Performance Management with Packetbeat & Elasticsearch

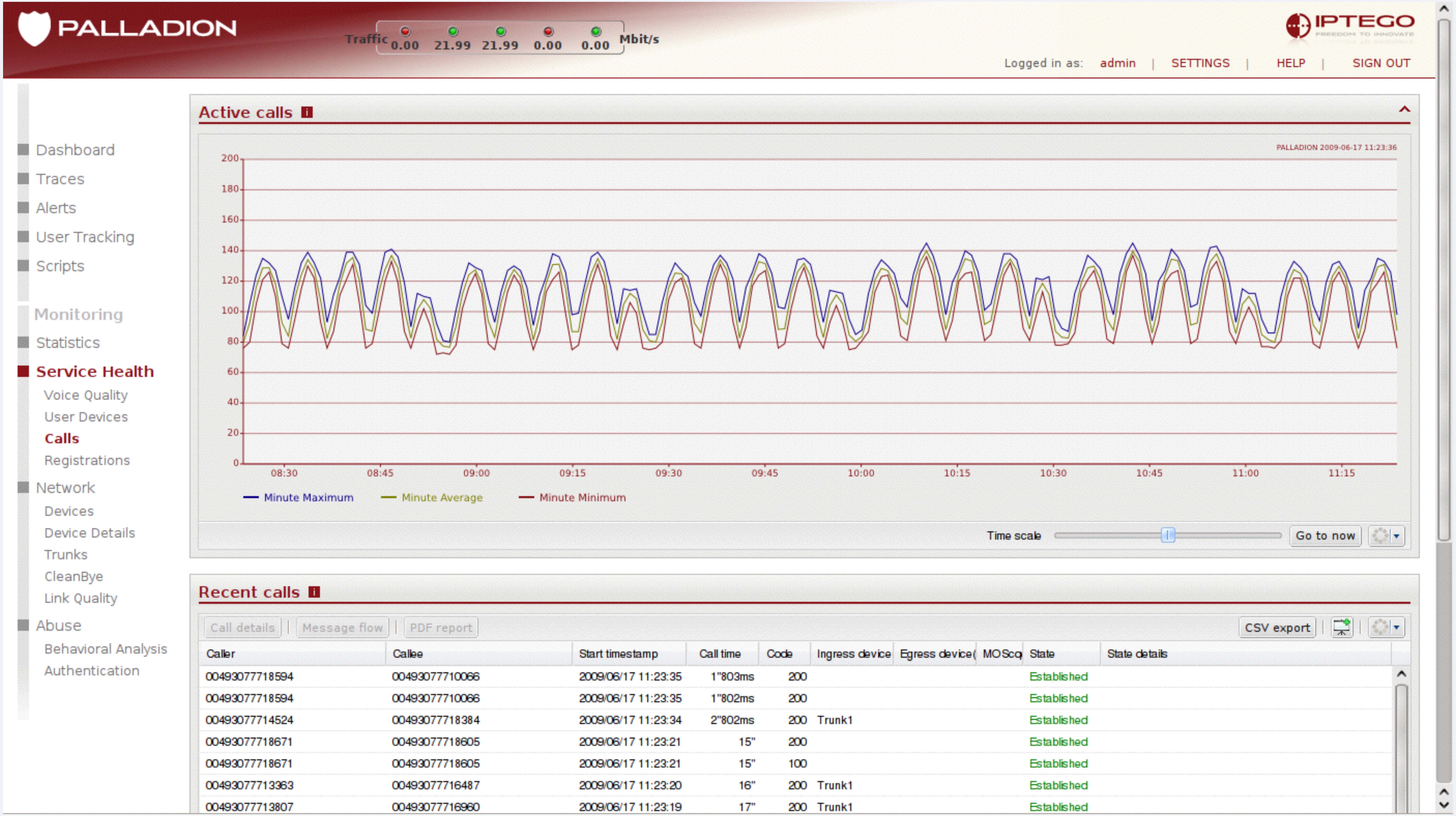
Tudor Golubenco
@tudor_g

KAMAILIO WORLD
CONFERENCE & EXHIBITION

```
$ whoami
```

Intro

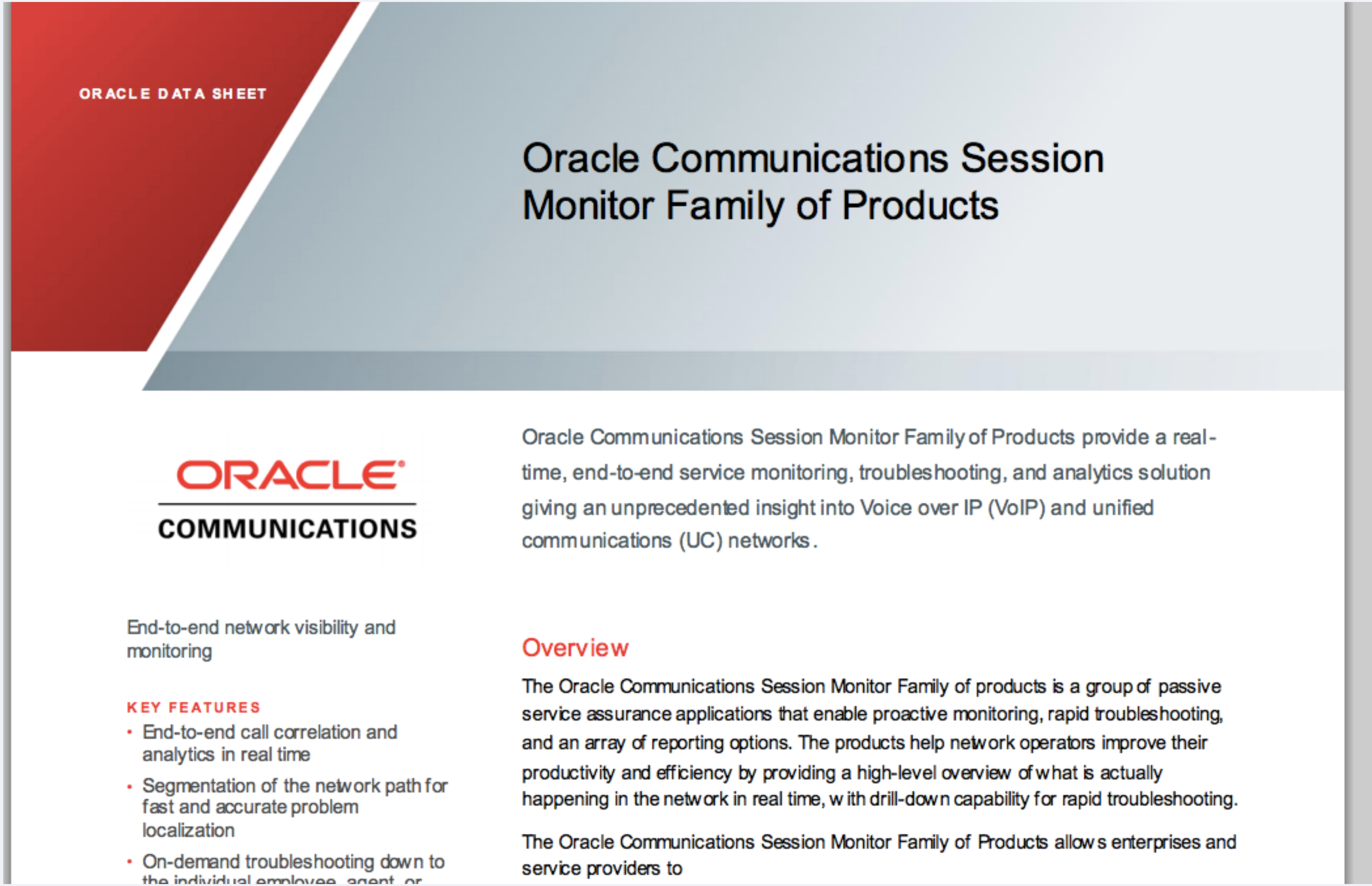
- Romanian living in Berlin
- Student @FOKUS in 2006, diploma project about handover in IMS networks
- Joined Iptego, a young VoIP company
  - CTO starting from 2009ish



