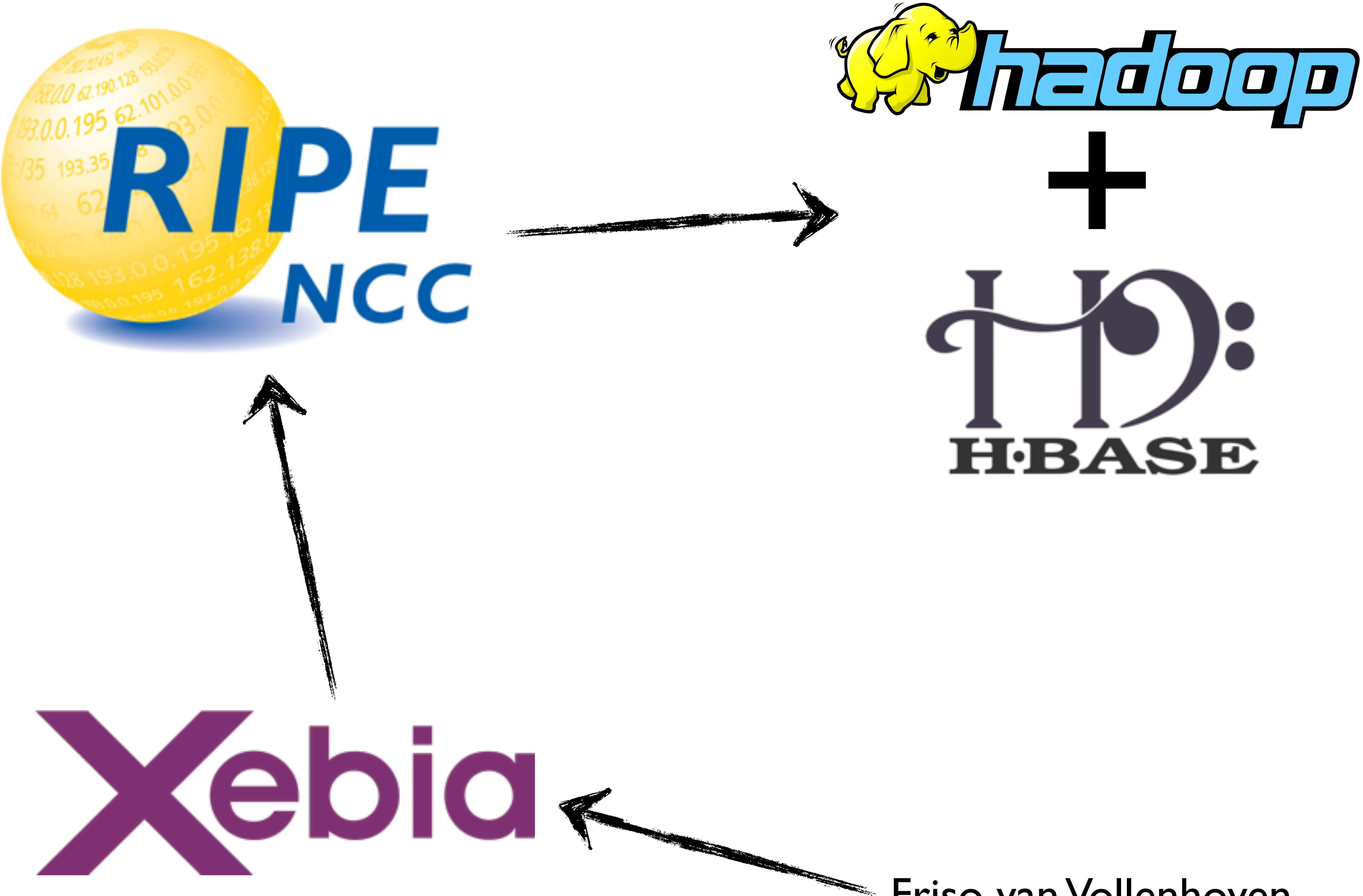
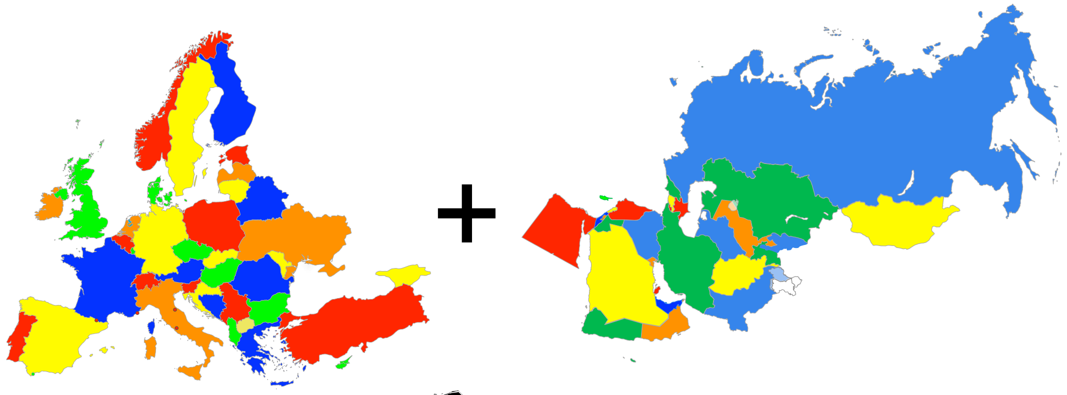
Analyzing the internet using Hadoop and Hbase

