

基于 Elastic Stack 的数据探索与分析

@medcl

About me

- 曾勇（Medcl）
- Developer @ Elastic
  - Follow Elasticsearch since v0.5, 2010
  - Joined Elastic since September, 2015
  - Now in Beats team
- @medcl
- medcl@elastic.co
- http://github.com/medcl
- Based in Changsha, Hunan, China



elastic

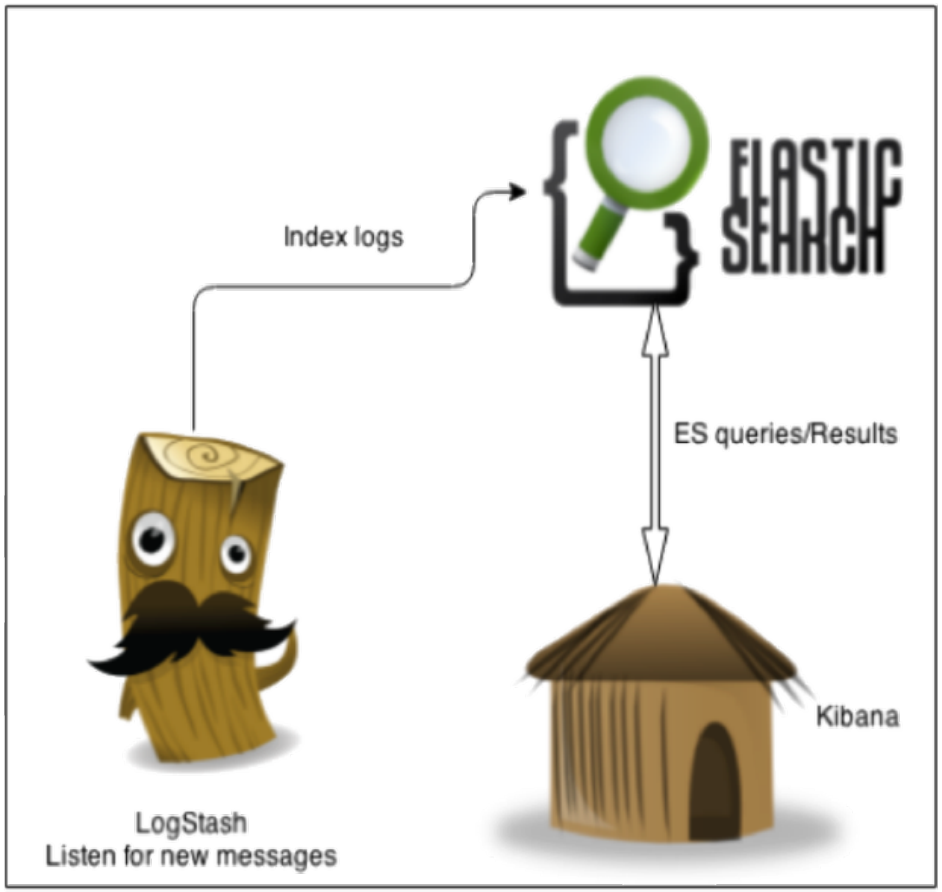
What's Elastic?

- A distributed startup company, since 2012
  - HQ: Mountain View, CA AND Amsterdam, Netherlands
  - With employees in 27 countries (and counting), spread across 18 time zones, speaking over 30 languages
- We are working on Open Source projects!
  - (Luckily some of them are popular, eg:elasticsearch)
- Offering support Subscription, X-pack, Cloud and Trainings
- Find us on: https://github.com/elastic and https://www.elastic.co



elastic

听说过 “ELK” 么?



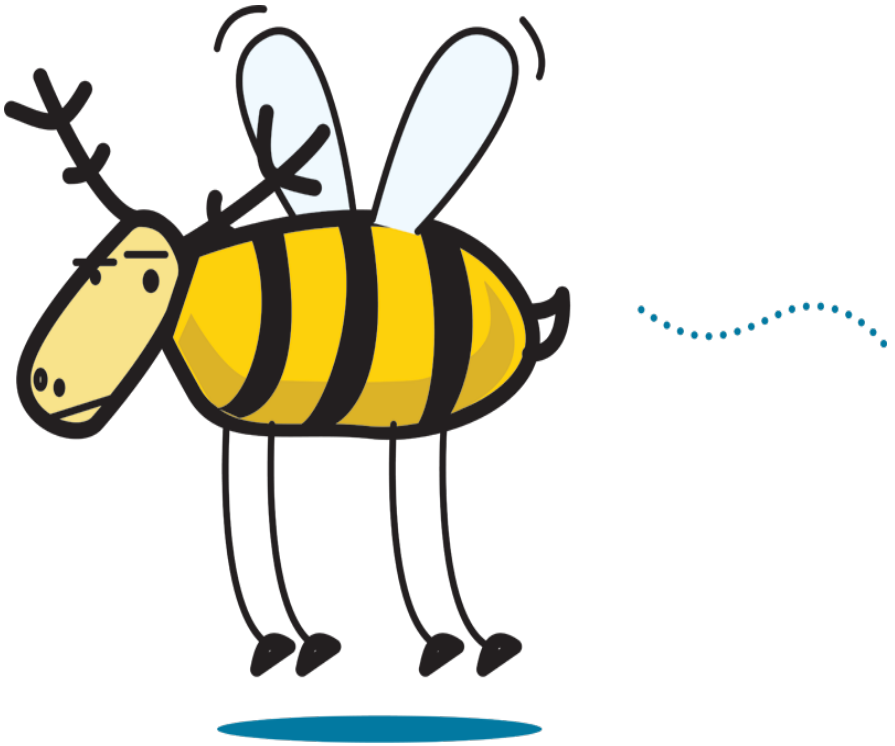


elastic

But ELK is out!



Beats & Packetbeat



ELKB? BELK? LKBE? BKEL?

elastic

Logo


ELASTIC
SEARCH


elasticsearch.


elasticsearch.


elasticsearch.






Kibana


kibana


kibana


kibana

Release Bonanza

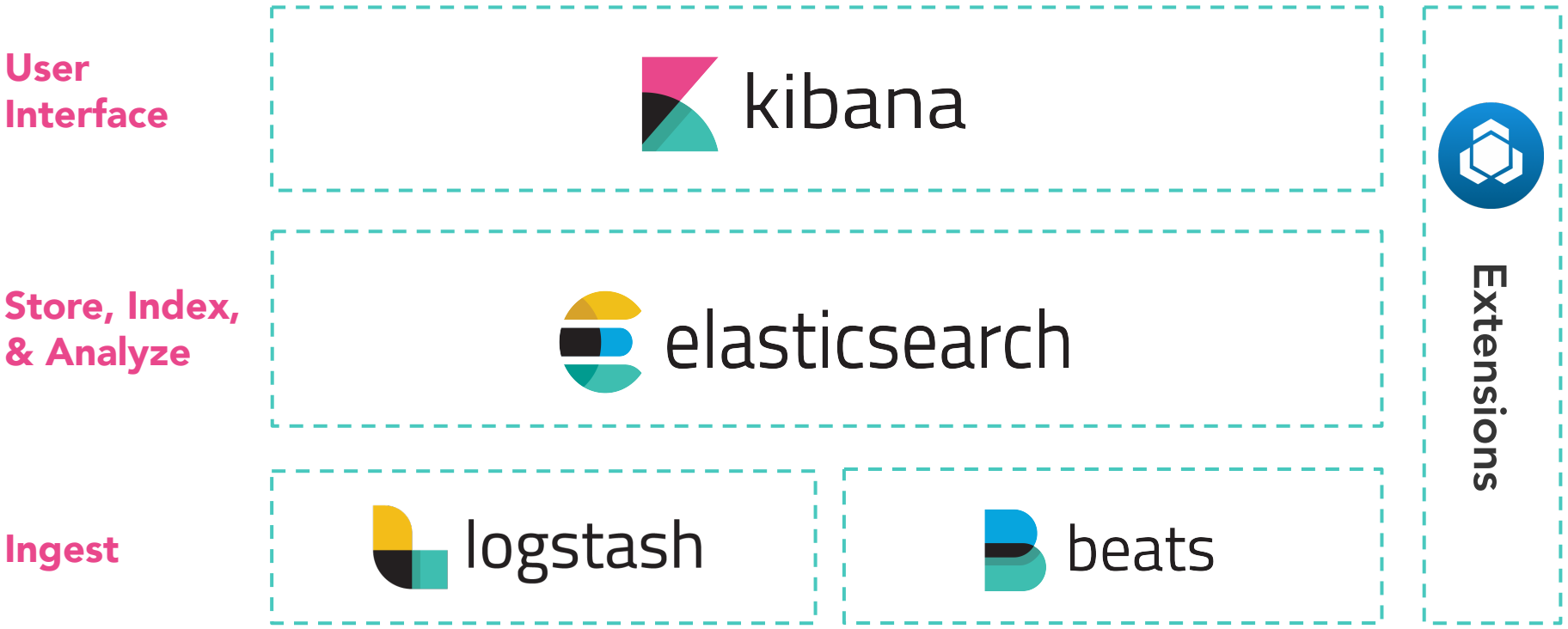
	Oct 28th	Nov 21st	Feb 2nd
es	2.0	2.1	2.2
kibana	4.2	4.3	4.4
ls	2.0	2.1	2.2
beats		1.0	1.1

elastic

It’s time to unite!

elastic

The “Elastic Stack” , stay together from v 5.0

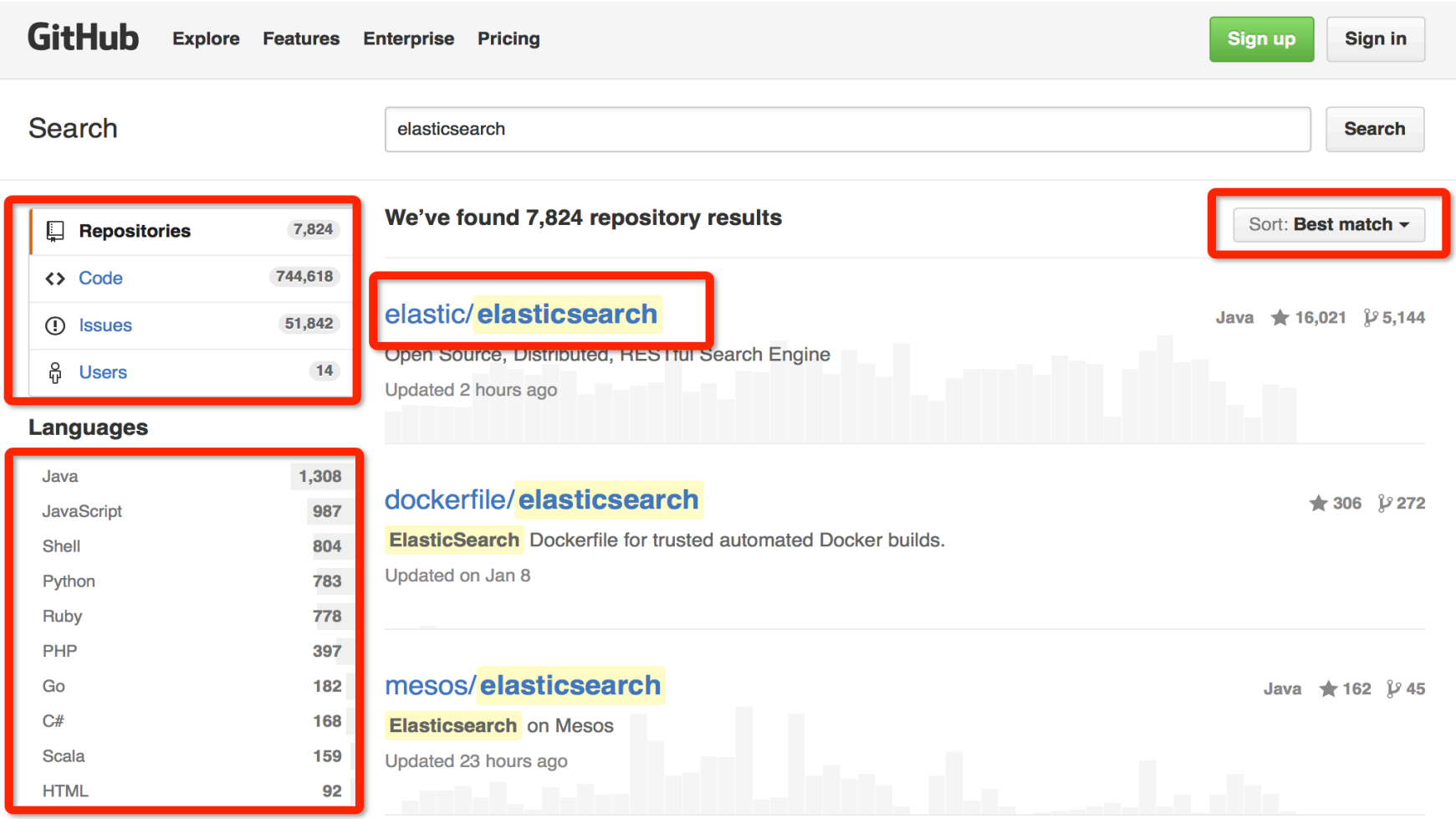


elastic

Elastic Stack
能做什么?

