

A solid green vertical bar is positioned to the left of the main title.

Stories from the Trenches at GoDaddy

Felix Gorodishter | @fgorodishter | felix@godaddy.com

3/8/2018

Who am I?

- Felix Gorodishter
 - Speak fluent Russian 😊
 - Diving since '96
 - With GoDaddy since '09
 - Currently Principal Architect
-
- Contact:
 - felix@godaddy.com
 - @fgorodishter



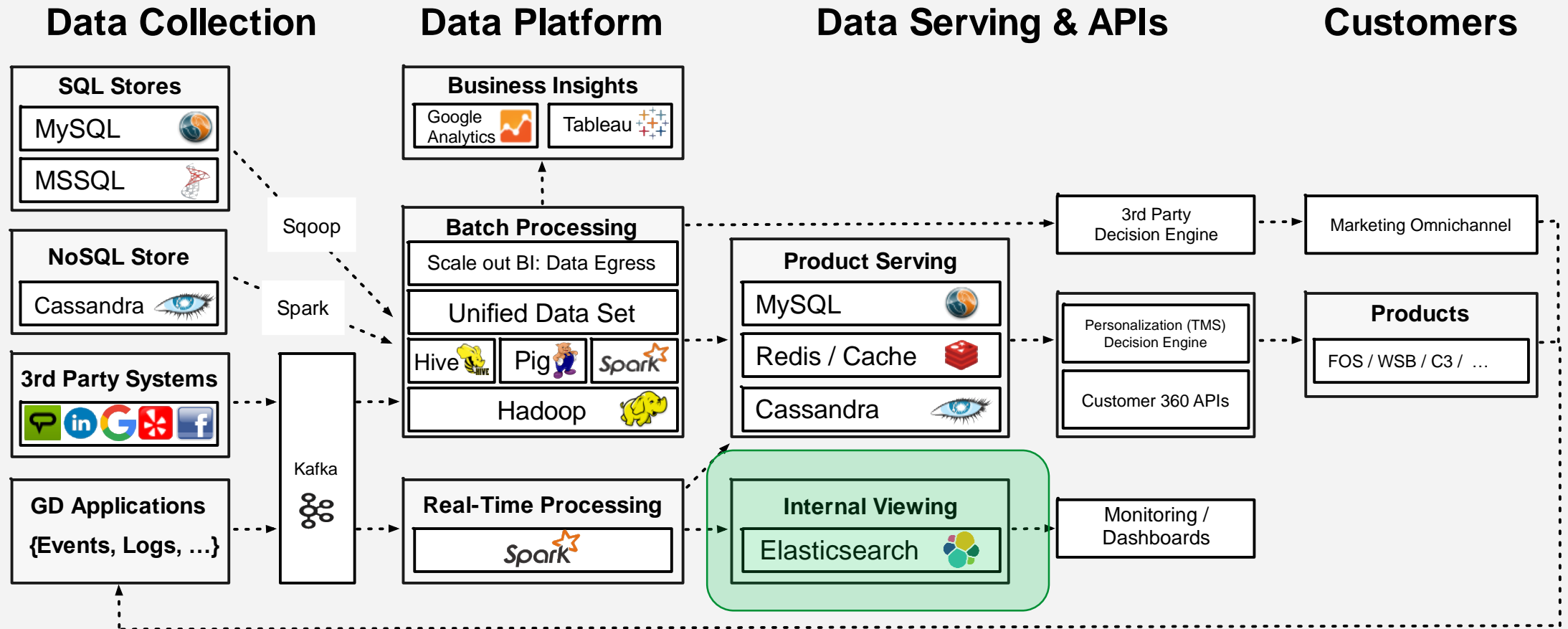


GoDaddy ...

Our vision is to radically shift the global economy toward life-fulfilling independent ventures.

- 17.3 M Customers worldwide (56 markets)
- 75 M Domains under management
- 10 M Websites hosted / 24 Datacenters
- 18 B DNS queries daily
- 2 B Attacks blocked monthly
- 85 K Servers
- 7000 Employees

