

elasticsearch

the missing intro

by Erik Rose



Part 1: Indexing & Querying



what it's good for

- ☁ Full-text search
- ☁ Big data
- ☁ Faceting
- ☁ Geographical queries

February 7th 2010 - January 20th 2013

Commits to master, excluding merge commits

Contribution Type: **Commits**



kimchy #1
3,819 commits / 1,299,650 ++ / 953,530 --

imotov #2
107 commits / 9,497 ++ / 3,280 --

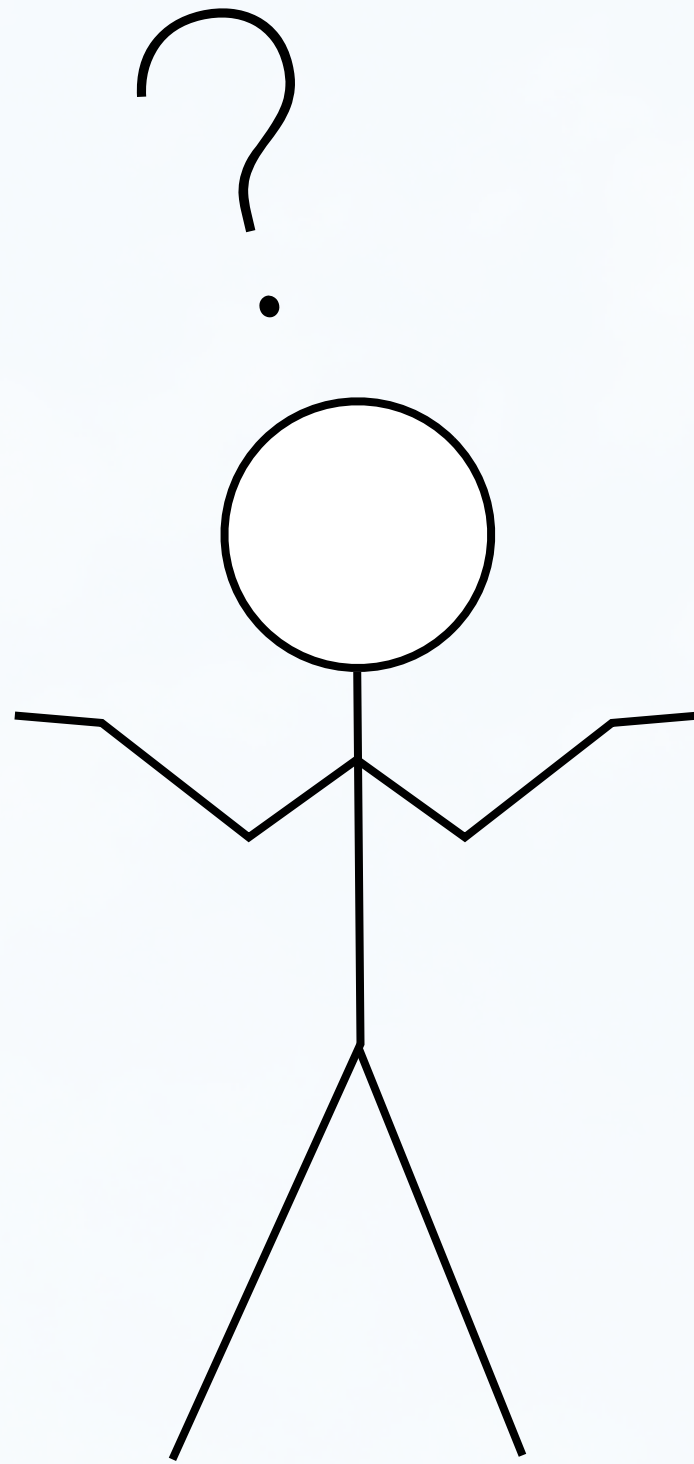
martijnvg #3
95 commits / 13,676 ++ / 4,953 --