elastic

Github: Enable Powerful Search For Both End-Users And Developers

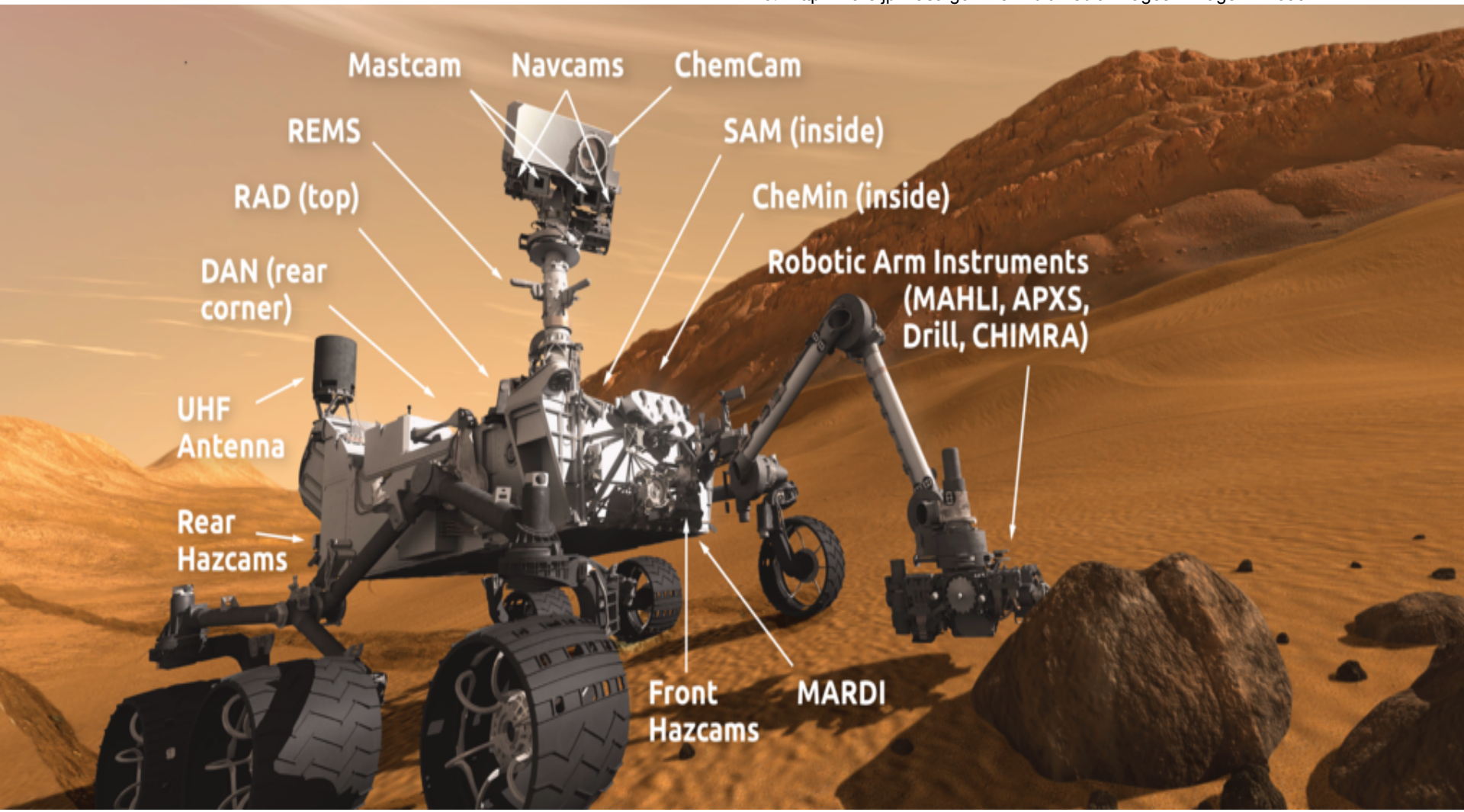


https://www.elastic.co/use-cases/github

elastic

NASA: Unlocking Interplanetary Datasets with Real-Time Search

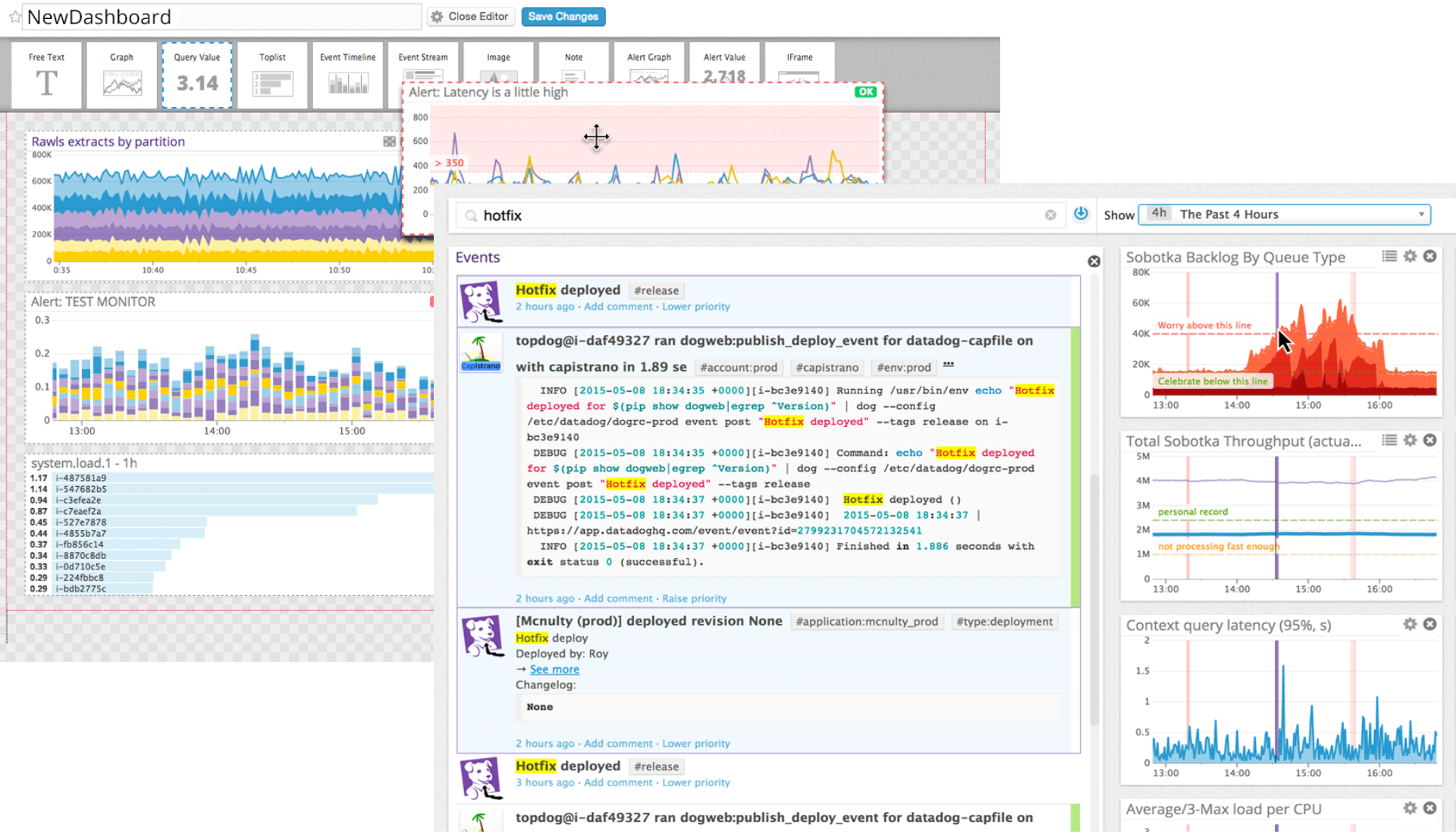
Pic：http://mars.jpl.nasa.gov/msl/multimedia/images/?ImageID=7693



https://www.elastic.co/elasticon/2015/sf/unlocking-interplanetary-datasets-with-real-time-search

elastic

Datadog：analysis metrics and time-series data



https://www.elastic.co/use-cases/data-dog

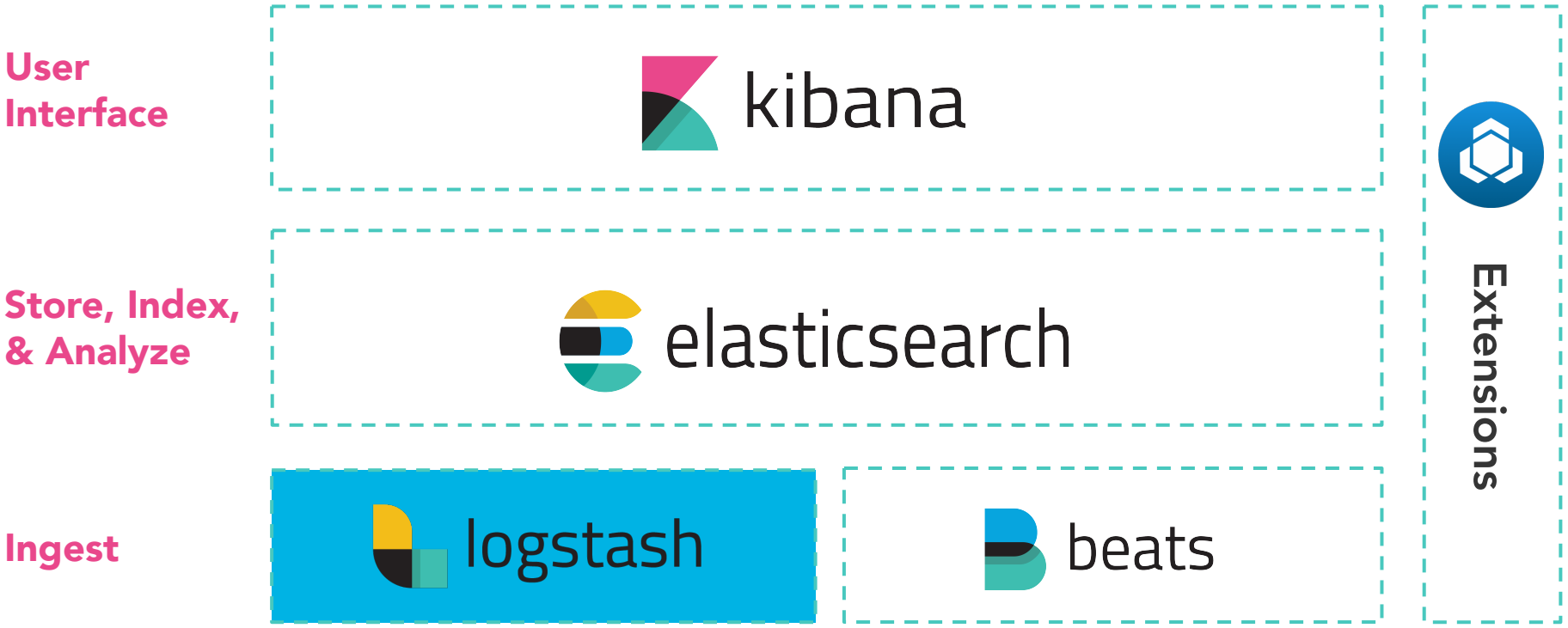
elastic

更多：

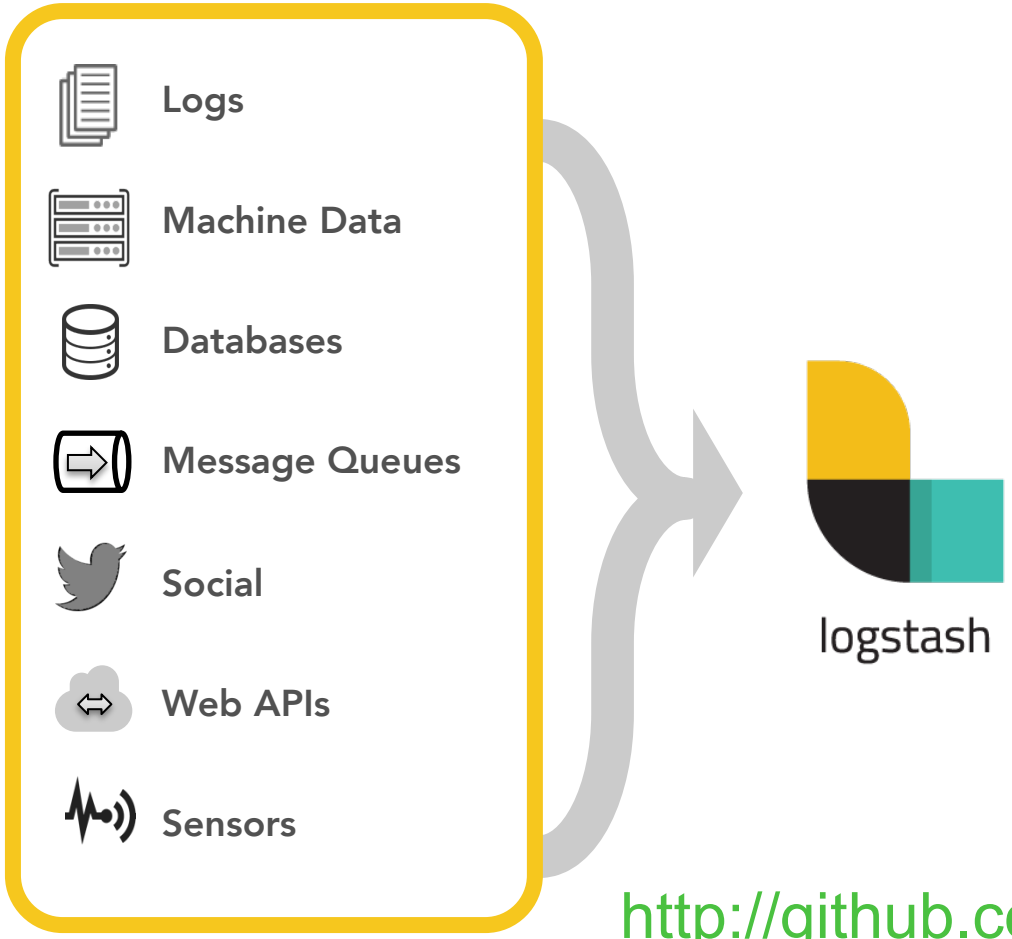
https://www.elastic.co/use-cases

elastic

The “Elastic Stack”



Logstash: Collect from diverse inputs

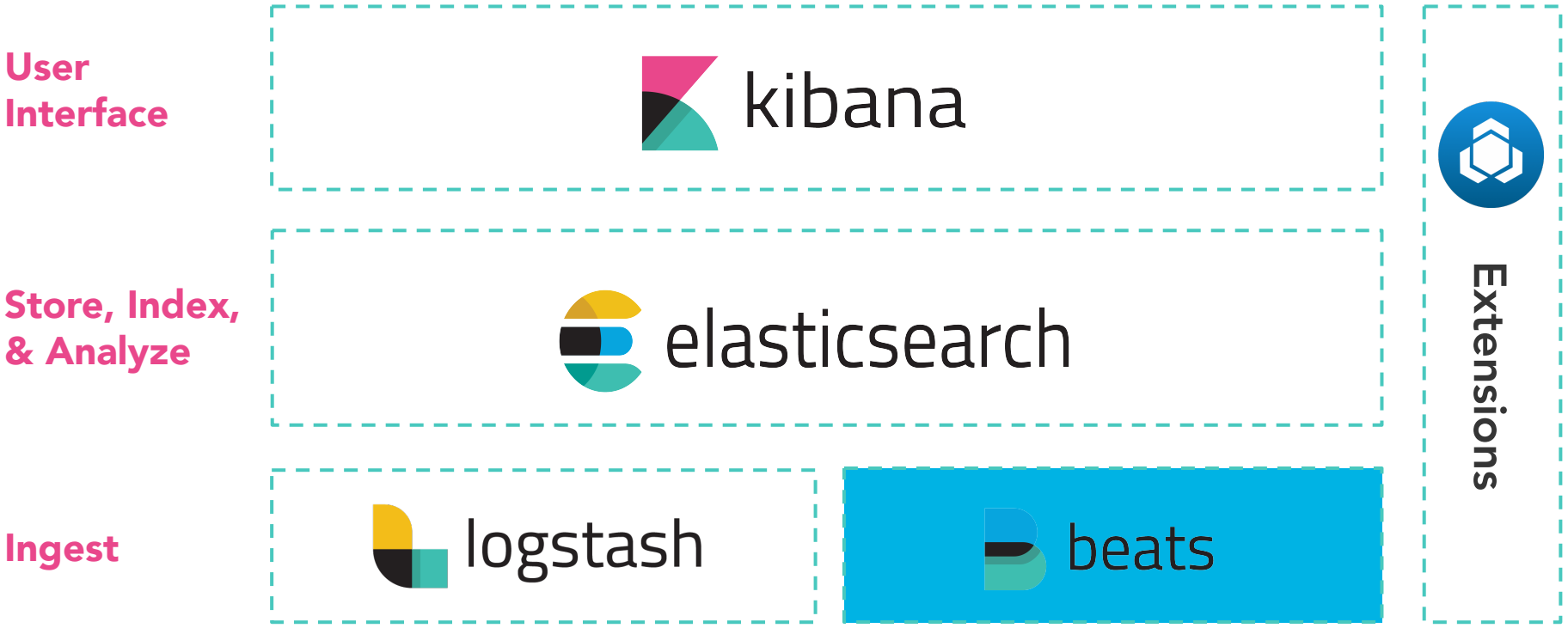


- Collects diverse sources
  - Logs + many others
  - Over 200 plugins
- Connects with live streams
  - Real-Time data
  - Wire / Transaction data
  - Full-Packet Network Capture

http://github.com/elastic/logstash

elastic

The “Elastic Stack”



beats

- Beats are lightweight shippers that collect and ship all kinds of **operational data** to Elasticsearch
  - Small application
  - Install as agent on your servers
  - Written in **Golang**
  - No runtime dependencies
  - Single purpose