Data Flows



11 PB

Managed HDFS

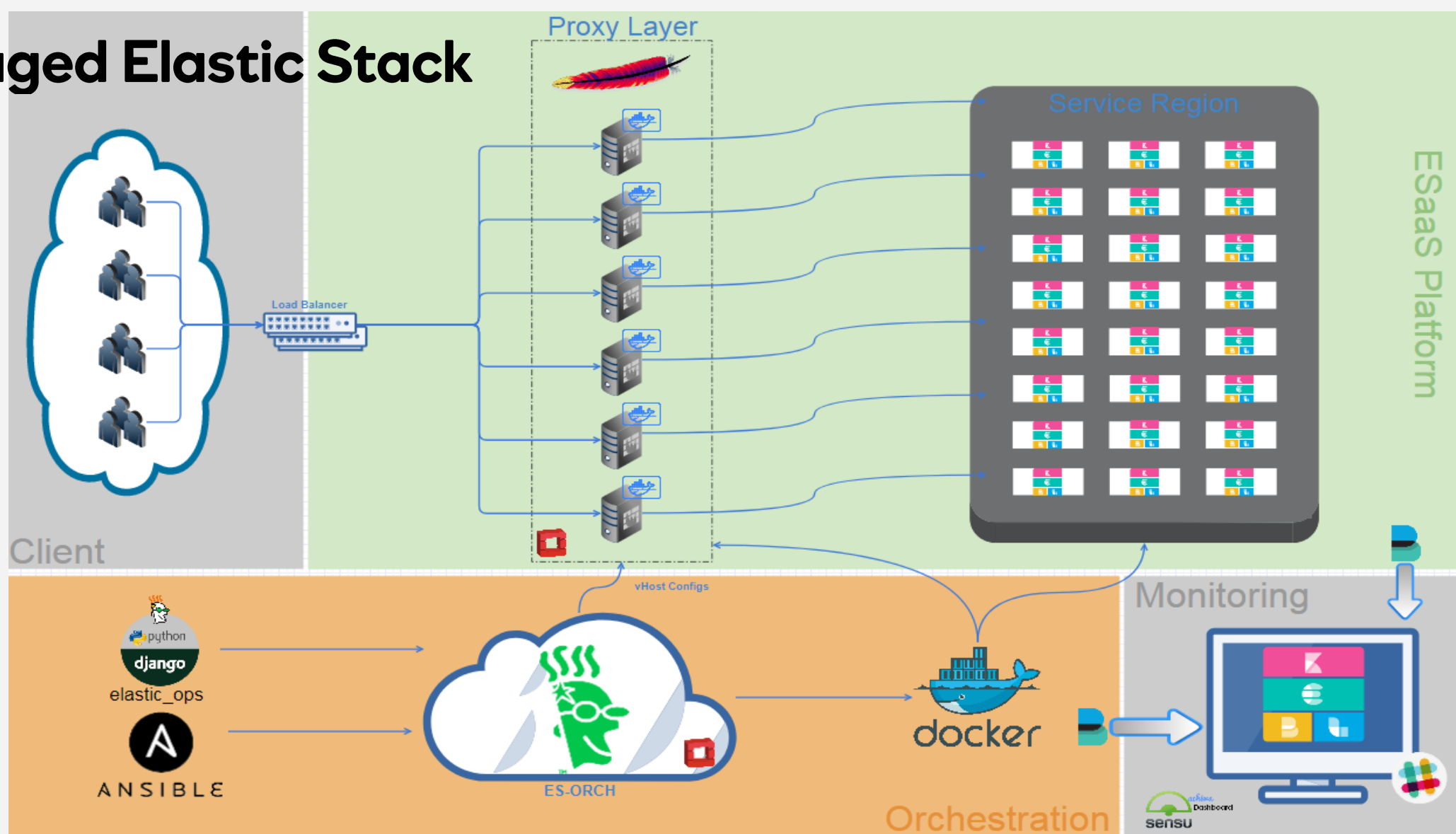
13 TB

New data per day in HDFS

200K

Messages per second

Managed Elastic Stack

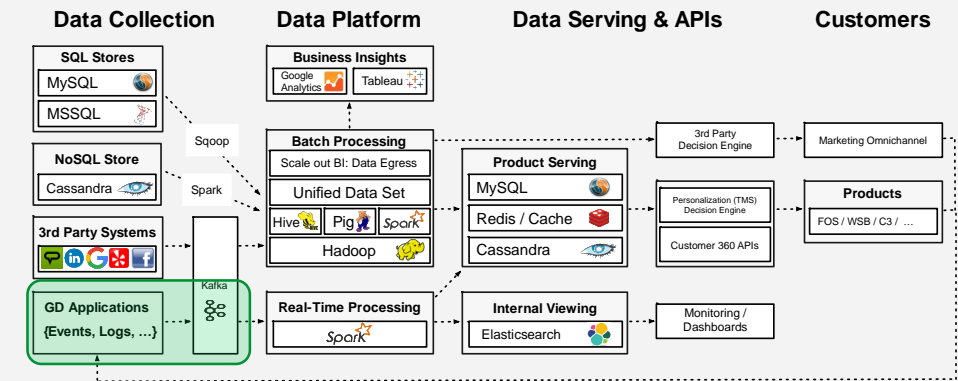


61 Managed Clusters

766 Containers

271 TB Indexed Data

Data Collection



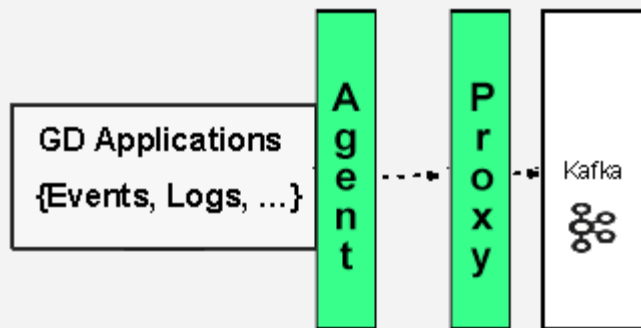
What we did

- Wrote agents for Linux and Windows *(in 2014)*
- Agent exposed local port on every server so teams can natively ship data over HTTP (and UDP too)

```
curl
-H "Content-Type: application/json"
-X POST
-d '{"fqdn":"'`hostname`'", "data":"felixtest"}'
http://localhost:<PORT>/v1/foo/bar
```

- Data from hosts is WRITE-ONLY into pipeline

Current State



Data Collection – for Operations

Our Agent(s) – CPM (Collector Process Manager)

- Operations/SRE needed more primitives
- Built it to be pluggable – Python on Linux & C# .NET on Windows
- Always ship base meta-data about sender
- Allow for tail or scheduled workloads

Linux

- /var/log/* (known useful stuff)
- /etc/passwd
- /etc/group & /etc/login.groups
- /etc/yum.conf & /etc/yum.repos.d/*
- rpm -qa
- yum check-update

Windows

- Application – event log
- System – event log
- Security – event log

Per Message Meta-data

```
[gd-linux-system-collector]
type = private
dc = phx3
env = staging
server_role = hypervisor
service_zone = phx-private-gen-zone-1
security_zone = mgmt
product_name = compute
```

If I were to do it today* ...

Evaluate open source tooling ... it has improved tremendously!



logstash



beats



fluentd

* We're currently evaluating tools



Winning Patching

Q: Are you patching?

A: Isn't that just magic?

Patching – What is it?