Friso van Vollenhoven
fvanvollenhoven@xebia.com



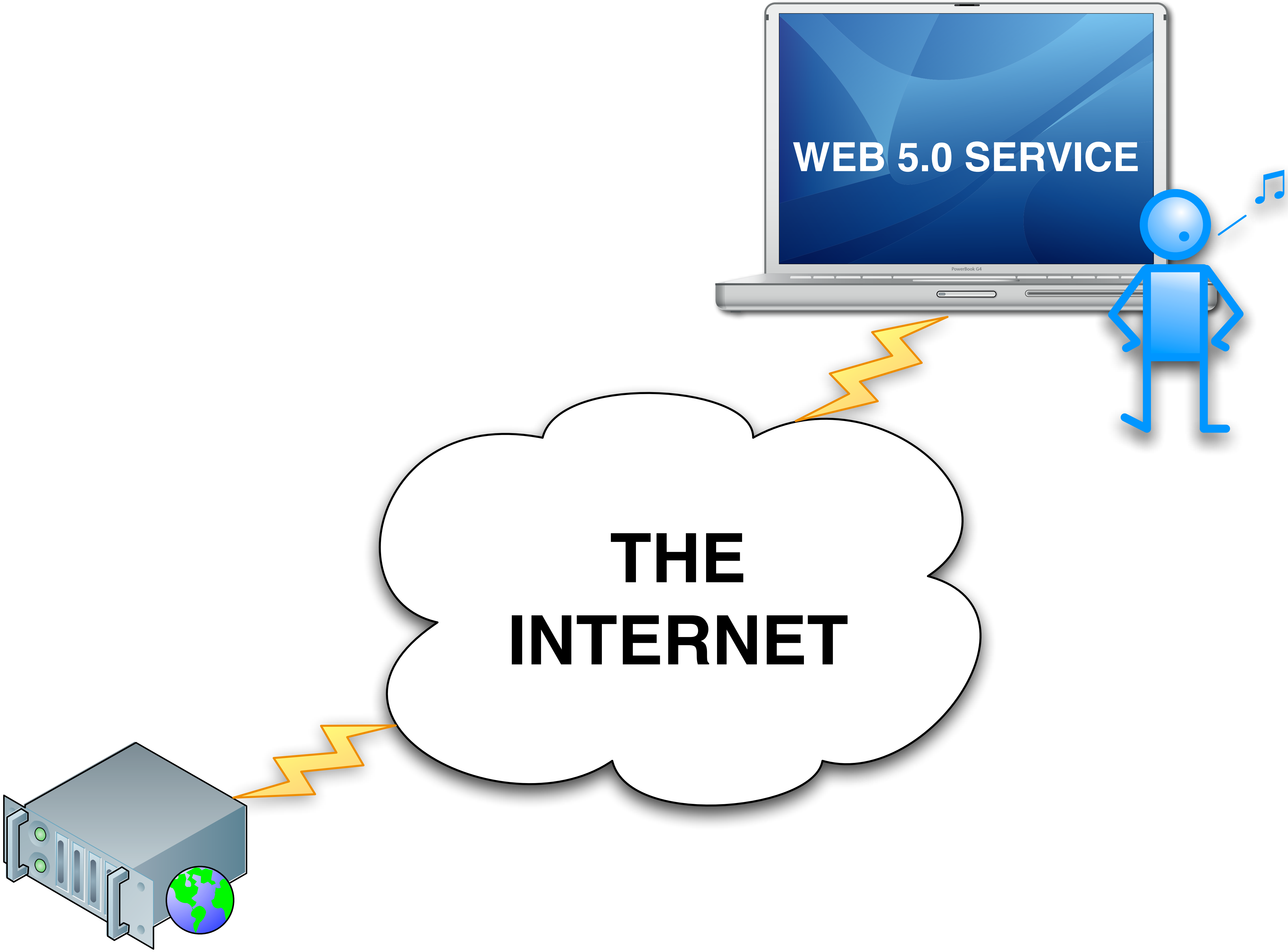
Friso van Vollenhoven
fvanvollenhoven@xebia.com