Caller	Callee	Start timestamp	Call time	Code	Ingress device	Egress device(	MOScq	State	State details
00493077718594	00493077710066	2009/06/17 11:23:35	1"803ms	200				Established	
00493077718594	00493077710066	2009/06/17 11:23:35	1"802ms	200				Established	
00493077714524	00493077718384	2009/06/17 11:23:34	2"802ms	200	Trunk1			Established	
00493077718671	00493077718605	2009/06/17 11:23:21	15"	200				Established	
00493077718671	00493077718605	2009/06/17 11:23:21	15"	100				Established	
00493077713363	00493077716487	2009/06/17 11:23:20	16"	200	Trunk1			Established	
00493077713807	00493077716960	2009/06/17 11:23:19	17"	200	Trunk1			Established	

Palladion

- Monitoring and troubleshooting for SIP (also RTP, RTCP, H.248, ENUM, Diameter, etc.)
- Iptego acquired by Acme Packet (2012)
- Acme Packet acquired by Oracle (2013)



ORACLE DATA SHEET

Oracle Communications Session Monitor Family of Products

ORACLE
COMMUNICATIONS

Oracle Communications Session Monitor Family of Products provide a real-time, end-to-end service monitoring, troubleshooting, and analytics solution giving an unprecedented insight into Voice over IP (VoIP) and unified communications (UC) networks.

End-to-end network visibility and monitoring

KEY FEATURES

- End-to-end call correlation and analytics in real time
- Segmentation of the network path for fast and accurate problem localization
- On-demand troubleshooting down to the individual employee, agent, or

Overview

The Oracle Communications Session Monitor Family of products is a group of passive service assurance applications that enable proactive monitoring, rapid troubleshooting, and an array of reporting options. The products help network operators improve their productivity and efficiency by providing a high-level overview of what is actually happening in the network in real time, with drill-down capability for rapid troubleshooting.

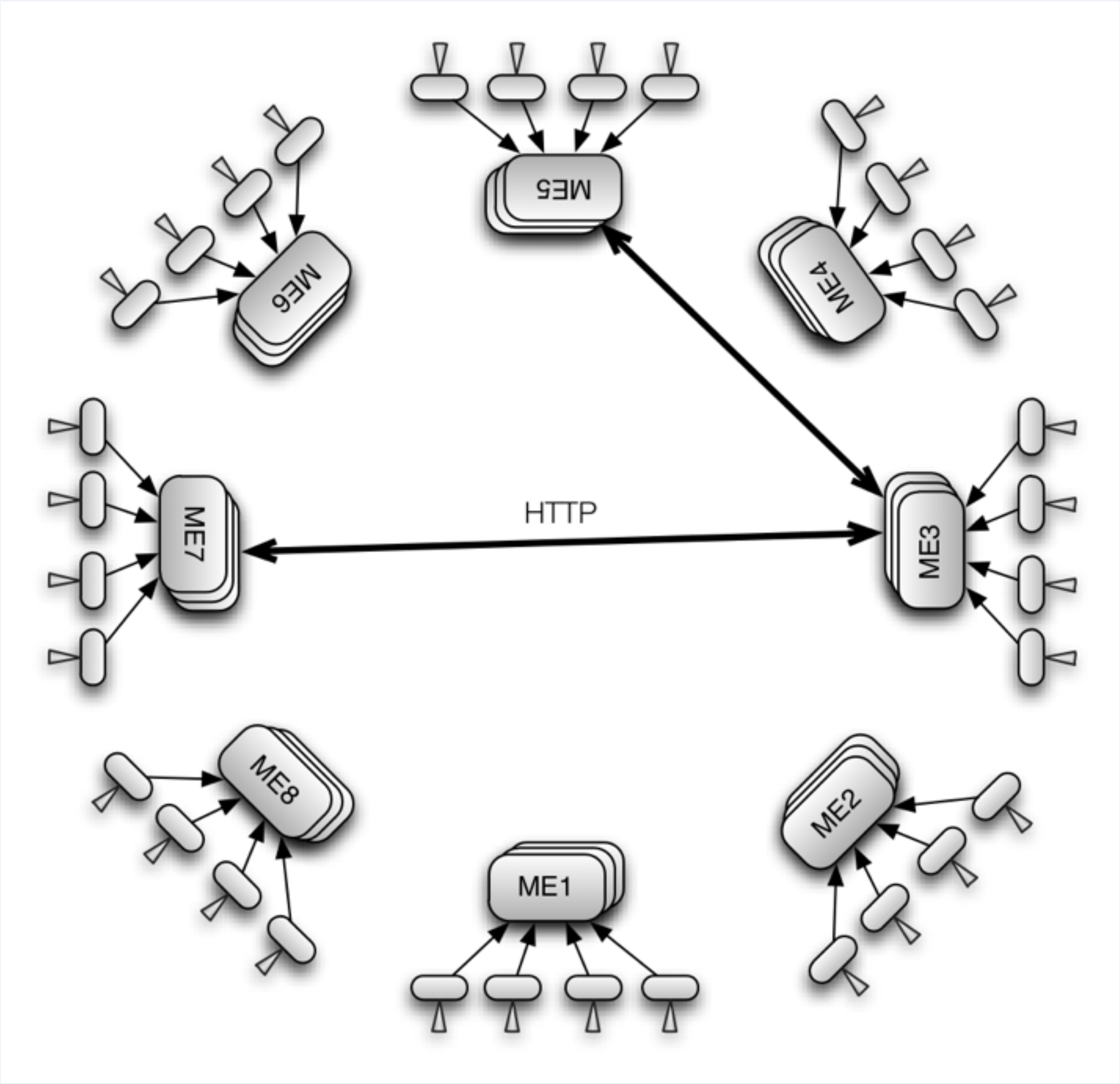
The Oracle Communications Session Monitor Family of Products allows enterprises and service providers to