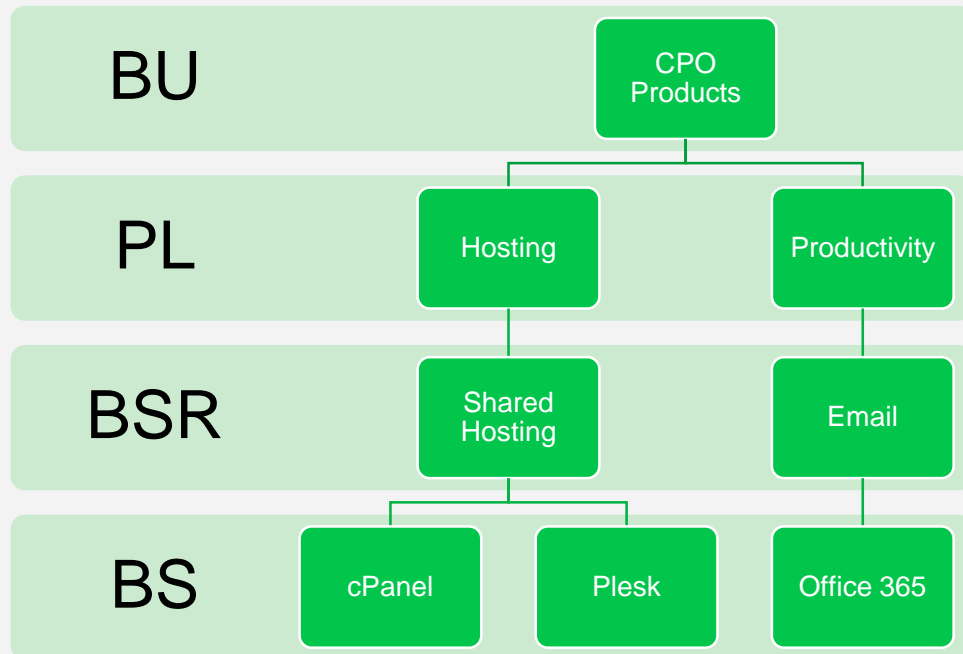
- Measure and report on the compliance and risk of our server fleet
- Support static and ephemeral infrastructure
- Support Windows & Linux
- Provide transparency in the data and collection
- Give the raw data to the teams
- Leverage the same data for ops to exec reporting



Business Service Mapping (BSM)

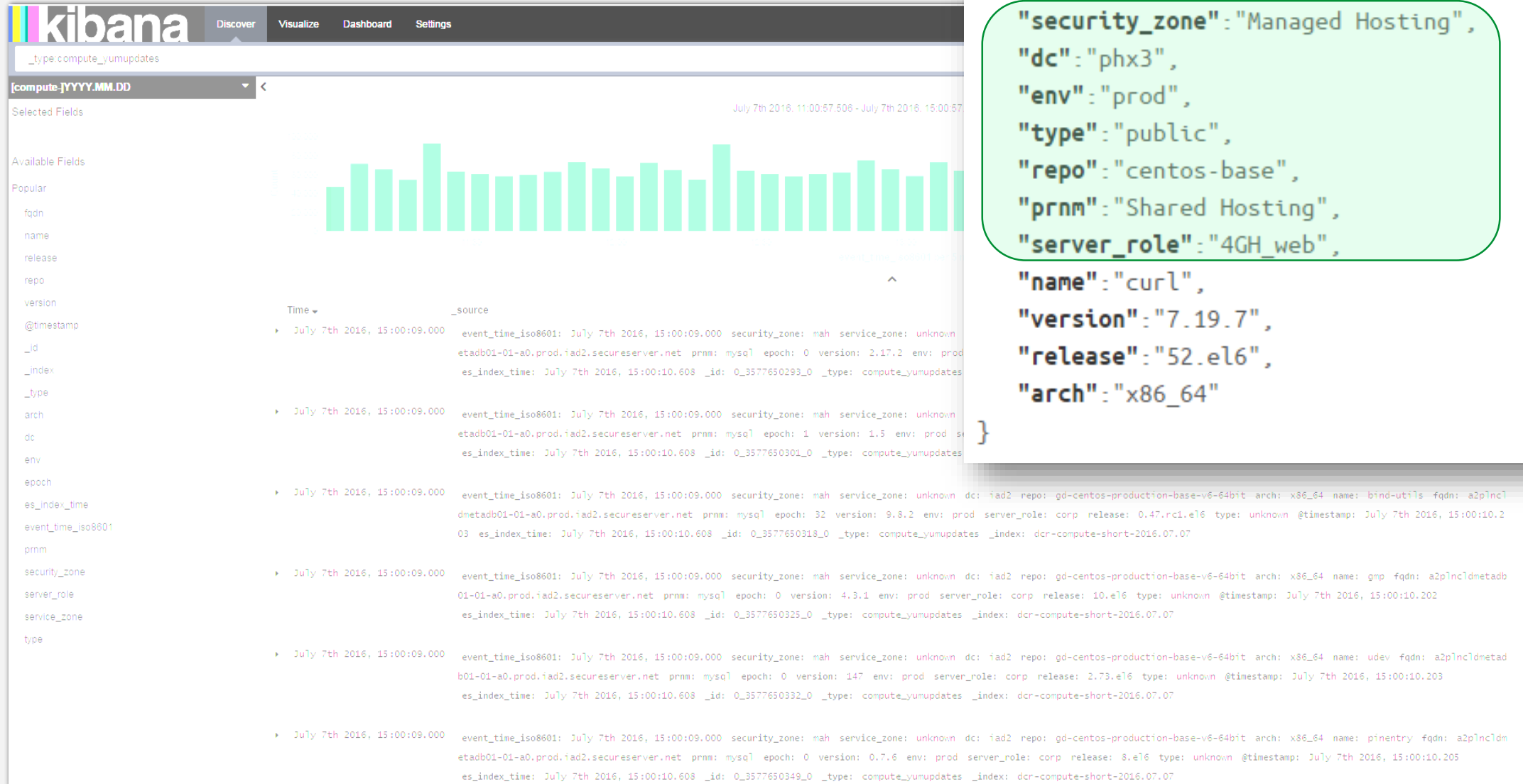
We leverage 4 layers:

- Business Unit (BU)
- Product Line (PL)
- Business Service Rollup (BSR)
- Business Service (BS)

