

Berlin 12-13 October 2017

{CODE}MOTION}

# { Monitoring Big Data Systems - Done “The Simple Way”

Demi Ben-Ari - VP R&D @  Panorays



# About Me



## Demi Ben-Ari, Co-Founder & VP R&D @ Panorays

- Google Developer Expert



- **Co-Founder of Communities:**

- **"Big Things" - Big Data, Data Science, DevOps**
- **Google Developer Group Cloud**
- **Ofek Alumni Association**



## **In the Past:**

- Sr. Data Engineer - Windward
- Team Leader & Sr. Java Software Engineer,  
Missile defense System - **"Ofek" – IAF**



# Agenda

- A lot of (NOT) funny Jokes
- Problem definition and Environment
- Monitoring pipeline solutions
  - Metrics
  - Datastores
  - Dashboards
  - Alerting
- Summary
- (Not going to address Service discovery and monitoring)



**Say "Distributed", Say "Big Data",  
Say....**



Google Cloud Platform

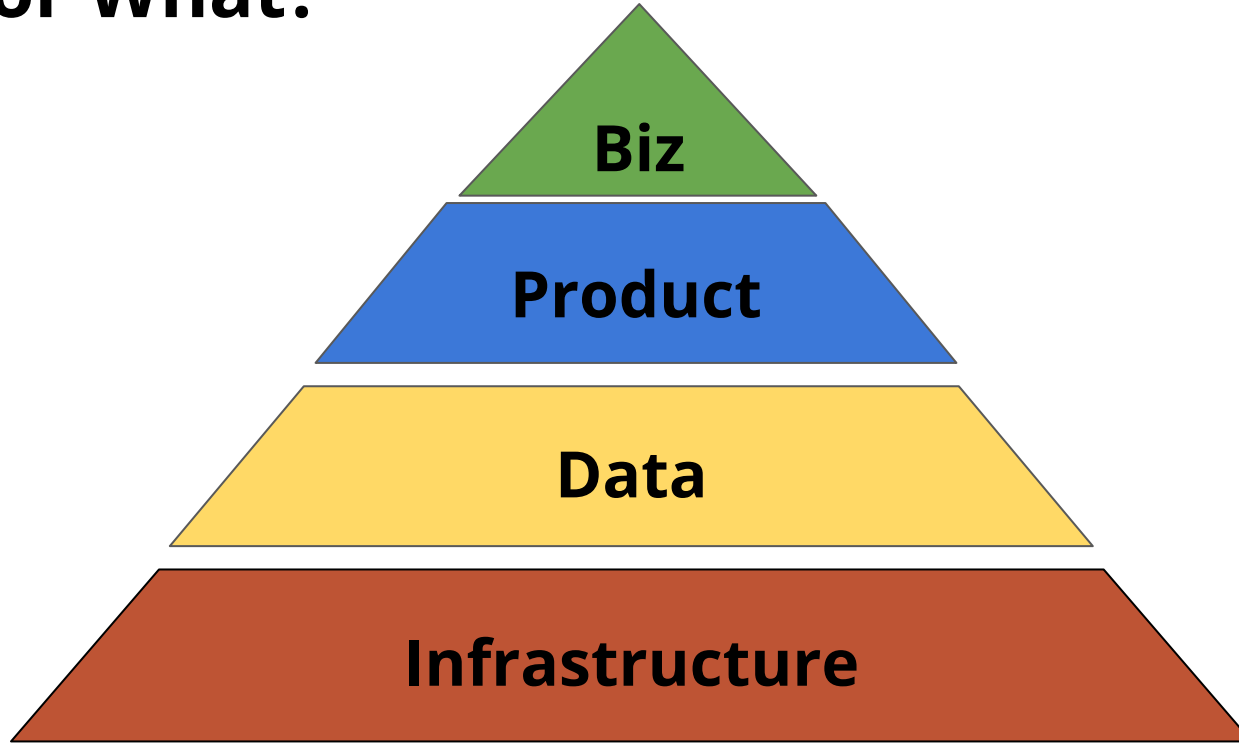
# What is Big Data (IMHO)? And What to Monitor?

- Systems involving the “3 Vs”:

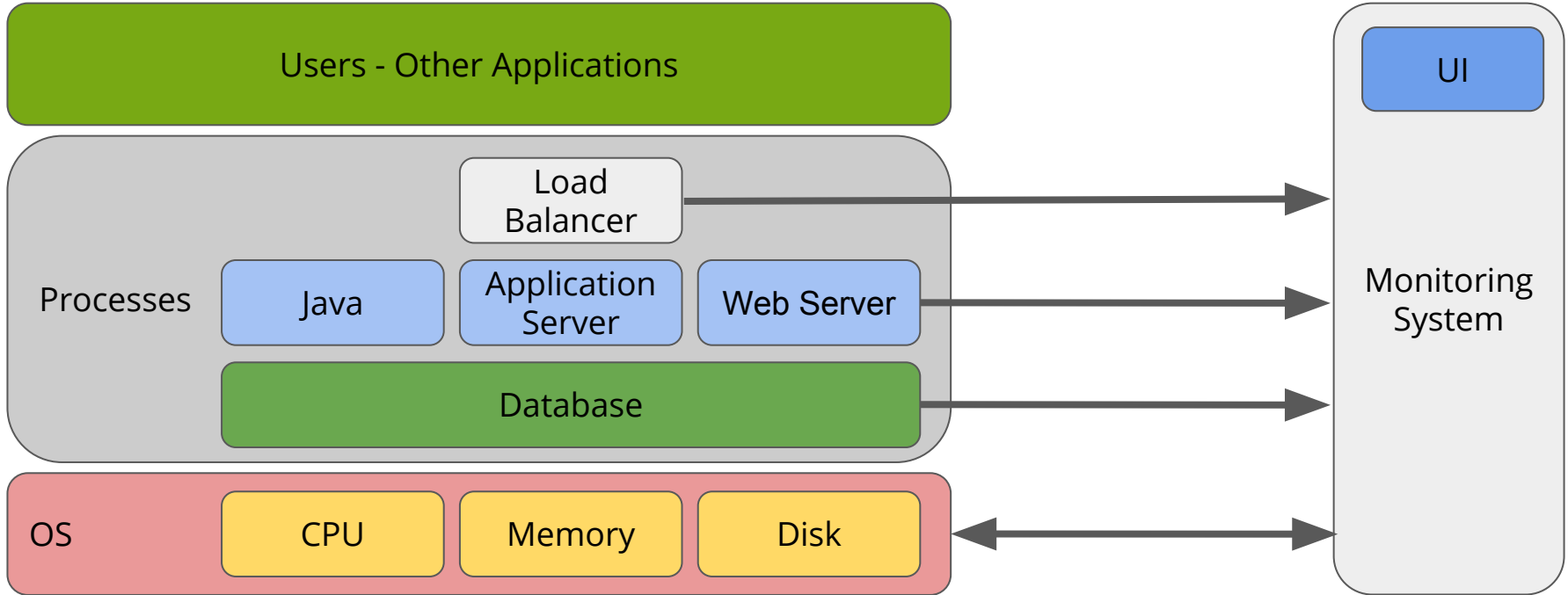
What are the right questions we want to ask?