s1monw #4
33 commits / 7,051 ++ / 5,804 --

Paikan #5
23 commits / 2,873 ++ / 398 --

lukas-vlcek #6
21 commits / 8,708 ++ / 430 --

one (insanely productive) man

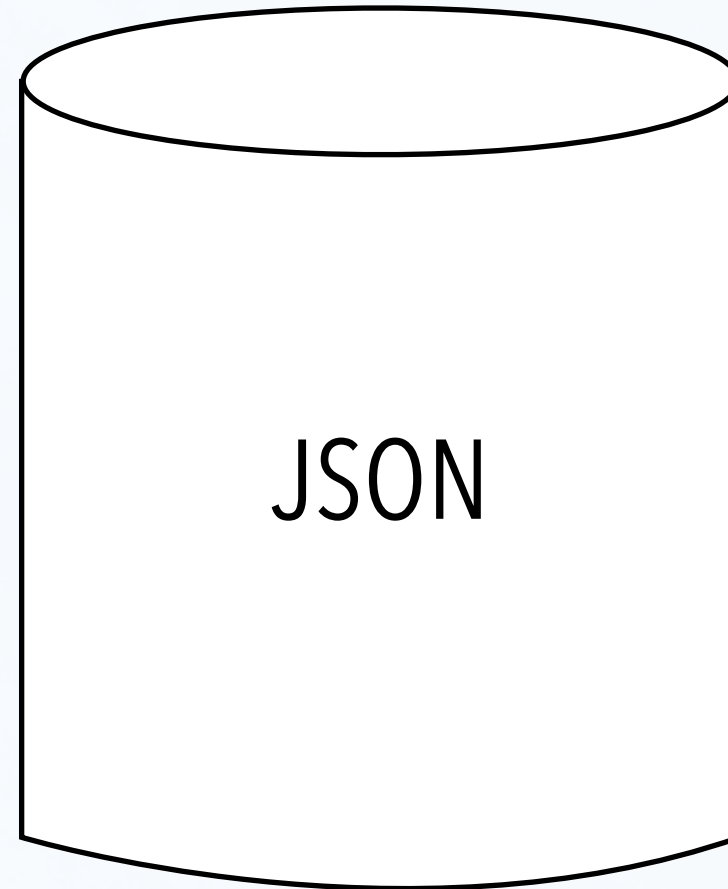