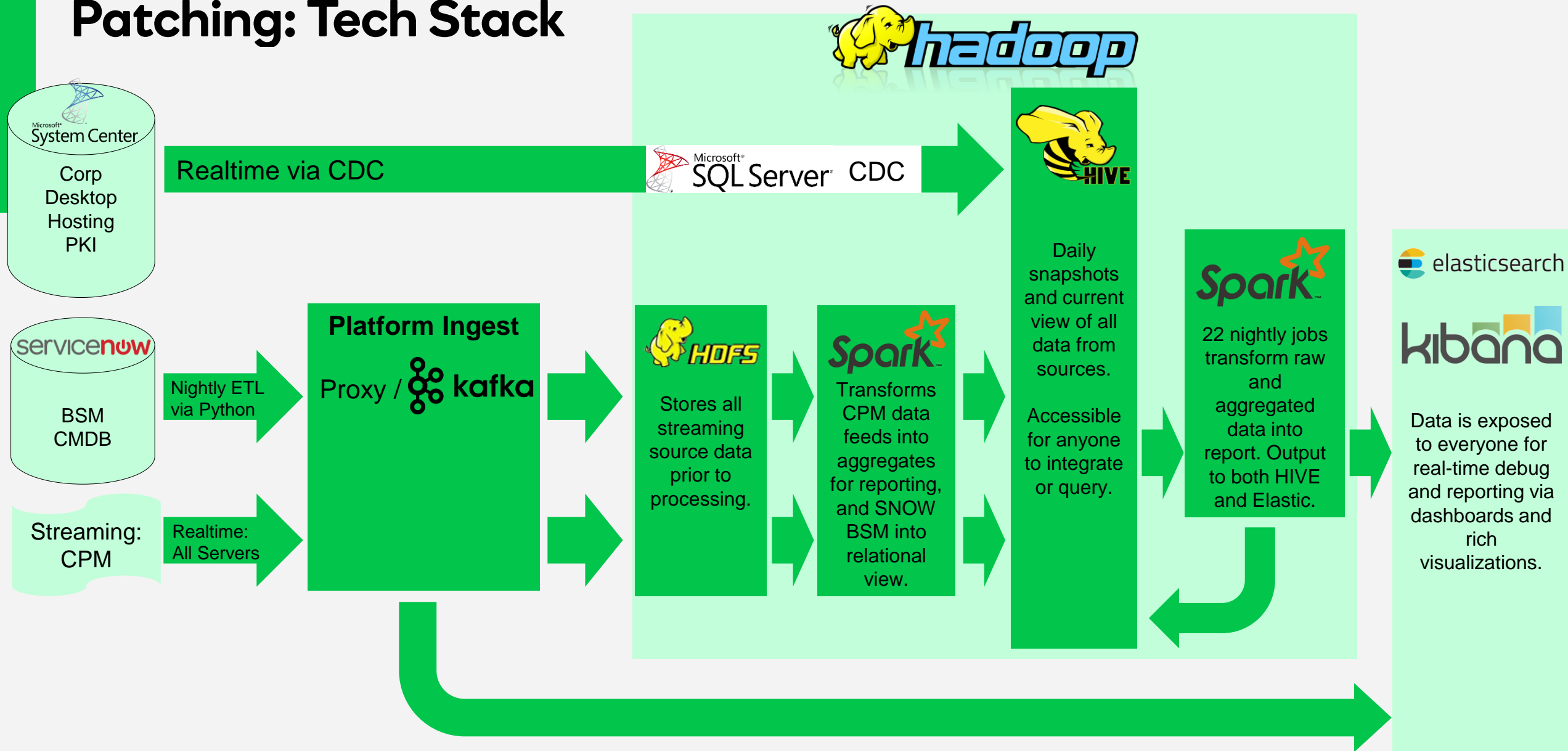


Patching

Once per hour, each host sends all available updates



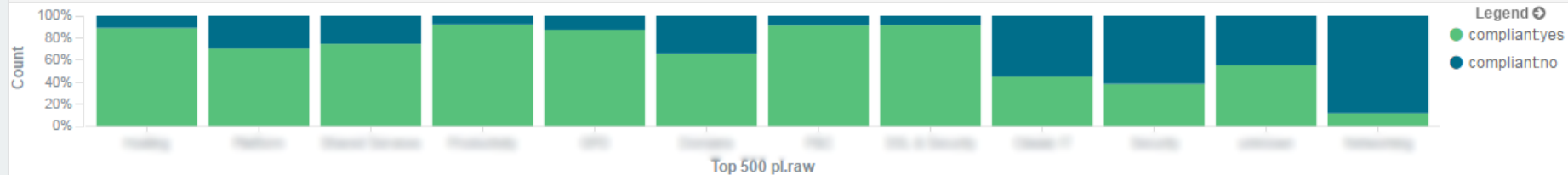
Patching: Tech Stack



Patching: PL Listing - Since Patch Tuesday

Top 500 bu.raw	Top 500 pl.raw	Count	Sum of compliant_int	Average complan
CPG Product: Rolling		7,701	6,856	0.89
CPG Product: Productivity		1,151	1,061	0.922
CPG Product: Smart		955	628	0.658
CPG Product: H&I		481	440	0.915
Customer: P&G		2,652	1,869	0.705
Customer: Intel		2,304	1,718	0.746
Customer: Gen 2		269	121	0.45
Customer: H&I		171	157	0.918
Customer: Intel		138	53	0.384
Customer: Rolling		17	2	0.118
Gen 2		794	694	0.874
unknown	unknown	67	37	0.552

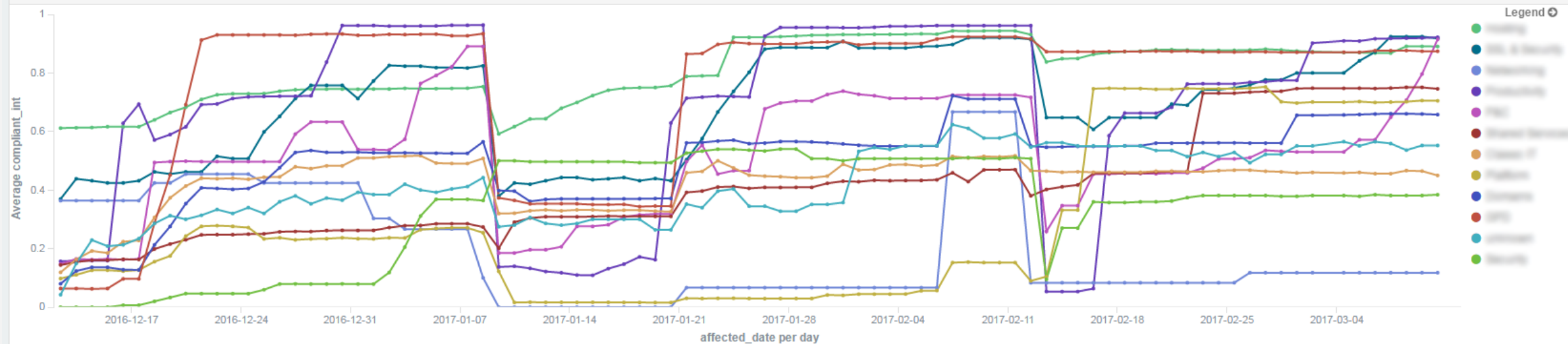
Patching: Product Lines Rollup - Since Patch Tuesday