Got complex with metrics

- Each new metric added complexity to the application (written C)
- Large number of metrics (~500K metrics)
- Each new feature and protocol needed to support all metrics

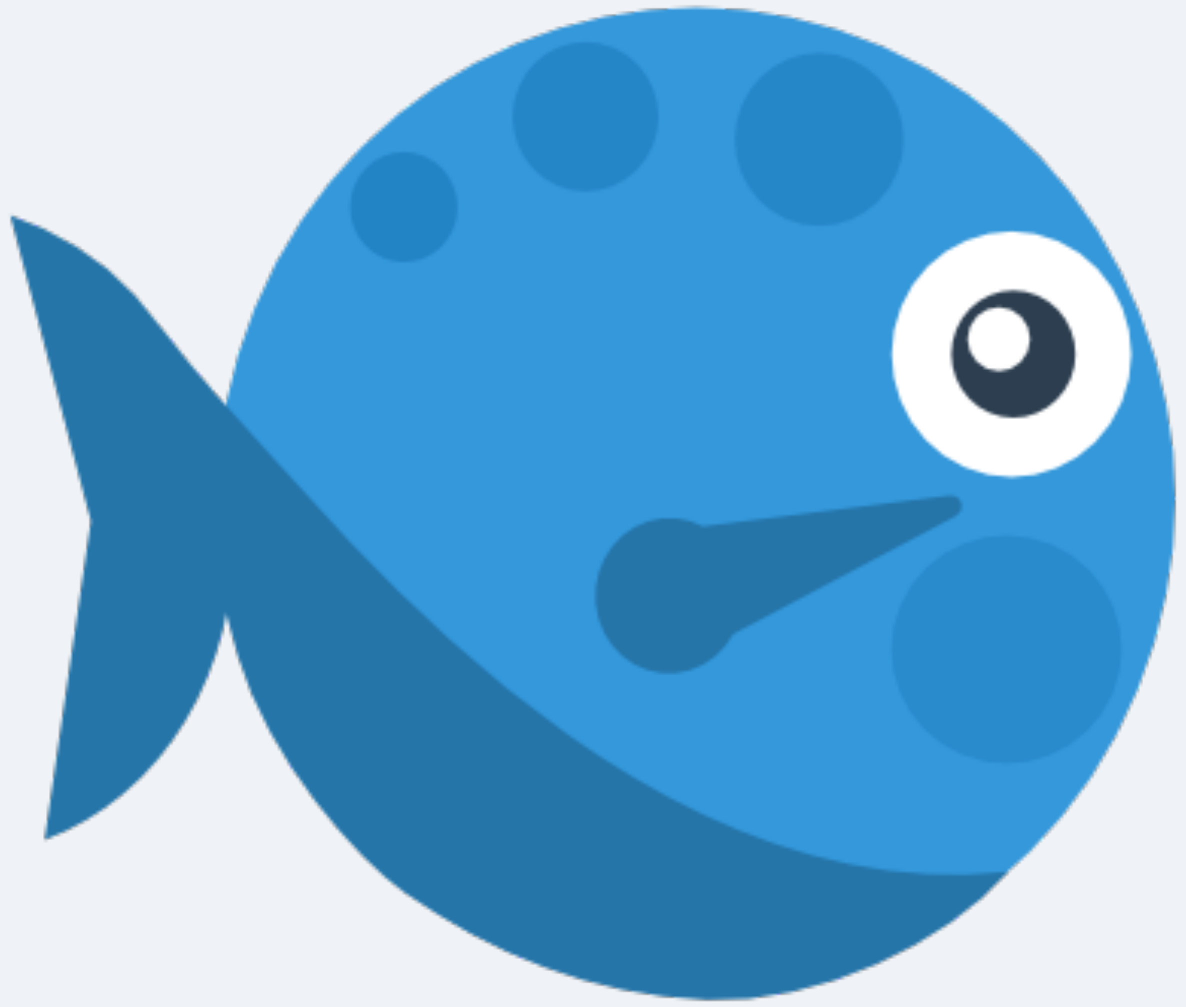
Got really complex with scalability

- We needed to show the global state (i.e. total number of active calls, end-to-end calls)
- Difficult when the data is distributed



It would be nice to have a system just like Palladion to monitor Palladion itself