- **Volume** - **How much?**
- **Velocity** - **How fast?**
- **Variety** - **What kind?** (Difference)

# Monitor What?

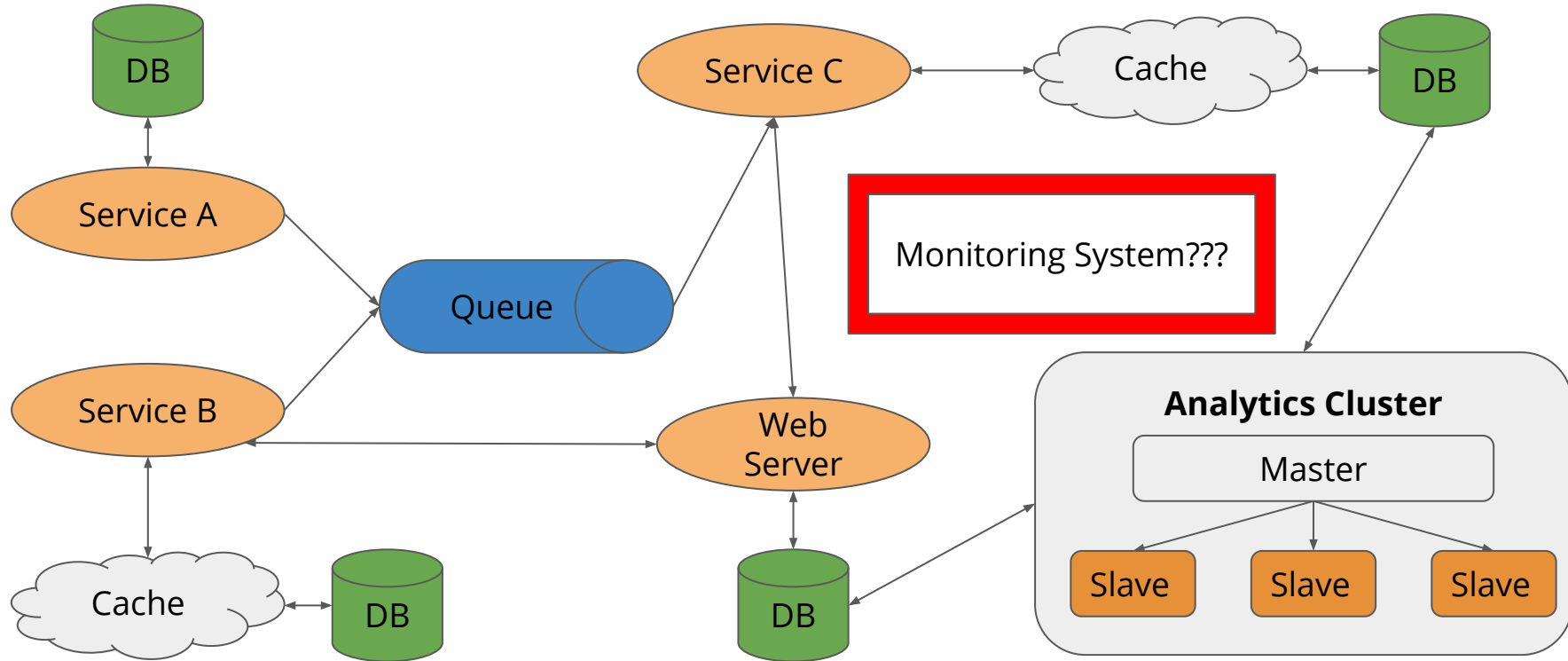


# Monolith Structure



Many times...all of this was on a single physical server!

# Distributed Microservices Architecture





Some basic concepts



# Basic Concepts

- **Monitoring**
- **White-box**
  - **internals**
- **Black-box**
  - **behavior**
- **Dashboard**
- **Alert**
- **Root cause**
- **Node and machine**
- **Deploy**
  - Any change to a service's running software or its configuration.
- **KPI** - Key Performance Indicator

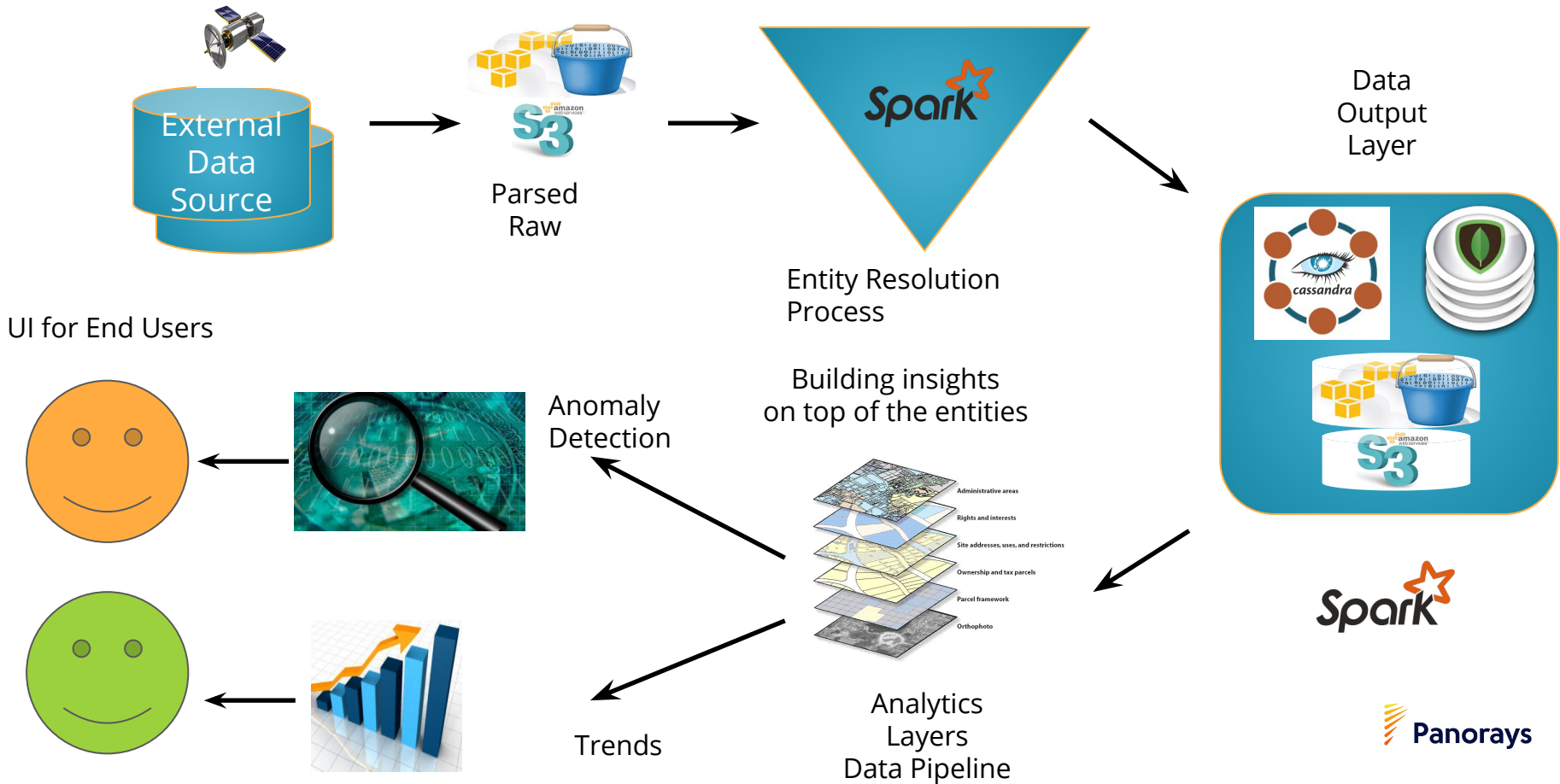
# Data flow and Environment (Use Case)