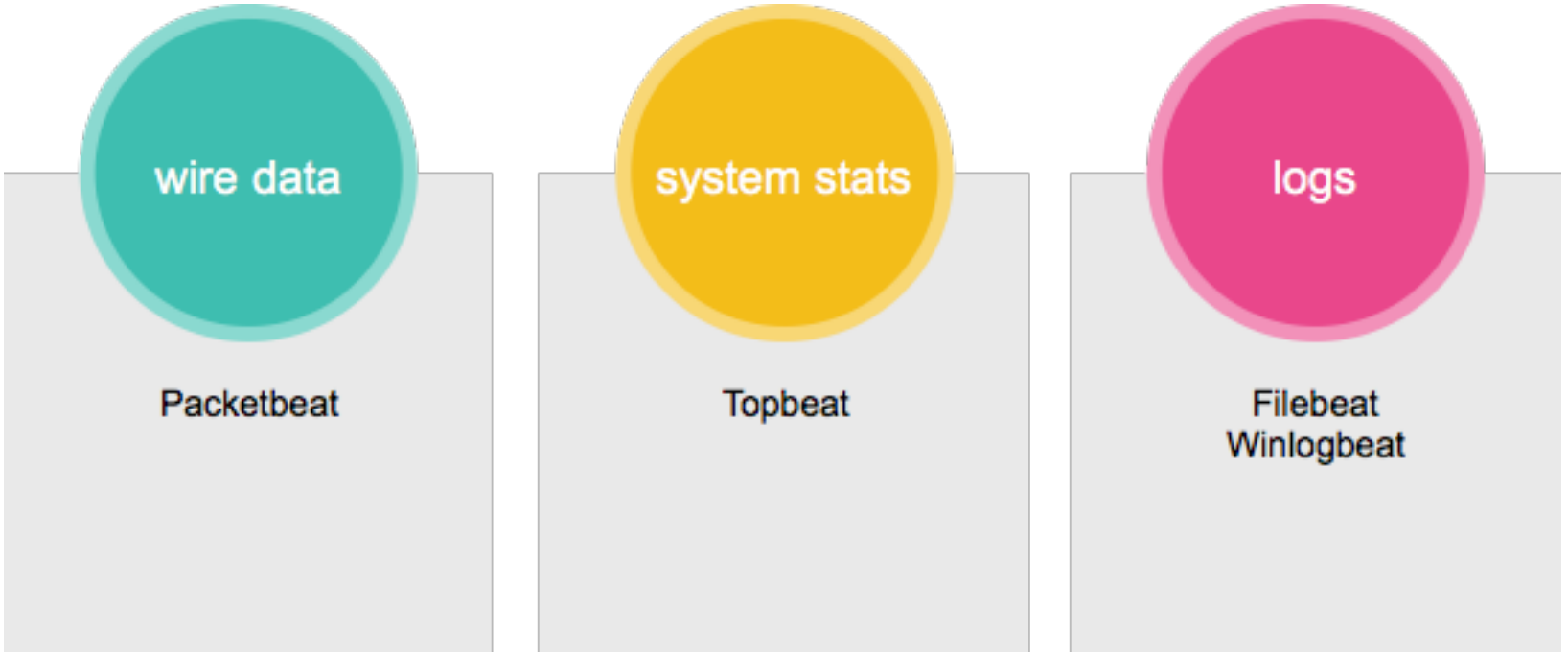




https://www.flickr.com/photos/8barbikes/17256970434/

http://github.com/elastic/beats

Examples of operational data



elastic

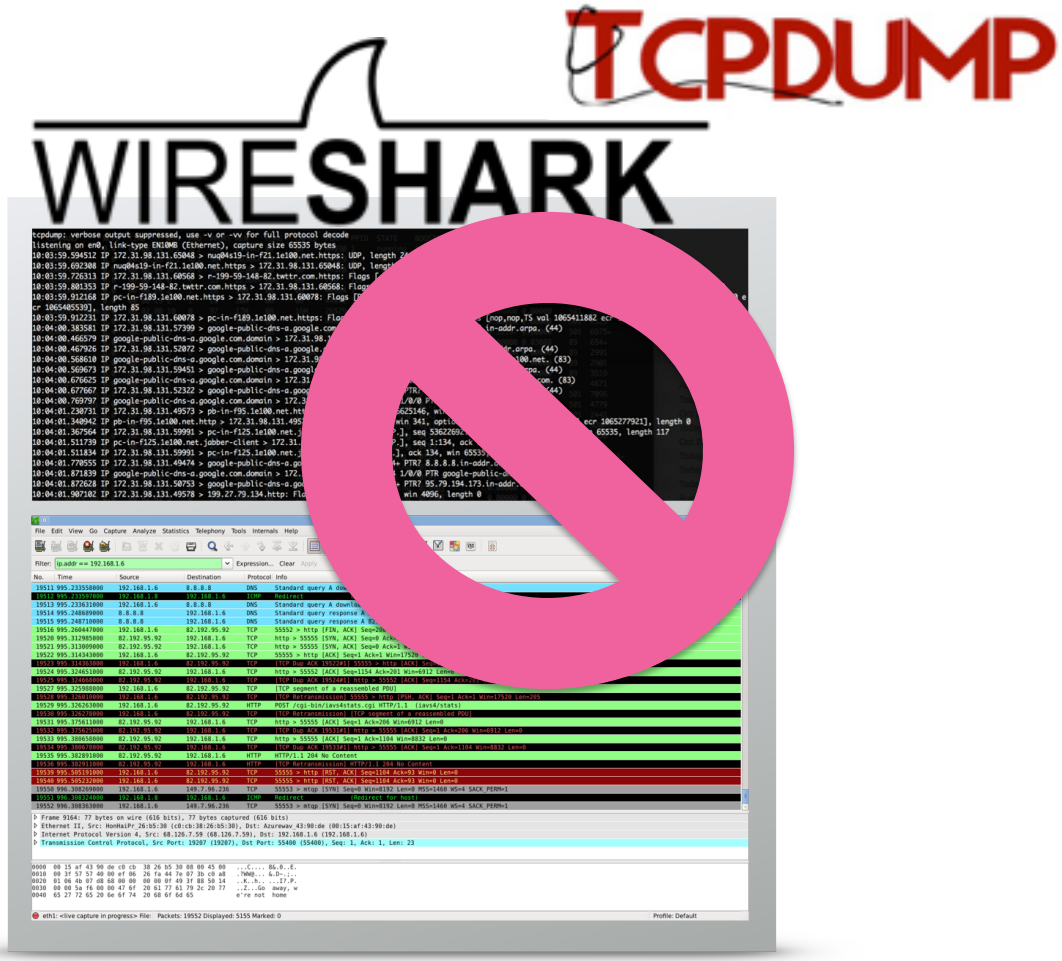
Packetbeat：Real-time application monitoring

Sniffs the traffic
between your servers,
parses the **application-level**
protocols on the fly.

Built-in protocols：

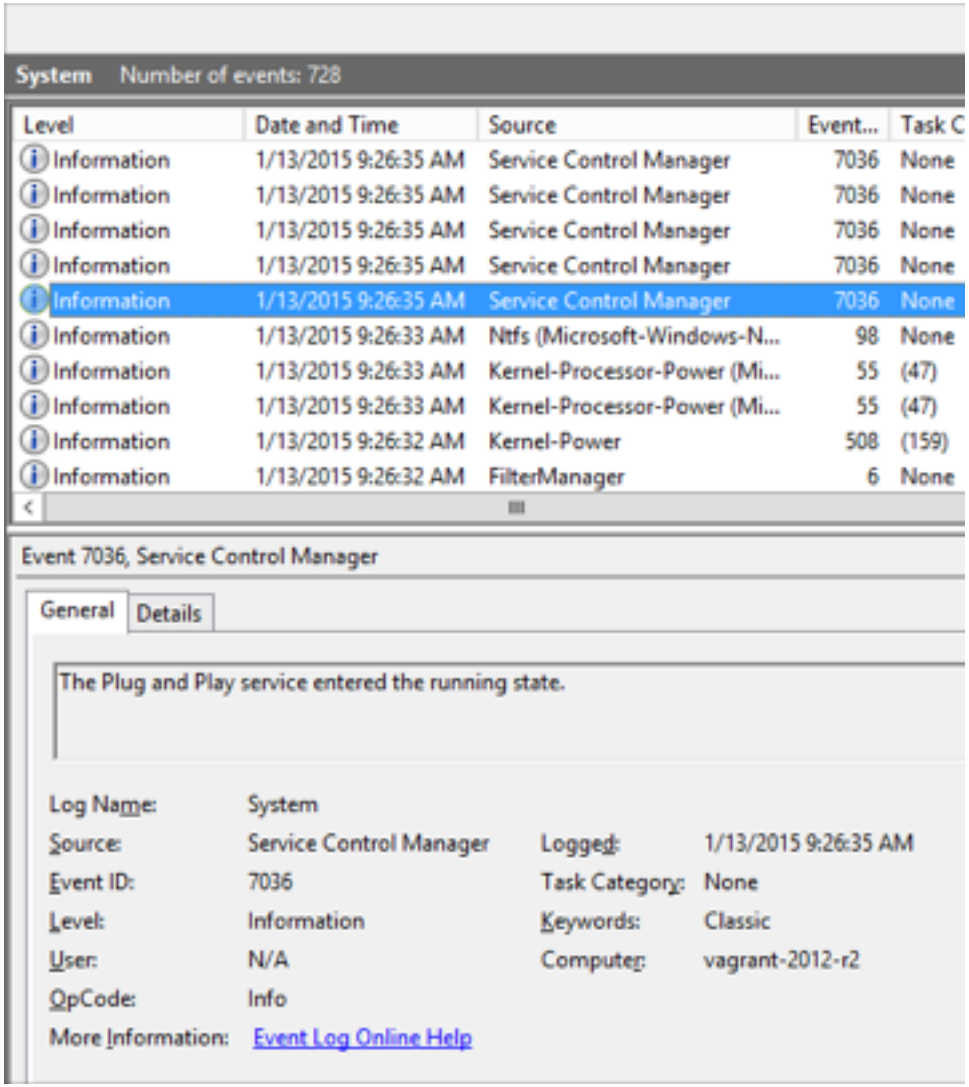
- HTTP
- MySQL
- PostgreSQL
- Redis
- Thrift-RPC
- MongoDB
- DNS
- Memcache
- ICMP
- AMQP
- …

Let’s go realtime!



elastic

winlogbeat!



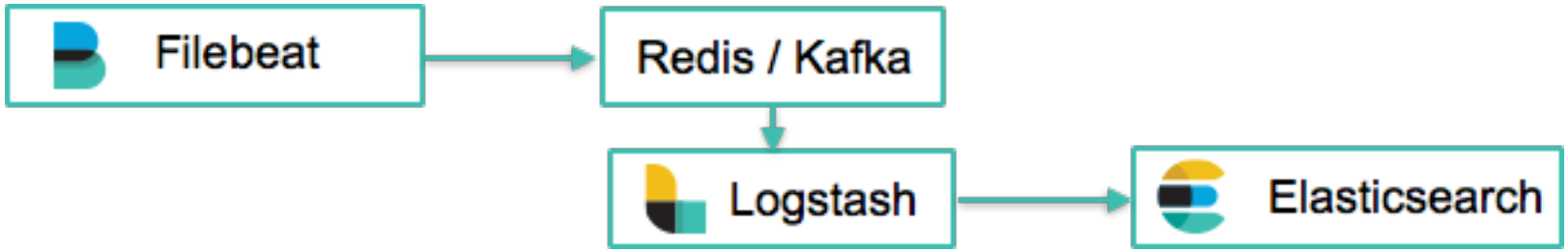
Forwards Windows Event logs to Elasticsearch

elastic

Filebeat

A more lightweight log shipper

- **Generic filtering**
  Flexibly reduce the amount of data sent of the wire and stored



elastic

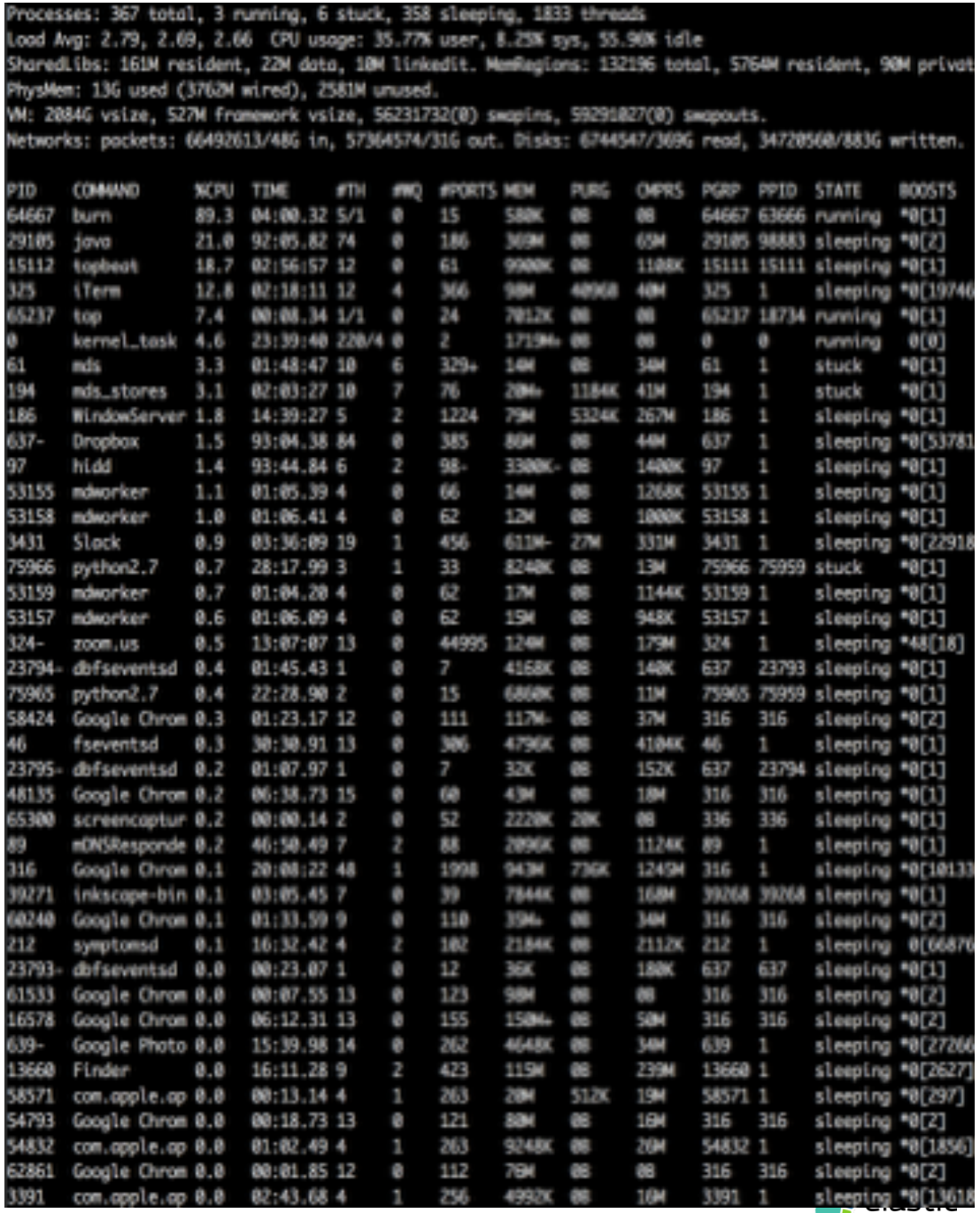
Topbeat

Like the Unix **top** command but sends the output periodically to Elasticsearch. Also works on Windows.

System wide
- system load
- total CPU usage
- …

Per process
- state
- name
- command line
- …

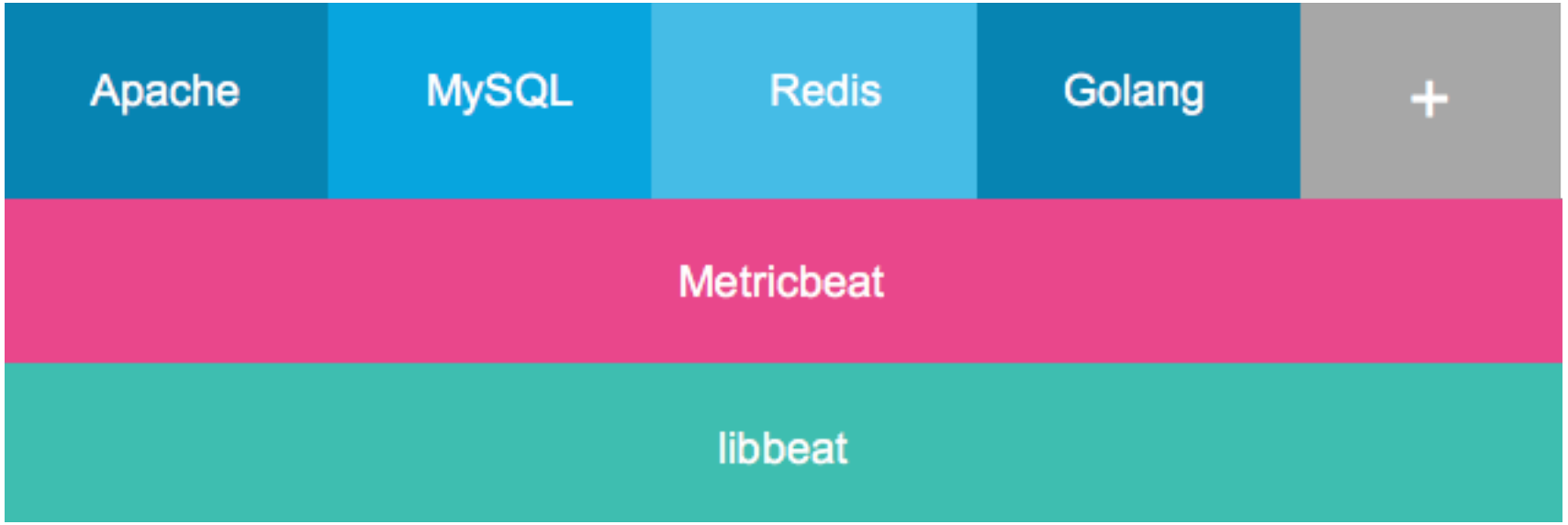
Disk usage
- available disks
- used, free space
- …



That's More!