Packetbeat

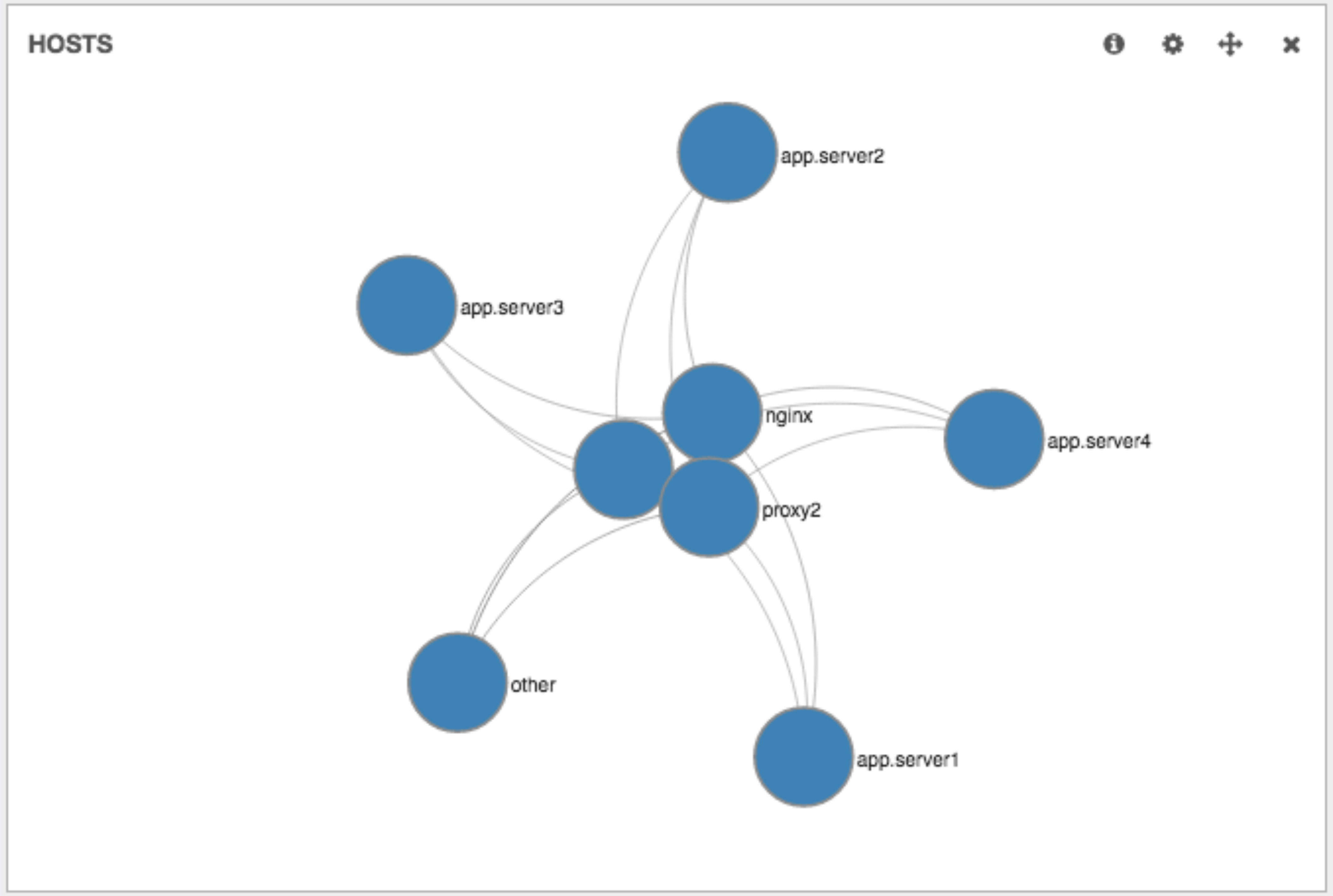


- Started by Monica Sarbu, first public version in 05.2014
- I joined full time 11.2014

monitoring and troubleshooting for distributed applications

Start from the communication

- The communication between components gets you the big picture
- Protocols are universal
- It’s objective
- No latency overhead



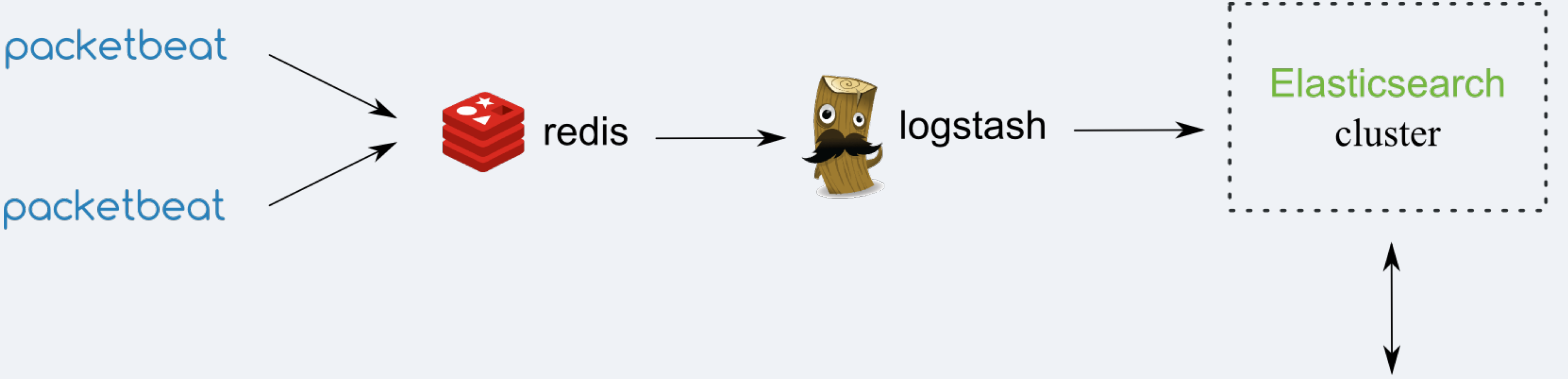
How it works

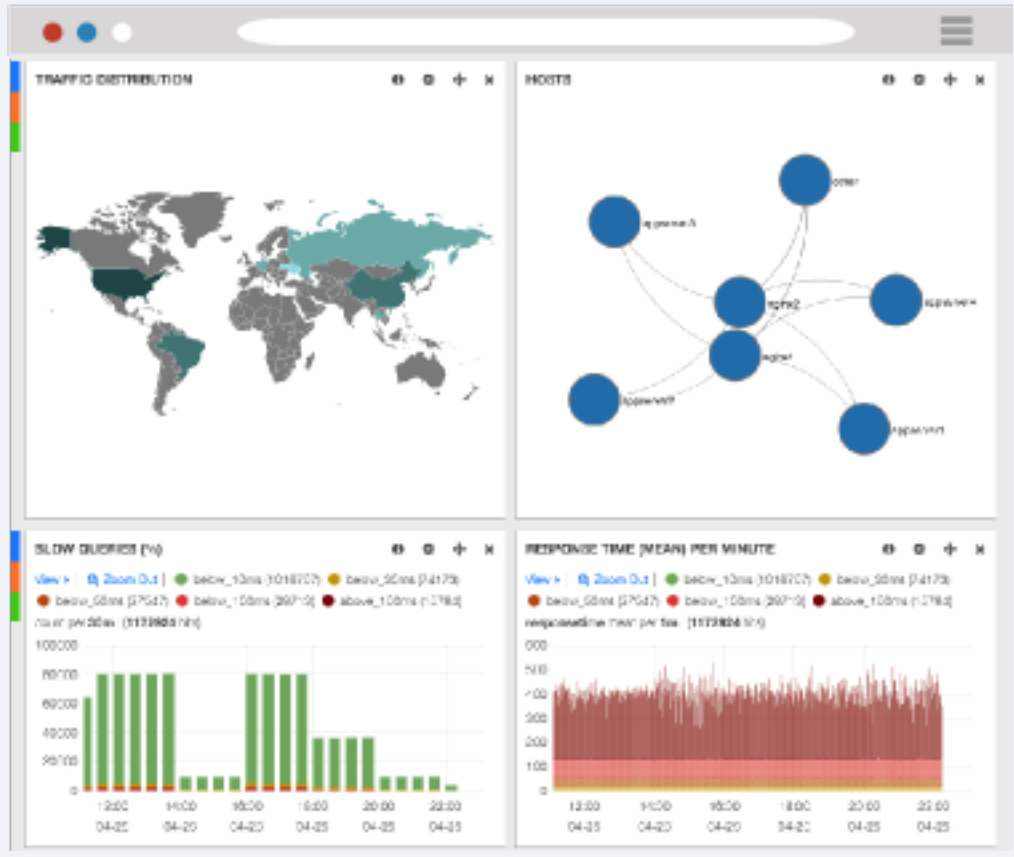
- Captures the wire traffic (libpcap, pfring, af_packet)
- Follows TCP streams, decodes HTTP, MySQL, PgSQL, Redis, Thrift-RPC
- Looks for requests, waits for the matching response
- Records response time, URLs, response codes, etc

```
$ packetbeat -e -d “publish”
```

```
{
  "client_ip": "127.0.0.1",
  "client_port": 46981,
  "ip": “127.0.0.1",
  "query": "select * from test",
  "method": "SELECT",
  "pgsql": {
    "error_code": "",
    "error_message": "",
    "error_severity": "",
    "iserror": false,
    "num_fields": 2,
    "num_rows": 2
  },
  "port": 5432,
  "responsetime": 12,
  "bytes_out": 95,
  "status": "OK",
  "timestamp": "2015-05-27T22:27:57.409Z",
  "type": "pgsql"
}
```

