

Scalable Monitoring with Apache Spark & friends





- Senior Software Engineer @ Tinder.
- Built monitoring pipeline at **Sony PlayStation**.
- An active contributor to **Grafana**.
- Speaker @ **GrafanaCon 2016**.

GitHub - <https://github.com/utkarshcmu>
Email – utkarsh.cmu@gmail.com



Monitoring

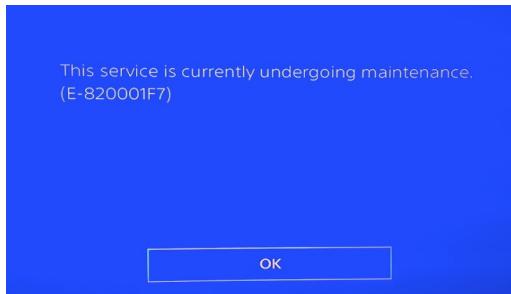
@



PlayStation™



PlayStation Outage!



Problems at Playstation Network

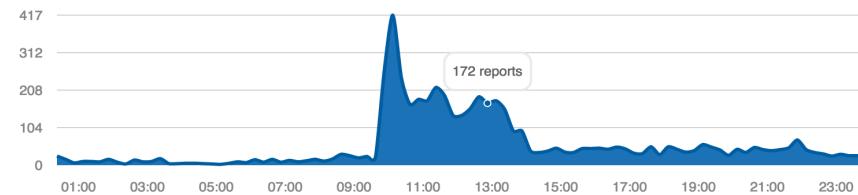
Published: 10/09/2015 3:22 p.m. By: downdetector.com

Playstation Network is having issues since 3:22 PM EDT. Are you also affected? Leave a message in the comments.

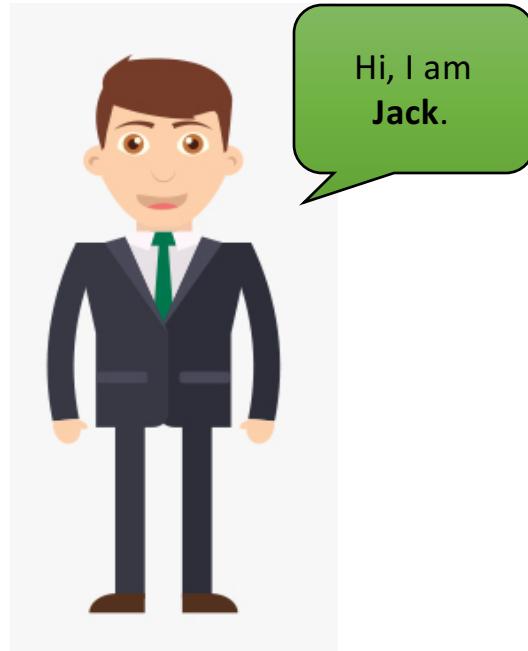
Most reported problems:

- Sign-in (67%)
- Game play (16%)
- Playstation Store (15%)

Oct. 9, 2015 Status overview



Sometime 3 years back...



POC on Monitoring



Requirements:

- 50,000 unique metrics from one source
- Data points every minute
- Roughly about **72 million data points per day**
- Data retention 60 days
- User friendly UI with possible customization



Monitoring Stack



Choosing the technology!

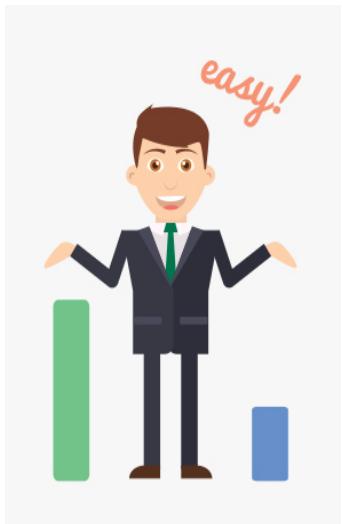


POC

Design & Architecture



POC Completed!



Mission accomplished!

1 metrics source
50,000 unique metrics
72 million data points per day



Metrics Onboarding



Team 1 Requirements:

- 100,000 unique metrics
- **About 200 million data points per day**

Team 2 Requirements:

- 400,000 unique metrics
- **About 600 million data points per day**

Team 3 Requirements:

- 500,000 unique metrics
- **About 2 billion data points per day**

Team 4 Requirements:

- 800,000 unique metrics
- **About 5 billion data points per day**

And more.....



POC

Design & Architecture



How to Scale?



Challenges:

- Multiple teams
- Millions of unique metrics
- Above 10 billion data points a day
- Process 3 million logs every minute and generate metrics
- Reprocessing of metrics and logs if needed
- **Provide real time monitoring for all of the above using GRAFANA!**

Should he continue with Graphite?
Should he ask to reduce metrics or datapoints?
How to dynamically scale Graphite?
Does Grafana support other datasources?
OpenTSDB / InfluxDB / KairosDB / Prometheus?
Support scaling Infrastructure to support variable load of metrics?



Strategy



Team 1 Requirements:

- 100,000 unique metrics
- About 200 million data points per day

Team 2 Requirements:

- 500,000 unique metrics
- About 2 billion data points per day

Team 3 Requirements:

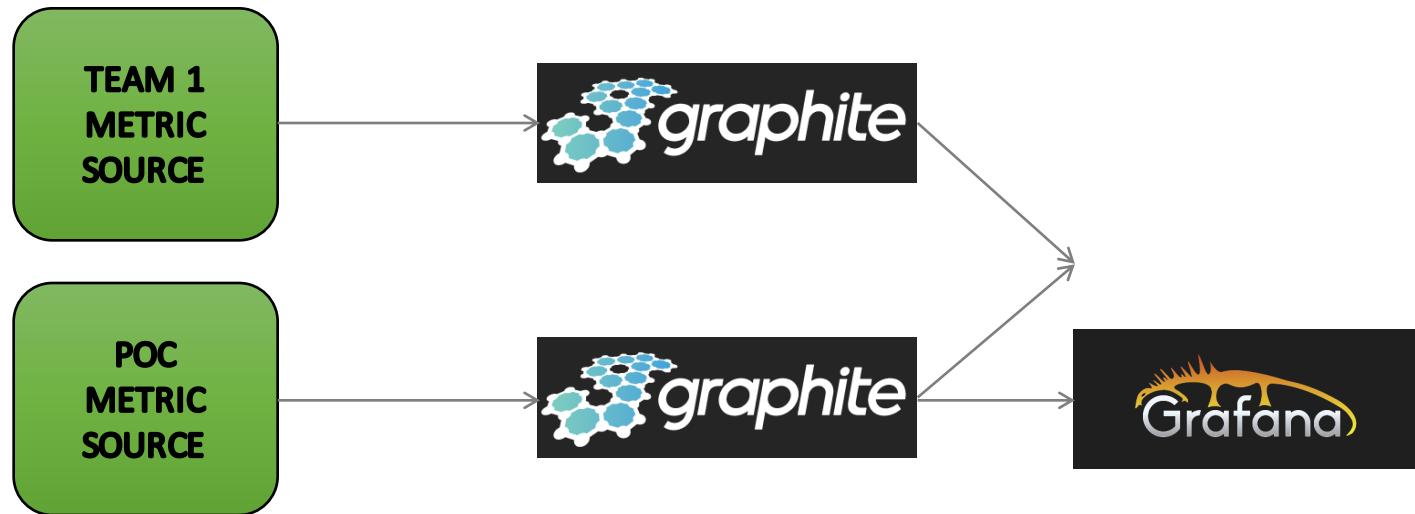
- 3 million logs a minute
- Generate metrics in real time

And more.....