Metricbeat: Connecting **Numb3rs**

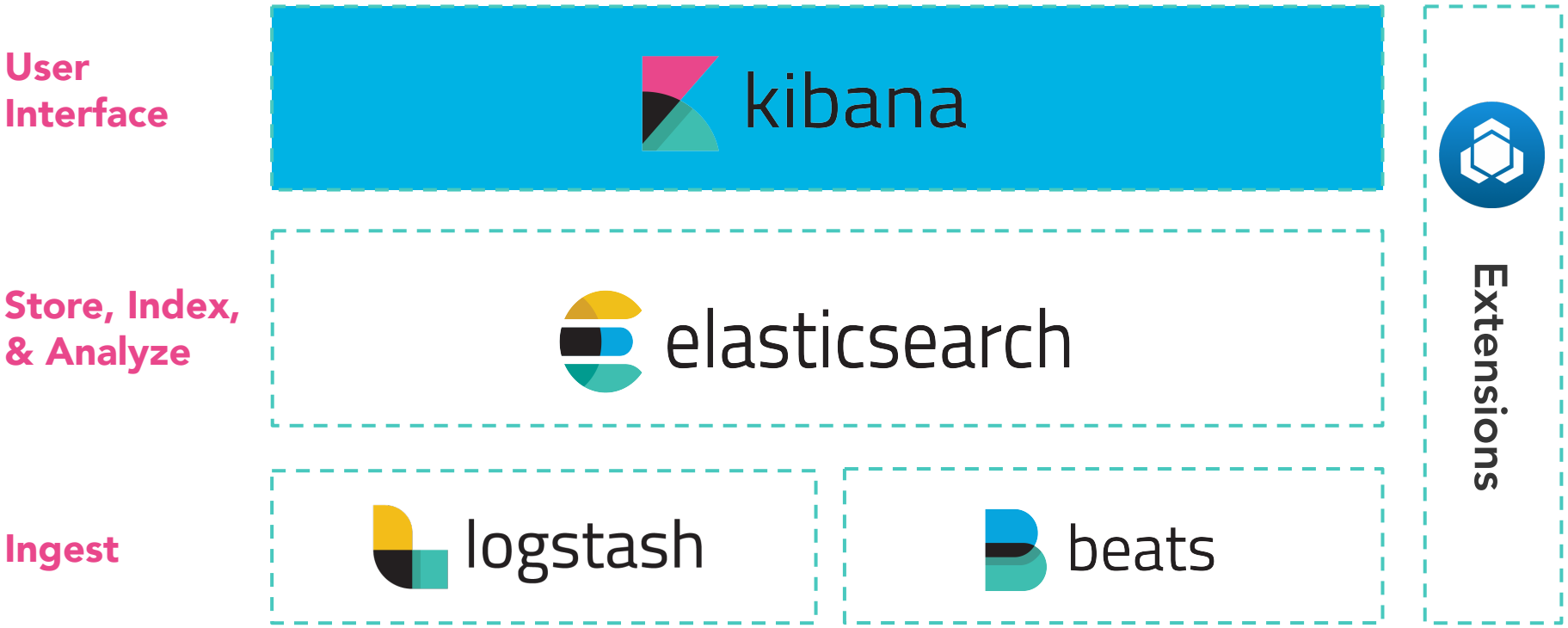
- Listens to the internal "beat" of systems via APIs.



http://github.com/elastic/beat-generator/

elastic

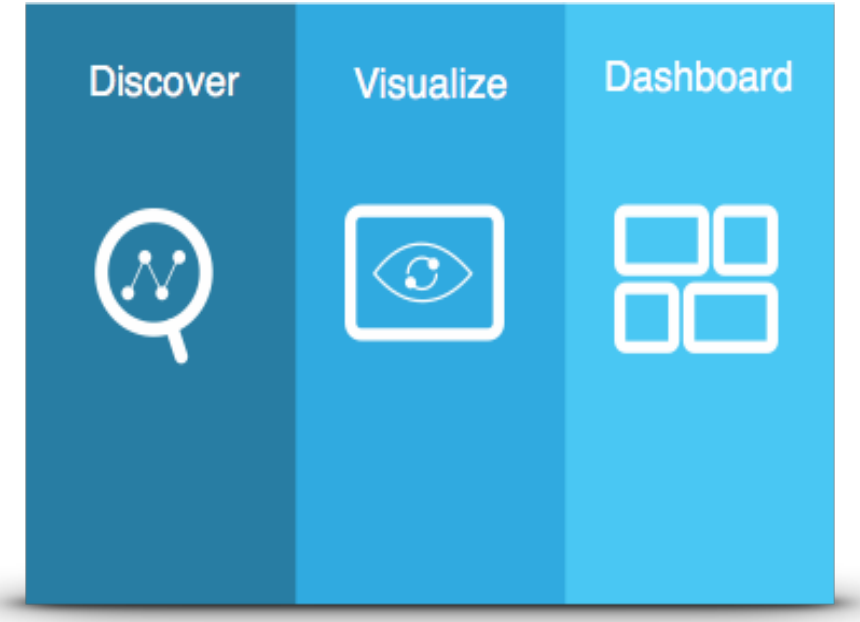
The "Elastic Stack"



What's Kibana?

kibana

Kibana is an
open source analytics
and **visualization platform**
designed to work with
Elasticsearch.

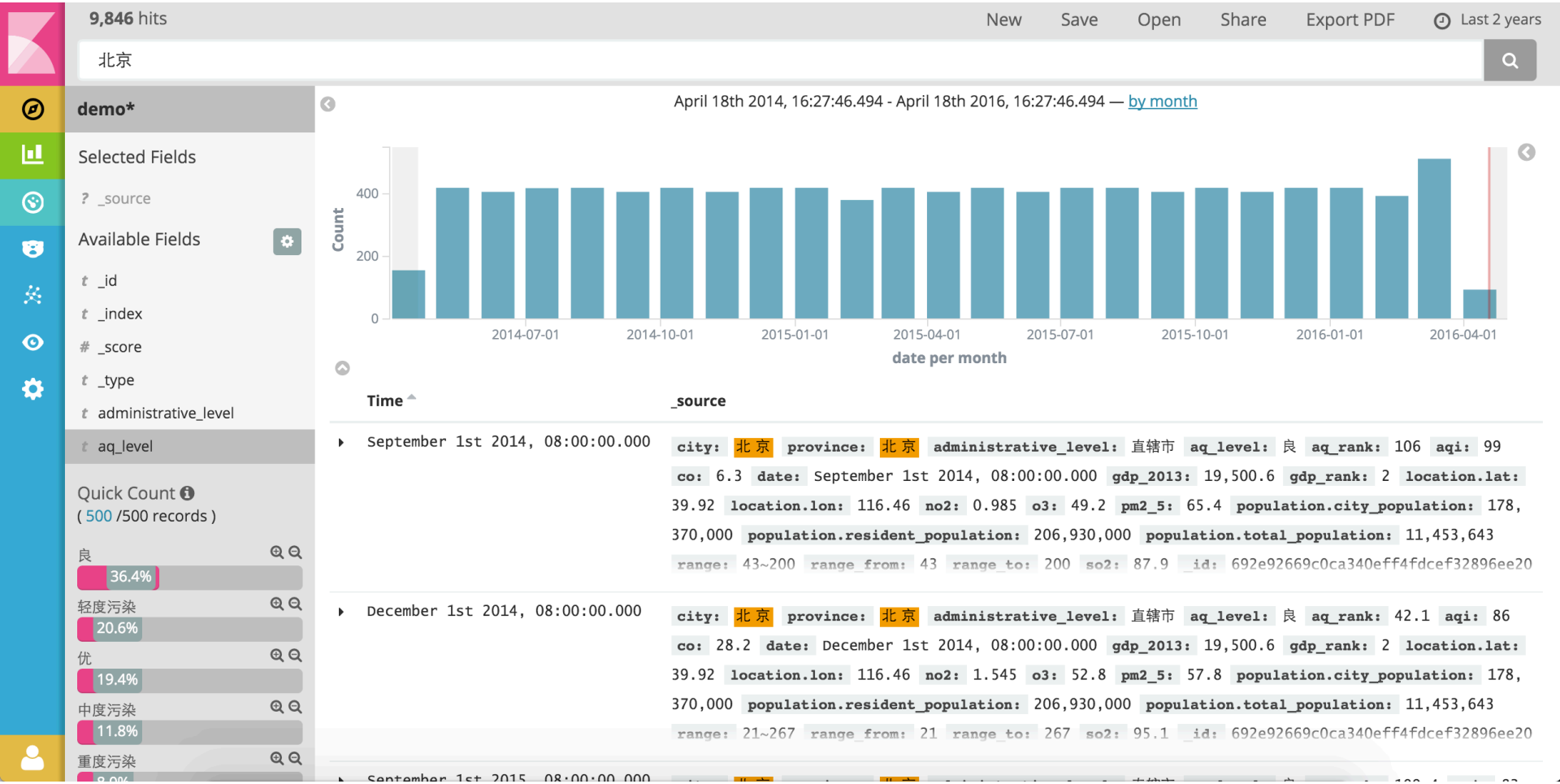


http://github.com/elastic/kibana

https://github.com/elastic/generator-kibana-plugin

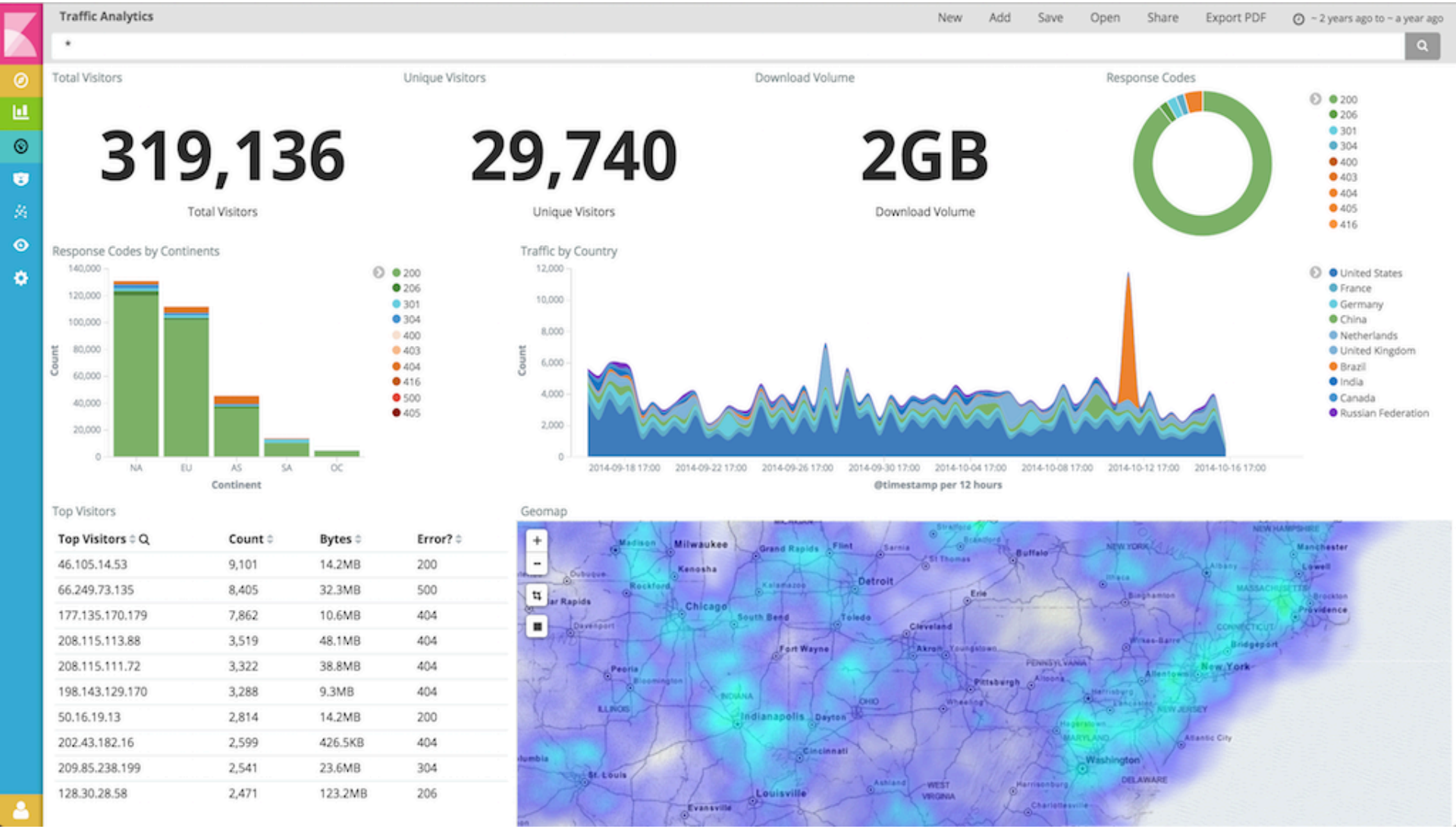
elastic

Search & Exploration



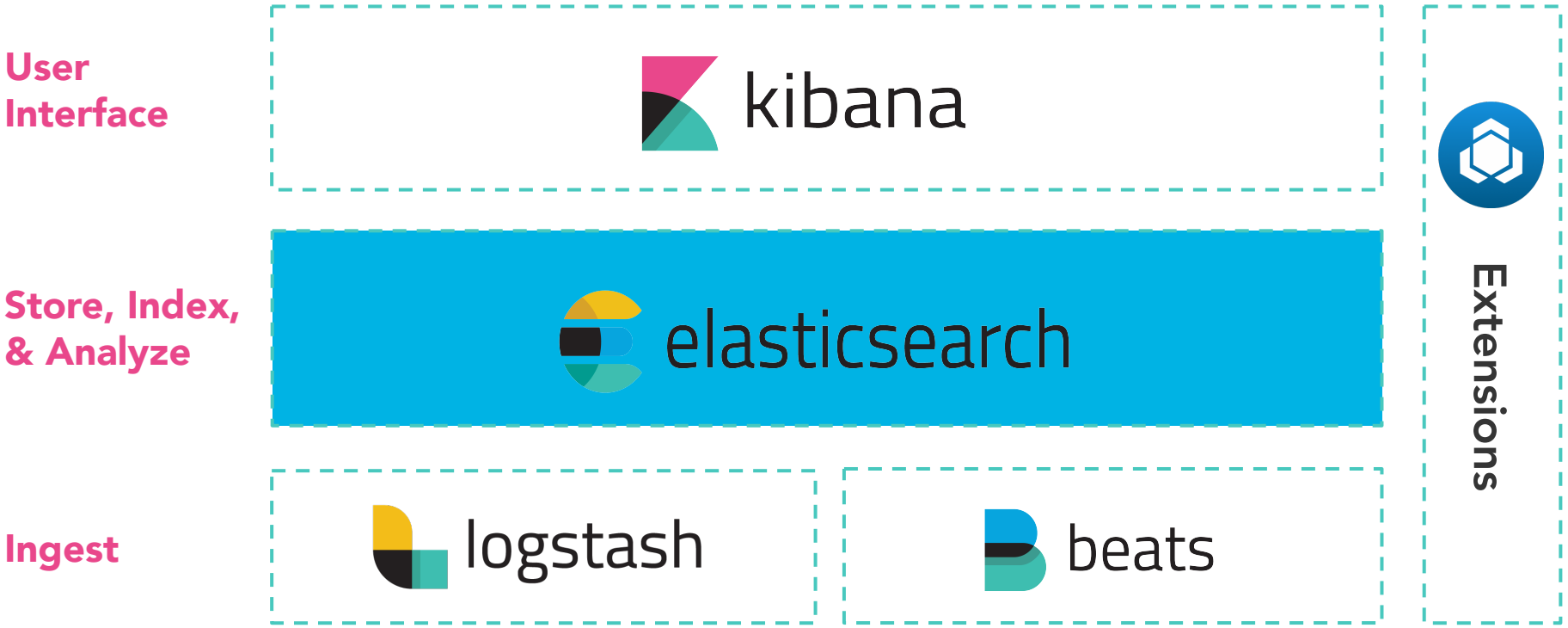
elastic

Visualization & Dashboard



elastic

The “Elastic Stack”



elastic

elasticsearch

Elasticsearch is an open source, distributed, scalable, highly available, document-oriented, RESTful, full text search engine with real-time search and analytics capabilities

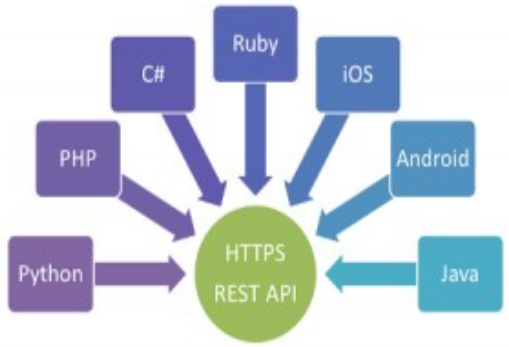
open source

Thomson Reuters: “107 clusters ~1747 nodes” @Elastic{ON}16

https://speakerdeck.com/elastic/thomson-reuters-research-journalism-finance-and-elastic

Netflix：”~150 clusters totaling ~3,500 nodes hosting ~1.3 PB of data”

http://techblog.netflix.com/2016/02/evolution-of-netflix-data-pipeline.html?m=1



JSON

- Real-time analytics
- Time series data analytics
- Logging analytics
- Security analytics
- Fraud detection
- Prediction modeling
- Recommendations
- …

http://github.com/elastic/elasticsearch

elastic


慢着，Elasticsearch 不是搜索引擎么？

You know for search, and analytics !

