Office 365 Big Data Applications using Open Source Technologies

Wesley Miao

PRICIPLE SW Engineering Manager, Office 365 Customer Fabric team at Microsoft Suzhou

- Office 365 Fun Facts
- Open Source Stack
- App 1: Service Monitoring
- App 2: Customer Insights
- App 3: Delve Analytics
- Spark Usage Patterns

Office 365 - Fun Facts

Outlook & O365 Core in Numbers...

CUSTOMERS

63 M MONTHLY ACTIVE USERS

27 M MONTHLY ACTIVE DEVICES

18 M monthly **calendar** users 3.5 M monthly **Yammer** users 2.6 M monthly **Groups** users 35 M mailboxes part of **First Release** 200 K monthly **Admin app** users 1.7 M monthly **Delve** users 500 K **Concierge** users 112 M migrated **consumer** mailboxes

VALUE

130 B EMAILS DELIVERED PER MONTH

1.3 M GROUPS CREATED PER MONTH

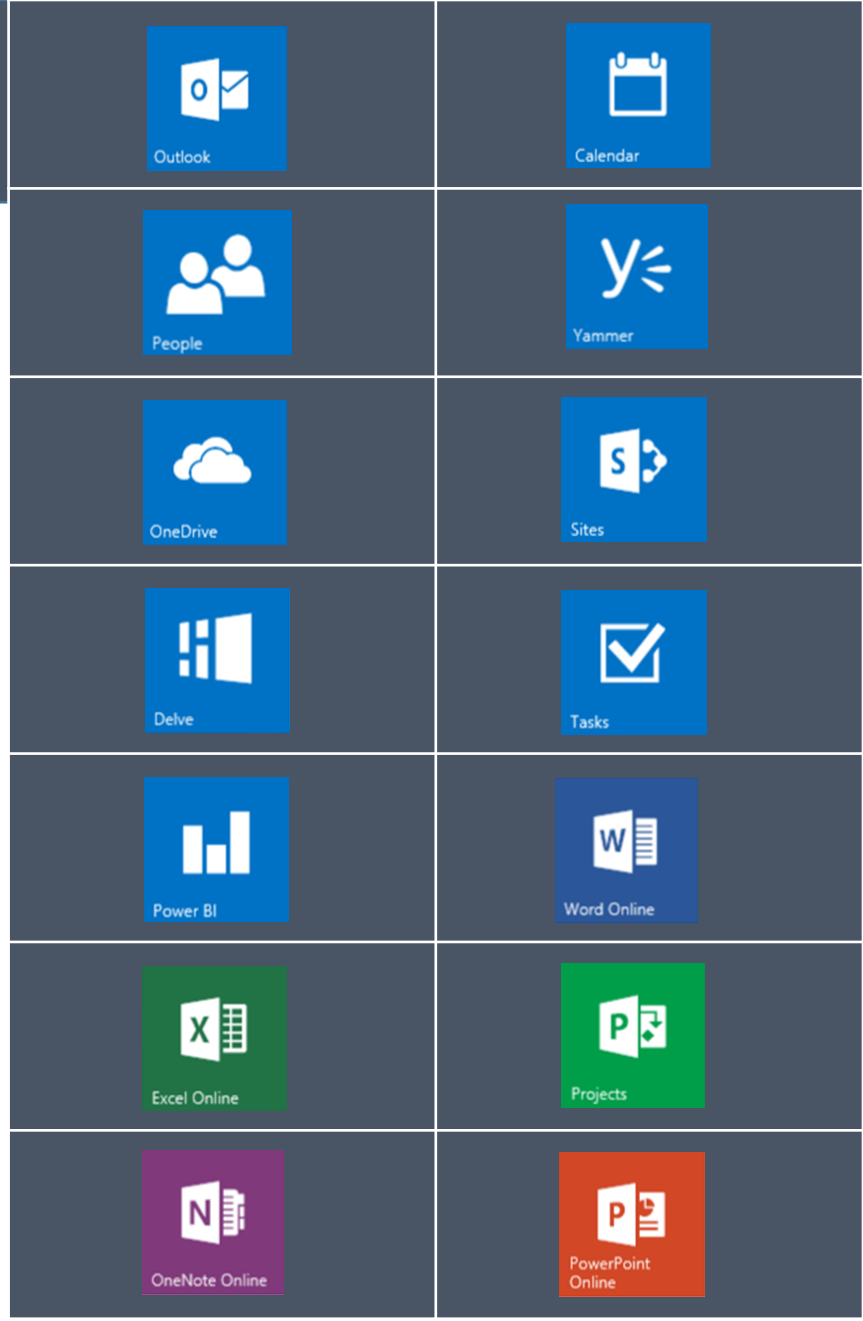
1.2 B **meetings** created monthly 55 B **spam** messages stopped monthly 90 s time taken for GoDaddy to provision a tenant 115 M consumer mailboxes **migrated** to XO1 0.6 s to **provision** a consumer mailbox 11 B user generated **searches** 99.97 % average **availability** 69 % improvement in **search latency** 185+ CSAT for concierge users (compared to 175 for CSS) 433 tenants whose experiences were improved using **Customer Fabric** insights