WINDWAR

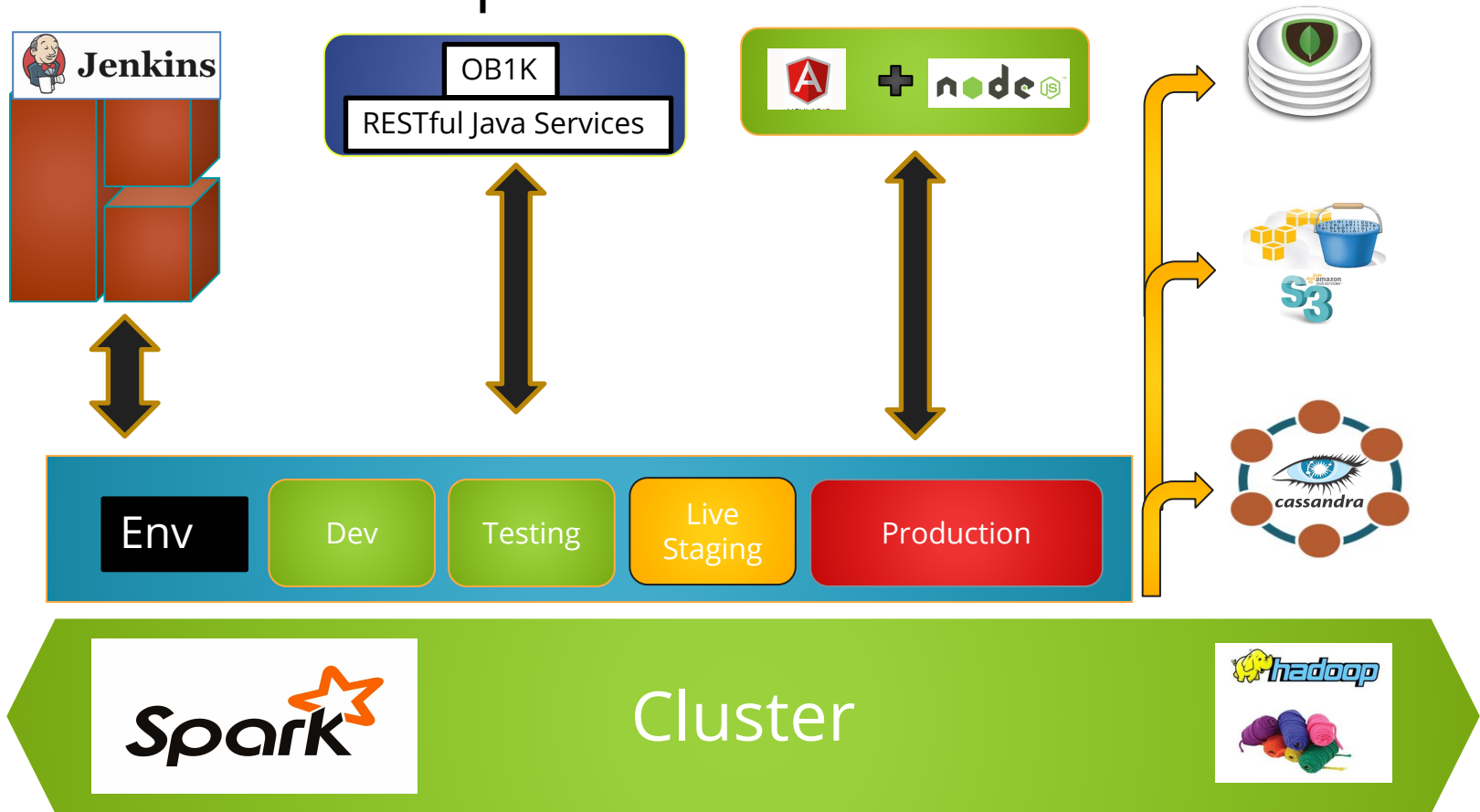
# Structure of the Data

- Maritime Analytics Platform
- Geo Locations + Metadata
- Arriving over time
- Different types of messages being reported by satellites
- Encoded (For compression reasons)
- Might arrive later than actually transmitted