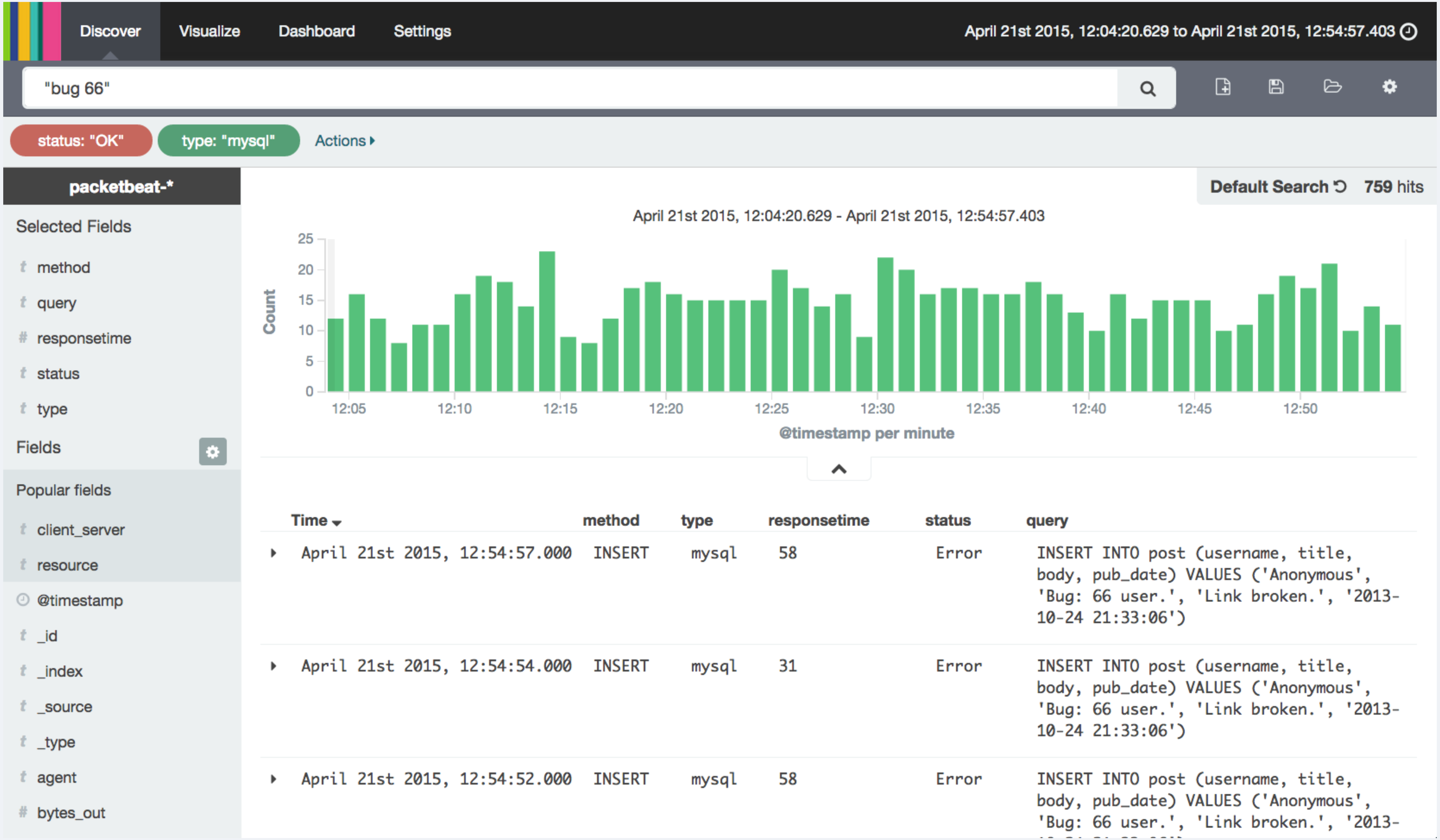
Packetbeat + ELK





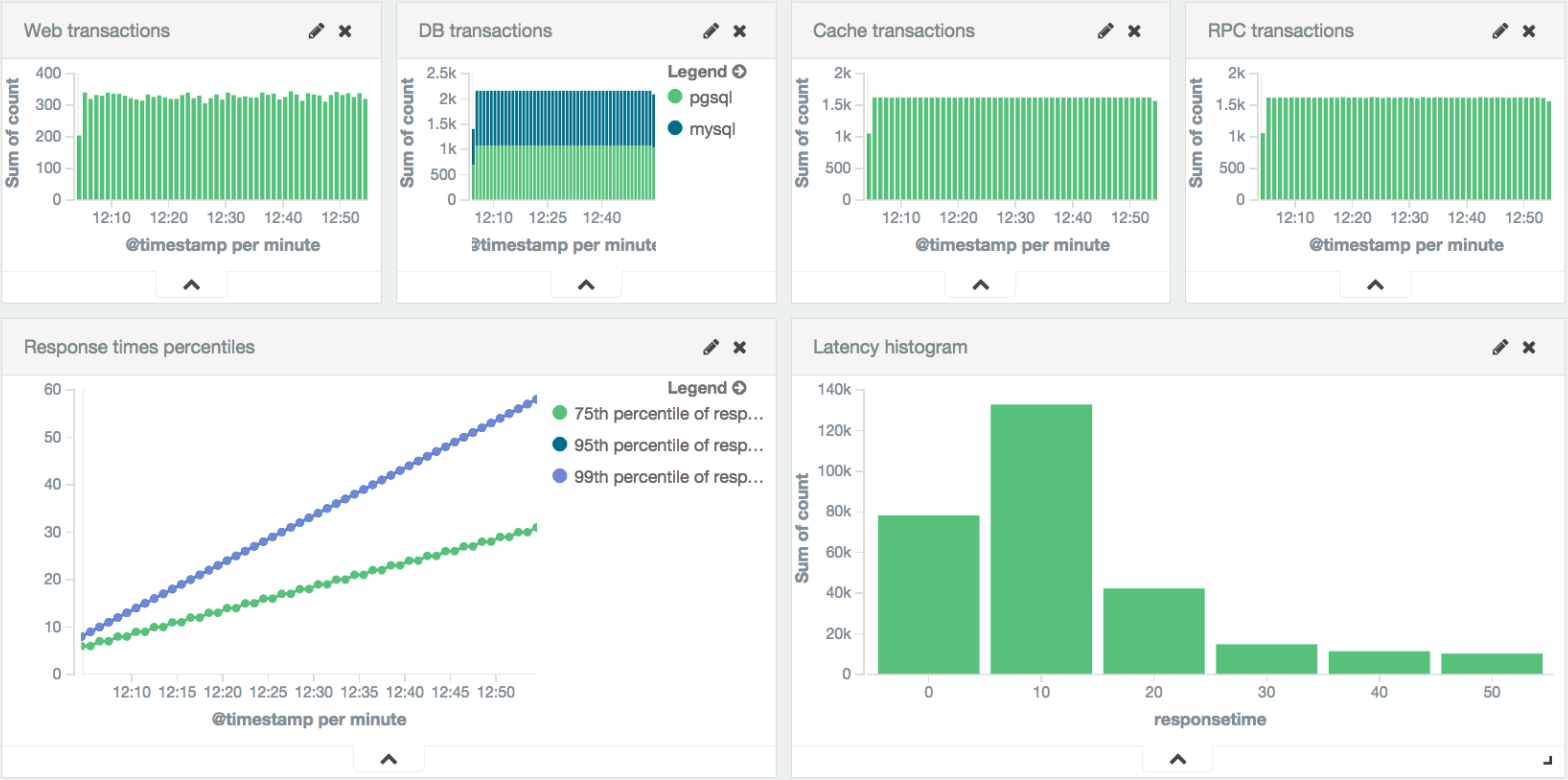
Why ELK?