the rest of us

data structures

HTTP

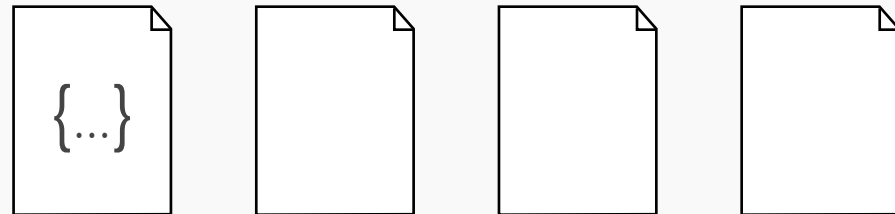
on port 9200



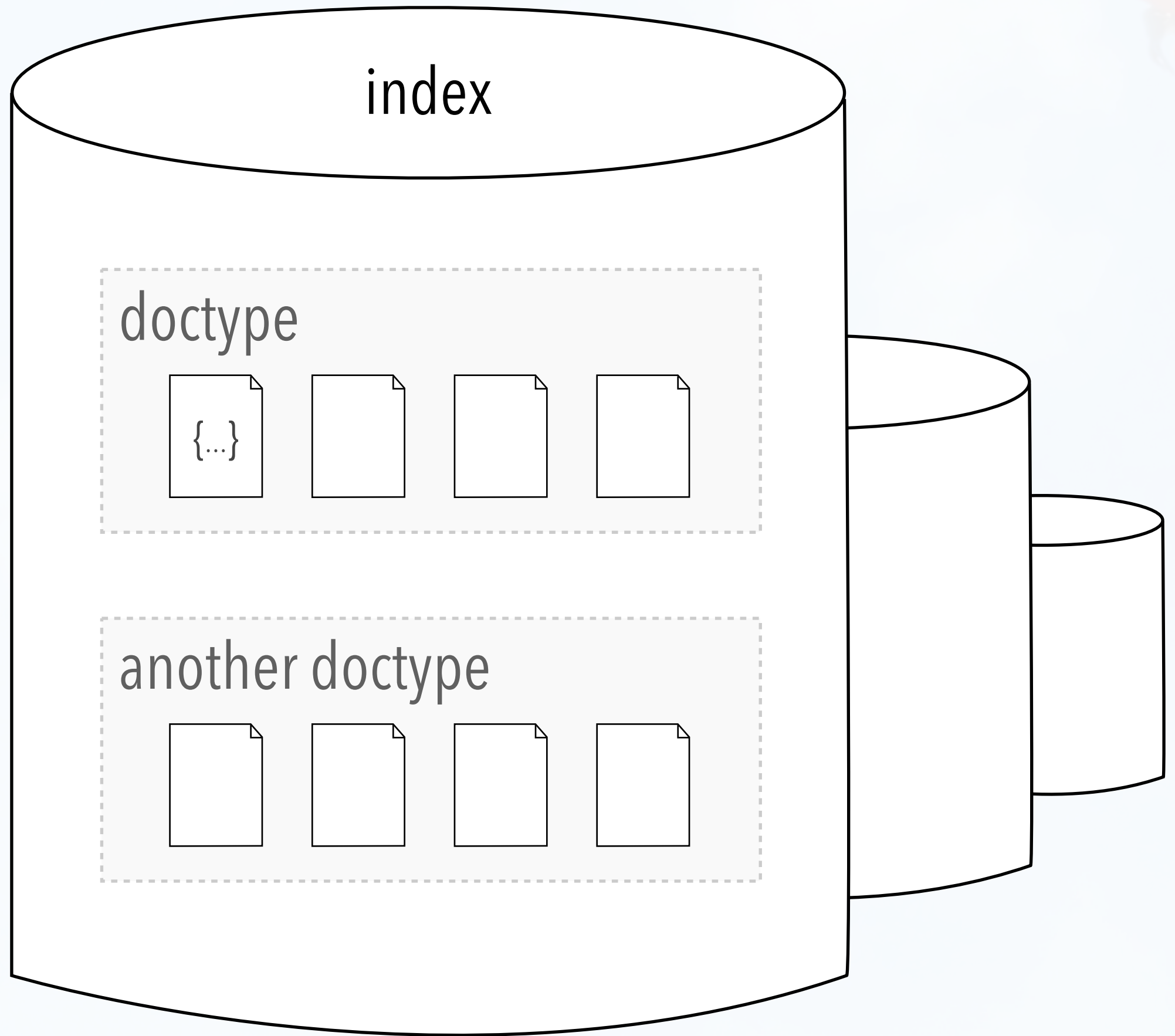
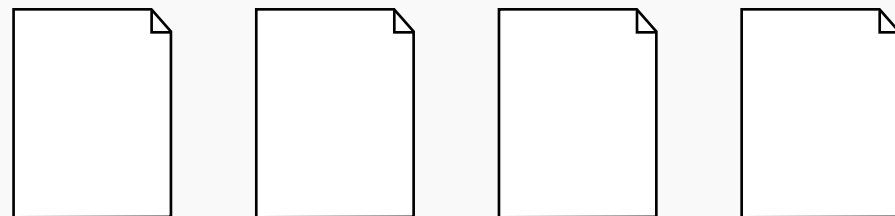
JSON