v0.09.0：~~Facets~~

v1.0.0：Aggregation

v2.0.0: Pipeline Aggregation

elastic

Aggregation

- Analytics

  柱状图、分布、统计、地理…

任何数据

能被查询到的数据就能被分析

接近实时

按需实时计算，~1s 刷新间隔

可嵌套组合

不像facets只有一级

elastic

Aggregation

Buckets：
Terms
Histogram
Geohash grids
…



Metrics：
min–avg-max
Stats
Cardinality
…



SELECT COUNT(*) , AVG(score) 〈--- Metrics
FROM `table`
GROUP BY province, city 〈--- Buckets

elastic

Aggregation == 3万英尺高空俯视 == Patterns



elastic



elastic

{ “city”:“北京” , “date”: “2016-02-08”,
“aq_level”: “严重污染”, “aq_rank”:68,
“aqi”: 391, “co”:115.5,
“no2”: 1.888, “o3”:62.2, “pm2_5”: 415.7,
“range”: “74~500”, “so2”: 523.5,
“location”: {“lat”:39.92, “lon”:116.46} }

数据来源：http://www.aqistudy.cn
elastic

Structure of an Aggregation

```
"aggregations" : {
    "<aggregation_name>" : {
        "<aggregation_type>" : {
            <aggregation_body>
        }
        [,"meta" : {  [<meta_data_body>] } ]?
        [,"aggregations" : { [<sub_aggregation>]+ } ]?
    }
    [,"<aggregation_name_2>" : { ... } ]*
}
```


elastic

空气质量统计（北京全年）

```
POST demo/_search?siz
{
   "query":{…}
   "aggs": {
       "aq_stats": {
           "terms": {
               "field":
               "size": 1
           }
       }
   }
}
```


```
{
  "aggregations": {
    "aq_stats": {
      "buckets": [
        {
          "key": "良",
          "doc_count": 118
        },
        {
          "key": "优",
          "doc_count": 102
        },
        {
          "key": "轻度污染",
          "doc_count": 64
        },
        {
          "key": "中度污染",
          "doc_count": 40
        },
        {
          "key": "重度污染",
          "doc_count": 21
        },
        {
          "key": "严重污染",
          "doc_count": 18
        }
      ]
    }
  }
}
```

平均空气质量统计（按城市）（

```
{
    "aggs": {
        "city_stats":
            "terms": {
                "field"
                "size":
            }
            ,"aggs": {
                "avg_pm2
                    "avg":
                        "fie
                    }
    }}}}
```


```
{
  "aggregations": {
    "city_stats": {
      "buckets": [
        {
          "key": "河源",
          "doc_count": 568,
          "avg_pm25": {
            "value": 38.366549295774654
          }
        },
        {
          "key": "佛山",
          "doc_count": 562,
          "avg_pm25": {
            "value": 42.66227758007118
          }
        },
        {
          "key": "汕头",
          "doc_count": 561,
          "avg_pm25": {
            "value": 37.265953654188955
          }
        },
        {
          "key": "泰安",
          "doc_count": 561,
          "avg_pm25": {
            "value": 76.28074866310159
          }
        }
```

30天空气质量趋势分析（Pip

```
{"aggs": {
   "qa_date_histo":{
       "date_histogram
            "field":"d
              "interval
        "aggs":{
            "the_avg":
              "avg":{
                   "fie
                } },
              "the_mov
                 "movi
                    "bu
                    "wi
                 }}}
}}}
```

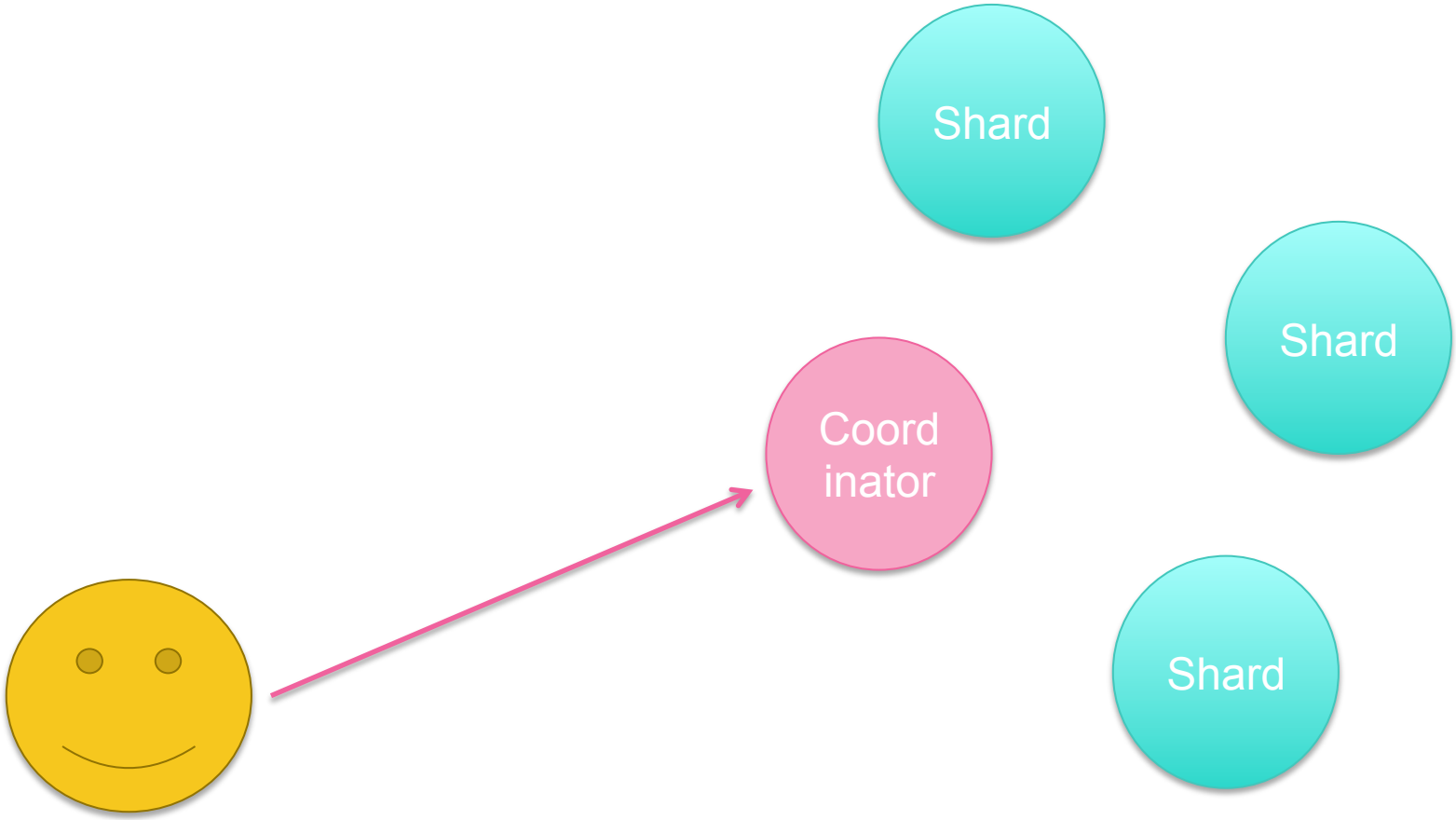

```
{
  "key_as_string": "2016-02-06T00:00:00.000Z",
  "key": 1454716800000,
  "doc_count": 189,
  "the_avg": {
    "value": 50.41798941798942
  },
  "the_movavg": {
    "value": 67.51189039993638
  }
},
{
  "key_as_string": "2016-02-07T00:00:00.000Z",
  "key": 1454803200000,
  "doc_count": 189,
  "the_avg": {
    "value": 65.5026455026455
  },
  "the_movavg": {
    "value": 66.97335424473356
  }
},
{
  "key_as_string": "2016-02-08T00:00:00.000Z",
  "key": 1454889600000,
  "doc_count": 189,
  "the_avg": {
    "value": 135.29470899470903
  },
  "the_movavg": {
    "value": 66.63231368035966
  }
```

Aggregation 工作原理

- Lucene Collector
- Optimized data structure
  - Compressed columnar datastore（previous FieldData，now DocValues）
  - Strings converted to enums（per segment）
- Single pass on your data，alone with the query
  - No matter how complex of your aggregation