Divide & Conquer



Design & Architecture

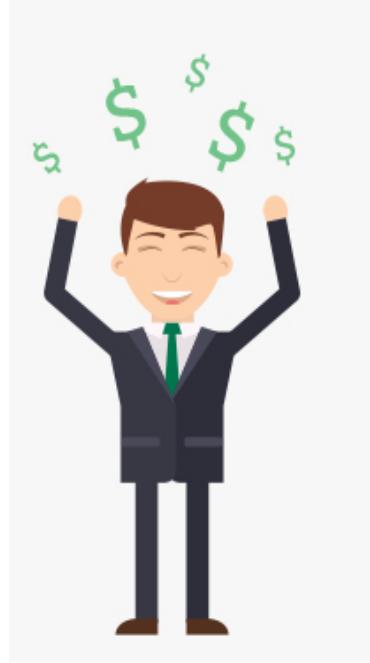


POC works for: Team 1 requirements:

1 metrics source	1 metrics source
50,000 unique metrics	100,000 unique metrics
72 million data points per day	200 million data points per day



Team 1 Conquered!



This strategy works! Bring it on!



Strategy



Team 1 Requirements:

- 100,000 unique metrics
- About 200 million data points per day

Team 2 Requirements:

- 500,000 unique metrics
- About 2 billion data points per day

Team 3 Requirements:

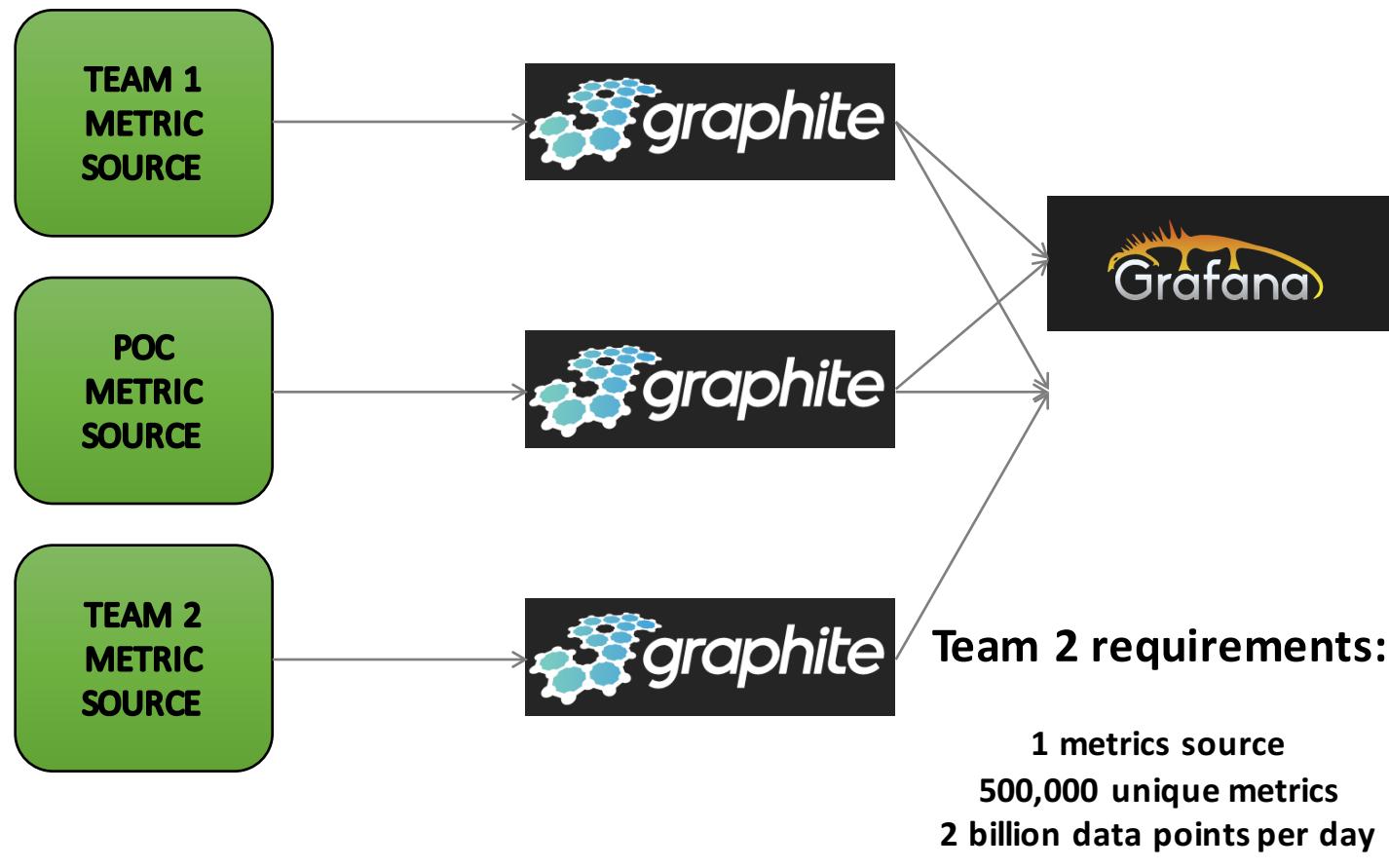
- 3 million logs a minute
- Generate metrics in real time

And more.....

