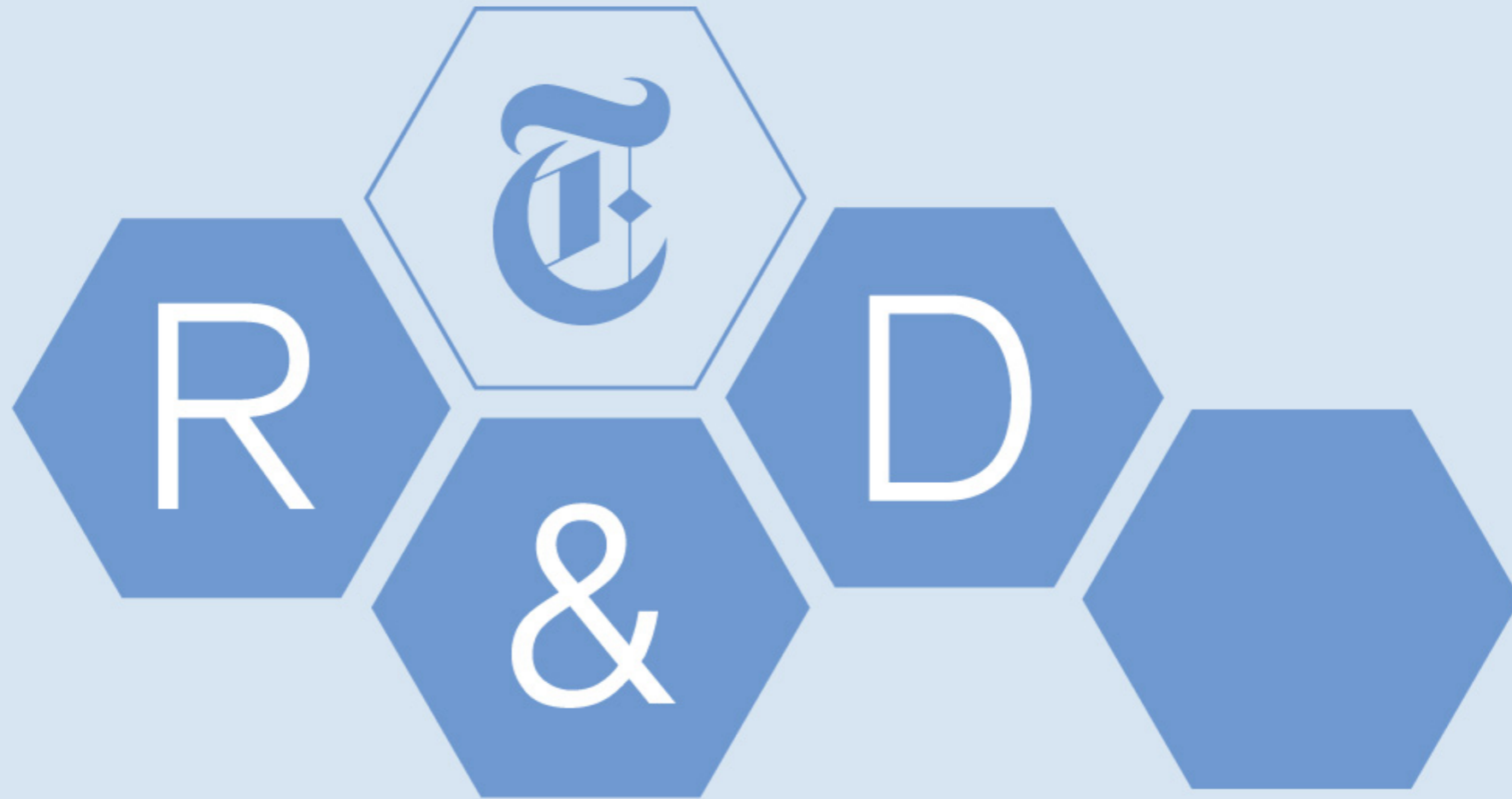


# Summary

- Overall, mongoDB is great for us because:
  - We can't anticipate clients' data sizes - need scalable DB.
  - Schema-less approach is critical for flexibility during research.
  - Map/reduce is a nice framework for customizable batch operations.
  - We do a lot of data collection, which mongo takes in stride.
  - Geo-indexing is super useful for any location-based projects.



Thanks!



THE NEW YORK TIMES  
RESEARCH & DEVELOPMENT

**Questions welcome at:  
[jakeporway@nytimes.com](mailto:jakeporway@nytimes.com)  
[@jakeporway](https://twitter.com/jakeporway)**