index

doctype



another doctype



IDs

```
6a8ca01c-7896-48e9-  
81cc-9f70661fcb32
```

diplodocus	333
duodenum	201
dwaal	500, 119

row row
row your
boat

0

row the
row boat

1

chicken
chicken
chicken

2

the
front
row

3

row → 0, 1, 3
boat → 0, 1
chicken → 2

row row
row your
boat

0

row the
row boat

1

chicken
chicken
chicken

2

the
front
row

3

row → 0, 1, 3
boat → 0, 1
chicken → 2

row row
row your
boat

0

row the
row boat

1

chicken
chicken
chicken

2

the
front
row

3

row → 0, 1, 3
boat → 0, 1
chicken → 2

row row
row your
boat

0

row the
row boat

1

chicken
chicken
chicken

2

the
front
row

3

row → 0, 1, 3
boat → 0, 1
chicken → 2

row row
row your
boat

0

row the
row boat

1

chicken
chicken
chicken

2

the
front
row

3

row → 0, 1, 3
boat → 0, 1
chicken → 2

row row
row your
boat

0

row the
row boat

1

chicken
chicken
chicken

2

the
front
row

3

row

→

doc

positions

0

[0, 1, 2]

1

[0, 2]

3

[2]

boat

→

0

[4]

1

[3]

chicken

→

2

[0, 1, 2]

row row
row your
boat

0

row the
row boat

1

chicken
chicken
chicken

2

the
front
row

3

row →

boat →

chicken →

doc

positions

0 [0, 1, 2]

1 [0, 2]

3 [2]

0 [4]

1 [3]

2 [0, 1, 2]

row row
row your
boat

0

row the
row boat

1

chicken
chicken
chicken

2

the
front
row

3

row

→

doc

positions

0

[0, 1, 2]

1

[0, 2]

3

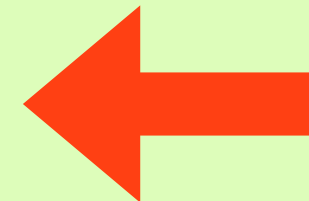
[2]

boat

→

0

[4]



?

1

[3]

chicken

→

2

[0, 1, 2]

row row
row your
boat

0

row the
row boat

1

chicken
chicken
chicken

2

the
front
row

3

row →

boat →

chicken →

doc

positions

0 [0, 1, 2]

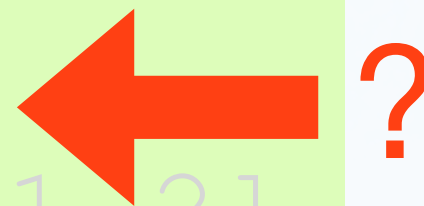
1 [0, 2]

3 [2]

0 [4]

1 [3]

2 [0, 1, 2]



The background of the image is a clear blue sky with several wispy white clouds scattered across it. The clouds are most prominent in the upper right and lower left areas, leaving a large clear blue space in the center.

analysis

stock analyzers

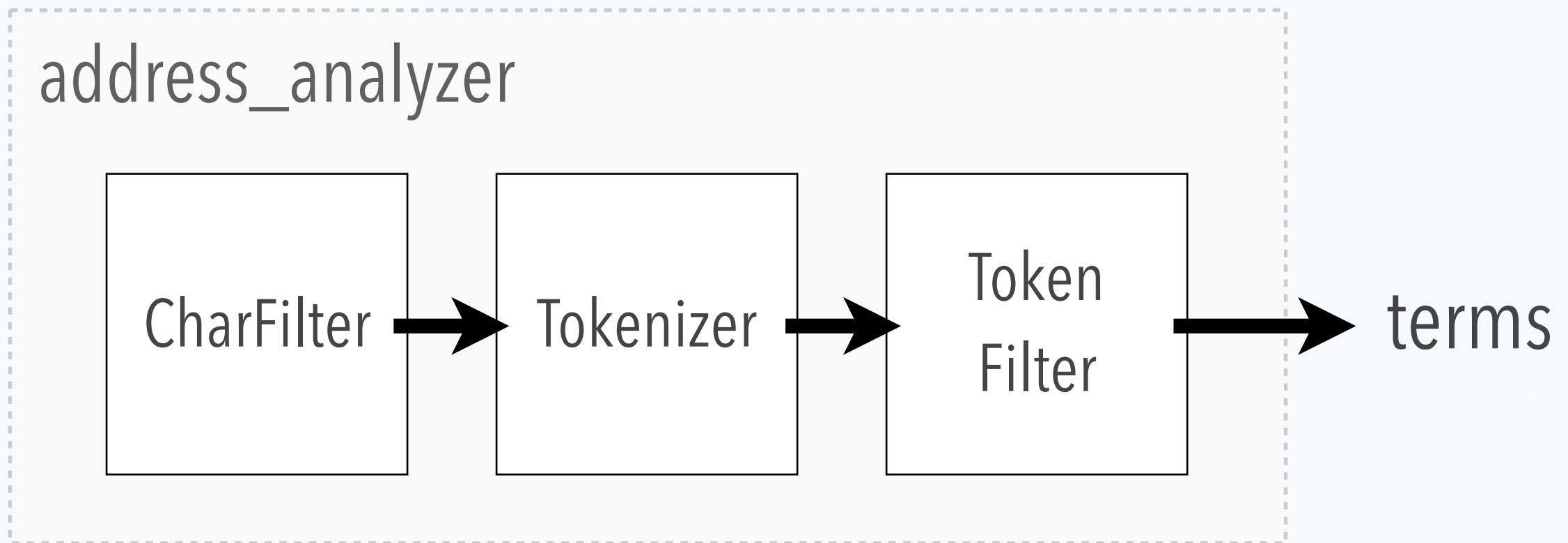
original:	Red-orange gerbils live at #43A Franklin St.
whitespace:	Red-orange, gerbils, live, at, #43A, Franklin, St.
standard:	red, orange, gerbils, live, 43a, franklin, st
simple:	red, orange, gerbils, live, at, a, franklin, st
stop:	red, orange, gerbils, live, franklin, st
snowball:	red, orang, gerbil, live, 43a, franklin, st