Divide & Conquer



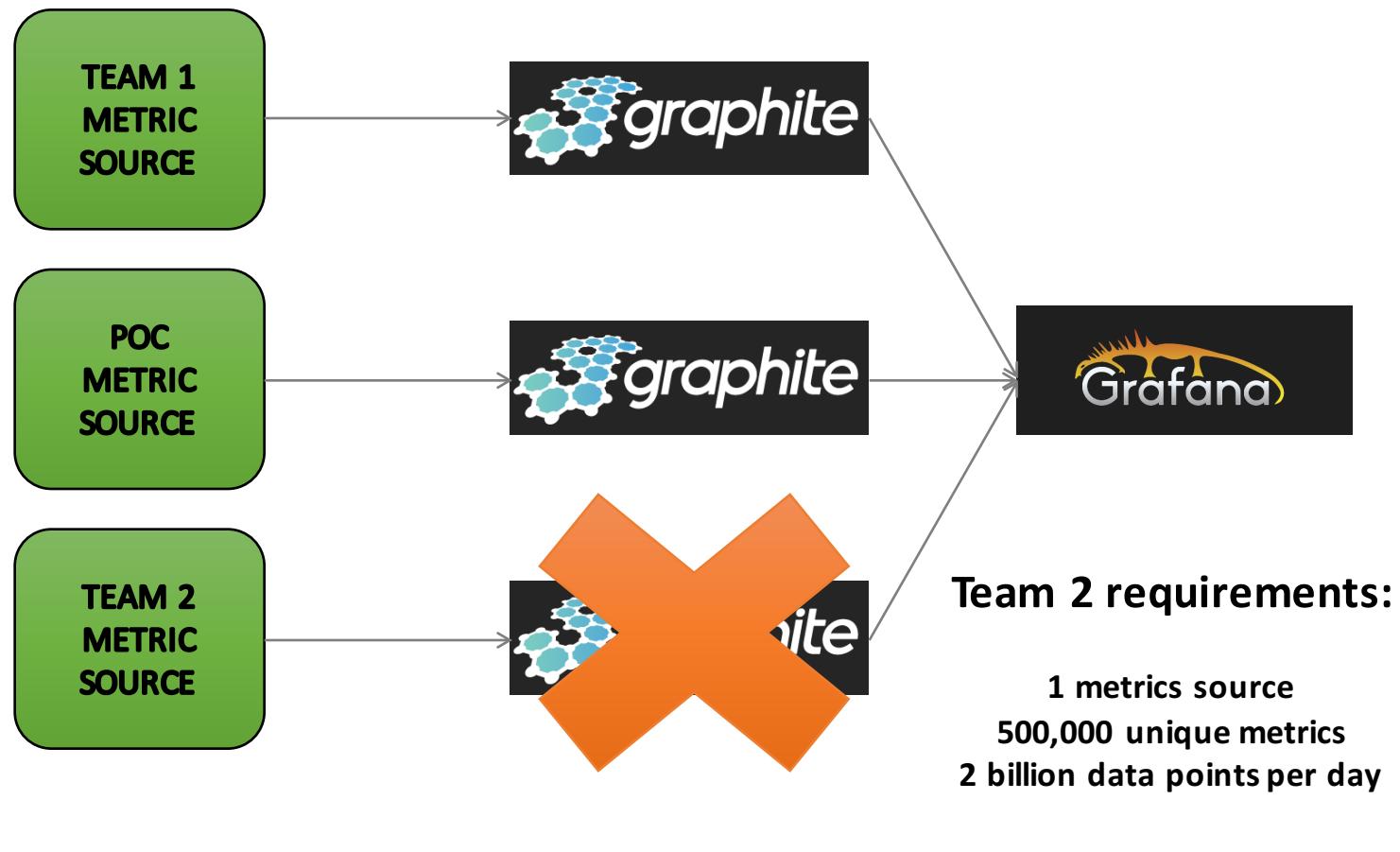
Design & Architecture

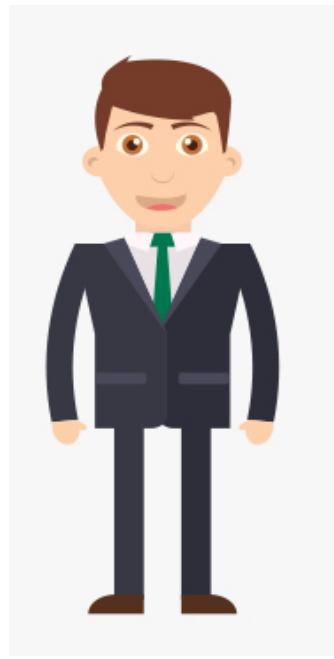


Team 2 Conquered!