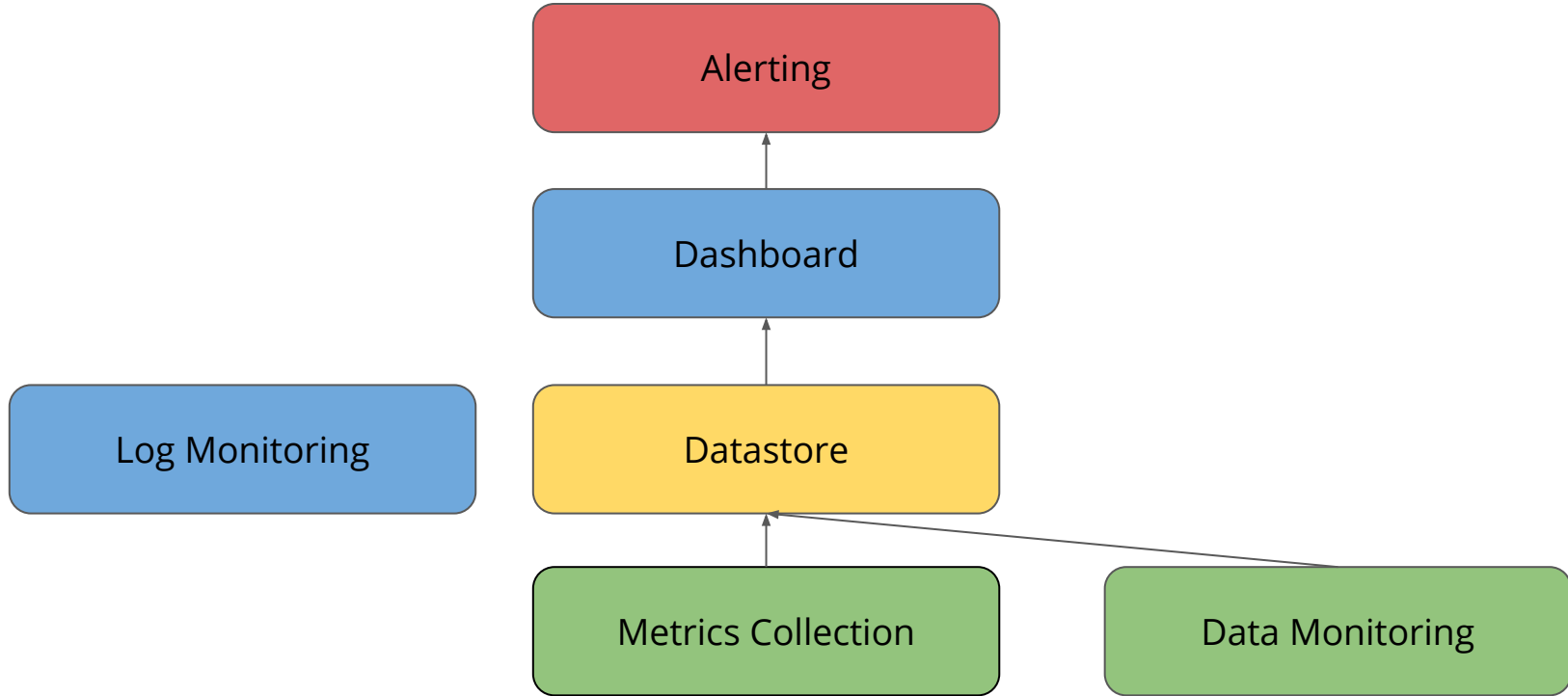
# Data Flow Diagram



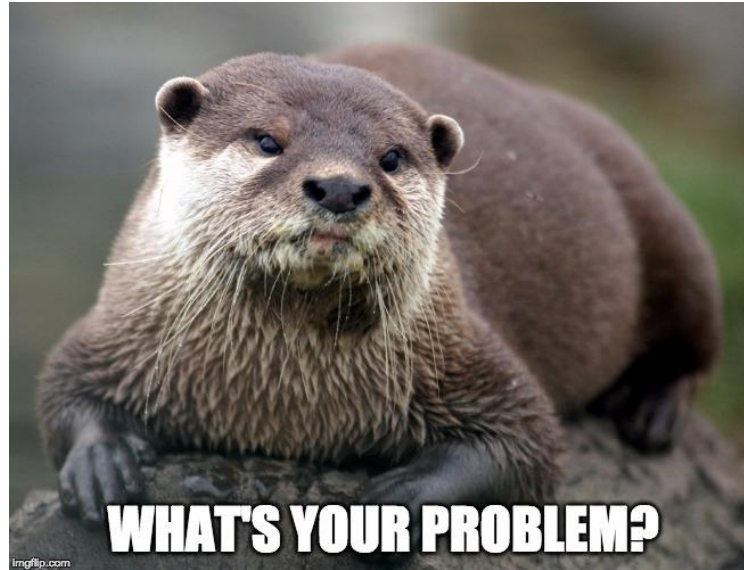
# Environment Description



# Monitoring Stack - Let's fill in the blanks



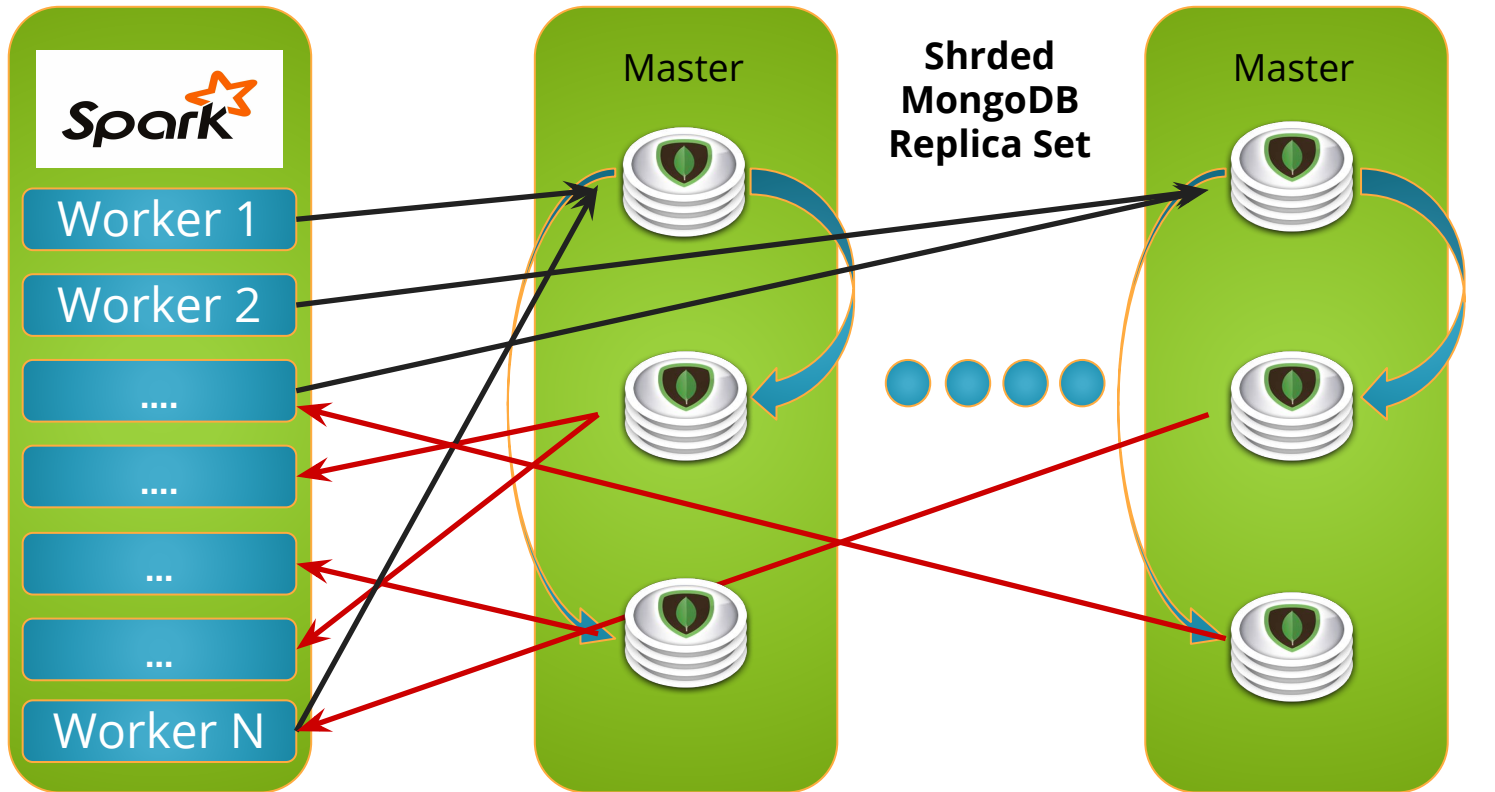
# Situations & Problems





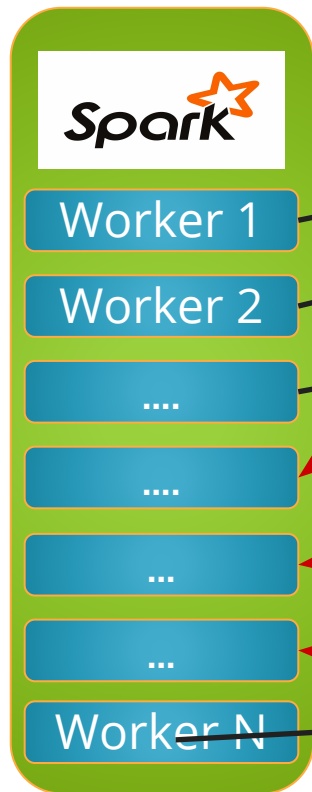
# MongoDB + Spark

Spark  
Cluster



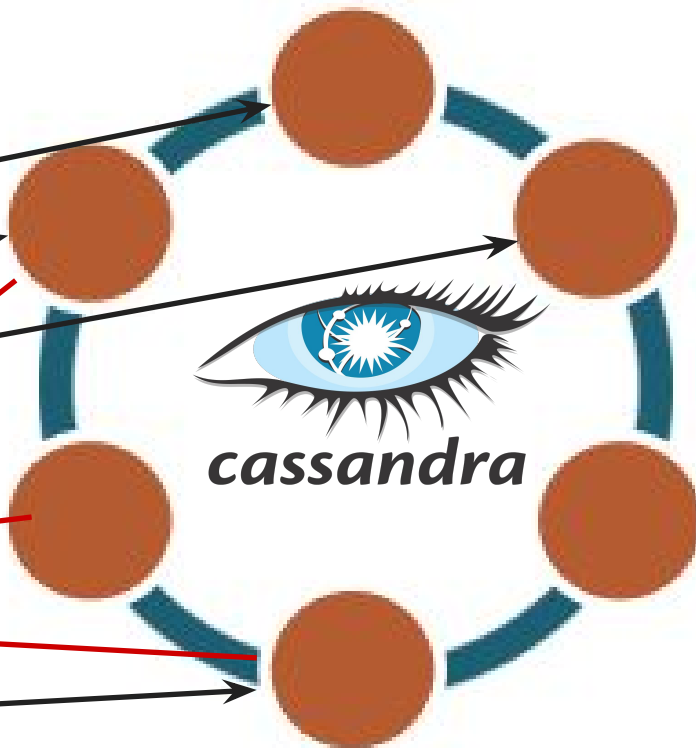
# Cassandra + Spark

Spark  
Cluster



Write →  
Read ←

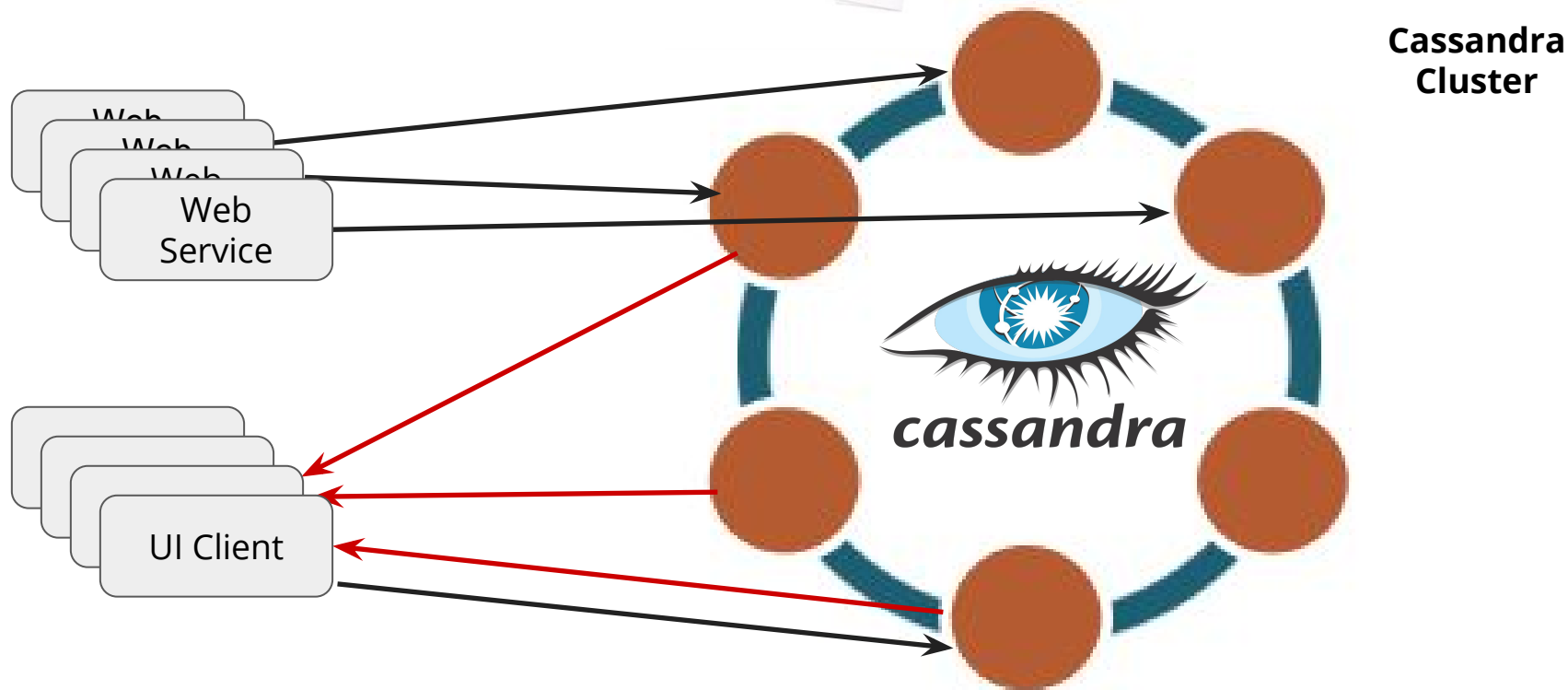
Cassandra  
Cluster



# Cassandra + Serving



Write →  
Read ←



# Problems

- Multiple physical servers
- Multiple logical services
- Want Scaling => More Servers
- Even if you had all of the metrics
  - You'll have an overflow of the data
- **Your monitoring becomes a “Big Data” problem itself**

# This is what “Distributed” really Means

The DevOps Guy  
(It might be you)



So...Solutions  
Let's Start!



# Monitoring Operation System



# Monitoring Operation System Metrics

- What to measure:
  - **CPU**
  - **Memory**
  - **Disk Space**
- How to measure:
  - [CollectD](#) or [StatsD](#) reporting to Graphite
  - [New Relic](#)
    - Nice and easy UI
    - Even the free account gives great tool