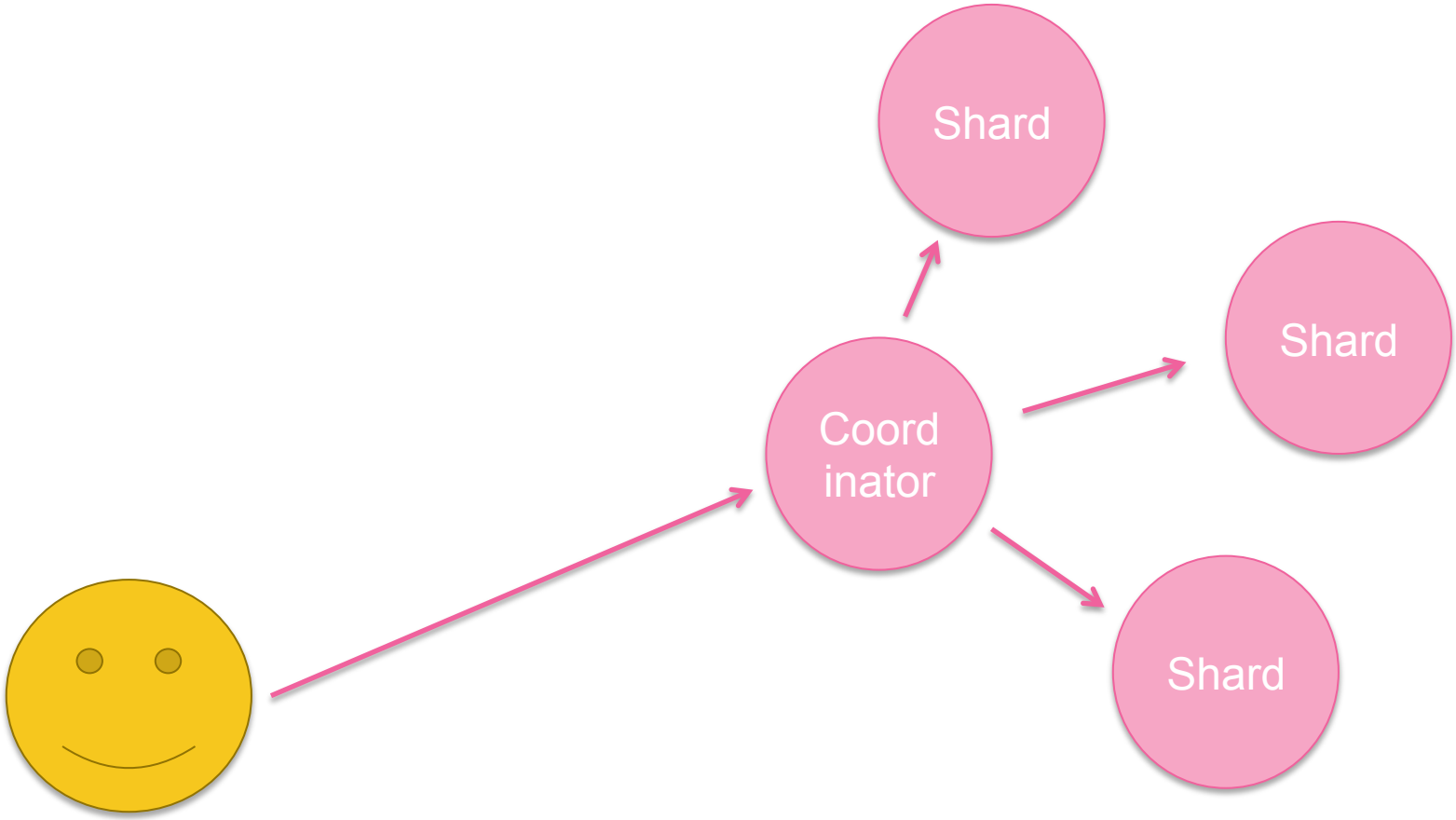
elastic

Aggregation 工作原理



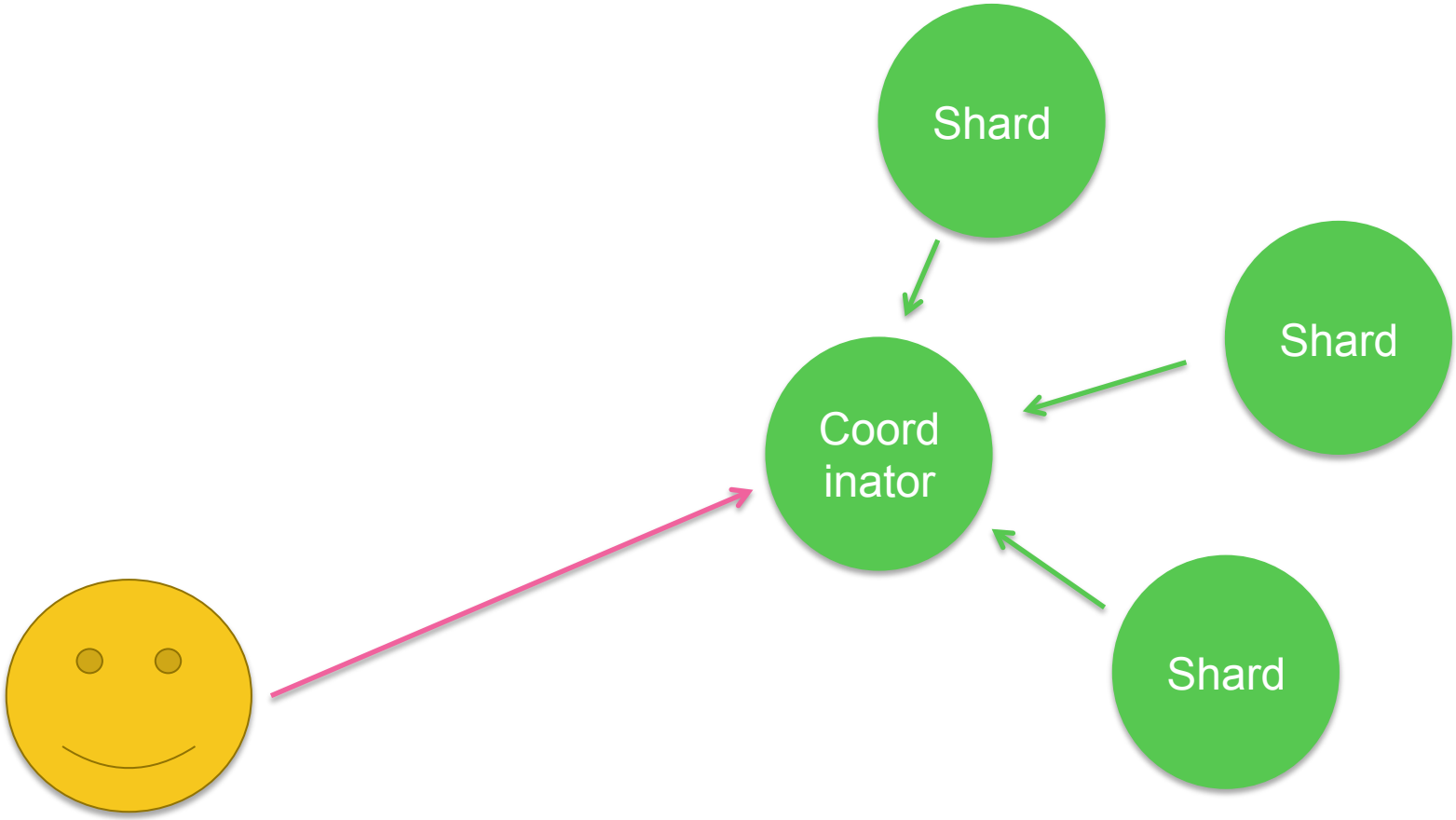
elastic

Aggregation 工作原理



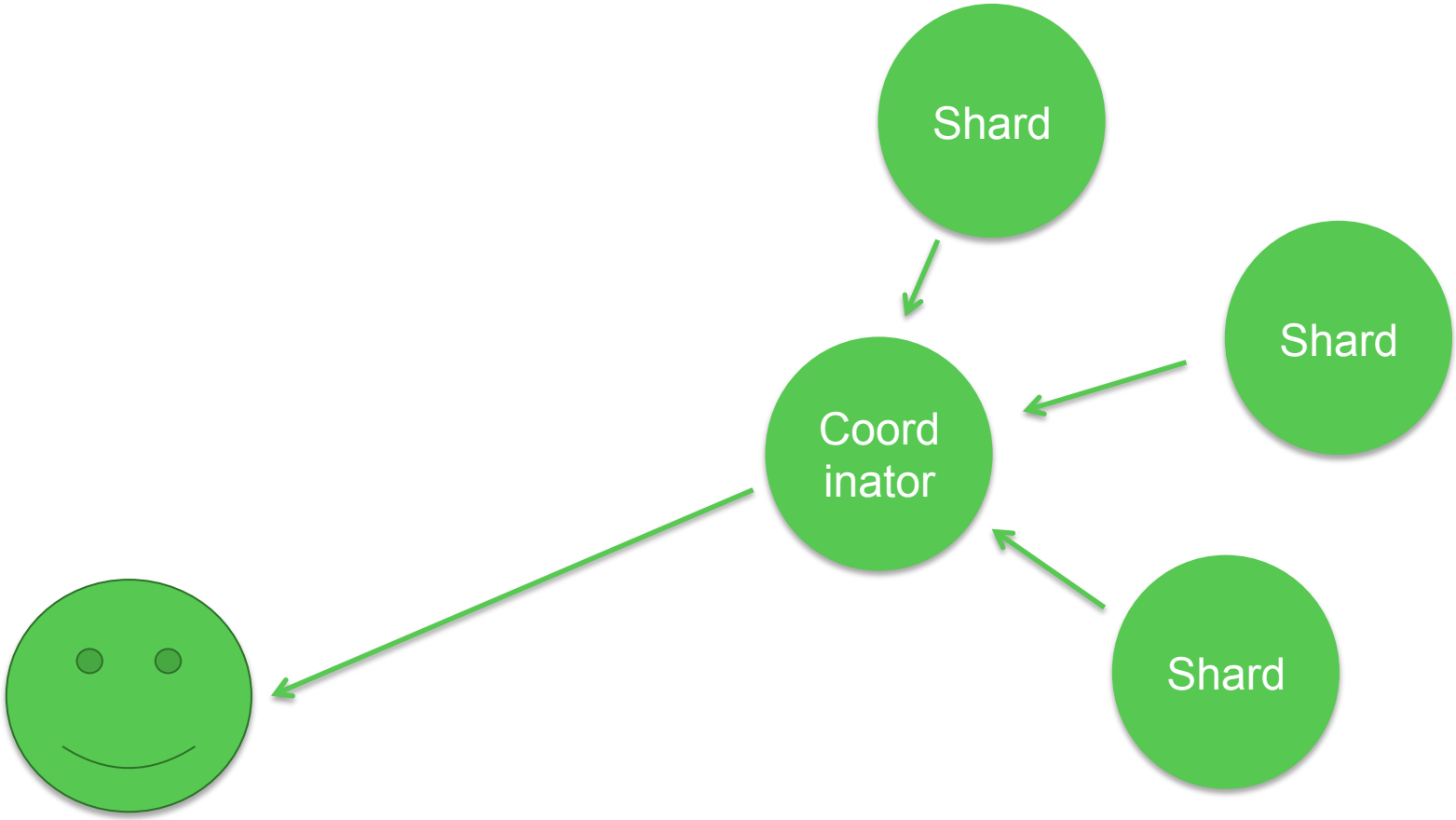
elastic

Aggregation 工作原理



elastic

Aggregation 工作原理

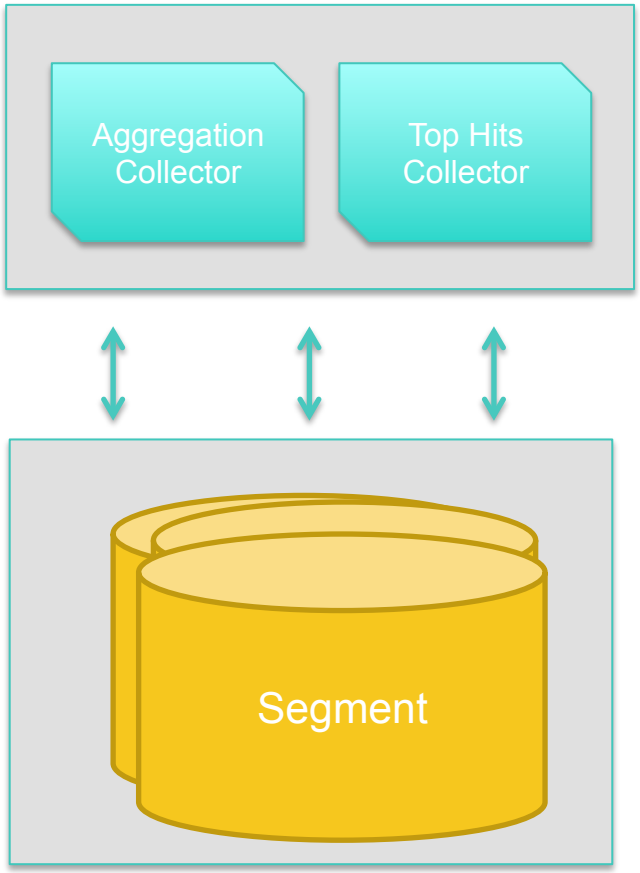


elastic

Aggregation 工作原理

Aggregation：
DocValues

DocID	Field
1	北京
2	上海
3	Beijing
4	上海



Lucene Index/ An ES Shard

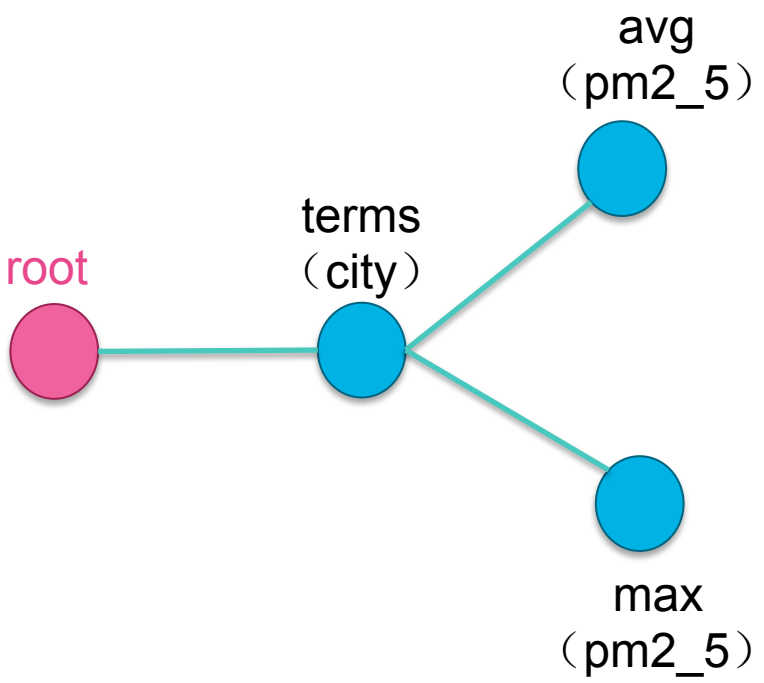
Search：
Invert Index

Term	DocID
北京	1，5，7
上海	2，4，6
广州	11，19，23
Beijing	3，12，13，15

elastic

Aggregation 工作原理

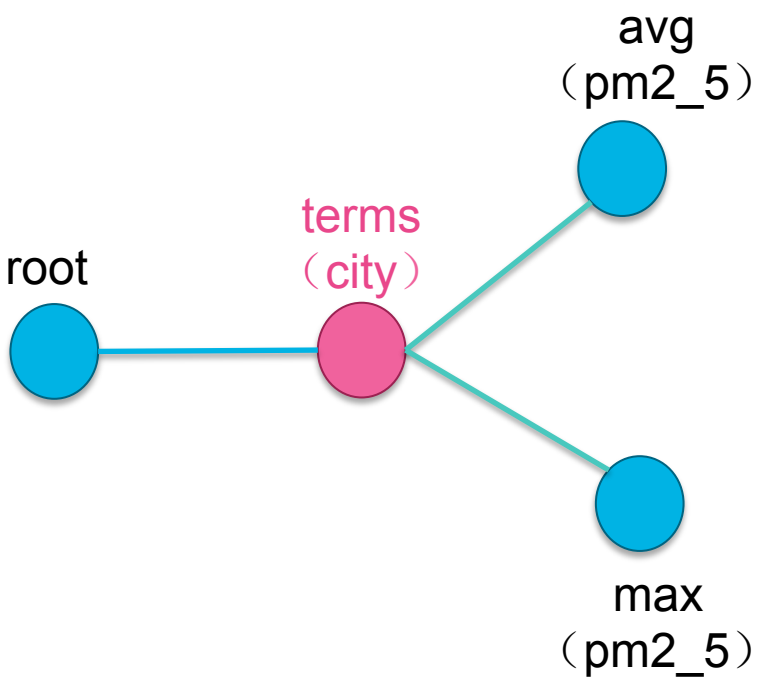
```
POST demo/_search?size=0
{
  "aggs": {
    "city_stats": {
      "terms": {
        "field": "city",
        "size": 10
      }
      ,"aggs": {
       "avg_pm25": {
        "avg": {
          "field": "pm2_5"
        }
       } ,
       "max_pm25": {
        "max": {
          "field": "pm2_5"
        }
       }
      } } }}
```



elastic

Aggregation 工作原理

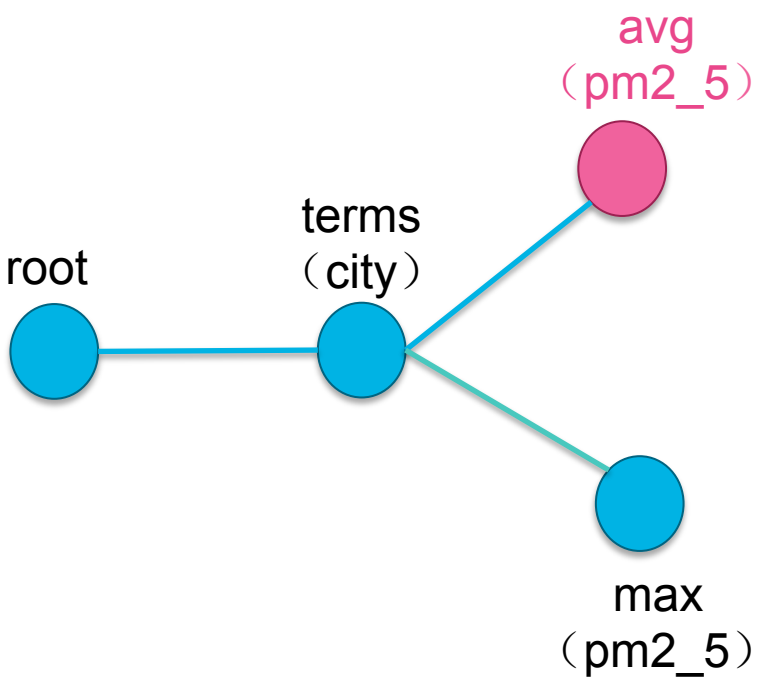
```
POST demo/_search?size=0
{
  "aggs": {
    "city_stats": {
      "terms": {
        "field": "city",
        "size": 10
      }
      ,"aggs": {
       "avg_pm25": {
        "avg": {
          "field": "pm2_5"
        }
       } ,
       "max_pm25": {
        "max": {
          "field": "pm2_5"
        }
       }
      } } }}
```



elastic

Aggregation 工作原理

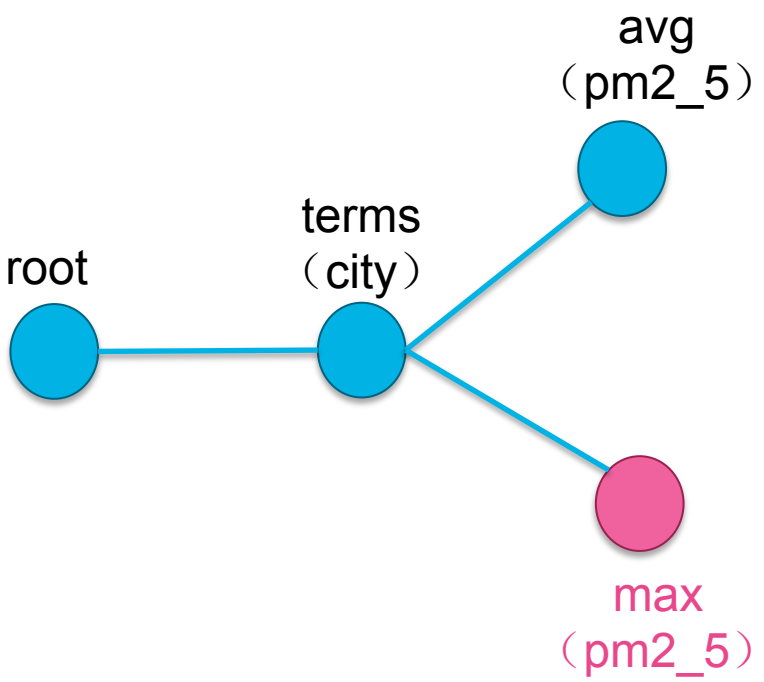
```
POST demo/_search?size=0
{
  "aggs": {
    "city_stats": {
      "terms": {
        "field": "city",
        "size": 10
      }
      ,"aggs": {
       "avg_pm25": {
        "avg": {
         "field": "pm2_5"
        }
       } ,
       "max_pm25": {
        "max": {
         "field": "pm2_5"
        }
       }
      } } }}
```



elastic

Aggregation 工作原理

```
POST demo/_search?size=0
{
  "aggs": {
    "city_stats": {
      "terms": {
        "field": "city",
        "size": 10
      }
      ,"aggs": {
       "avg_pm25": {
        "avg": {
          "field": "pm2_5"
        }
       } ,
       "max_pm25": {
        "max": {
          "field": "pm2_5"
        }
       }
      } } }}
```

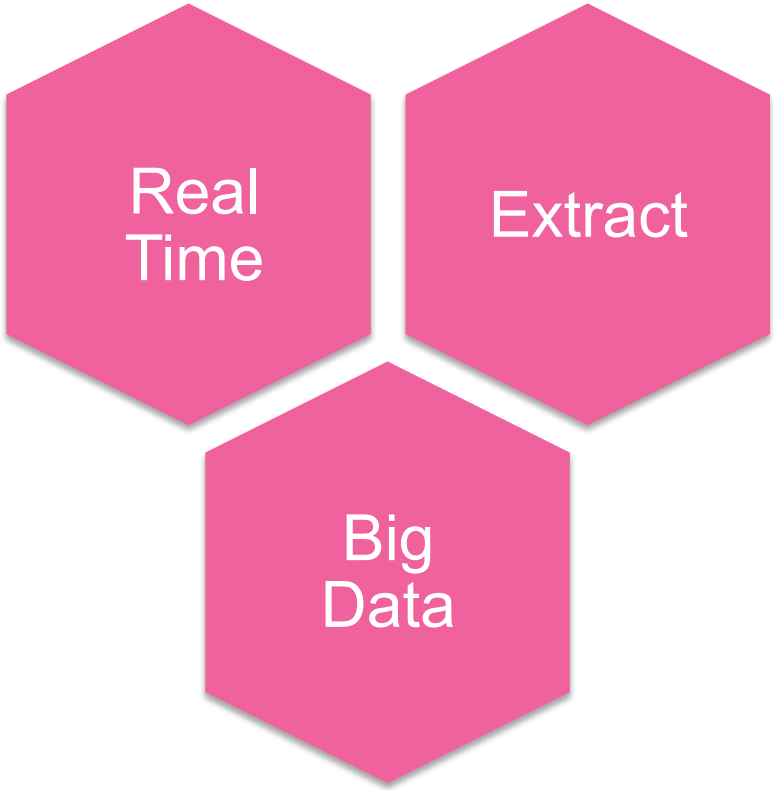


elastic

What's more?