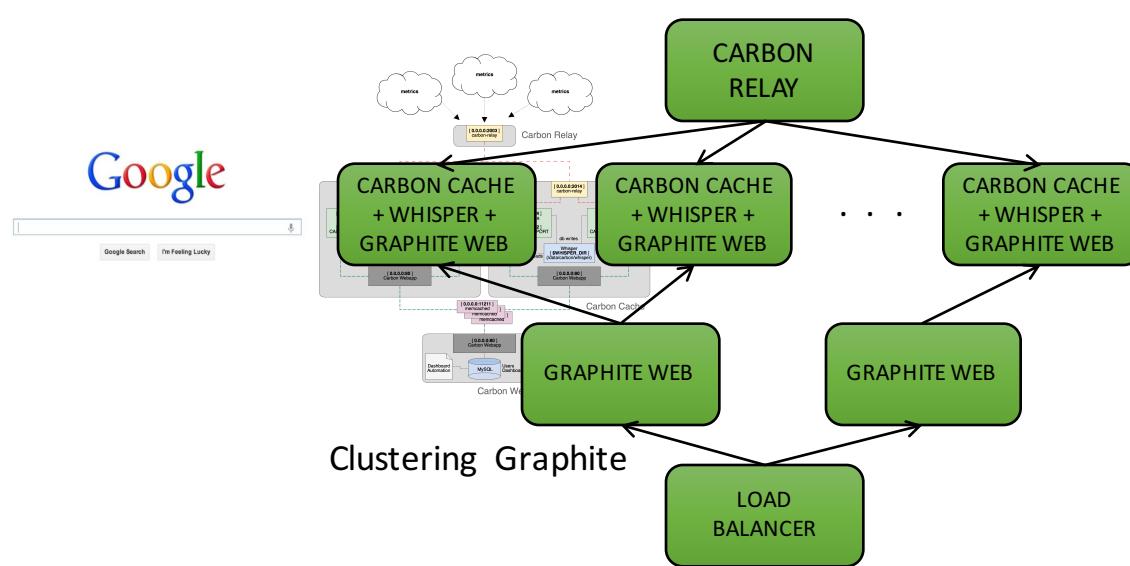


Design & Architecture

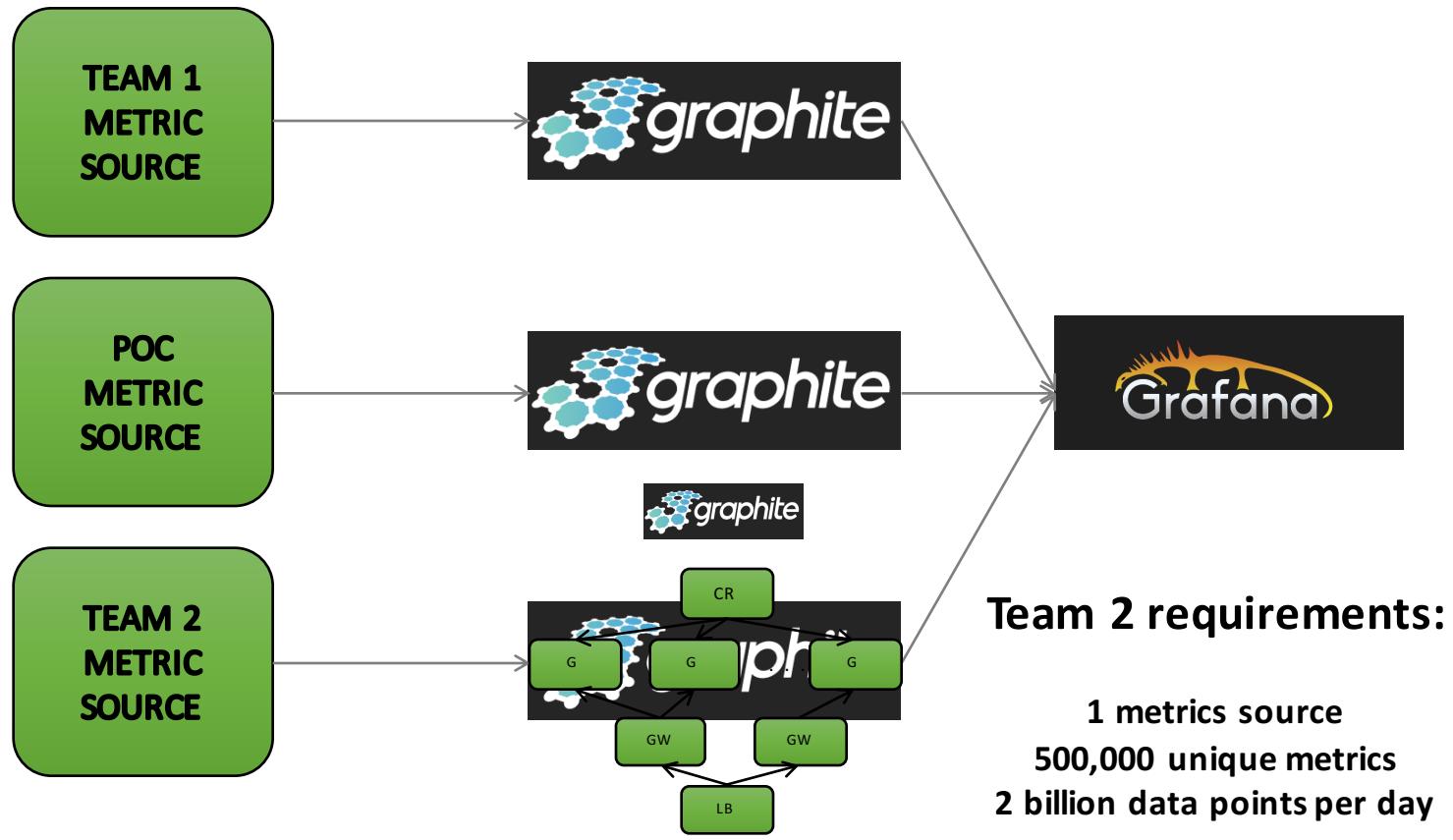




Scaling Graphite



Design & Architecture



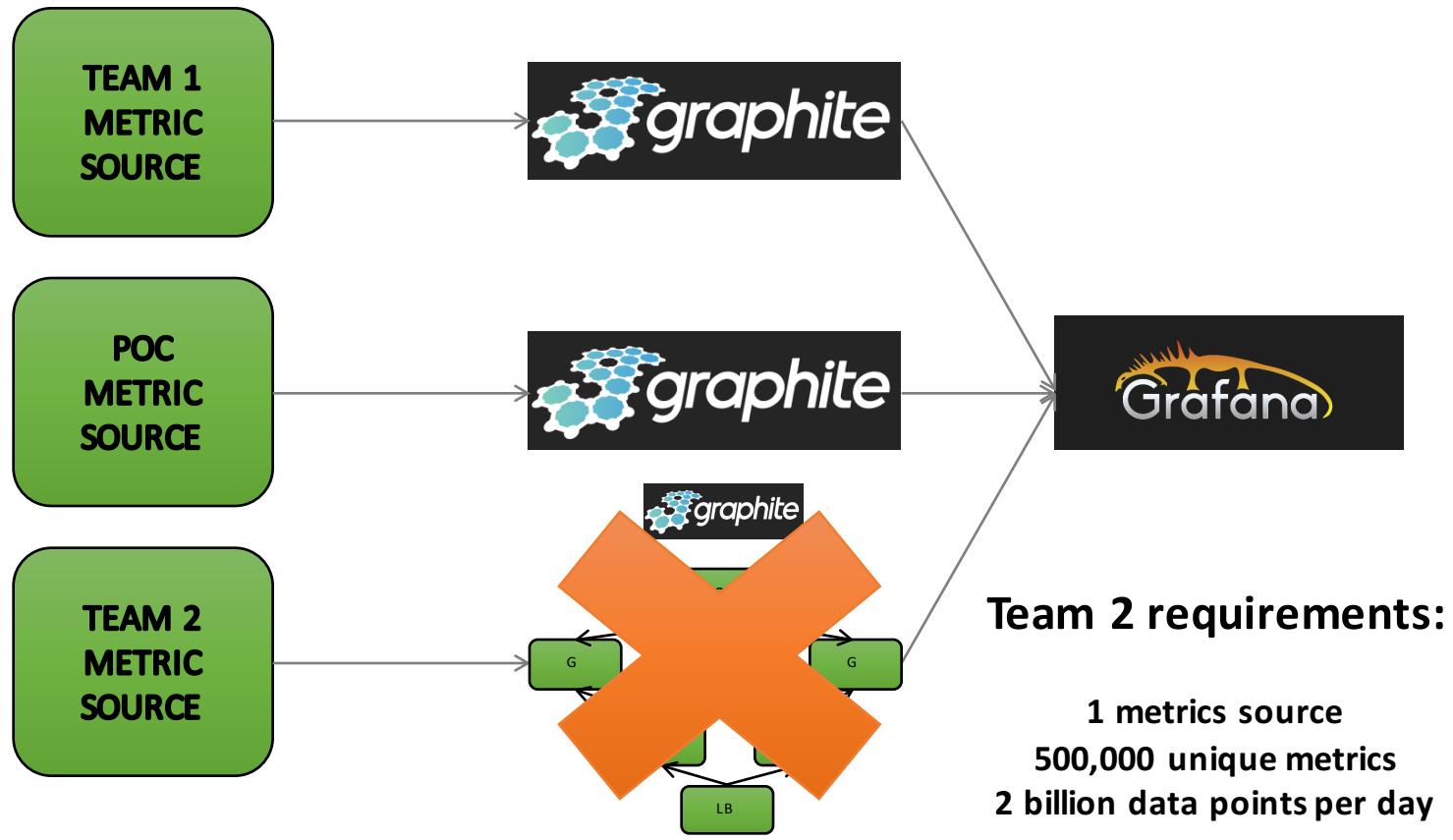
Team 2 Conquered!



But..... Happiness lasted only for a month ☹



Design & Architecture



Scalable Alternatives To Graphite



Update on InfluxDB Clustering, High-Availability and Monetization

Paul Dix — March 10, 2016

[Redacted content block]

Update: Since I wrote this post we've delivered on the things I've promised.

- We have continued to improve our open source platform with [88 new features and 133 bug fixes](#) to InfluxDB. This includes performance enhancements and all new query functionality like Holt-Winters, moving averages, and killing long running queries.
- We created [influx-relay](#) as a pure open source option for high-availability setups.
- We released managed clusters of InfluxDB on [InfluxCloud](#) on April 19th.
- On September 8th we made an affordable [on-premise InfluxEnterprise](#) offering at the promised price of \$399/month.

"How does InfluxData make money?" That's a question I've been asked many times over the course of the last year or so. I've never had a good answer, so I'm going to try and do better.

TSDB

A cartoon illustration of a character with brown hair, wearing a blue t-shirt with a large white "KDB" logo, green cargo pants, and a blue baseball cap. The character is standing next to a large, stylized blue "KDB" logo. To the right of the character, there is a white rounded rectangle containing text and another smaller illustration of a character.

Recent Posts

[TL;DR InfluxDB Tech Tip](#)
December 01, 2016 — Dr. 1st, 2016

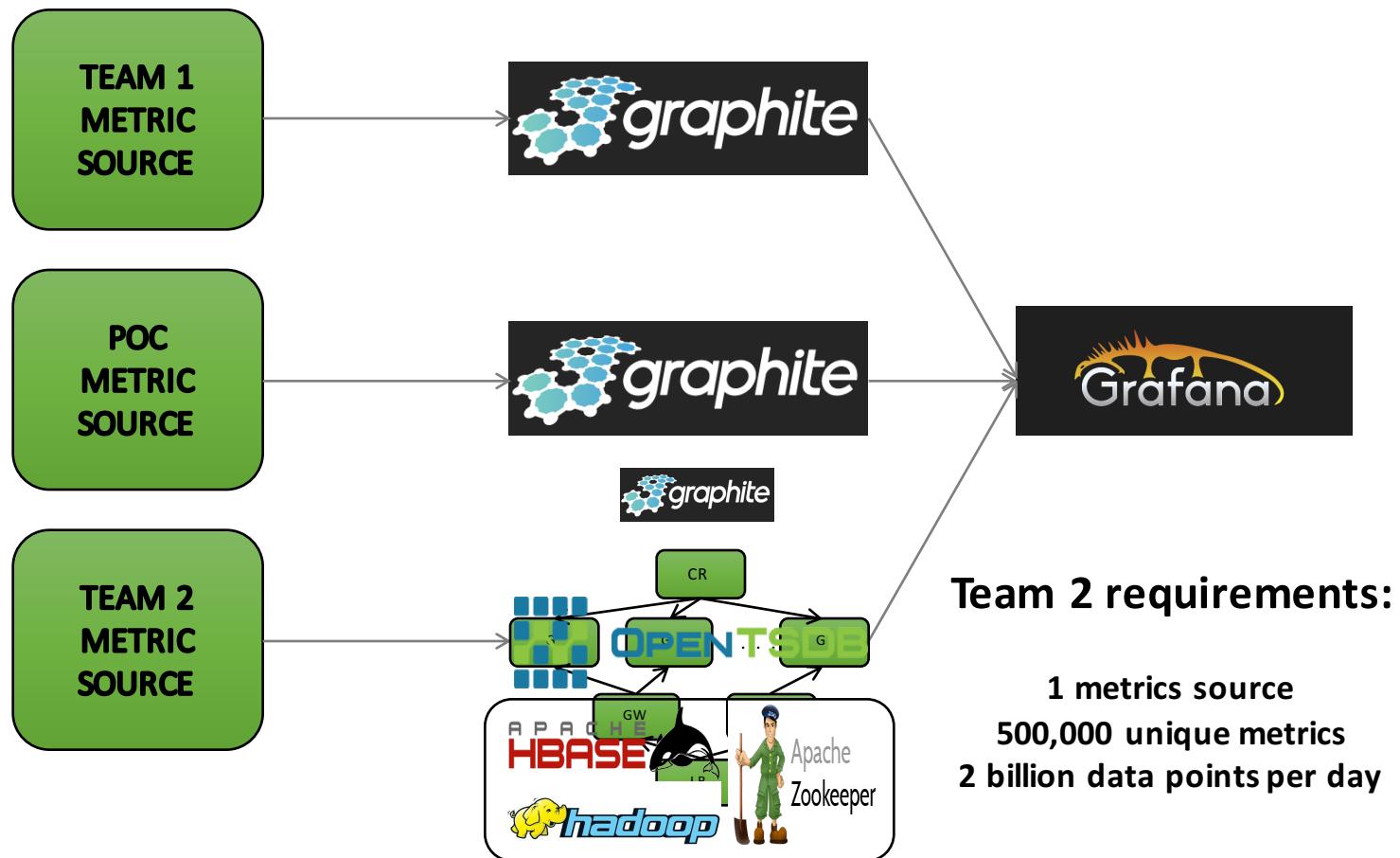
[How to spin up the TICK in a Kubernetes instance](#)
November 30th, 2016