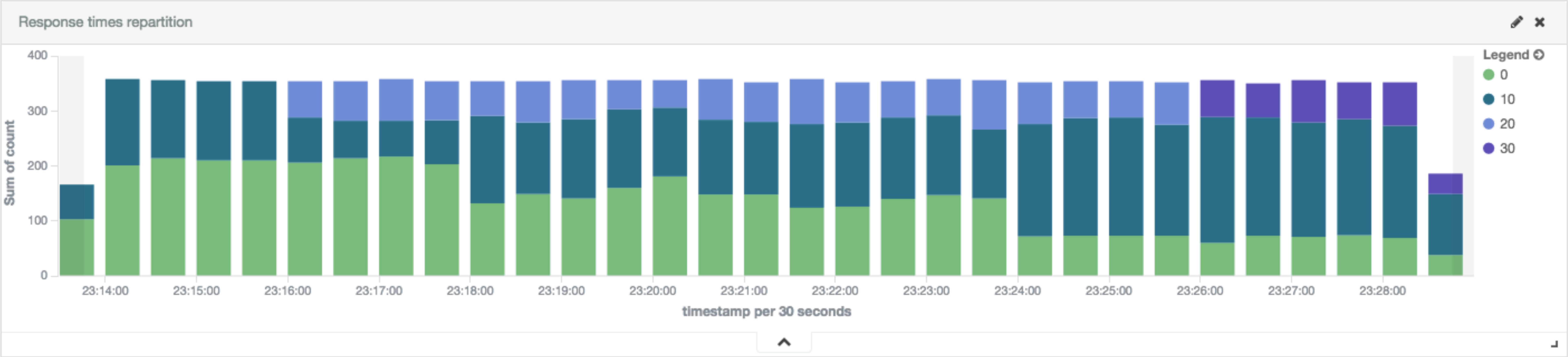
- Already proven to scale and perform for logs
- Clear and simple flow for the data
- “Send the code where the data is, not the other way around”
- Powerful features that become simple:
  - Drilling down to the transactions related to a peak
  - Top N features are trivial
  - Slicing by different dimensions is easy



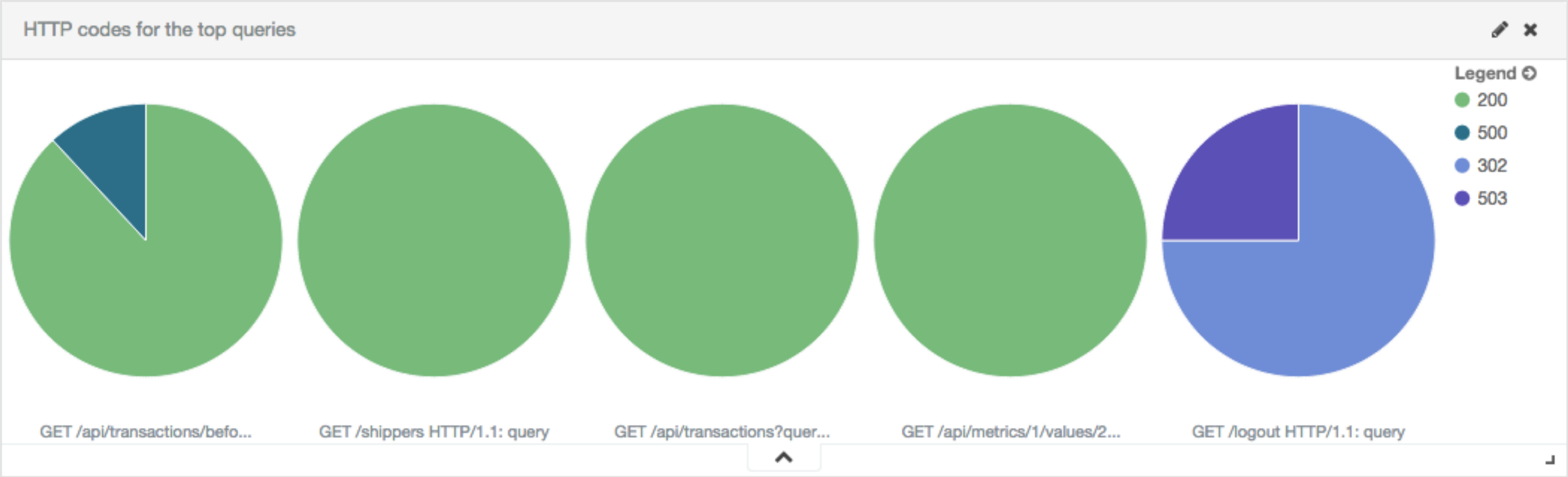
"bug 66"

Time	method	type	responsetime	status	query
April 21st 2015, 12:54:57.000	INSERT	mysql	58	Error	INSERT INTO post (username, title, body, pub_date) VALUES ('Anonymous', 'Bug: 66 user.', 'Link broken.', '2013-10-24 21:33:06')
April 21st 2015, 12:54:54.000	INSERT	mysql	31	Error	INSERT INTO post (username, title, body, pub_date) VALUES ('Anonymous', 'Bug: 66 user.', 'Link broken.', '2013-10-24 21:33:06')
April 21st 2015, 12:54:52.000	INSERT	mysql	58	Error	INSERT INTO post (username, title, body, pub_date) VALUES ('Anonymous', 'Bug: 66 user.', 'Link broken.', '2013-