- 近似算法（Approximate algorithms）
  - 唯一值：Cardinality
    - Hyperloglog++
  - 百分位：Percentile
    - TDigest

Fixed memory！



- 控制效率和内存占用
  - Terms
    - Breath_first collect mode
  - Sampler
    - Max docs per shard

- 更多有趣的Aggregation！
  - Significant Terms Aggregation
    - The uncommonly common
  - Geohash grid

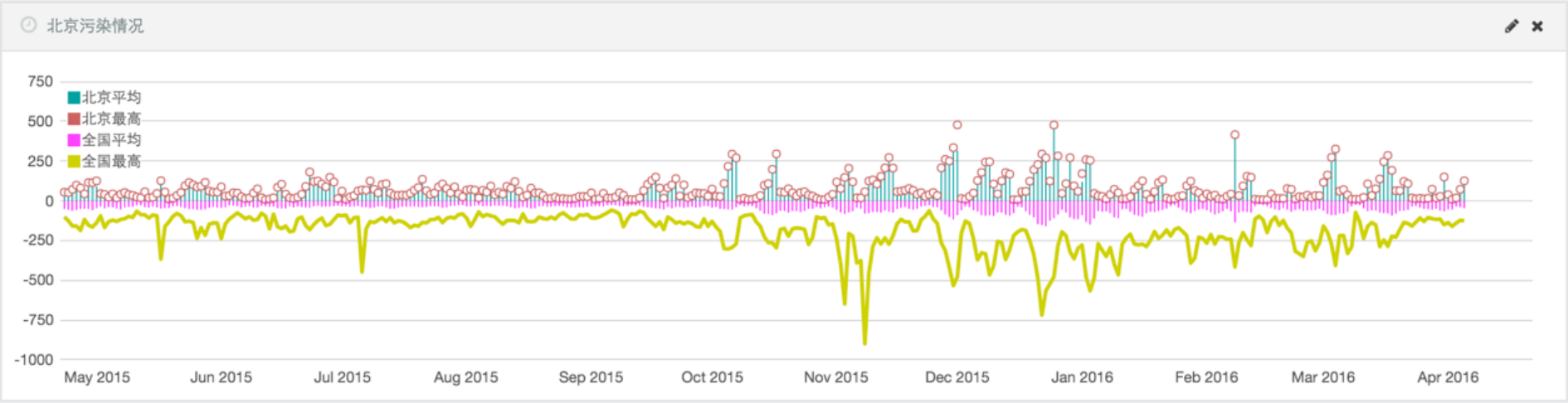
elastic


说好的图形界面呢？

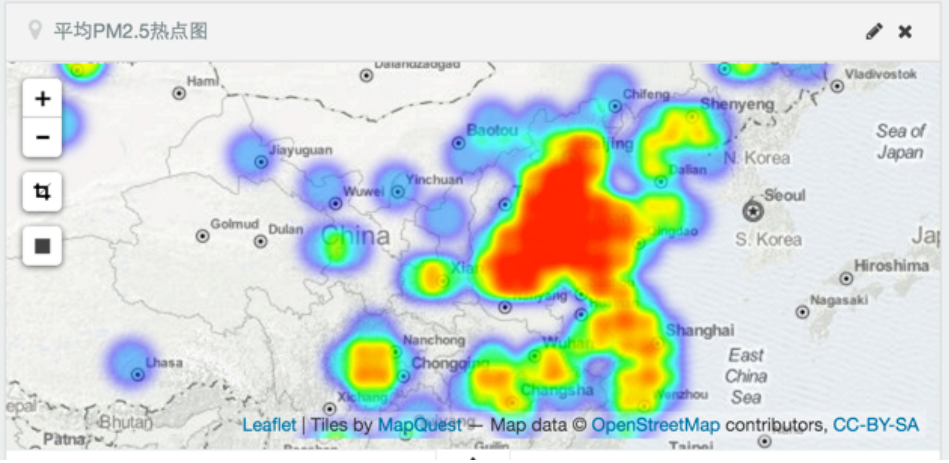
DEMO

Kibana、Timelion、Graph

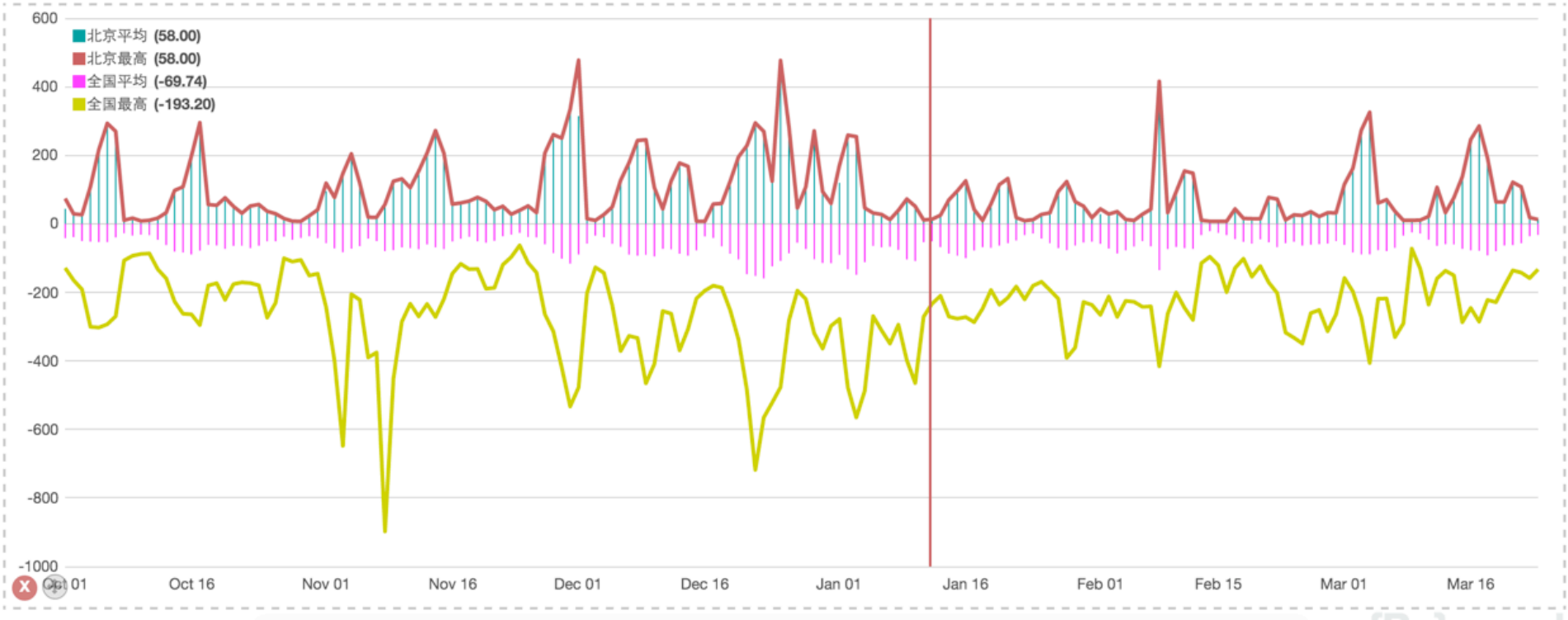
elastic


北京污染情况
750
500
250
0
-250
-500
-750
-1000
北京平均
北京最高
全国平均
全国最高
May 2015
Jun 2015
Jul 2015
Aug 2015
Sep 2015
Oct 2015
Nov 2015
Dec 2015
Jan 2016
Feb 2016
Mar 2016
Apr 2016


污染累计排行云图
大同
郑州
廊坊
诸暨
中山
阳泉
宿迁
保定
菏泽
三亚
肇庆
九江
焦作
聊城
衡水
邯郸
江门
平顶山
长治
芜湖
常熟
无锡
汕头
长春
荣成
湛江
海口

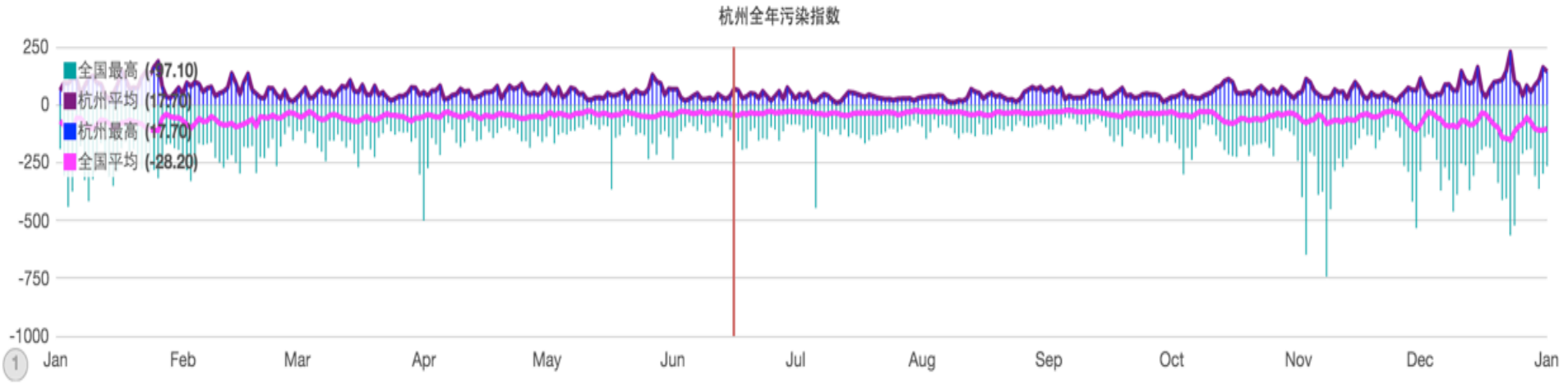

平均PM2.5热点图
Haml
Chifeng
Shenyeng
Vladivostok
Sea of Japan
Baotou
Jiayuguan
N. Korea
Yinchuan
Wuwei
Seoul
Golmud
Dulan
China
S. Korea
Hiroshima
Nagasaki
Shanghai
East China Sea
Nanchong
Chongqing
Lhasa
Changsha
Bhutan
Patna
Leaflet | Tiles by MapQuest — Map data © OpenStreetMap contributors, CC-BY-SA

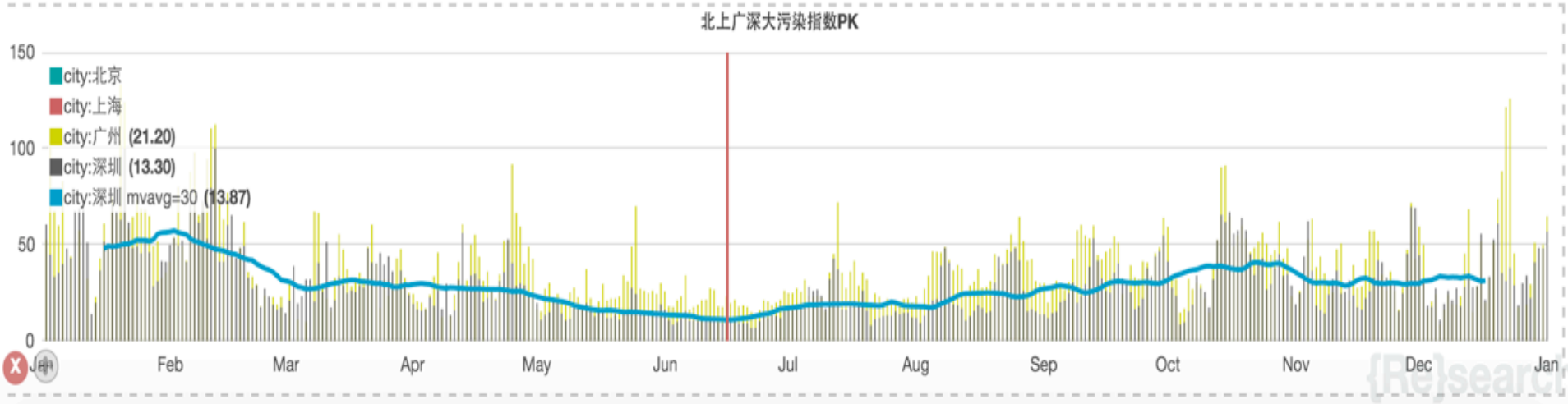
elastic


北京平均 (58.00)
北京最高 (58.00)
全国平均 (-69.74)
全国最高 (-193.20)
600
400
200
0
-200
-400
-600
-800
-1000
Oct 01
Oct 16
Nov 01
Nov 16
Dec 01
Dec 16
Jan 01
Jan 16
Feb 01
Feb 15
Mar 01
Mar 16