[InfluxDB Week in Review](#)
28, 2016 — November 21, 2016

[TL;DR InfluxDB Tech Tip](#)



Design & Architecture



Team 2 Conquered!



Finally!



Strategy



Team 1 Requirements:

- 100,000 unique metrics
- About 200 million data points per day

Team 2 Requirements:

- 500,000 unique metrics
- About 2 billion data points per day

Team 3 Requirements:

- 3 million logs a minute
- Generate metrics in real time

And more.....

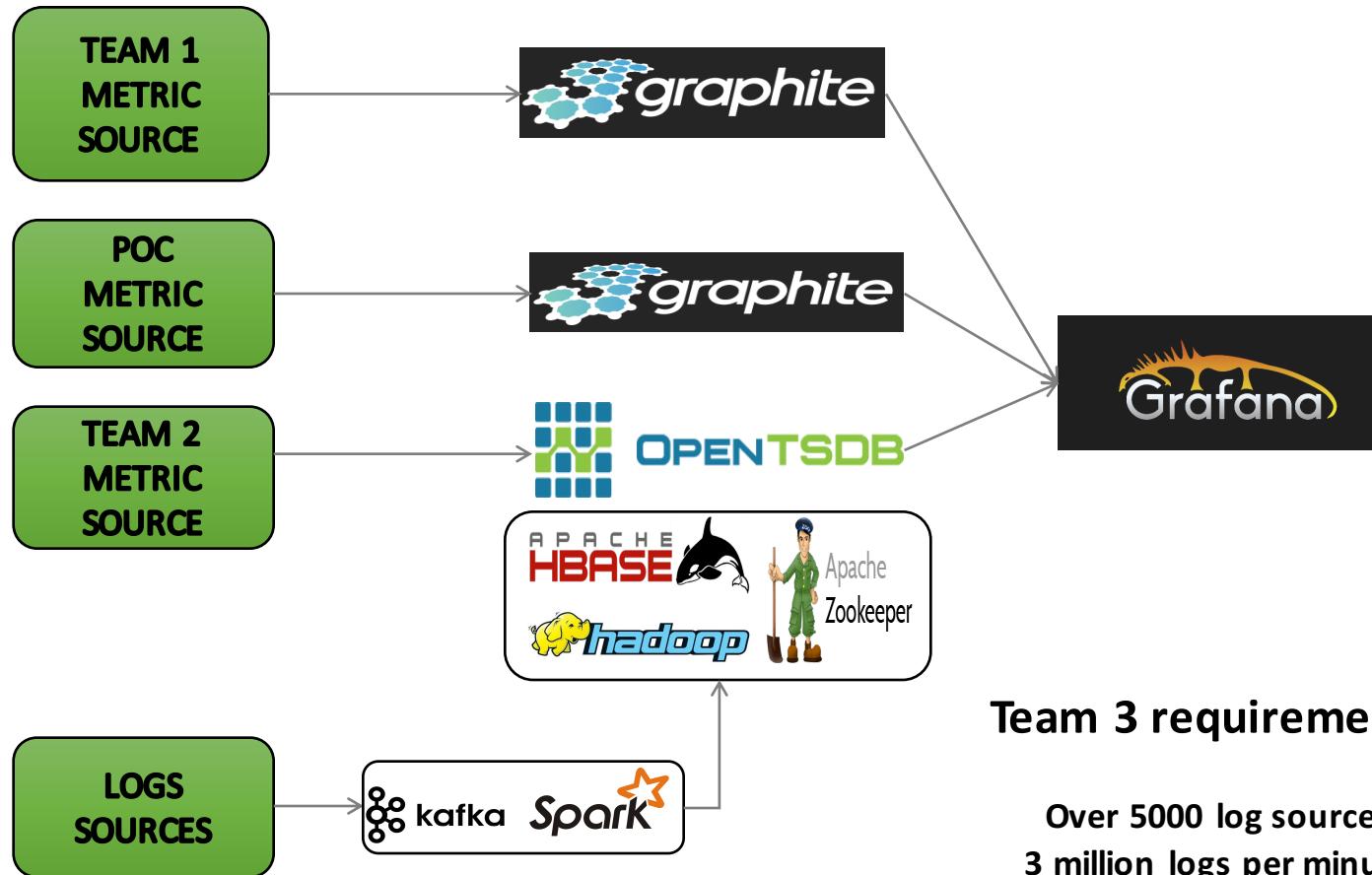
Divide & Conquer



How to process logs at scale?



Design & Architecture



Team 3 requirements:

Over 5000 log sources
3 million logs per minute



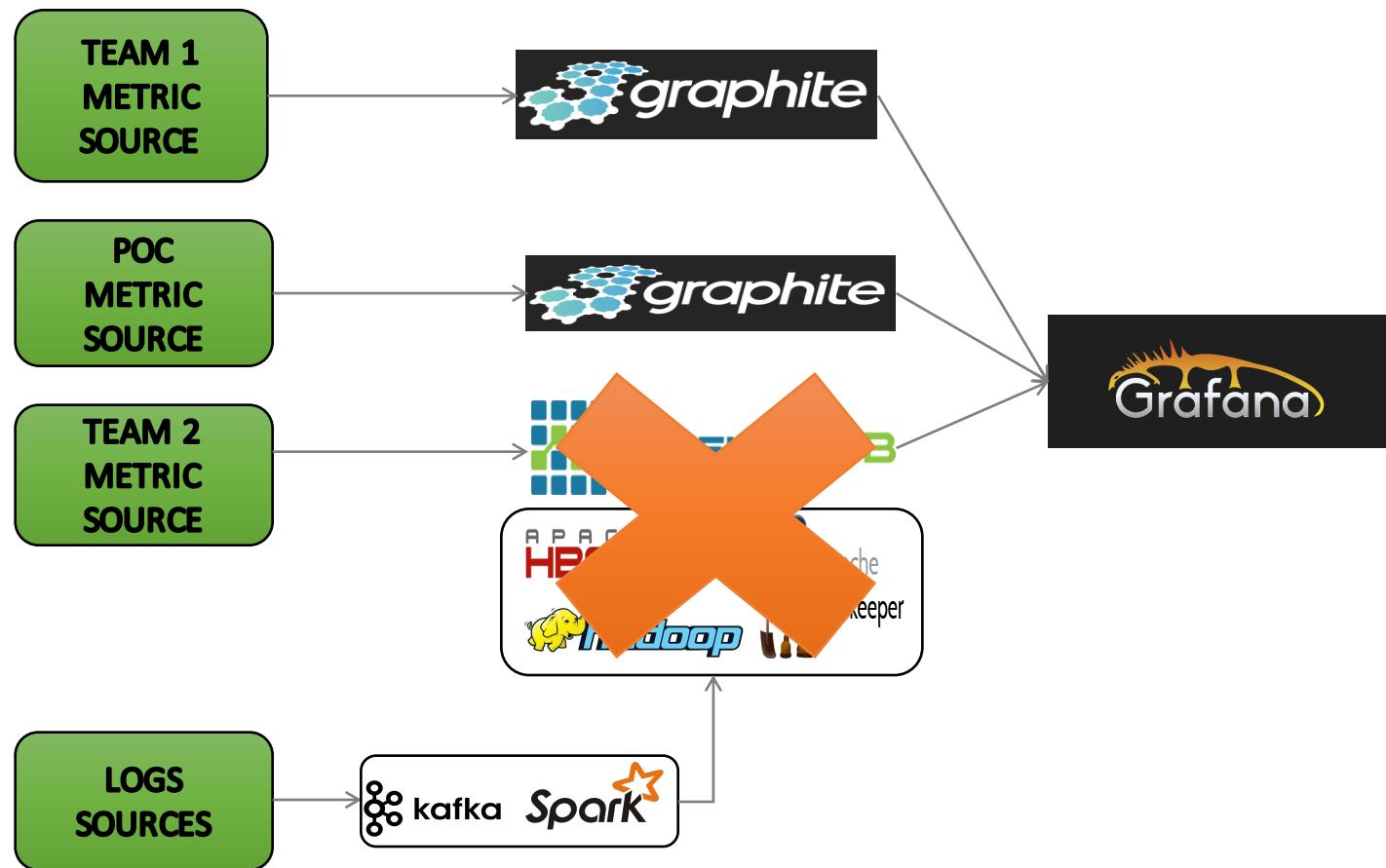
Team 3 Conquered!