    - Alerting of thresholds





Google Cloud Platform



Some help  
from “the Cloud”



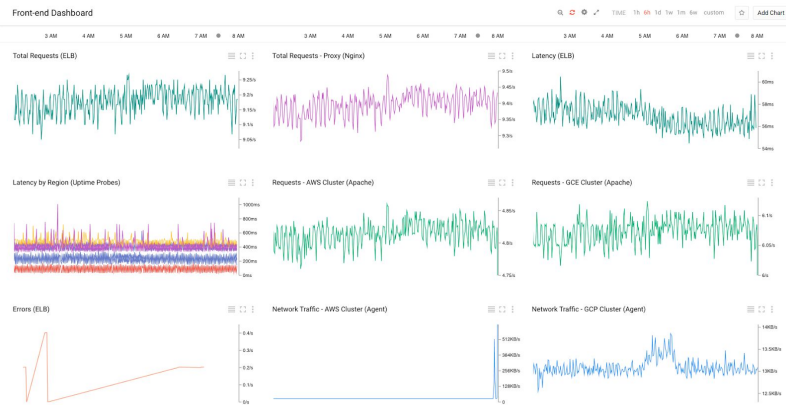
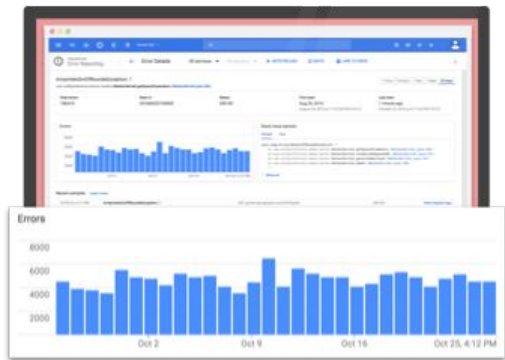
# Google Cloud Platform StackDriver



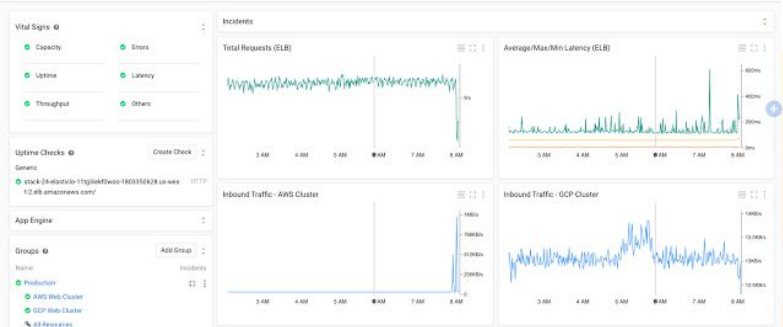
Google Cloud Platform



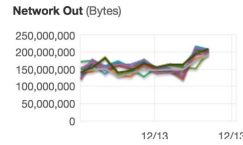
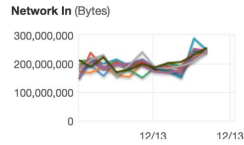
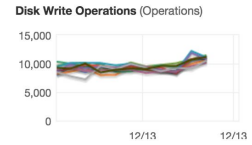
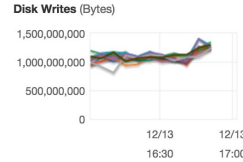
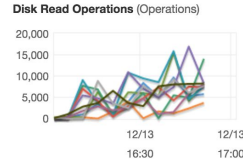
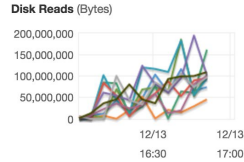
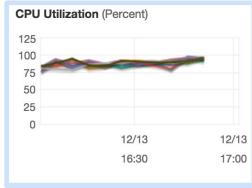
Stackdriver



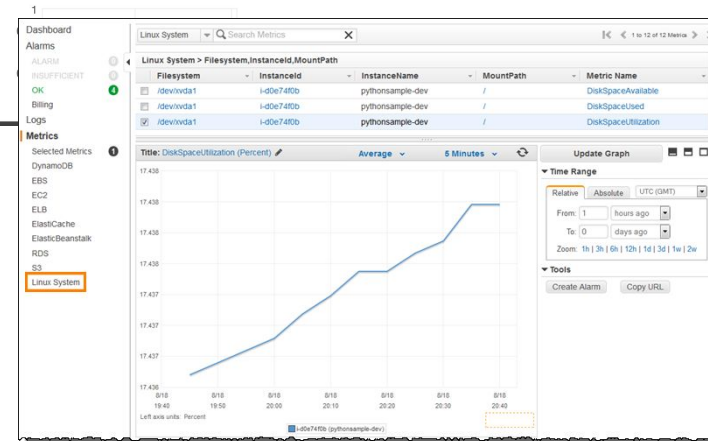
Google Stackdriver - Demo



# Amazon Web Services CloudWatch



Status Check Failed (Any) (Count)



# Report to Where?

- **We chose:**
- [Grafite](#) ([InfluxDB](#)) + [Grafana](#)
- Can correlate System and Application metrics in one place :)



 Grafite

 Elasticsearch

 Cloudwatch

 Prometheus

 InfluxDB

 & More



# Report to Where?

- Save DevOps efforts if you're willing to Pay :)
- **Hosted Graphite**
  - <https://www.hostedgraphite.com/>
- Throwing the "Big Data" volume monitoring problem at someone else

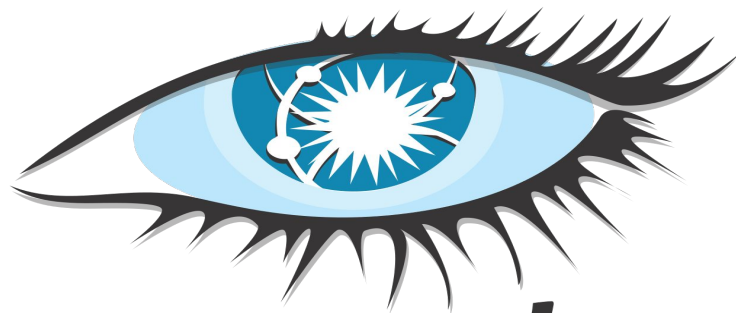
Connections  
Connections...