elastic

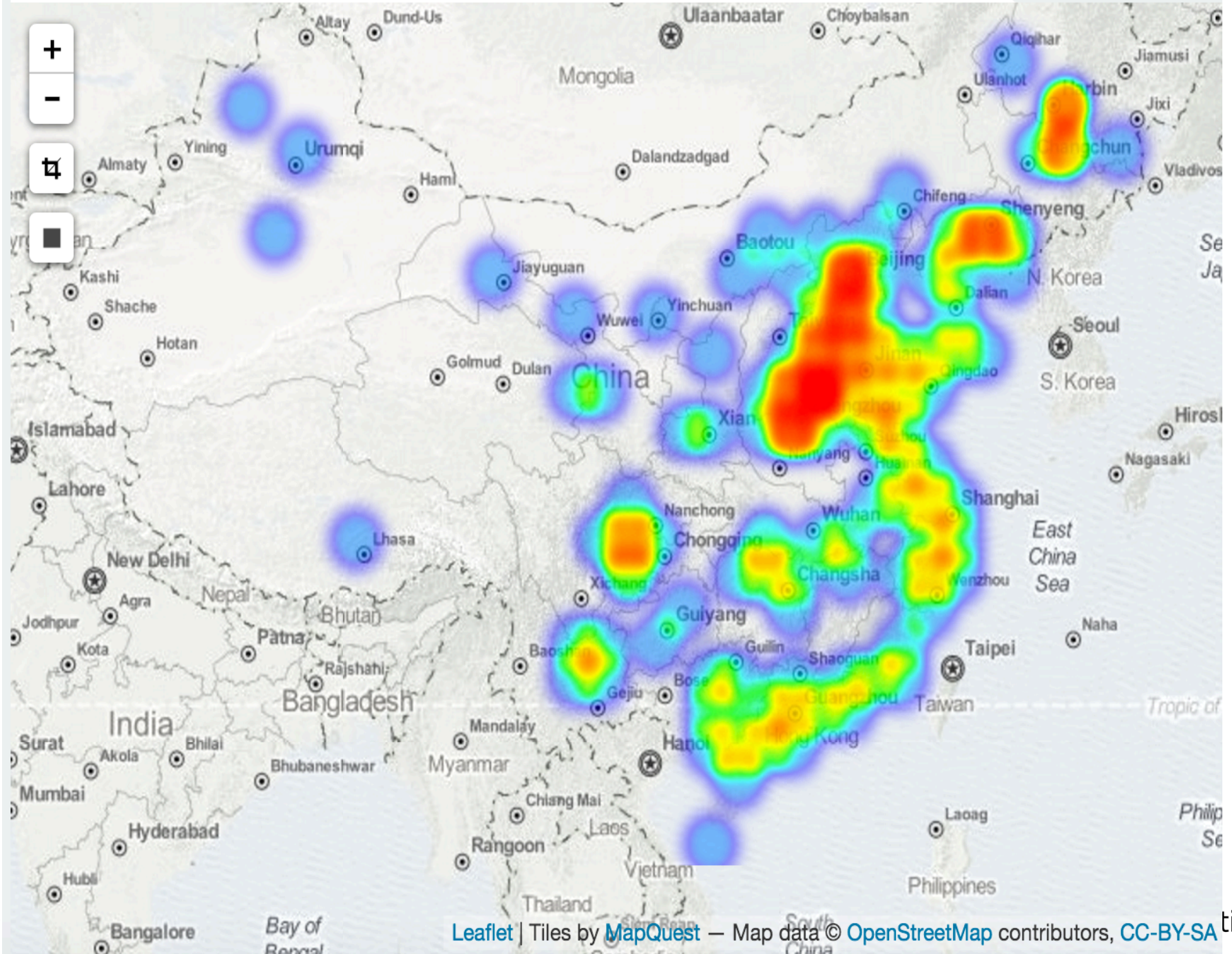
DEMO



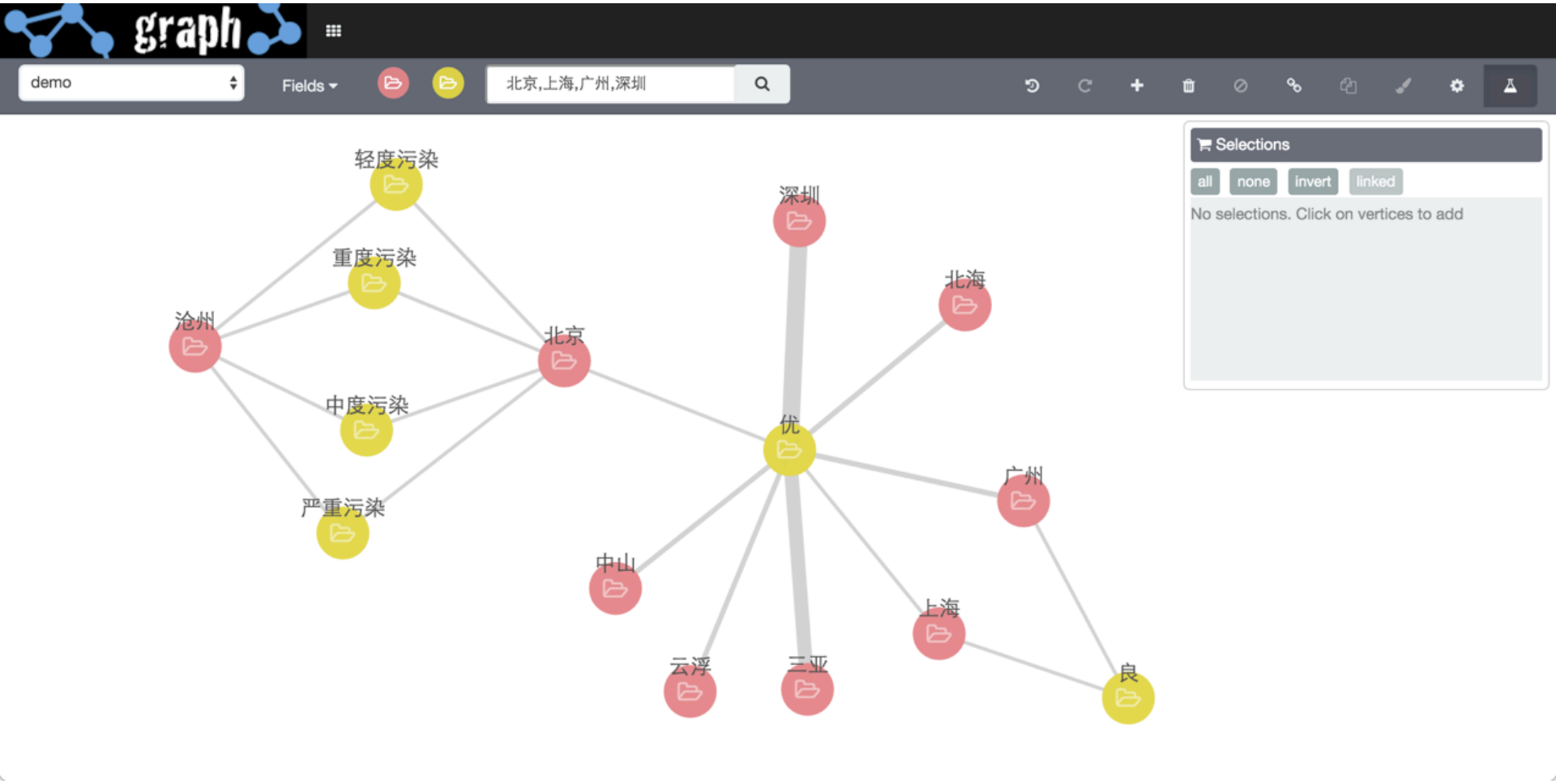


elastic

平均PM2.5热点图

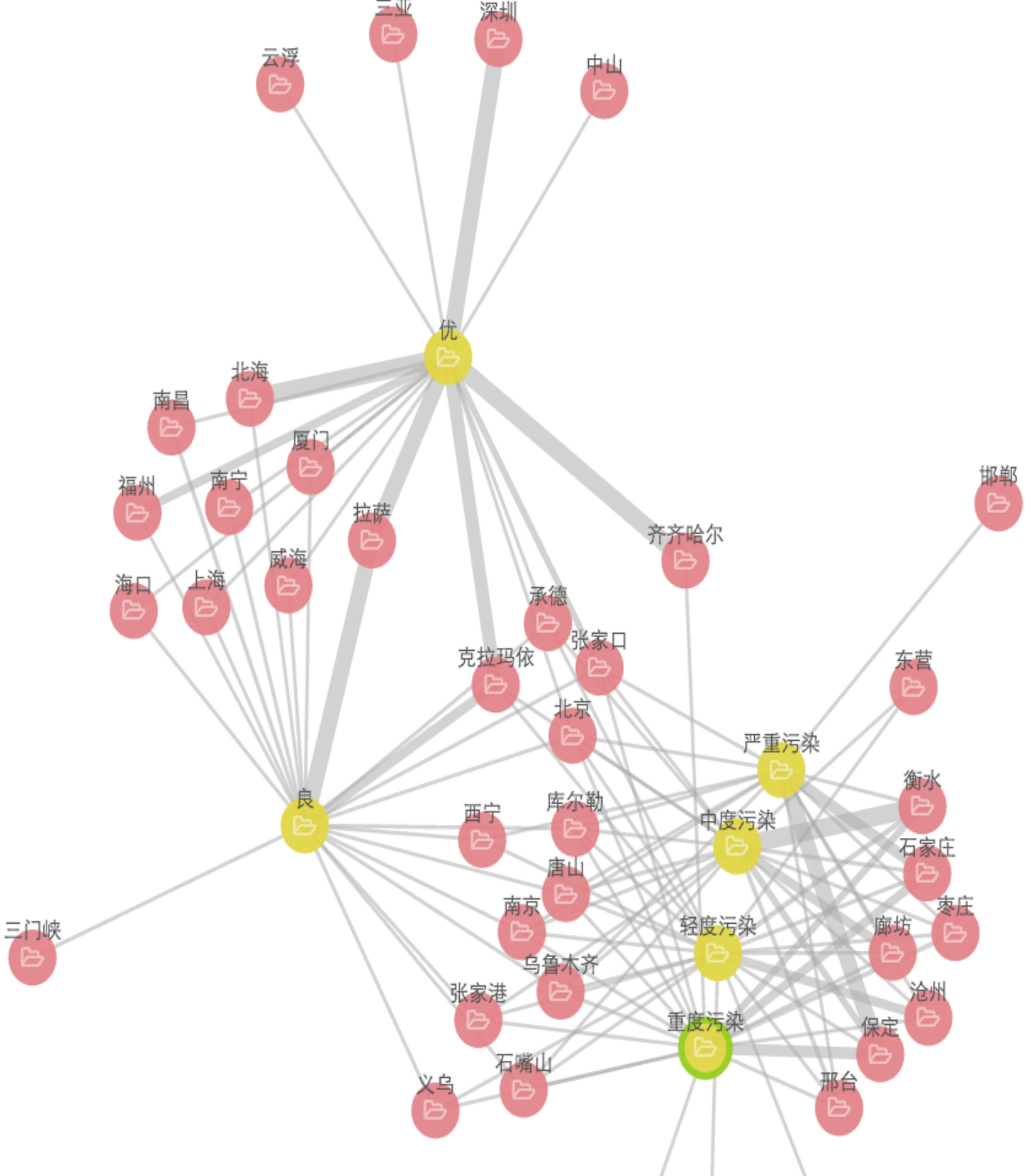


Leaflet | Tiles by MapQuest — Map data © OpenStreetMap contributors, CC-BY-SA


graph
demo
Fields
北京,上海,广州,深圳
Selections
all
none
invert
linked
No selections. Click on vertices to add
轻度污染
重度污染
沧州
北京
中度污染
严重污染
深圳
北海
优
广州
中山
上海
云浮
三亚
良

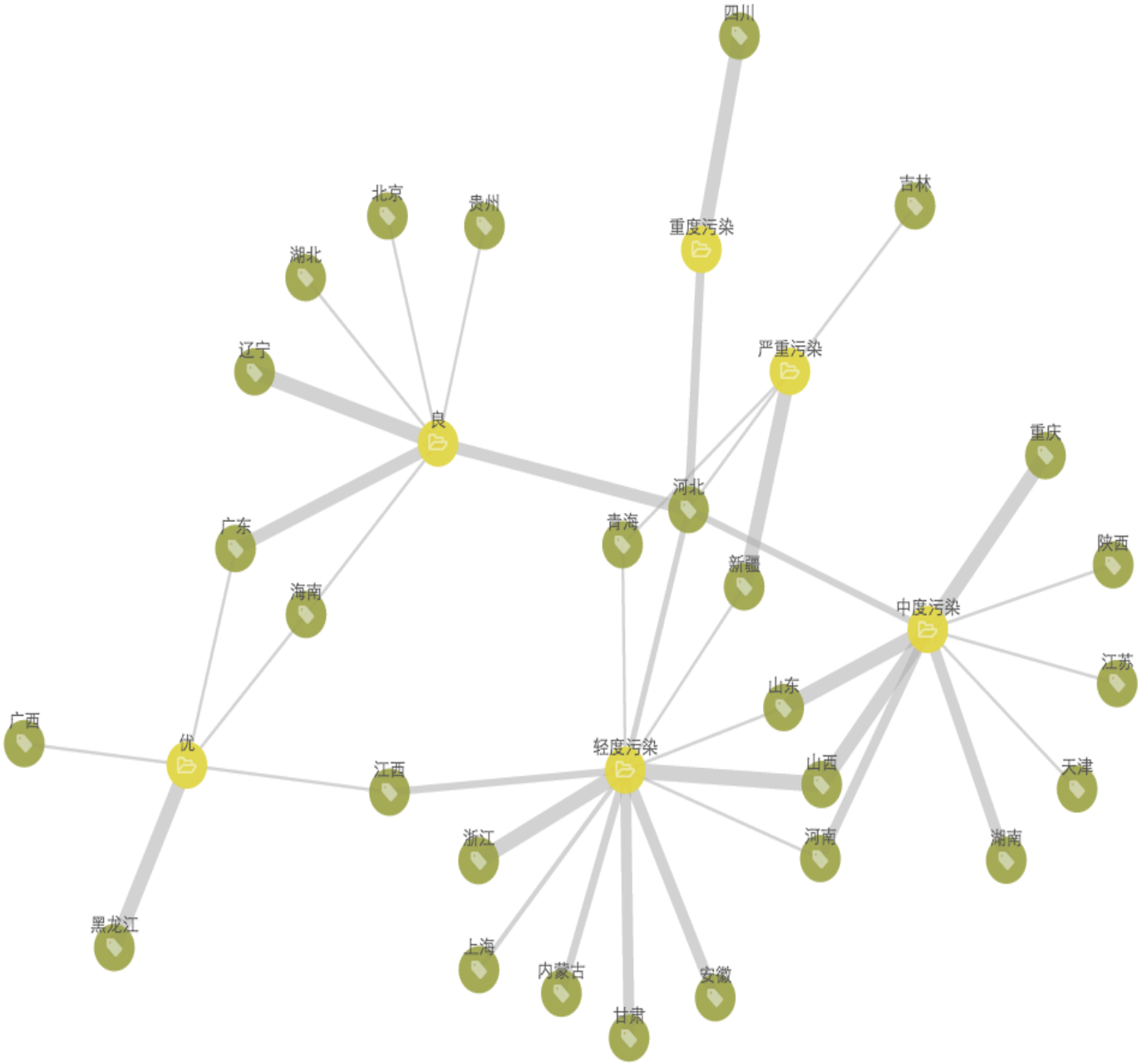
elastic

Dem


三亚
深圳
云浮
中山
优
北海
南昌
厦门
福州
南宁
拉萨
邯郸
齐齐哈尔
威海
海口
上海
承德
张家口
克拉玛依
东营
北京
严重污染
衡水
良
西宁
库尔勒
中度污染
石家庄
唐山
南京
枣庄
三门峡
轻度污染
廊坊
乌鲁木齐
沧州
张家港
重度污染
保定
石嘴山
义乌
邢台

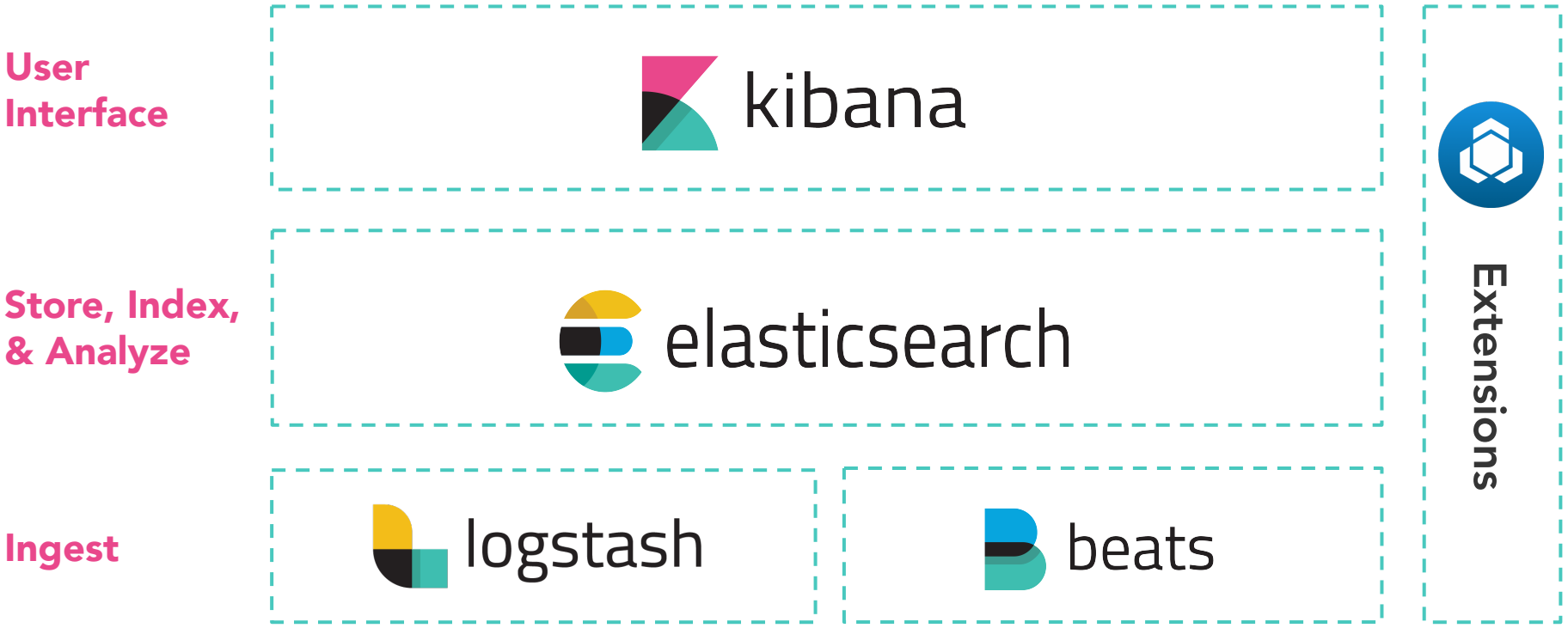
tic

Dem


四川
北京
贵州
吉林
重度污染
湖北
辽宁
严重污染
良
重庆
河北
广东
青海
陕西
新疆
海南
中度污染
江苏
山东
广西
优
轻度污染
江西
山西
天津
浙江
河南
湖南
黑龙江
上海
内蒙古
安徽
甘肃

elastic

The "Elastic Stack"



elastic

Community

- 源码 & Issue: http://github.com/elastic/
- 英文社区: http://discuss.elastic.co
- 中文社区: http://elasticsearch.cn
- 官方 QQ 群: 190605846
- 下载: https://www.elastic.co/downloads
- 博客: https://www.elastic.co/blog
- 线下活动: http://elasticsearch.meetup.com/
- IRC: #elasticsearch, #logstash, #kibana, #beats
- 官方 Twitter @elastic



elastic



More questions?

欢迎来 Elastic 展台找我!

Thanks!

elastic

附录

elastic

ES 基本操作

索引（插入数据）

```
POST demo/pm2_5/1
{
"city":"北京" , "date": "201
"aq_level": "严重污染", "aq_
"no2": 1.888, "o3":62.2, "p
"so2": 523.5, "location": {
}
```

```
{
  "_index": "demo",
  "_type": "pm2_5",
  "_id": "1",
  "_version": 1,
  "_shards": {
    "total": 2,
    "successful": 1,
    "failed": 0
  },
  "created": true
}
```


elastic

基本操作

获取数据

```
GET demo/pm2_5/1
```


```
{
  "_index": "demo",
  "_type": "pm2_5",
  "_id": "1",
  "_version": 1,
  "found": true,
  "_source": {
   "aq_level": "严重污染“,   "aq_rank": 68,
"aqi": 391,   "city": "北京“,   "co": 115.5,
"date": "2016-02-08“,   "province": "北京",
    "range": "74~500“, "so2": 523.5
  }
}
```


elastic

基本操作

删除数据

```
DELETE demo/pm2_5/1
```


elastic

搜索

通过GET参数进行搜索

```
GET demo/pm2_5/_search?q=北京
```


elastic

搜索

查询北京的 PM2.5 数据

```
POST demo/pm2_5/_search
{
  "query": {
    "match": {
      “city”: “北京"
    }
  }
}
```


elastic

搜索

查询北京的 PM2.5 数据

```
POST demo/pm2_5/_search
{"query": {"bool": {
        "must": [
          {"term": {"city": {"value": "北京"}}},
          {"term": {"date": {"value": "2016-02-08"}}}]}}
}
```


elastic

分析

Aggregation

```
POST twitter/tweet/_search
{
  "aggs" : {
    "uers_stats" : {
      "terms" : { "field" : "user" }
    }
  }
}
```