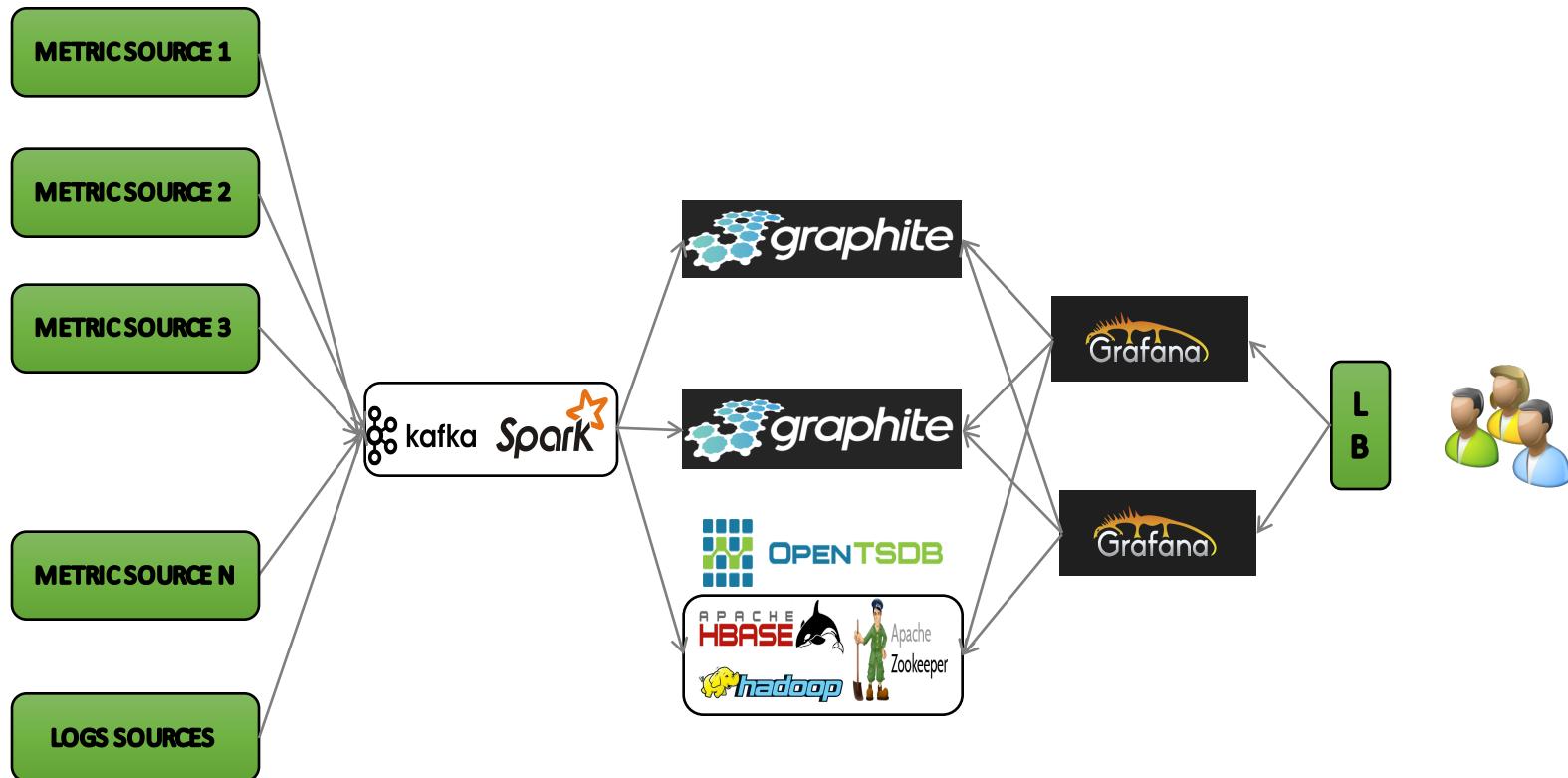
But One day..



Design & Architecture



Design & Architecture



Metrics & Logs Sources



Too many sources!

All different formats!

More Spark Streaming jobs!



Writing a Spark Streaming job!

Spark Documentation

```
public final class JavaDirectKafkaWordCount {
    private static final Pattern SPACE = Pattern.compile(" ");

    public static void main(String[] args) throws Exception {
        if (args.length < 2) {
            System.err.println("Usage: JavaDirectKafkaWordCount <brokers> <topics>\n" +
                "  <brokers> is a list of one or more Kafka brokers\n" +
                "  <topics> is a list of one or more kafka topics to consume from\n");
            System.exit(1);
        }

        StreamingExamples.setStreamingLogLevels();

        String brokers = args[0];
        String topics = args[1];

        // Create context with a 2 seconds batch interval
        SparkConf sparkConf = new SparkConf().setAppName("JavaDirectKafkaWordCount");
        JavaStreamingContext jssc = new JavaStreamingContext(sparkConf, Durations.seconds(2));

        Set<String> topicsSet = new HashSet<>(Arrays.asList(topics.split(",")));
        Map<String, String> kafkaParams = new HashMap<>();
        kafkaParams.put("metadata.broker.list", brokers);

        // Create direct kafka stream with brokers and topics
        JavaPairInputDStream<String, String> messages = KafkaUtils.createDirectStream(
            jssc,
            String.class,
            String.class,
            StringDecoder.class,
            StringDecoder.class,
            kafkaParams,
            topicsSet
        );

        // Get the lines, split them into words, count the words and print
        JavaDStream<String> lines = messages.map(Tuple2::_2);
        JavaDStream<String> words = lines.flatMap(x -> Arrays.asList(SPACE.split(x)).iterator());
        JavaPairDStream<String, Integer> wordCounts = words.mapToPair(s -> new Tuple2<>(s, 1));
        wordCounts.reduceByKey((i1, i2) -> i1 + i2);
        wordCounts.print();

        // Start the computation
        jssc.start();
        jssc.awaitTermination();
    }
}
```