Patching: Product Lines - Since Patch Tuesday



Patching: Product Lines Compliant Percentages





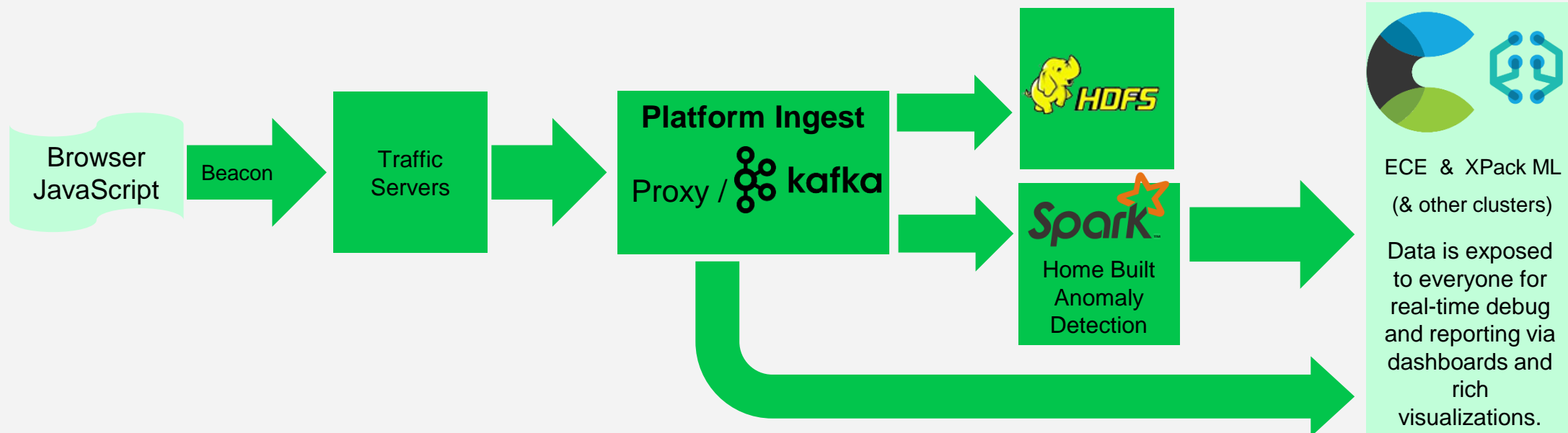
RUM / User Events

Q: Isn't that just GA?

RUM / User Events

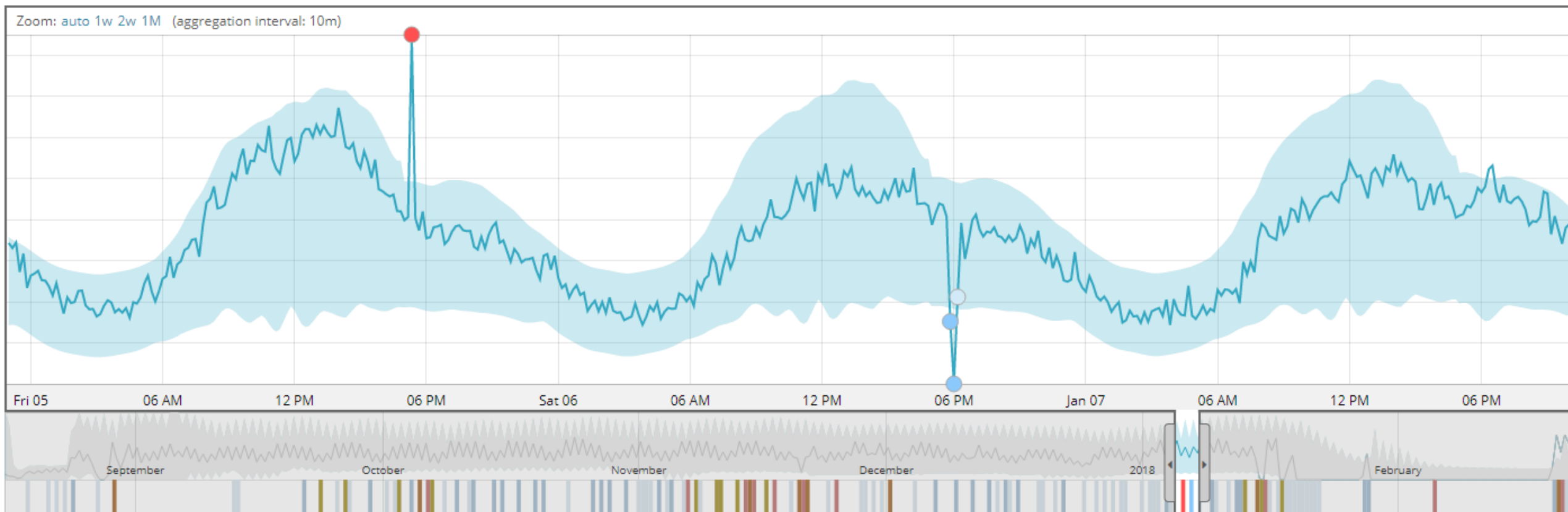
Our JS – Traffic2

- 100% fidelity clickstream / event data
- Ability for teams to act quickly on streamed data
- Ability to join data to other datasets – ie. network monitoring / flow
- Support for our split testing & personalization frameworks