RIPE NCC = One of five Regional Internet Registrars

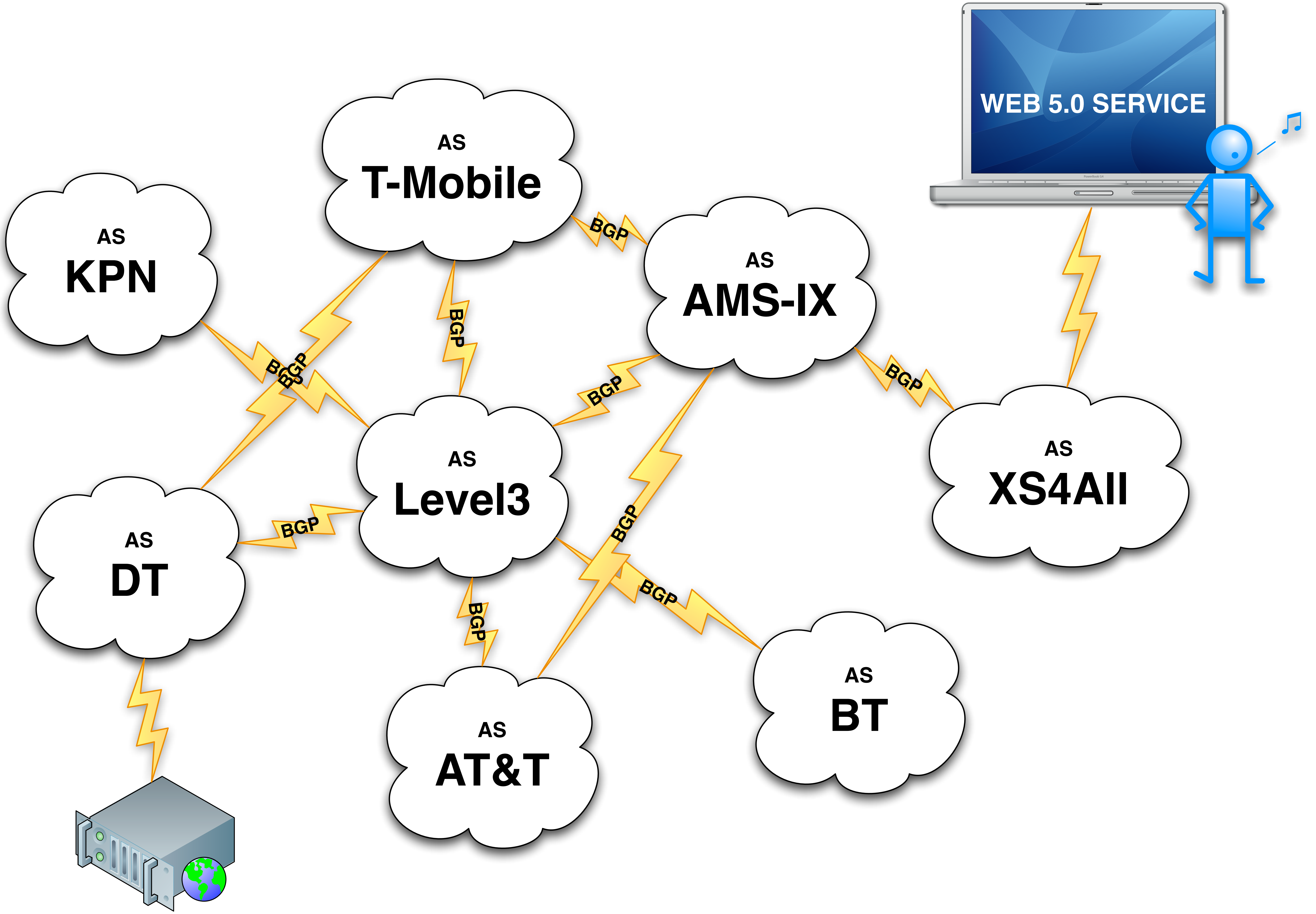
Allocate IP address ranges to organizations (e.g. ISPs, network operators, etc.)