Lazer – Wrapper Library for Streaming

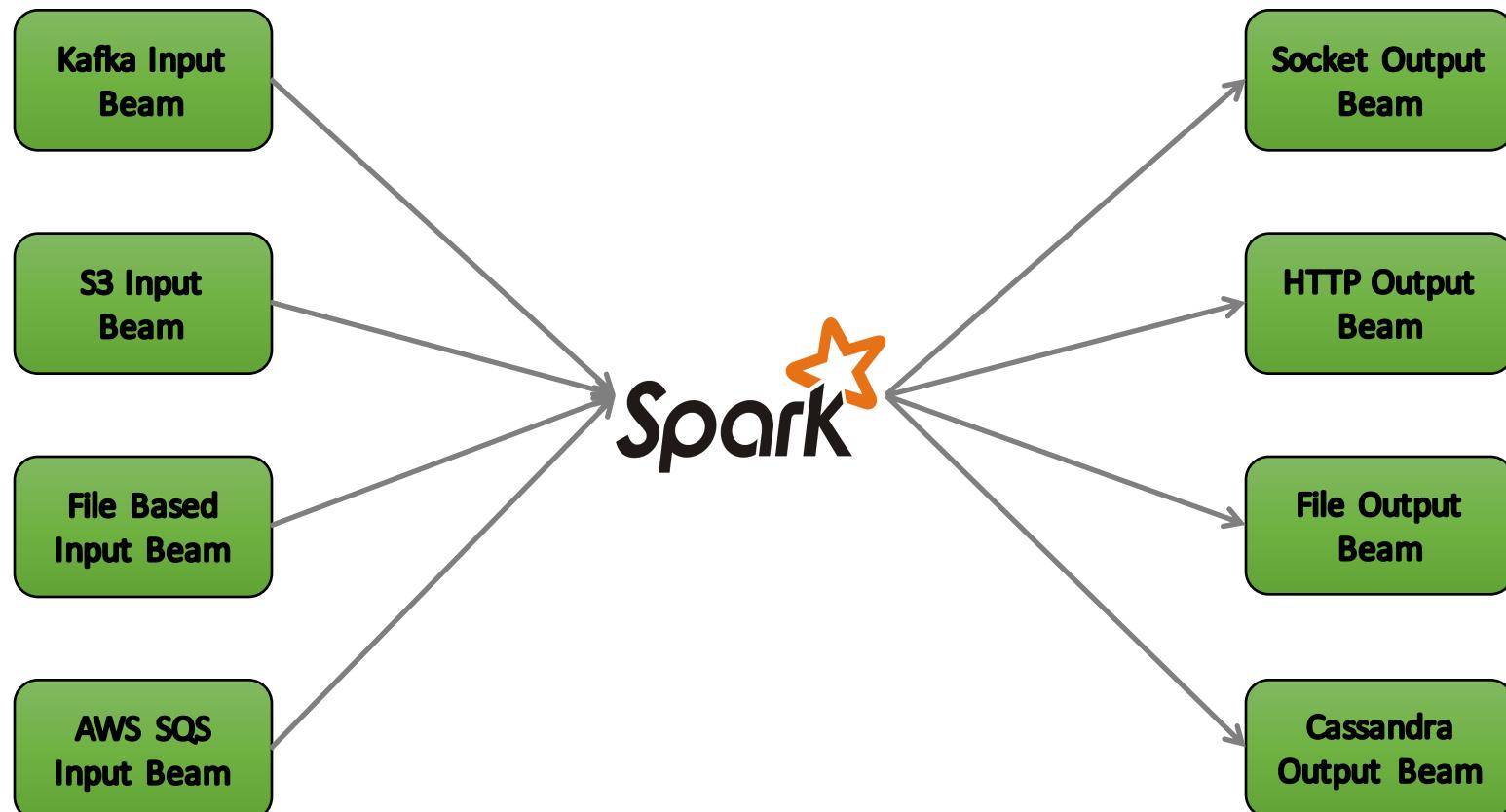
```
public class JavaDirectKafkaWordCount {

    private static final Pattern SPACE = Pattern.compile(" ");

    public static void main(String args[]) {
        Lazer laser = new Lazer("exampleBeams/events.json");
        InputBeam[] inputBeams = laser.getInputBeams();
        JavaDStream<String> lines = inputBeams[0].read();
        JavaDStream<String> words = lines.flatMap(x -> Arrays.asList(SPACE.split(x)).iterator());
        JavaPairDStream<String, Integer> wordCount = words.mapToPair(s -> new Tuple2<>(s, 1)).reduceByKey((i1, i2) -> i1 + i2);
        wordCount.print();
        laser.startAndAwaitTermination();
    }
}
```



Lazer – Wrapper library for Streaming



Lazer – Wrapper library for Streaming

```
events.json
```

```
1 {  
2   "beam.settings": {  
3     "name": "KafkaLogCountExample",  
4     "duration": 10  
5   },  
6   "spark.settings": {  
7     "spark.streaming.backpressure.enabled": true,  
8     "spark.executor.memory": "4g"  
9   },  
10  "input.settings": [  
11    {  
12      "class": "io.github.utkarshcmu.input.KafkaInputBeam",  
13      "config": {  
14        "topics": [  
15          "topicA",  
16          "topicB"  
17        ],  
18        "params": {  
19          "metadata.broker.list": "kafka:9092",  
20          "group.id": "consumerGroupName",  
21          "auto.offset.reset": "lastest"  
22        }  
23      }  
24    }  
25  ],  
26  "output.settings": [  
27    {  
28      "class": "io.github.utkarshcmu.output.SocketOutputBeam",  
29      "config": {  
30        "host": "graphite",  
31        "port": 2003  
32      }  
33    }  
34  ]  
35 }
```

```
public class JavaDirectKafkaWordCount {  
  
    private static final Pattern SPACE = Pattern.compile(" ");  
  
    public static void main(String args[]) {  
        Lazer laser = new Lazer("exampleBeams/events.json");  
        InputBeam[] inputBeams = laser.getInputBeams();  
        JavaDStream<String> lines = inputBeams[0].read();  
        JavaDStream<String> words = lines.flatMap(x -> Arrays.asList(SPACE.split(x)).iterator());  
        JavaPairDStream<String, Integer> wordCount = words.mapToPair(s -> new Tuple2<s, 1>).reduceByKey((i1, i2) -> i1 + i2);  
        wordCount.print();  
        laser.startAndAwaitTermination();  
    }  
}
```



Lazer – Open sourced now!



<https://github.com/utkarshcmu/lazer>



Some numbers

- More than **3 million unique metrics** supported
 - creation and deletion happens all the time
- More than **11 billion data points** written per day
 - across all TSDBs
- Processing about **40 billion events** per day
 - logs and metrics events in near real time (within 30 seconds)
- More than 3000 requests per minute to Grafana dashboards
 - around **7000 requests** in during outages



Lessons Learned



Strategy



Divide & Conquer



Look for alternatives!



Choose scalable components!