User Events

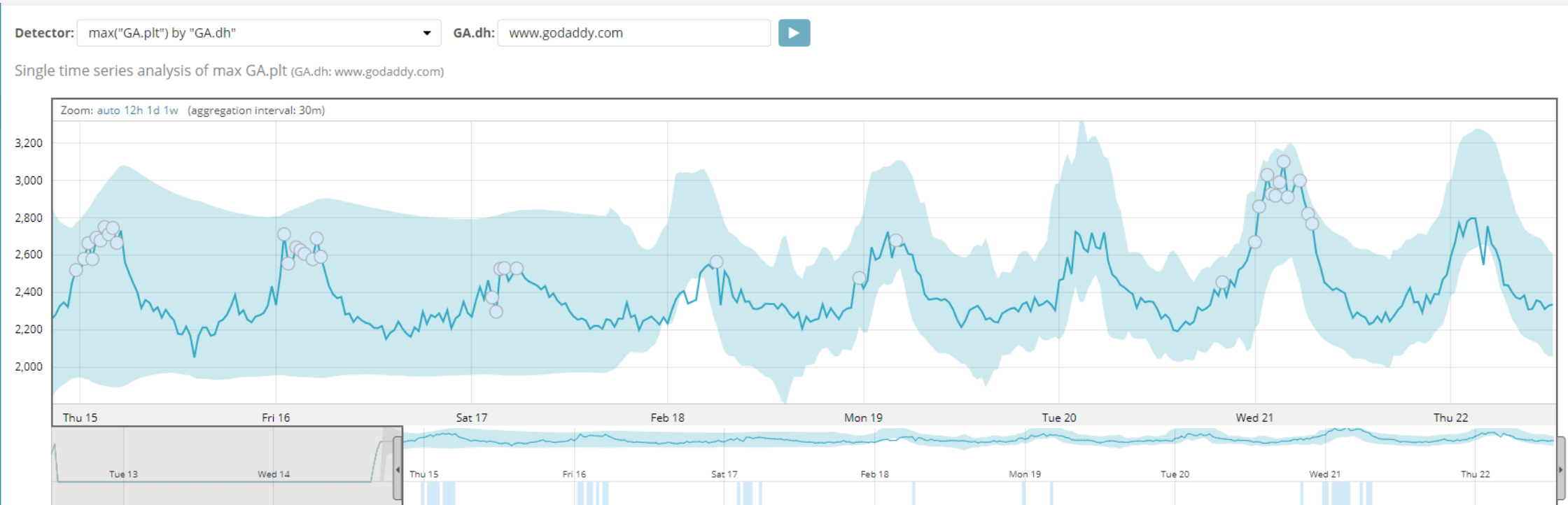
- Products track every aspect of customer interaction / lifecycle via Traffic2
- Recently started to analyze this via Elastic X-Pack ML



RUM - Facets / Findings

- Analyze by Source Geo → Datacenter → Page/Site
- 75th Percentile is most useful for ML on this dataset
- Top 1000 sites are interesting – but unique every hour/day
- We leverage Advanced ML job with aggregations:
 - date_histogram by 5m
 - terms agg top N
 - percentiles 75 for page load time (and other timings)

```
{
  "cts": "1519286437470",
  "dh": "woo.godaddysites.com",
  "dl": "woo.godaddysites.com/foo",
  "dp": "/foo",
  "ua": {
    "orig": "Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 ...",
    "browser": {
      "...",
    },
  },
  "os": {
    "name": "Windows 7",
    "...",
  },
},
"ds": "2000",
"ClientIp": "172.17.249.139",
"ClientIp_Geo": "...",
"plt": 2718,
"dns": 5,
"tcp": 195,
"srt": 178,
"pdt": 445,
"rrt": 730,
"dit": 1628,
"dct": 1629,
"@timestamp": "2018-02-22T08:00:25.844Z"
}
```



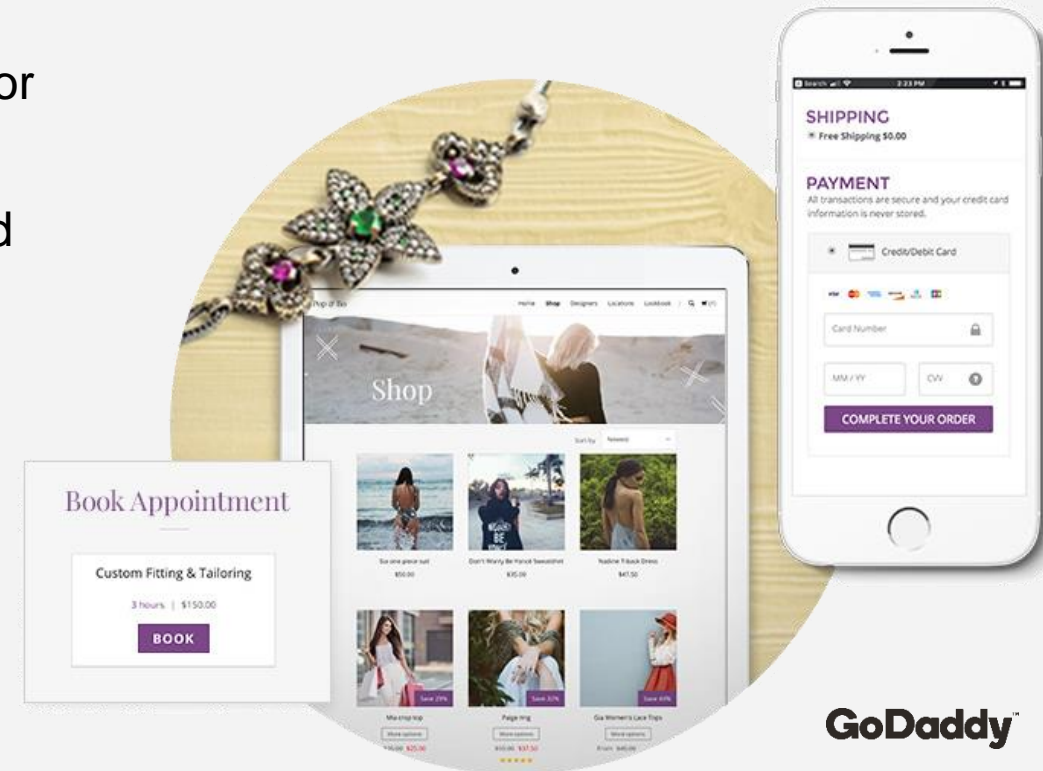
Blog Post:
<http://x.co/MLAgg>
by [Rich Collier](#)



Business KPIs

GoCentral KPIs

- GoCentral team moves extremely fast – 13,500 code/config deployments in 2017
 - *“If you don’t stop and look around once in awhile, you could miss it.” – Ferris Bueller*
- Free trial product so we analyze by cohorts
 - If I bought January 1st, on January 14th I’ll be in the 14 day cohort
- Business level KPIs are trailing indicators
 - Activate – when customer setup the product they signed up for
 - Publish – when customer launches their initial website
 - Conversion – when customer switches from Free Trial to Paid
 - Auto Renew – whether account has auto-conversion enabled