Provide near real time network information based on measurements



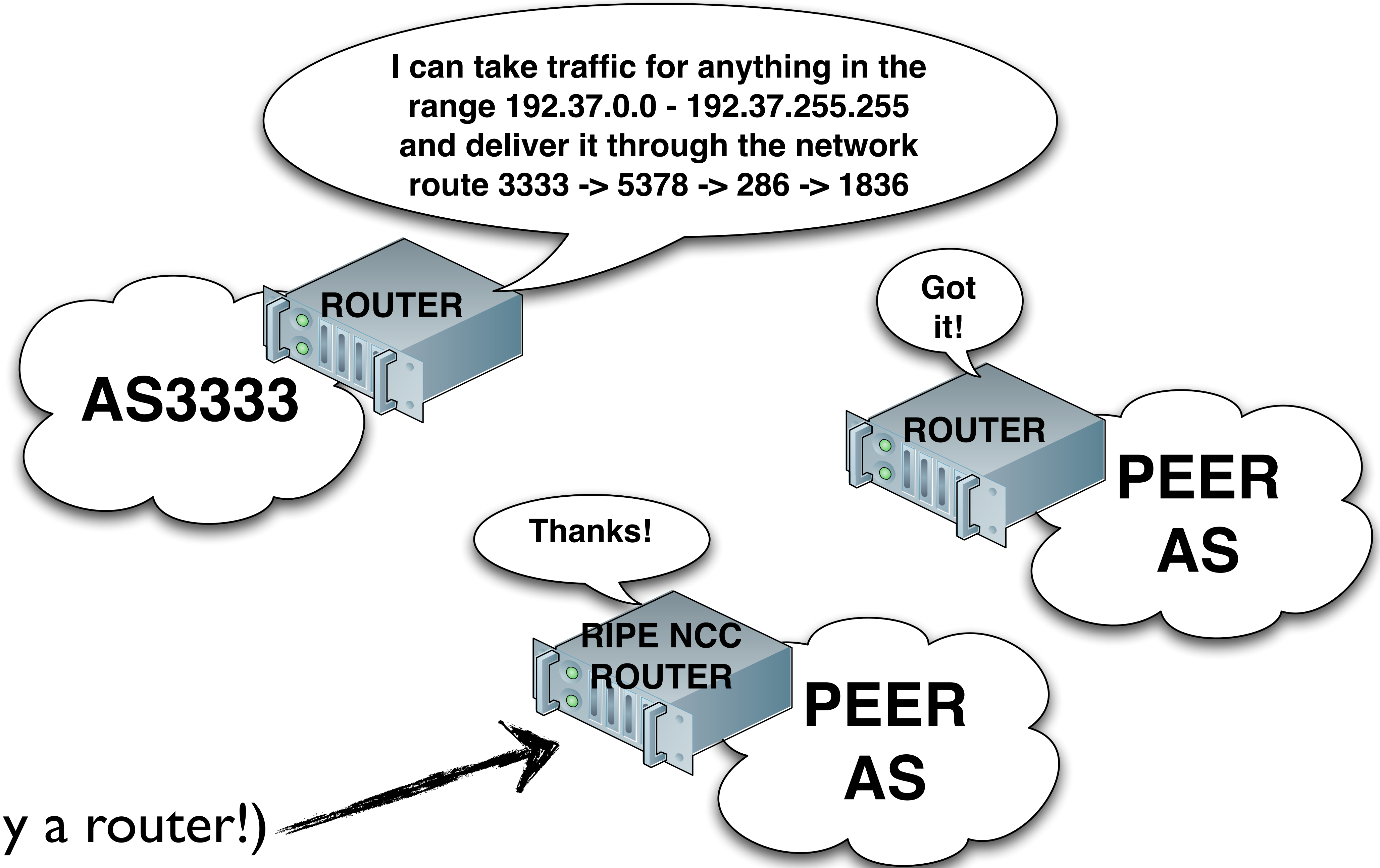
WEB 5.0 SERVICE

THE INTERNET


WEB 5.0 SERVICE
AS
T-Mobile
AS
KPN
AS
AMS-IX
AS
Level3
AS
XS4All
AS
DT
AS
AT&T
AS
BT
BGP

Our Data

```
BGP4MP|980099497|A|193.148.15.68|3333|192.37.0.0/16|3333 5378 286 1836|IGP|193.148.15.140|0|0||NAG||
```


I can take traffic for anything in the range 192.37.0.0 - 192.37.255.255 and deliver it through the network route 3333 -> 5378 -> 286 -> 1836
ROUTER
AS3333
Got it!
ROUTER
PEER AS
Thanks!
RIPE NCC ROUTER
PEER AS

(not really a router!)

Our Data

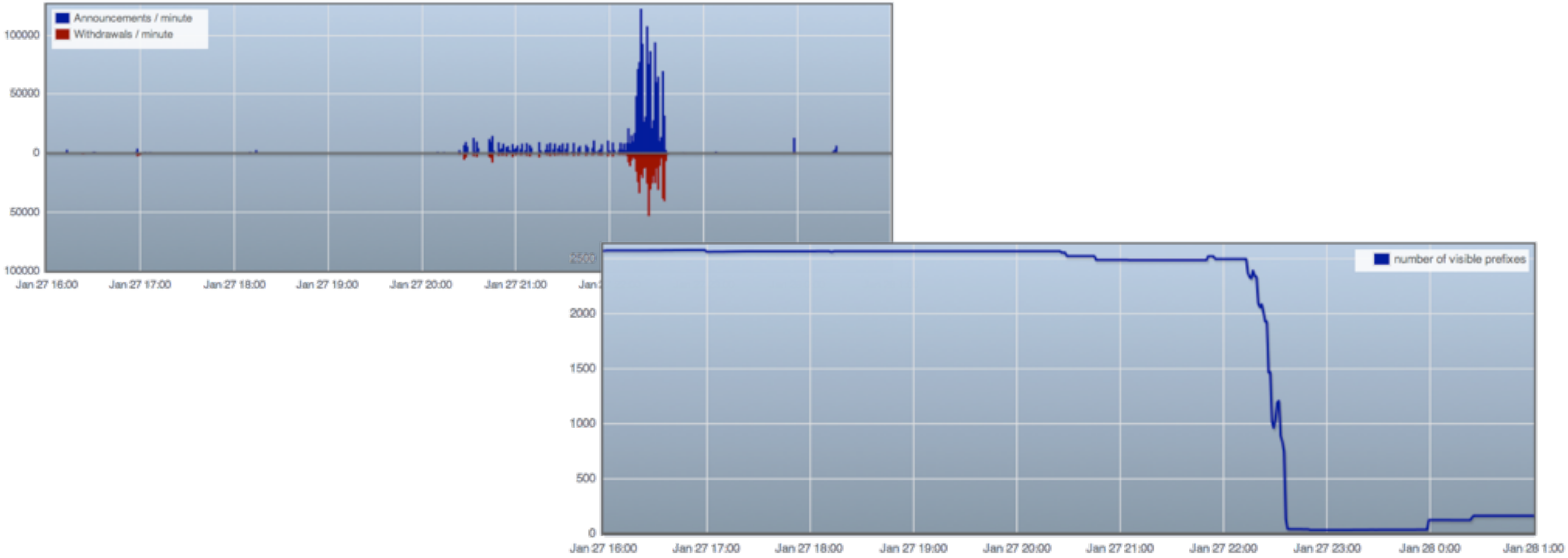
- 15 active data collection points (the not really routers), collecting updates from ~600 peers
- BGP updates every 5 minutes, ~80M / day
- Routing table dumps every 8 hours, ~9M entries / dump