(Subject to effort and time)



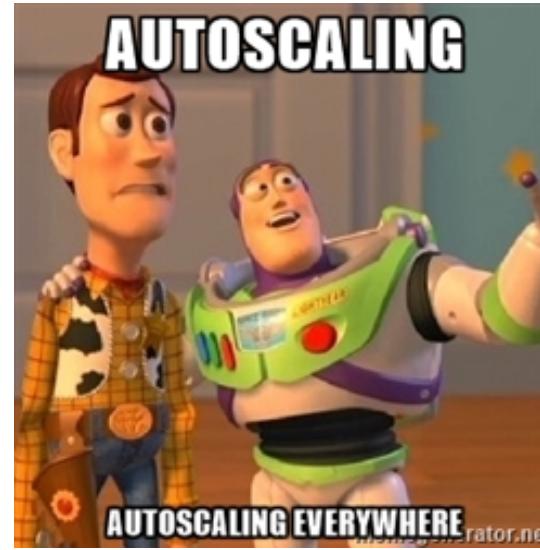
Automation



ANSIBLE

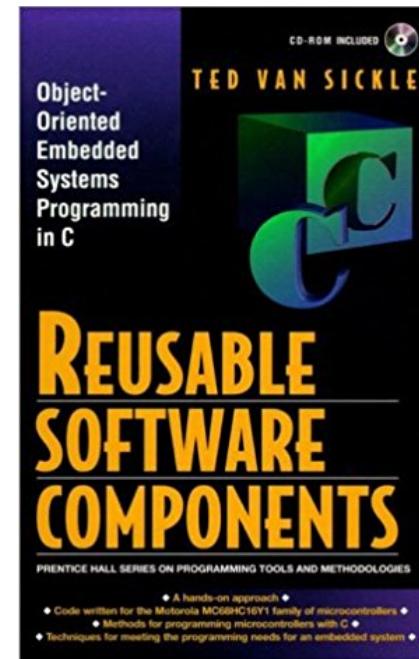


CHEF™

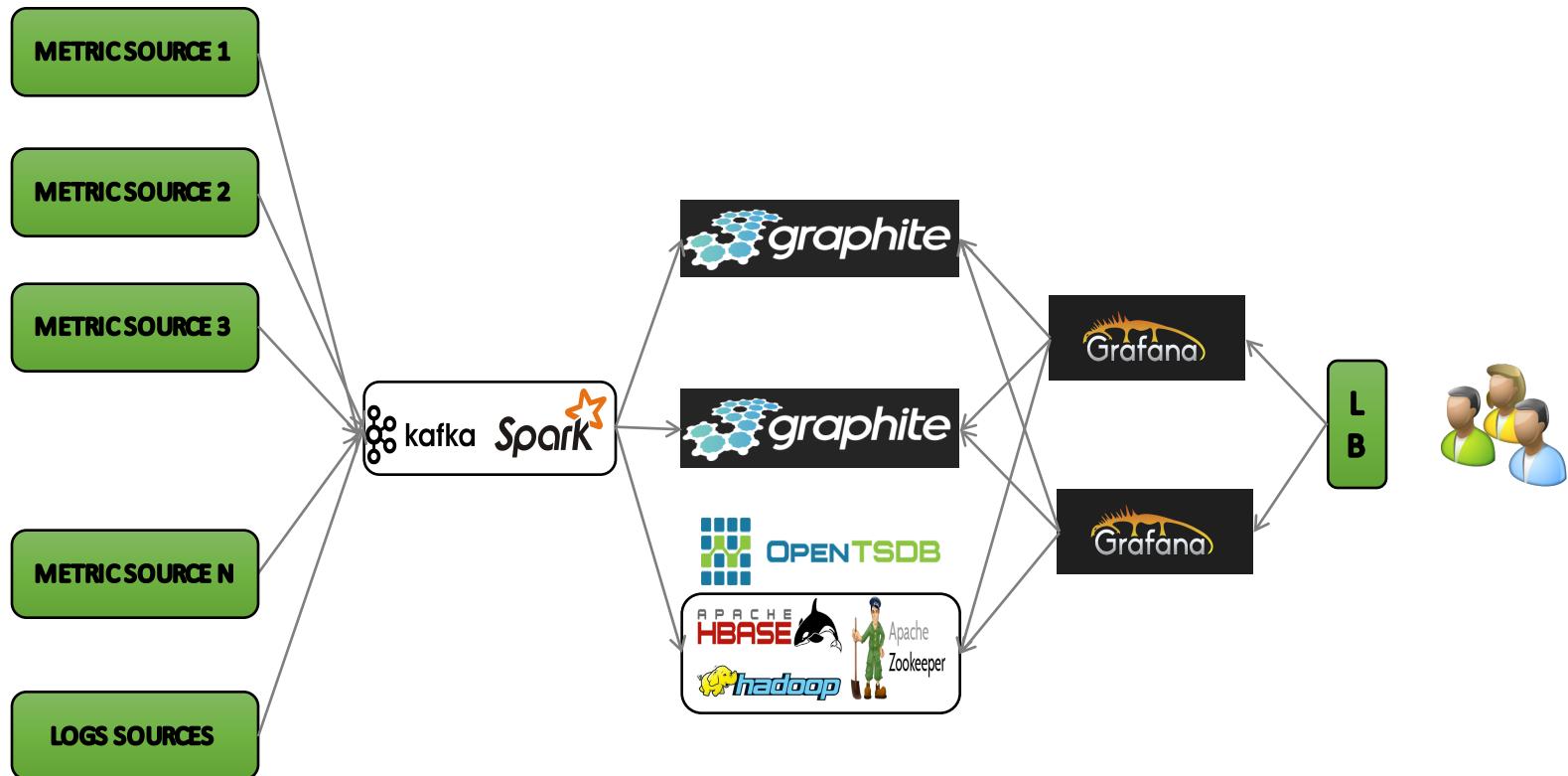


Re-use components

- Libraries
- SDKs
- Common artifacts

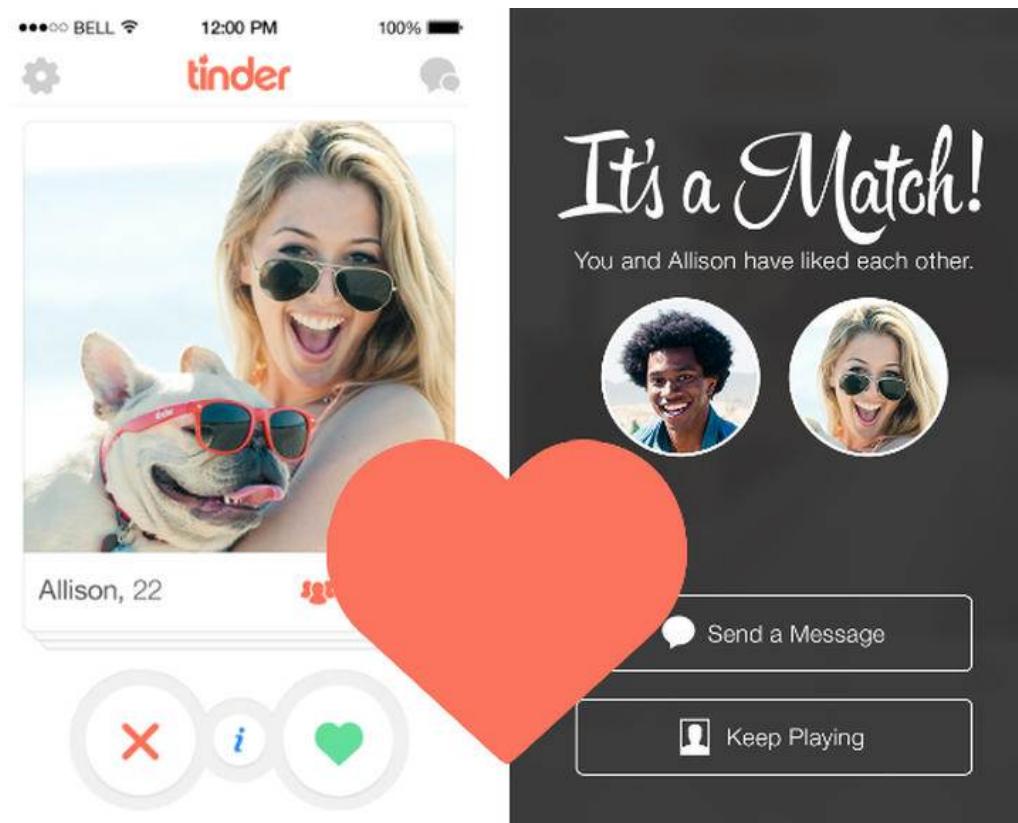


Scalable Monitoring Platform





Happy Users!





We are hiring!

Email – utkarsh.bhatnagar@gotinder.com