- ☁ stopwords
- ☁ stemming
- ☁ punctuation
- ☁ case-folding

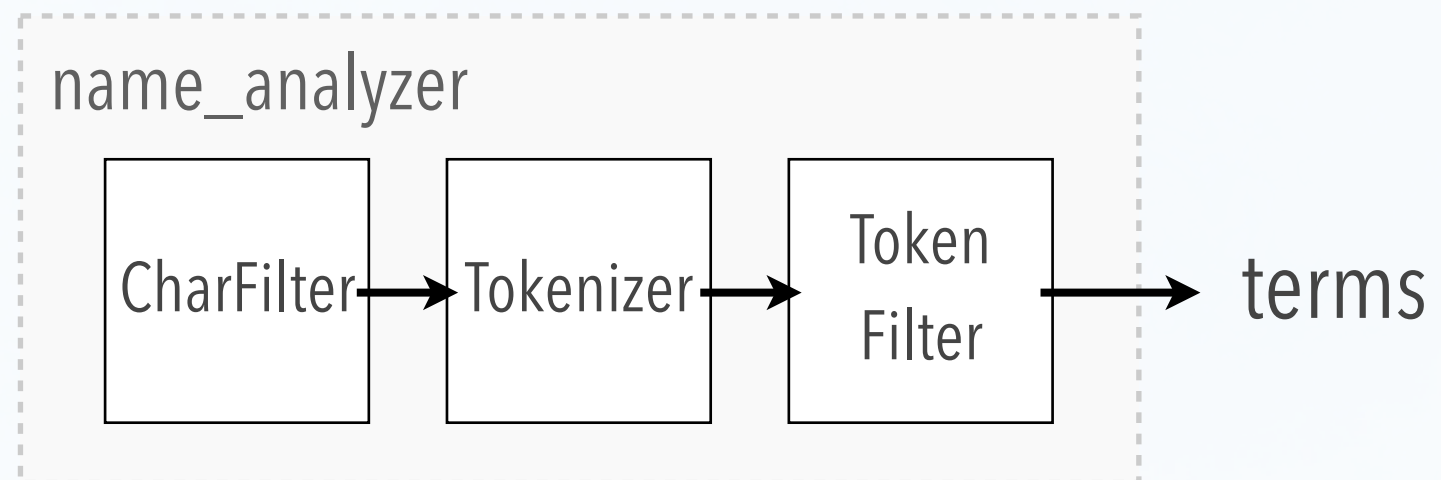
```
curl -XGET -s 'http://localhost:9200/_analyze?
analyzer=whitespace&pretty=true' -d 'Red-
orange gerbils live at #43A Franklin St.'
```

```
{
  "tokens" : [ {
    "token" : "Red-orange",
    "start_offset" : 0,
    "end_offset" : 10,
    "type" : "word",
    "position" : 1
  }, {
    "token" : "gerbils",
    "start_offset" : 11,
    "end_offset" : 18,
    "type" : "word",
    "position" : 2
  }, ...
```

```
'address':  
  {'type': 'string',  
   'analyzer': 'address_analyzer'}
```




```
'analysis': {
  'analyzer': {
    'name_analyzer': {
      'type': 'custom',
      'tokenizer': 'name_tokenizer',
      'filter': ['lowercase']
    }
  },
  'tokenizer': {
    'name_tokenizer': {
      'type': 'pattern',
      'pattern': "[^a-zA-Z']+"
    }
  }
}
```