FAQ

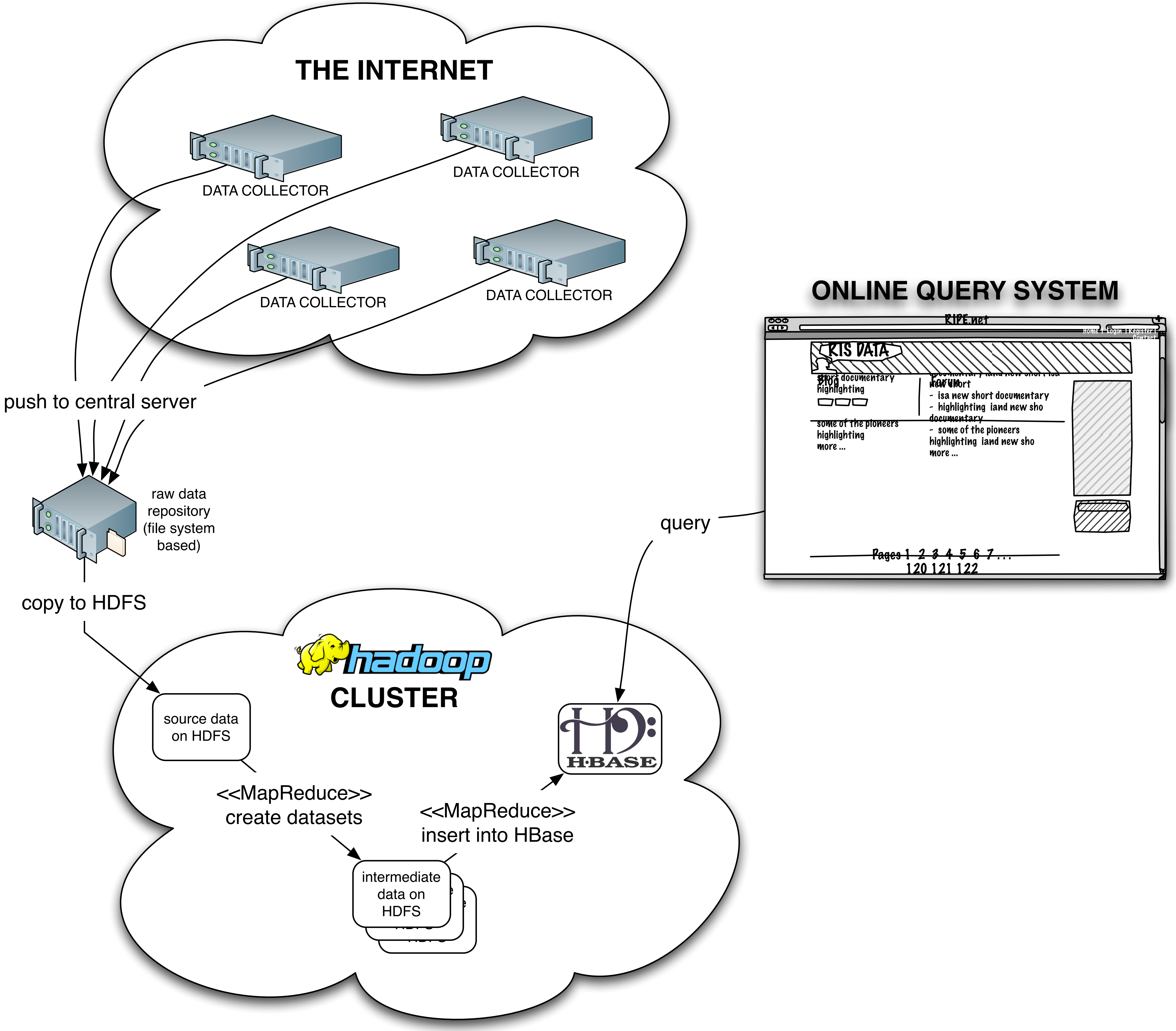
- What are all updates that are somehow related to AS3333 for the past two weeks?
- How many updates per second were announcing prefixes originated by AS286 between 11:00 and 12:00 on February 2nd in 2005?
- Which AS adjacencies have existed for AS3356 in the past year?
- What was the size of the routing table of the peer router at 193.148.15.140 two years ago?