# Drivers to Datastores

- Actions they usually do:
  - Open connection
  - Apply actions
    - Select, Insert, Update, Delete
  - Close connection
- Do you monitor each?
  - **Hint:** Yes!!!! Hell Yes!!!
- Creating a wrapper in any programming language and reporting the metrics
  - Count, execution times, errors...

# Monitoring Cassandra



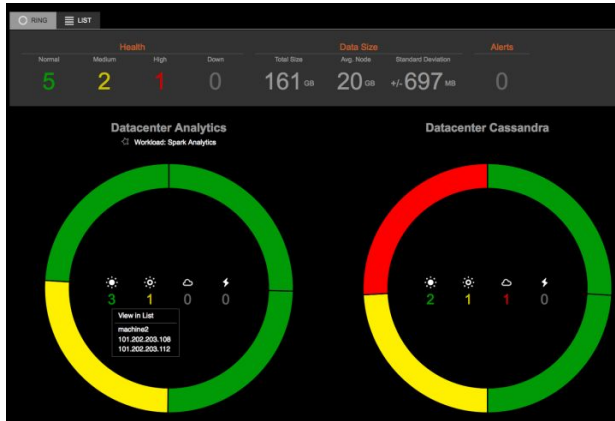
***cassandra***

DATASTAX 



# Monitoring Cassandra

- [OpsCenter](#) - by DataStax



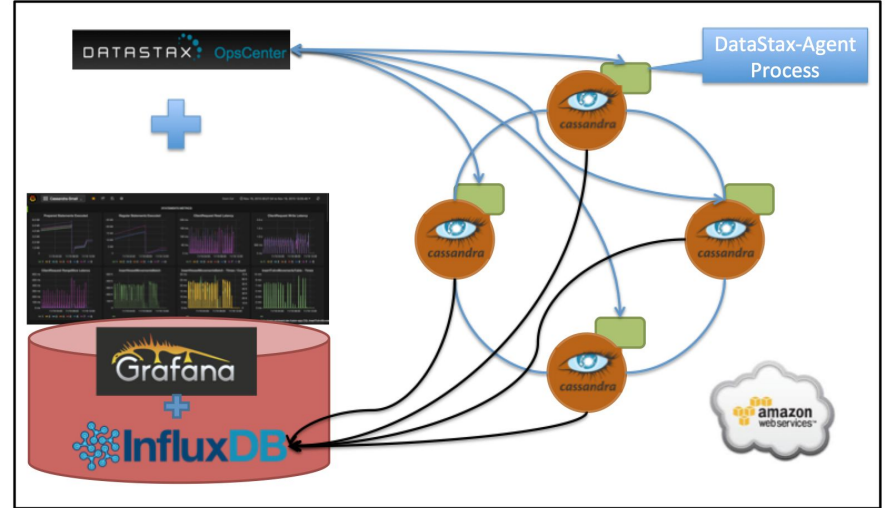
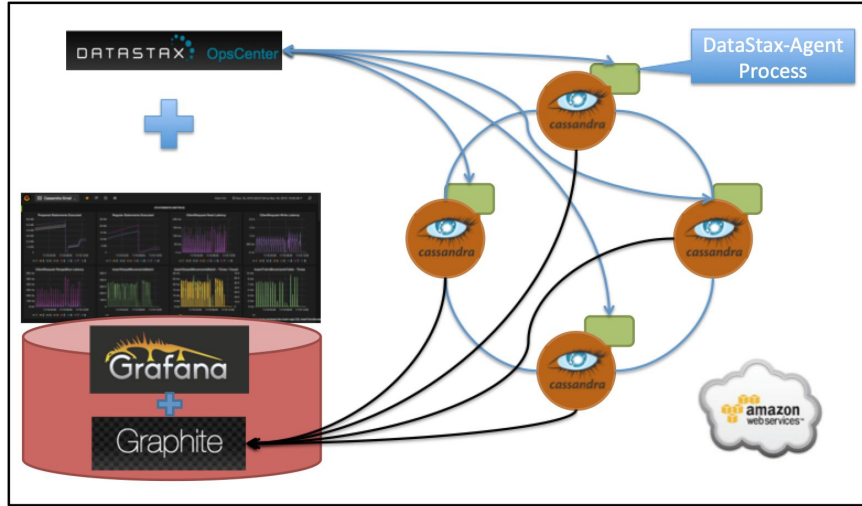
# Monitoring Cassandra

- Is the enough?...

We can connect it to Graphite also (**Blog:** [“Monitoring the hell out of Cassandra”](#))

- Plug & Play the metrics to Graphite - Internal Cassandra mechanism
- Back to the basics: [dstat](#), [iostat](#), [iotop](#), [jstack](#)

# Monitoring Cassandra





# Monitoring Spark



# What to monitor in an Apache Spark Cluster

- Application execution
- Resource consumption and allocation
- Task Failures
- Environment and Amount of servers
- Physical OS metrics
- Infrastructure services

# Ways to Monitoring Spark

- [Grafana-spark-dashboards](#)
  - Blog:  
<http://www.hammerlab.org/2015/02/27/monitoring-spark-with-graphite-and-grafana/>
- **Spark UI** - Online on each application running
- **Spark History Server** - Offline (After application finishes)
- [Spark REST API](#)
  - Querying via inner tools to do ad-hoc monitoring
- **Back to the basics:** [dstat](#), [iostat](#), [iotop](#), [jstack](#)
- Blog post by Tzach Zohar - [“Tips from the Trenches”](#)
- <http://spark.apache.org/docs/latest/monitoring.html>

# Monitoring Your Data



# Data Questions? What should be measure

- Did all of the computation occur?
  - Are there any data layers missing?
- How much data do we have? **(Volume)**
- Is all of the data in the Database?
- **Data Quality Assurance**



# Data Answers!

- **KISS** (Keep it simple stupid)
- **Jenkins + Maven (JUnit)** for the rescue
- Creating a maven “monitoring” project.
  - Running scheduled tasks, each for the relevant data source
    - Database data existence
    - S3 files existence
    - Data flow that keeps on coming from sensors
    - (Any other data source that you can imagine...)
  - Scheduled task that write amount metrics to Graphite -> Dashboards
  - Report task execution to Graphite



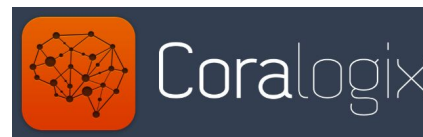
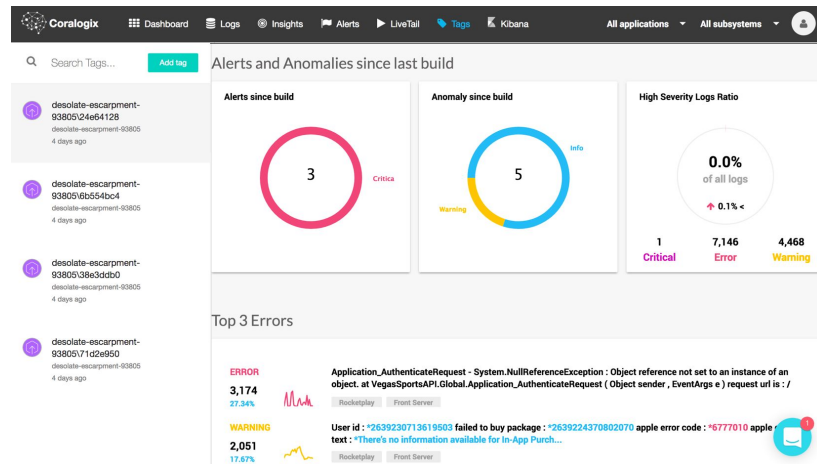
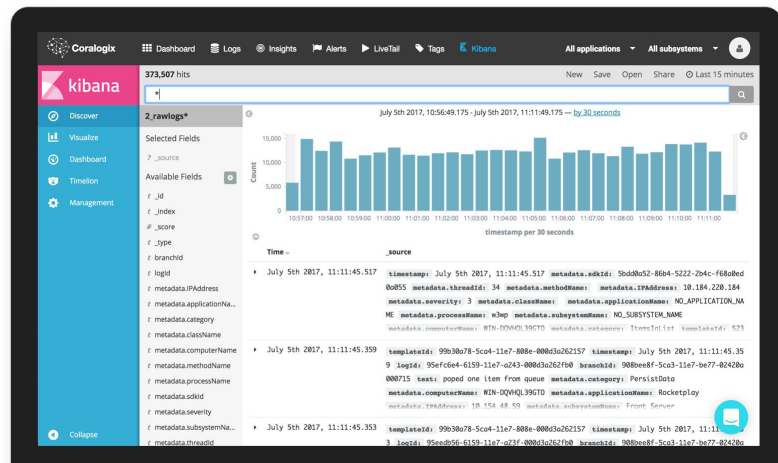
# Data Answers!