GoCentral KPIs

PySpark Approach:

```
## Build Dataset
```

```
1. df = df \  
2.     .withColumn('cohort_activate', \  
3.     F.when((F.datediff(df.activate_date, df.signup_date) <= cohort), 1).otherwise(0)) \  
4.     .withColumn('cohort_end_date', F.date_add(df.signup_date, cohort))
```

```
5. df2 = df.groupBy(df.cohort).agg({"cohort_activate" : "avg"})
```

```
## Ingest into ES
```

```
6. df2.write.format("org.elasticsearch.spark.sql").mode("append").save("index/type")
```

```
## Ask ML to process
```

```
7. start_payload = {"end": latest_available_date}  
8. requests.post(url + '/_xpack/ml/anomaly_detectors/' + job + '/_open')  
9. requests.post(url + '/_xpack/ml/datafeeds/datafeed-' + job + '/_start', json=start_payload)
```

* This is a code fragment

Create a new job

Job Details

Analysis Configuration

Datafeed

Edit JSON

Data Preview

bucket_span ⓘ

1d

summary_count_field_name ⓘ

record_count

categorization_field_name ⓘ

Detectors ⓘ

cohort_activate
mean(cohort_activate) by cohort



+ Add Detector

Influencers ⓘ

- ☒ cohort
- ☐ doc_id.keyword
- ☒ free_trial_signup_string.keyword
- ☒ record_count

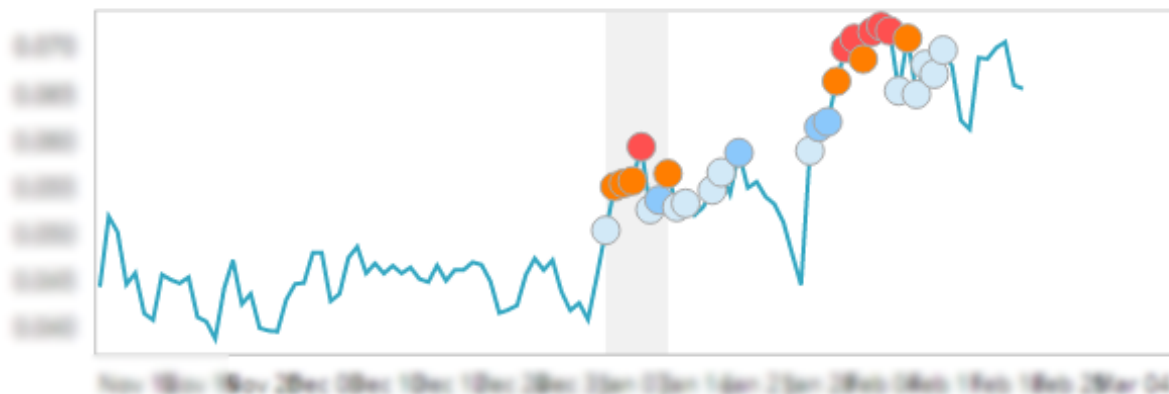
Custom influencer

+ Add

Anomalies

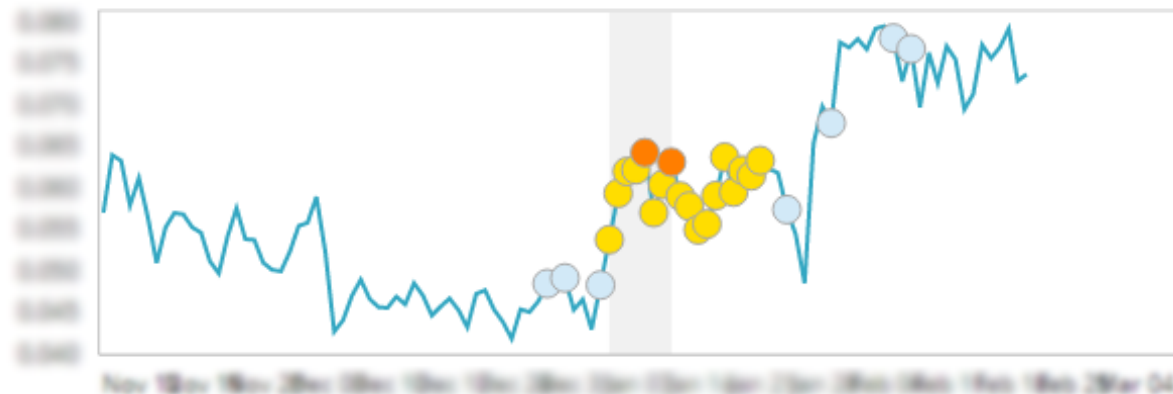
mean(cohort_conversion) by cohort - cohort 32 ⓘ

[View](#)



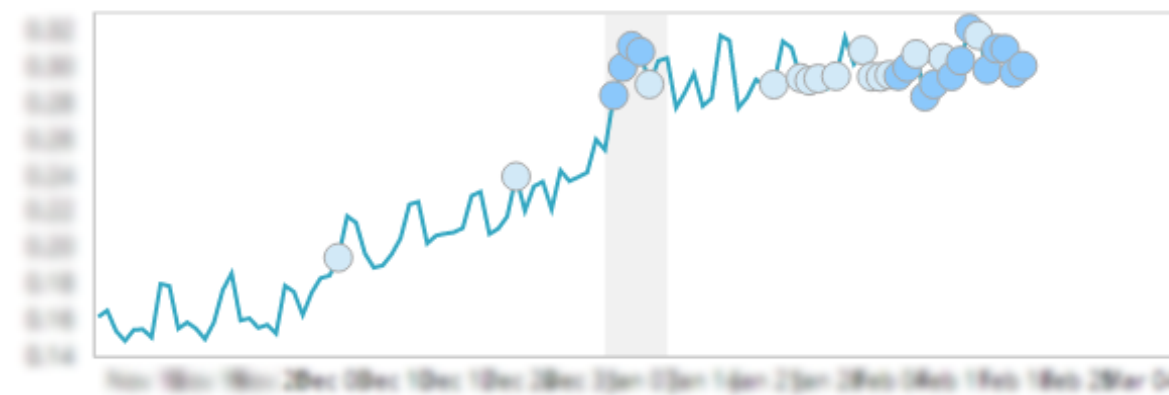
mean(cohort_auto_renew) by cohort - cohort 32 ⓘ

[View](#)