Not so frequently asked:

What happened in Egypt?



http://stat.ripe.net/egypt



THE INTERNET

DATA COLLECTOR

DATA COLLECTOR

DATA COLLECTOR

DATA COLLECTOR

push to central server

raw data repository (file system based)

copy to HDFS

hadoop CLUSTER

source data on HDFS

<<MapReduce>> create datasets

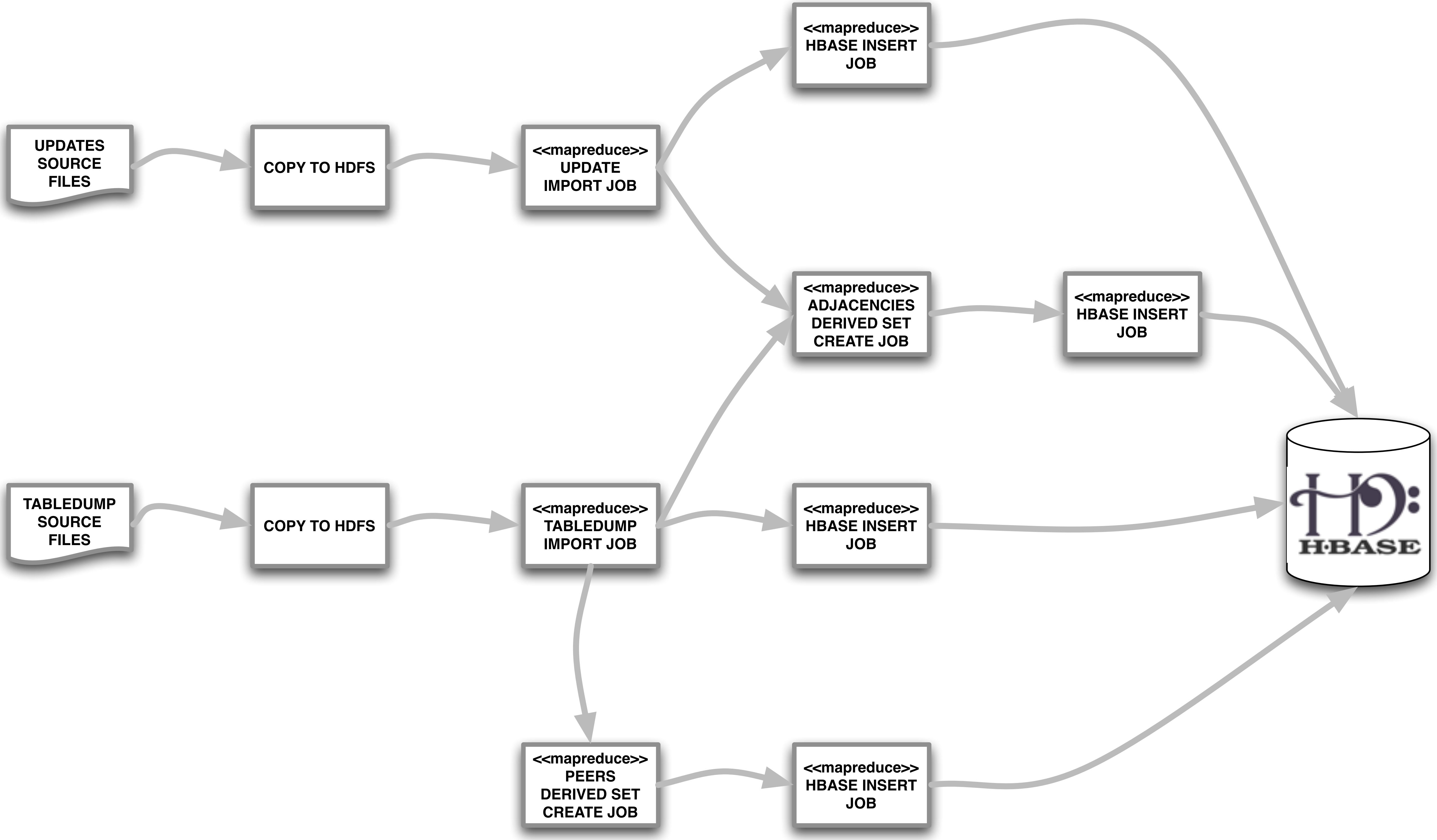
intermediate data on HDFS

<<MapReduce>> insert into HBase

query

ONLINE QUERY SYSTEM

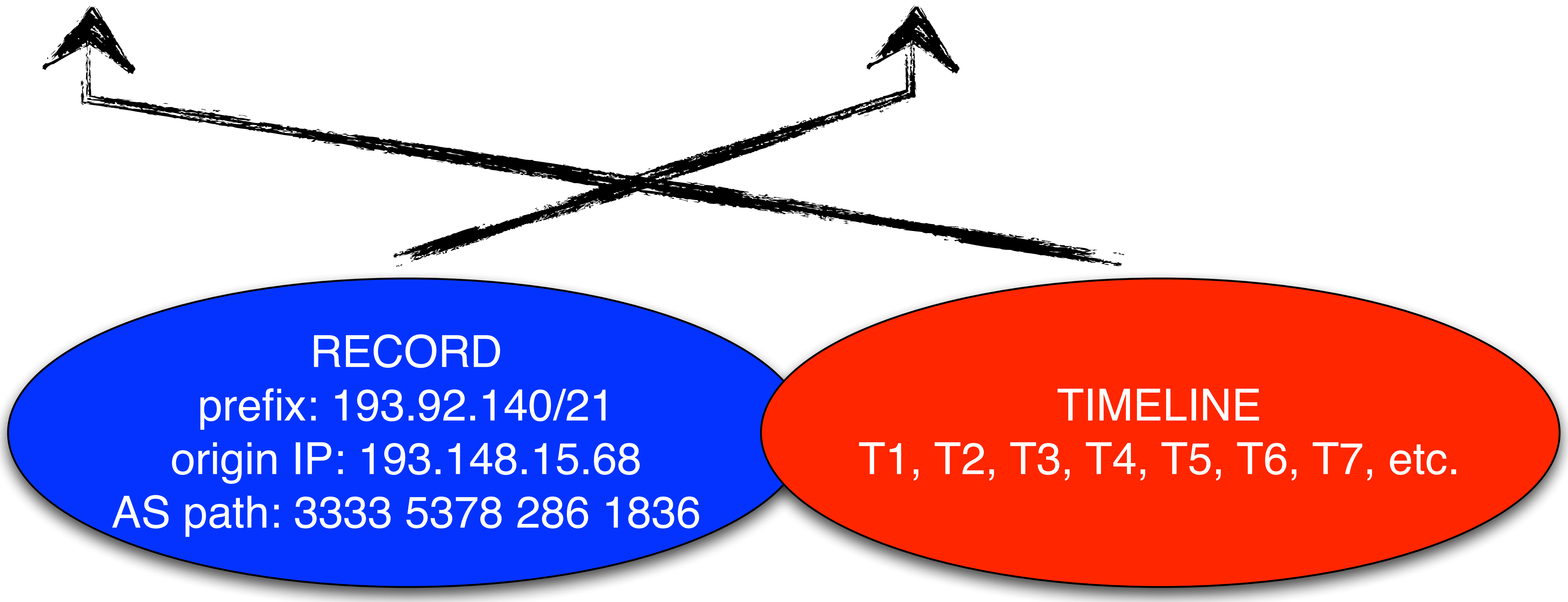
Import


UPDATES SOURCE FILES
COPY TO HDFS
<<mapreduce>> UPDATE IMPORT JOB
<<mapreduce>> HBASE INSERT JOB
<<mapreduce>> ADJACENCIES DERIVED SET CREATE JOB
<<mapreduce>> HBASE INSERT JOB
TABLEDUMP SOURCE FILES
COPY TO HDFS
<<mapreduce>> TABLEDUMP IMPORT JOB
<<mapreduce>> HBASE INSERT JOB
HBASE
<<mapreduce>> PEERS DERIVED SET CREATE JOB
<<mapreduce>> HBASE INSERT JOB

Import

- Hadoop MapReduce for parallelization and fault tolerance
- Chain jobs to create derived datasets
- Database store operations are idempotent

```
BGP4MP 980099497 A|193.148.15.68|3333|192.37.0.0/16|3333 5378 286 1836|IGP|193.148.15.140|0|0||NAG||
```



RECORD
prefix: 193.92.140/21
origin IP: 193.148.15.68
AS path: 3333 5378 286 1836

TIMELINE
T1, T2, T3, T4, T5, T6, T7, etc.

Inserting new data points means read-modify-write:

- find record
- fetch timeline
- merge timeline with new data
- write merged timeline

Schema and representation

ROW KEY	CF: DATA	CF: META
[index bytes]	[record]:[timeline] [record]:[timeline] [record]:[timeline] [record]:[timeline]	exists = Y
[index bytes]	[record]:[timeline] [record]:[timeline] [record]:[timeline] [record]:[timeline]	exists = Y

Schema and representation

- Multiple indexes; data is duplicated for each index
- Protobuf encoded records and timelines
- Timelines are timestamps or pairs of timestamps denoting validity intervals
- Data is partitioned over time segments (index contains time part)
- Indexes have an additional discriminator byte to avoid extremely wide rows
- Keep an in-memory representation of the key space for backwards traversal
- Keep data hot spot (most recent two to three weeks) in HBase block cache