- **The method doesn't really matter, as long as you:**
  - Can follow the results over time
  - Know what your **data flow**, know what **might fail**
  - It's easy for anyone to **add more monitoring**  
(For the ones that add the new data each time...)
  - It don't trust others to add monitoring  
(It will always end up the **DevOps's** "fault" -> No monitoring will be applied)

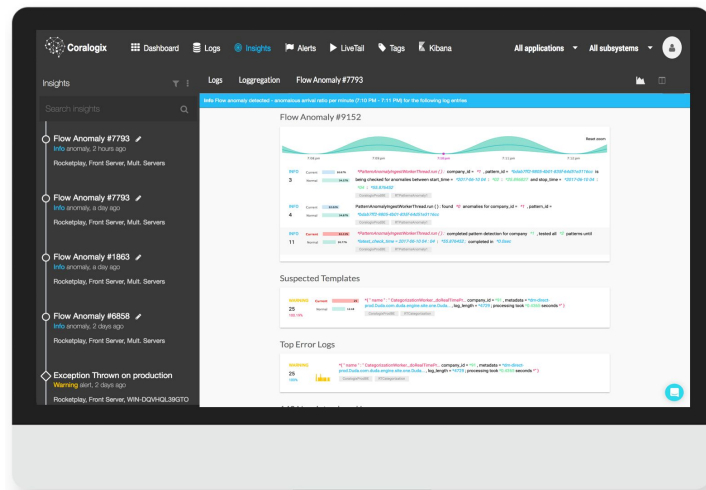
# Logging? Monitoring?



# Coralogix

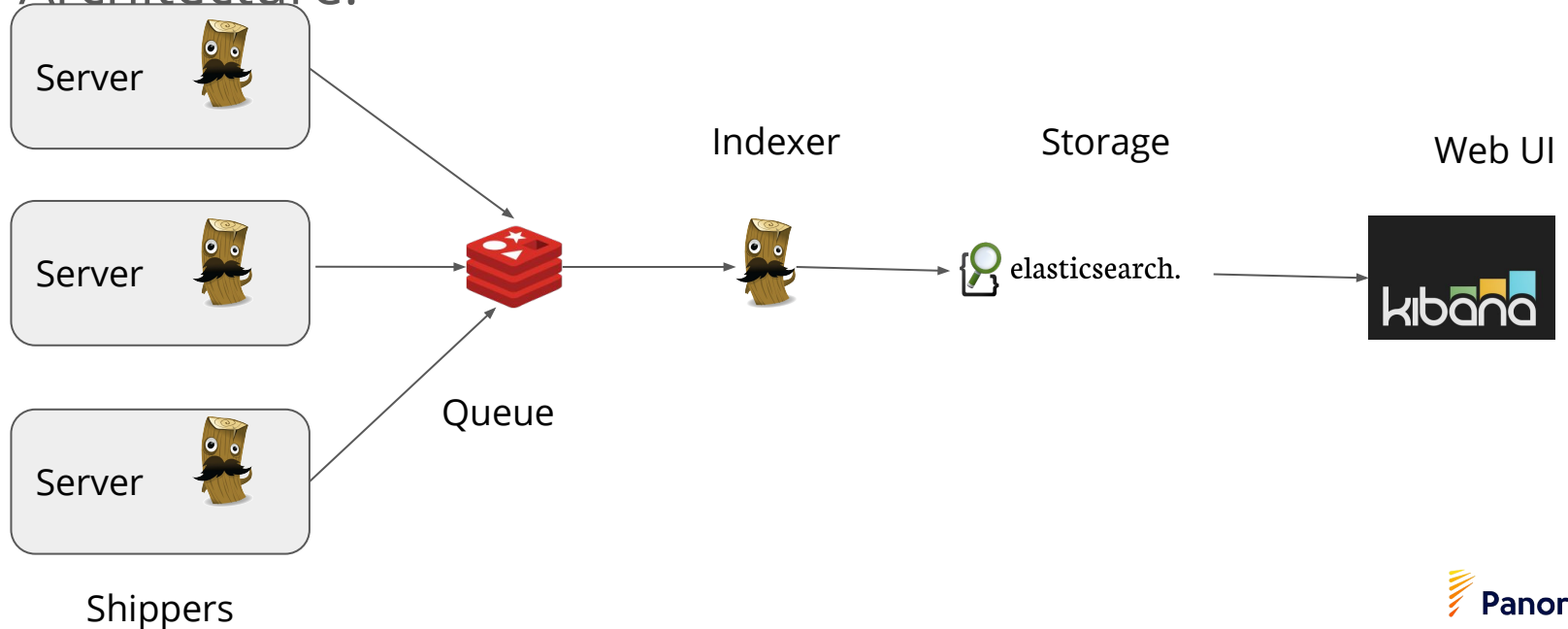


# Coralogix

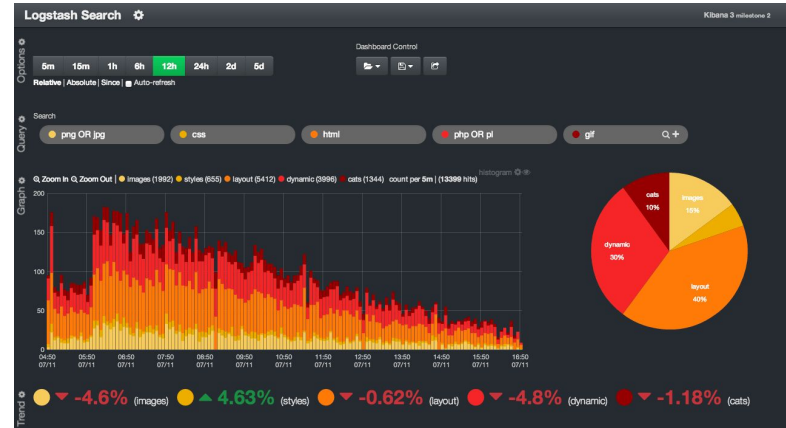


# ELK - Elasticsearch + Logstash + Kibana

- [Elastic](#)
- Architecture:



# ELK - Elasticsearch + Logstash + Kibana



Did someone say  
“Dashboard”?