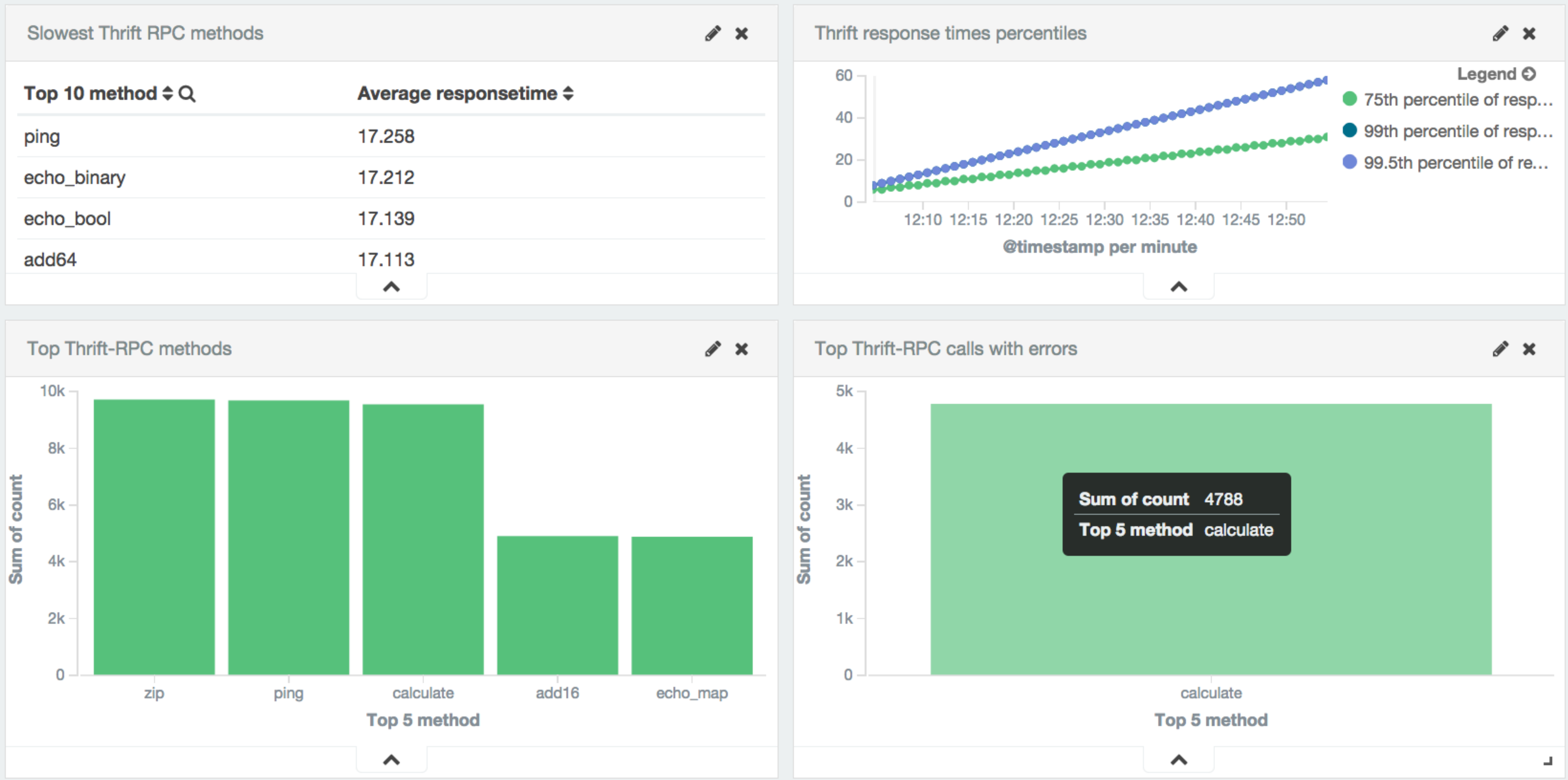

Web transactions
Sum of count
400
300
200
100
0
12:10
12:20
12:30
12:40
12:50
@timestamp per minute
DB transactions
Sum of count
2.5k
2k
1.5k
1k
500
0
12:10
12:25
12:40
@timestamp per minute
Legend
● pgsql
● mysql
Cache transactions
Sum of count
2k
1.5k
1k
500
0
12:10
12:20
12:30
12:40
12:50
@timestamp per minute
RPC transactions
Sum of count
2k
1.5k
1k
500
0
12:10
12:20
12:30
12:40
12:50
@timestamp per minute
Response times percentiles
60
50
40
30
20
10
0
12:10
12:15
12:20
12:25
12:30
12:35
12:40
12:45
12:50
@timestamp per minute
Legend
● 75th percentile of resp…
● 95th percentile of resp…
● 99th percentile of resp…
Latency histogram
Sum of count
140k
120k
100k
80k
60k
40k
20k
0
0
10
20
30
40
50
responsetime



Response times repartition



HTTP codes for the top queries



Slowest Thrift RPC methods

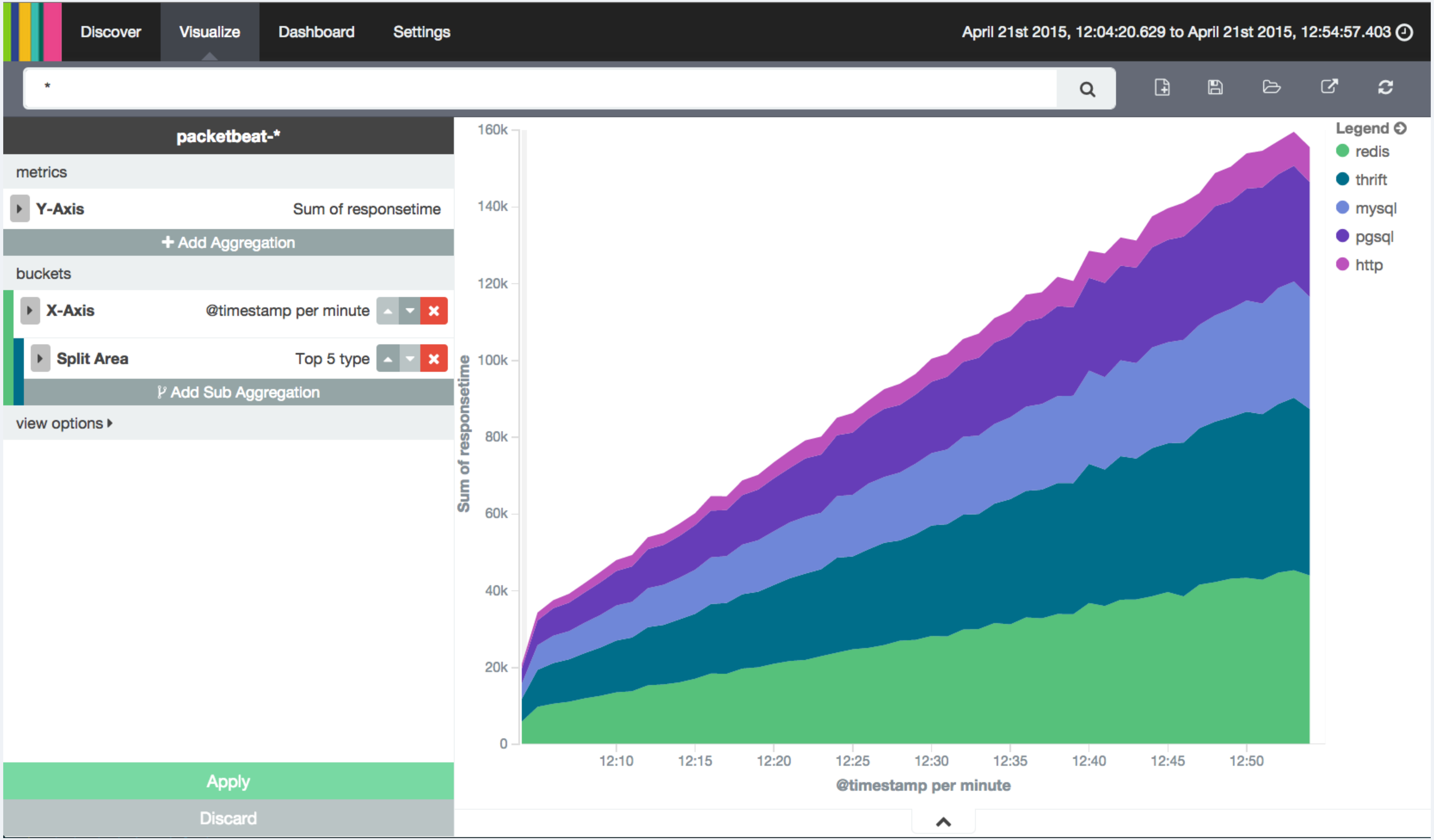
Top 10 method	Average responsetime
ping	17.258
echo_binary	17.212
echo_bool	17.139
add64	17.113