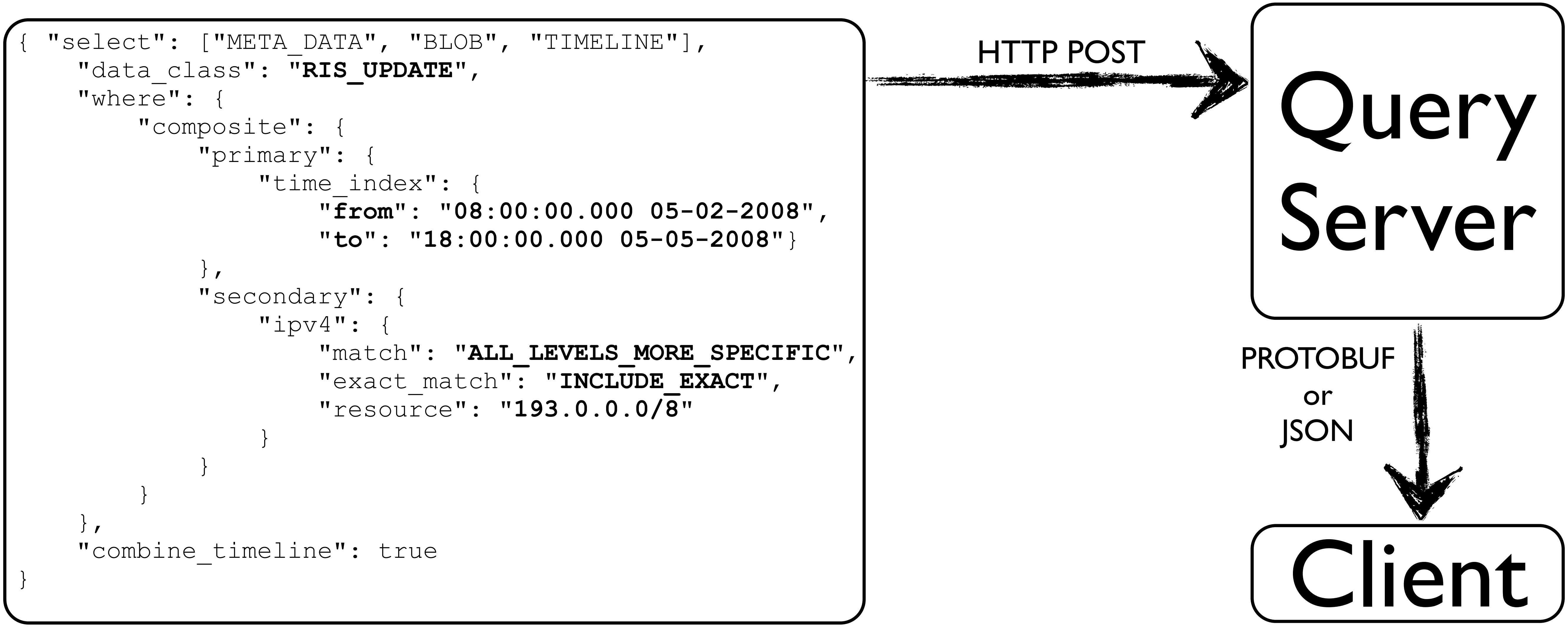
Schema and representation

Strategy:

- Make sure data hotspot fits in HBase block cache
- Scale out seek operations for (higher latency) access to old data

Result:

- 10x 10K RPM disk per worker node
- 16GB heap space for region servers



```
{ "select": ["META_DATA", "BLOB", "TIMELINE"],
    "data_class": "RIS_UPDATE",
    "where": {
        "composite": {
            "primary": {
                "time_index": {
                    "from": "08:00:00.000 05-02-2008",
                    "to": "18:00:00.000 05-05-2008"}
            },
            "secondary": {
                "ipv4": {
                    "match": "ALL_LEVELS_MORE_SPECIFIC",
                    "exact_match": "INCLUDE_EXACT",
                    "resource": "193.0.0.0/8"
                }
            }
        }
    },
    "combine_timeline": true
}
```

- In-memory index of all resources (IPs and AS numbers)
- Internally generates lists ranges to scan that match query predicates
- Combines time partitions into a single timeline before returning to client
- Runs on every RS behind a load balancer
- Knows which data exists

Hadoop and HBase: some experiences

Blocking writes because of memstore flushing and compactions.

Tuned these (amongst others):

```
hbase.regionserver.global.memstore.upperLimit
hbase.regionserver.global.memstore.lowerLimit
hbase.hregion.memstore.flush.size
hbase.hregion.max.filesize
hbase.hstore.compactionThreshold
hbase.hstore.blockingStoreFiles
hbase.hstore.compaction.max
```

Hadoop and HBase: some experiences

Dev machines: dual quad core, 32GB RAM, 4x 750GB 5.4K RPM data disks, 150GB OS disk

Prod machines: dual quad core, 64GB RAM, **10x 600GB 10K RPM data disks**, 150GB OS disk

See a previously *IO bound* job become *CPU bound* because of expanded IO capacity.

Hadoop and HBase: some experiences

Garbage collecting a 16GB heap can take quite a long time.

Using G1*. Full collections still take seconds, but no missed heartbeats.

* New HBase versions have a fix / feature that works with the CMS collector. RIPE NCC is looking into upgrading.

Hadoop and HBase: some experiences

Estimating required capacity and machine profiles is hard.

Pick a starting point. Monitor. Tune. Scale. In that order.

Hadoop and HBase: some experiences

Other useful tools

- Ganglia
- CFEngine / Puppet
- Splunk
- Bash

Hadoop and HBase: some experiences

Awesome community!

(mailing lists are a big help when in trouble)

Q&A?

Friso van Vollenhoven
fvanvollenhoven@xebia.com