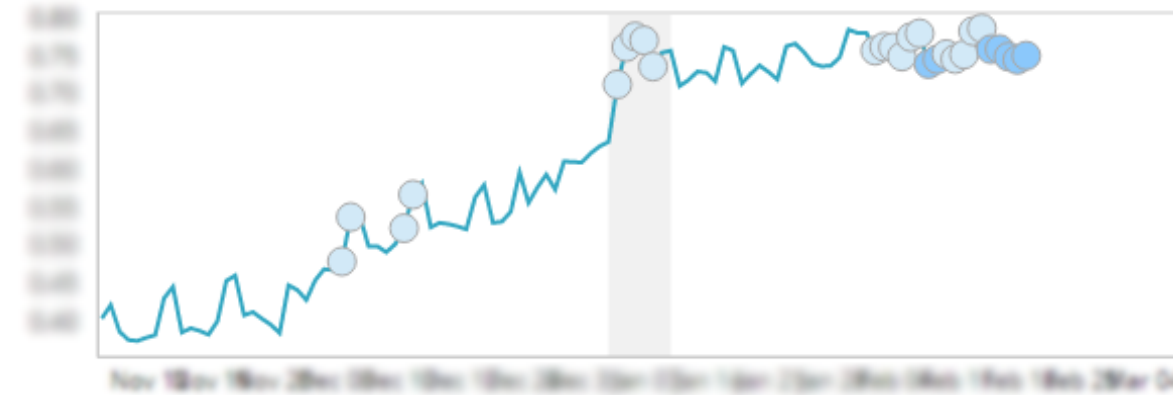
mean(cohort_publish) by cohort - cohort 32 ⓘ

[View](#)



mean(cohort_activate) by cohort - cohort 32 ⓘ

[View](#)



Severity threshold:

⚠ minor ▾

Interval:

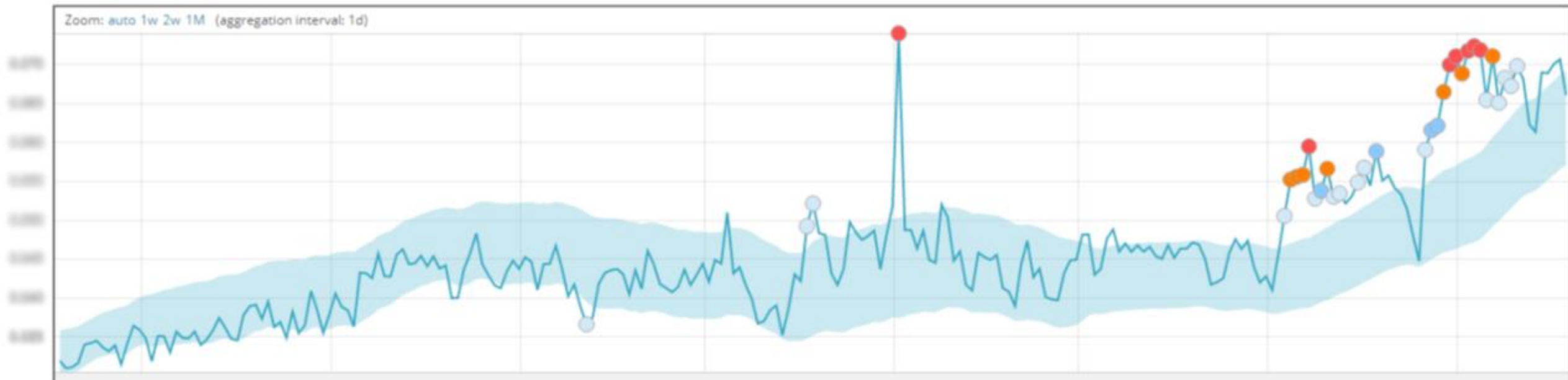
Auto ▾

time ▾	max severity ▾	detector ▾	found for ▾	influenced by ▾	actual ▾	typical ▾	description ▾	job ID ▾
▶ January 7th 2018	⚠ 89	mean(cohort_conversion) by cohort	32	cohort: 32 free_trial_signup_string.keyword: 2017-12-07 record_count: 10000	10000	10000	↑ 1.4x higher	gocentral-kpi-bycohort-aggr-cor
▶ January 4th 2018	⚠ 66	mean(cohort_conversion) by cohort	32	cohort: 32 free_trial_signup_string.keyword: 2017-12-04 record_count: 10000	10000	10000	↑ 1.3x higher	gocentral-kpi-bycohort-aggr-cor

GoCentral KPIs

Detector: cohort: 

Single time series analysis of avg cohort_conversion (cohort: 32)



- Model Plot FTW!

Key Learnings



Hardest part is your data

Bulk of projects was spent figuring out what data was actionable versus vanity and formatting to take best advantage of ML.



Business likes model plots

The visualization with a model-plot is extremely convincing / powerful. Use it where you can afford.

* Advanced jobs require JSON config!



Alerting / Watcher is Hard

Plan to spend time on watcher configuration, especially for advanced notification. Its getting better, but still more to do.



Make data ingest idempotent

Leverage custom document `_id` field so you can reload same data easily.

(Applies to Hive & other data stores - eg. ability to overwrite a full partition)



Advanced jobs are your friend

We found ourselves running most ML workloads on Advanced jobs due to their power in configuring and enabling model plot (see below).



Wait for updates to bake

Good practice for any prod workload, but ECE is new and has more moving pieces.

Build a dev cluster and upgrade at-will for new features – especially in ML!



You will try & retry jobs

Tooling is powerful, but figuring out the right mix of detectors, influencers, etc is dataset specific. Set aside a sprint or two for this.



Have a strategy for data gaps

There are times real-time data will stop flowing. Have a strategy for how you will deal with gaps.



Leverage the Elastic team

We've had an incredible relationship with the Elastic team.

But wait, there's more!

Replaced our SIEM with Elastic + Hadoop

Monitoring & Alert Correlation / Analysis

Team Metrics: CI/CD Pipelines / Code Health / Etc.

Fleet-wide Meltdown / Spectre Analysis

We can't wait for...



Elastic APM



Guess what We're hiring!

x.co/jobplz | godaddy.com/jobs

Arizona, California (SD, LA, SF, Sunnyvale), Iowa, Massachusetts, Washington, and more!

Questions?

Felix Gorodishter
@fgorodishter
felix@godaddy.com