synonyms

```
"filter": {  
  "synonym": {  
    "type": "synonym",  
    "synonyms": [  
      "albert => albert, al",  
      "allan => allan, al"  
    ]  
  }  
}
```

original query: Allan Smith
after synonyms: [allan, al] smith

original query: Albert Smith
after synonyms: [albert, al] smith

querying


```
{
  "bool" : {
    "must" : {
      "term" : { "user" : "fred" }
    },
    "must_not" : {
      "range" : {
        "age" : { "from" : 12, "to" : 21 }
      }
    },
    "should" : [
      {
        "term" : { "tag" : "crunchy" }
      },
      {
        "term" : { "tag" : "elasticsearch" }
      }
    ],
    "minimum_number_should_match" : 1,
    "boost" : 1.0
  }
}
```

Queries

- match
- multi_match
- bool
- boosting
- ids
- custom_score
- custom_boost_factor
- constant_score
- dis_max
- field
- filtered
- flt
- flt_field
- fuzzy
- has_child
- has_parent
- match_all
- mlt
- mlt_field
- prefix
- query_string
- range
- span_first
- span_near
- span_not
- span_or
- span_term
- term
- terms
- top_children
- wildcard
- nested
- custom_filters_score
- indices
- text
- geo_shape

Filters

- and
- bool
- exists
- ids
- limit
- type
- geo_bbox
- geo_distance
- geo_distance_range
- geo_polygon
- geo_shape
- has_child
- has_parent
- match_all
- missing
- not
- numeric_range
- or
- prefix
- query
- range
- script
- term
- terms
- nested

filters

- ☁ Boolean
- ☁ Fast
- ☁ Cacheable

queries

- ☁ Fuzzy, scoring
- ☁ Slower
- ☁ Not cacheable

```
curl -XGET -s 'http://localhost:9200/blog/_search?pretty=true' -d \  
{  
  "query": {  
    "filtered": {  
      "filter": {  
        "term": {  
          "category": "rants"  
        }  
      },  
      "query": {  
        "bool": {  
          "should": [  
            {  
              "match_phrase": {  
                "body": "fix your little red wagon"  
              }  
            },  
            {  
              "match": {  
                "body": "fix your little red wagon"  
              }  
            }  
          ]  
        }  
      }  
    }  
  }  
}
```

libraries

☁ ~~ps~~

☁ pyelasticsearch

```
query = {'query': {  
    'filtered': {  
        'query': {  
            'match': {'name': 'test tester'}  
        },  
        'filter': {  
            'range': {  
                'age': {'from': 27, 'to': 37}  
            }  
        }  
    }  
}}}}  
es.search(query, index='people')
```

☁ elasticsearch

```
print [item['title'] for item in  
    searcher.query(title__text='cookie')  
    .filter(topics='websites')]
```


thank you

twitter: ErikRose
erik@mozilla.com

Part 2: Sunday at 1:10pm

Background image by Tim and Julie Wilson: <https://secure.flickr.com/photos/secondtree/>.

This presentation is noncommercial sharealike in accordance with that image's license.