Thrift response times percentiles

60
40
20
0
12:10
12:15
12:20
12:25
12:30
12:35
12:40
12:45
12:50
@timestamp per minute
Legend
● 75th percentile of resp...
● 99th percentile of resp...
● 99.5th percentile of re...

Top Thrift-RPC methods

Top Thrift-RPC calls with errors



April 21st 2015, 12:04:20.629 to April 21st 2015, 12:54:57.403

*

packetbeat-*

metrics

Y-Axis Sum of responsetime

buckets

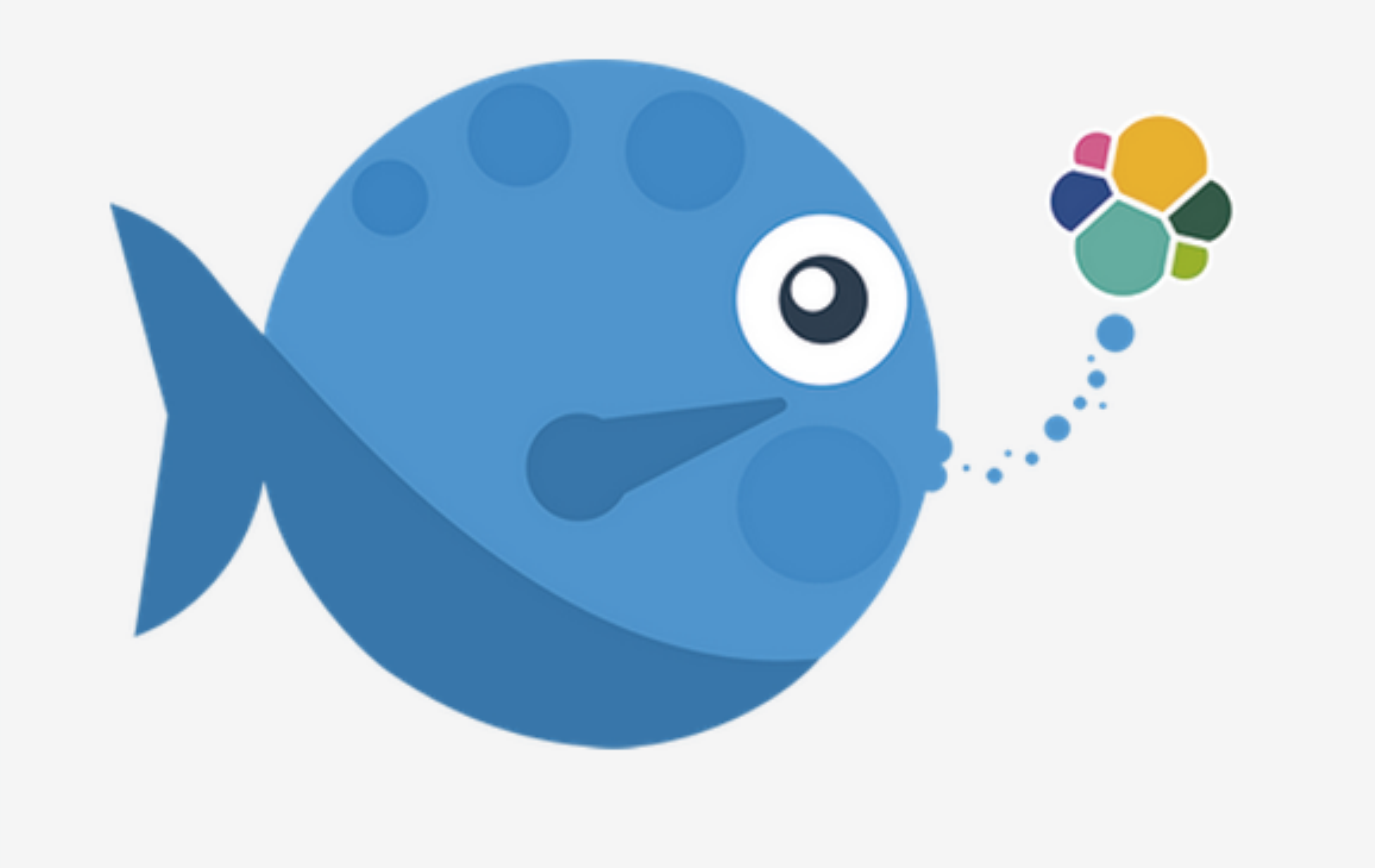
X-Axis @timestamp per minute

Split Area Top 5 type

Future plans

- Packet data is just the beginning
- Other sources of operational data:
  - OS readings: CPU, memory, IO stats
  - Code instrumentation
  - API gateways
  - Common servers internal stats (Nginx, Elasticsearch, Kamailio)

Joining Elastic



```
from __future__ import beats
```

The Beats

- Packetbeat - data from the wire
- Filebeat (Logstash-Forwarder) - data from log files
- Future:
  - Topbeat - CPU, mem, IO stats
  - Metricsbeat - arbitrary metrics from nagios/sensu like scripts
  - RUMbeat - data from the browser
  - Kamiliobeat (?)

Stay in touch

- @tudor_g / @packetbeat
- https://discuss.elastic.co/c/beats
- Sign up for the webinar:
  - https://www.elastic.co/webinars/beats-platform-for-leveraging-operational-data