<http://go.funpic.hu>



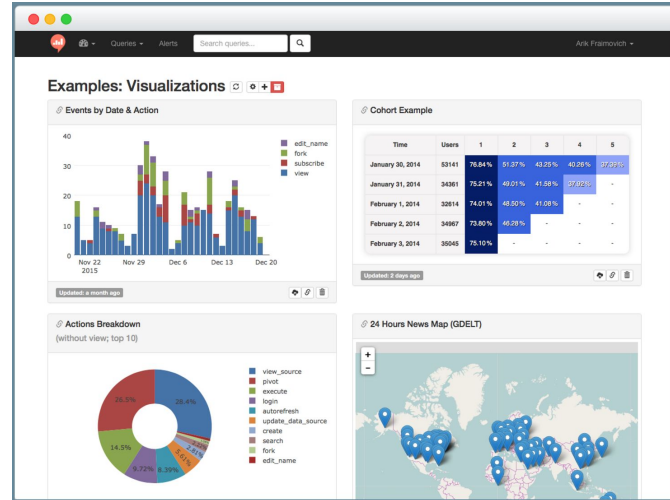
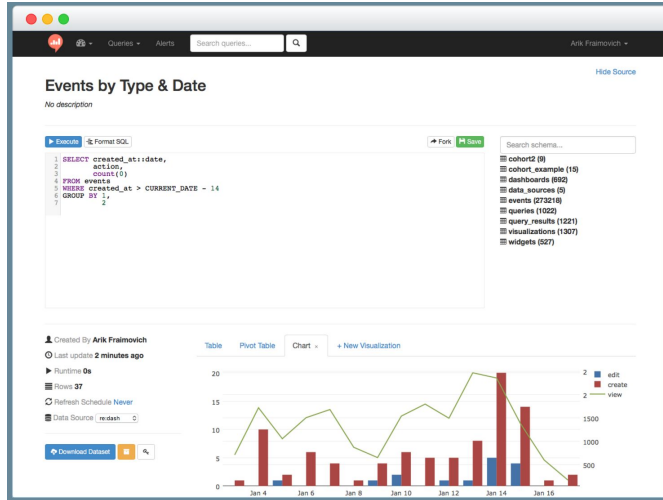
# Redash



- <http://redash.io/>
  - Open Source: <https://github.com/getredash/redash>
  - Came out as one of many Open source tool by Everything.me
  - Created and Maintained by [Arik Fraimovich](#) (You rock!)
  - Written in Python
  - Has an on-premise and hosted solution
- 
- At [Panorays](#) we also use it for Alerting via Its integration with Slack



# Redash - Screenshots



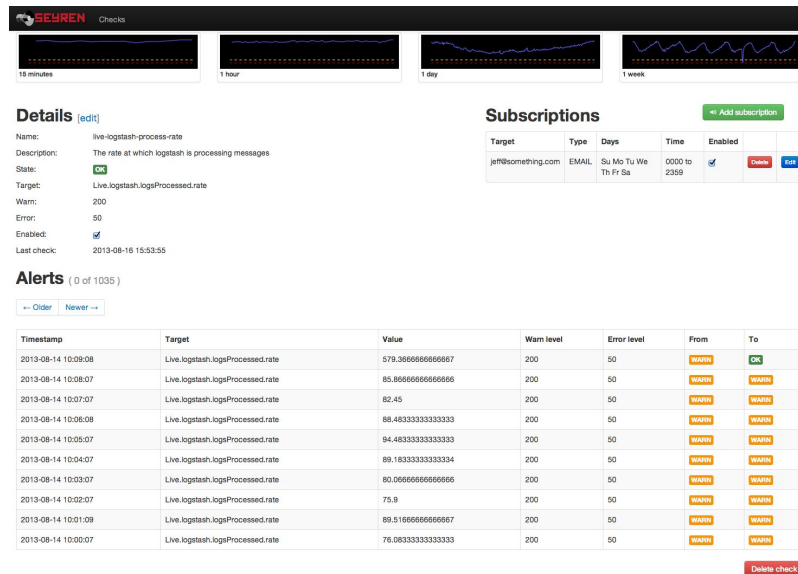
Alerting



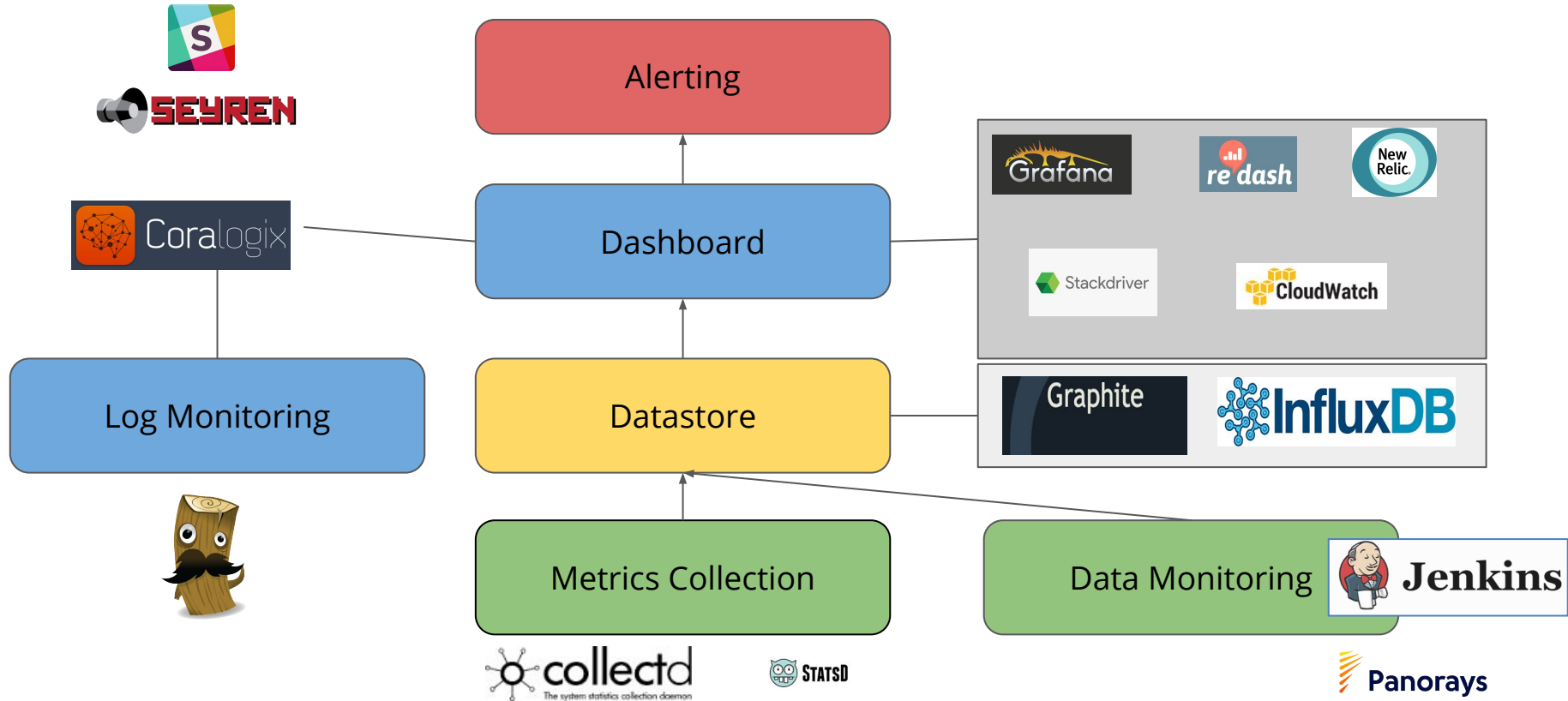
# Alerting



- [Syren](#) - Open source
- Reporting to:
  - [Email](#), [Flowdock](#), [HipChat](#), [HTTP](#), [Hubot](#), [IRCCat](#), [PagerDuty](#), [Pushover](#), [SLF4J](#), [Slack](#), [SNMP](#), [Twilio](#)



# Summary - Monitoring Stack



Not my problem...or is it?



So...Who does the monitoring in our company?



<https://imgflip.com/i/18kw1>

# Conclusions

- Correlating **Application and System** metrics!!!!
- Ask the right **monitoring questions** -> **answer with Dashboards**
- **KISS - simple is key**, what's hard, we tend not to do at all
- **Alert** about what you can actually react to
  - (And to the relevant person)
- Measure whatever you can
  - only way to know if you're improving
- **Monitor your business KPIs too**



# Conclusions

- If all of what I've said is not enough...

**Graphs are fricking cool!**



# Questions?





- LinkedIn
- Twitter: @demibenari
- Blog:  
<http://progexc.blogspot.com/>
- **demibenari@gmail.com**

- "Big Things" Community
- Meetup, YouTube, Facebook,  
Twitter
- GDG Cloud

# Thank You





Mind the  
Attack Surface

# Resources

- [Monitoring distributed systems](#) - A case study in how Google monitors its complex systems