ENABLERS

102 K PROVISIONED SERVERS

160 PB CONSUMED STORAGE

3258 provisioned **DAGs** in 57 **datacenters** across 14 different **countries** 48 regular trains **deployed** 26 T events processed for **security detection** $194 K in **bug bounty** payouts 108 **Fast Trains** ran through the year 85 % drop in machine **Vulnerabilities** 435 M **CA WFs** executed per month 1.6 M pre-checkin **topologies** deployed per month $245 M saved in **HW costs** via perf improvements 1000 alerts to engineers eliminated through **Red Alerts** 16 K perf regressions throttled by **PUMA** 70 K writes/sec processed by **Customer Fabric** platform 700 auto recovery actions executed by **Red Alerts** 100 % of Exchange users on boarded onto **Torus** 9 T **AD queries** per month 300 K **database moves** per day 15 K rack **failover/failbacks**




kafka


APACHE
Spark™

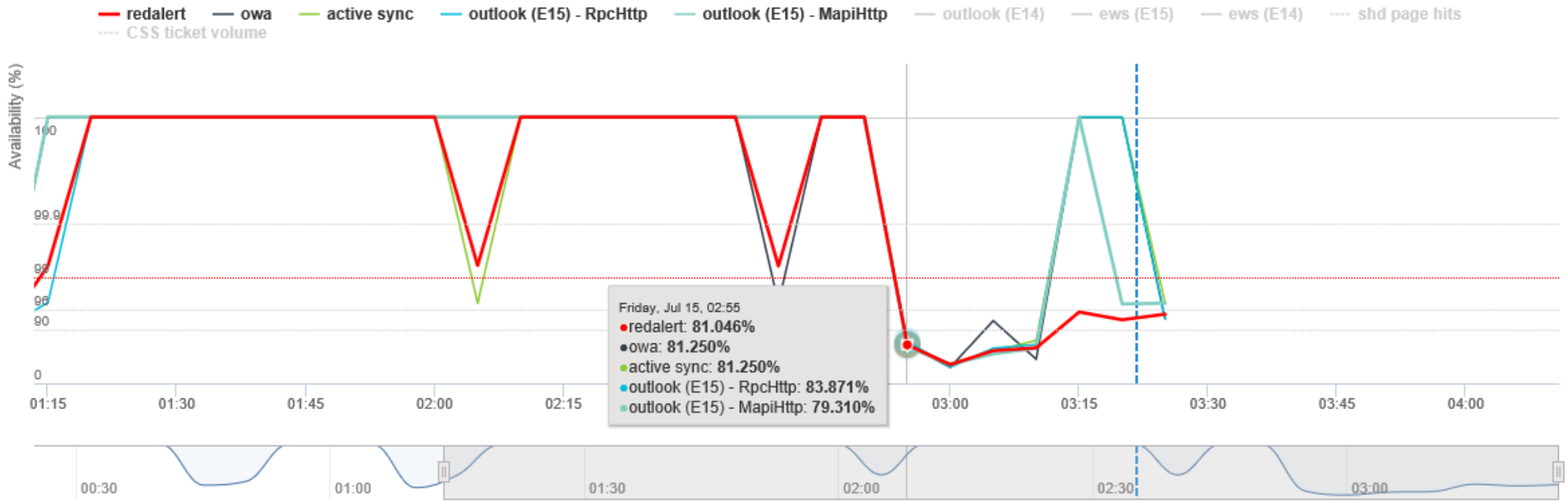

cassandra

App 1: Service Monitoring

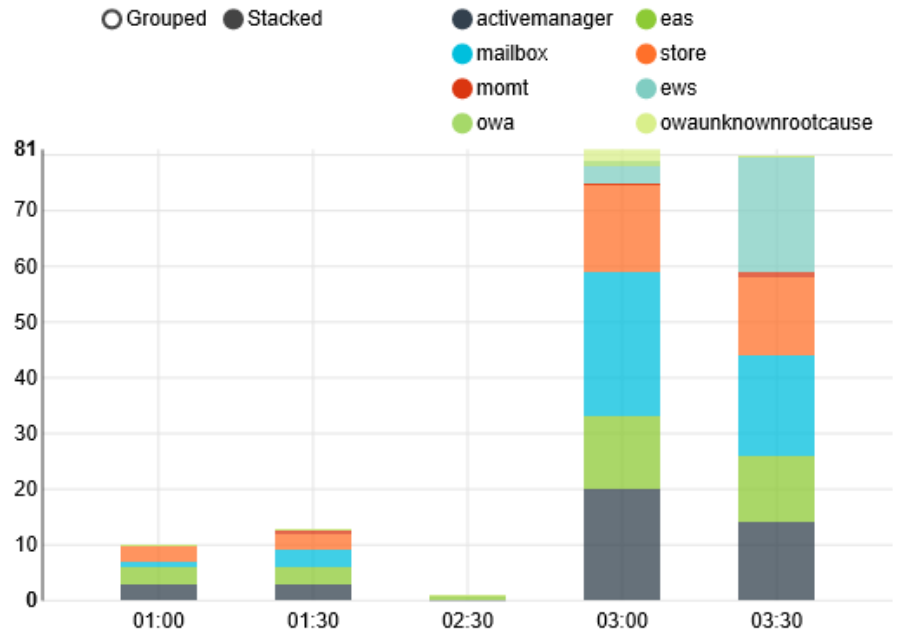
Office365 Service Monitoring

- Active Monitoring (Synthetic)
  - Local Active Monitoring
  - External Active Monitoring

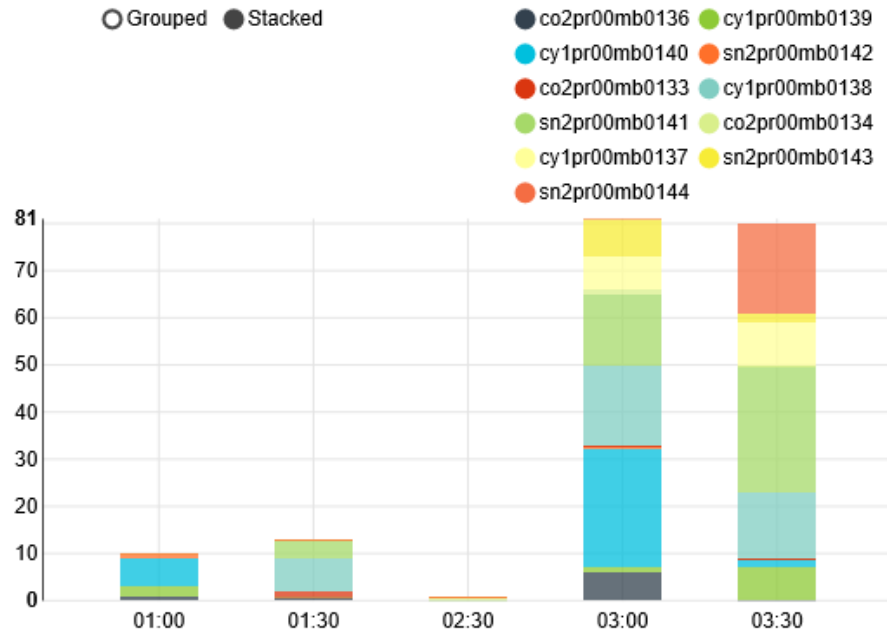
- Passive Monitoring (Real traffic)

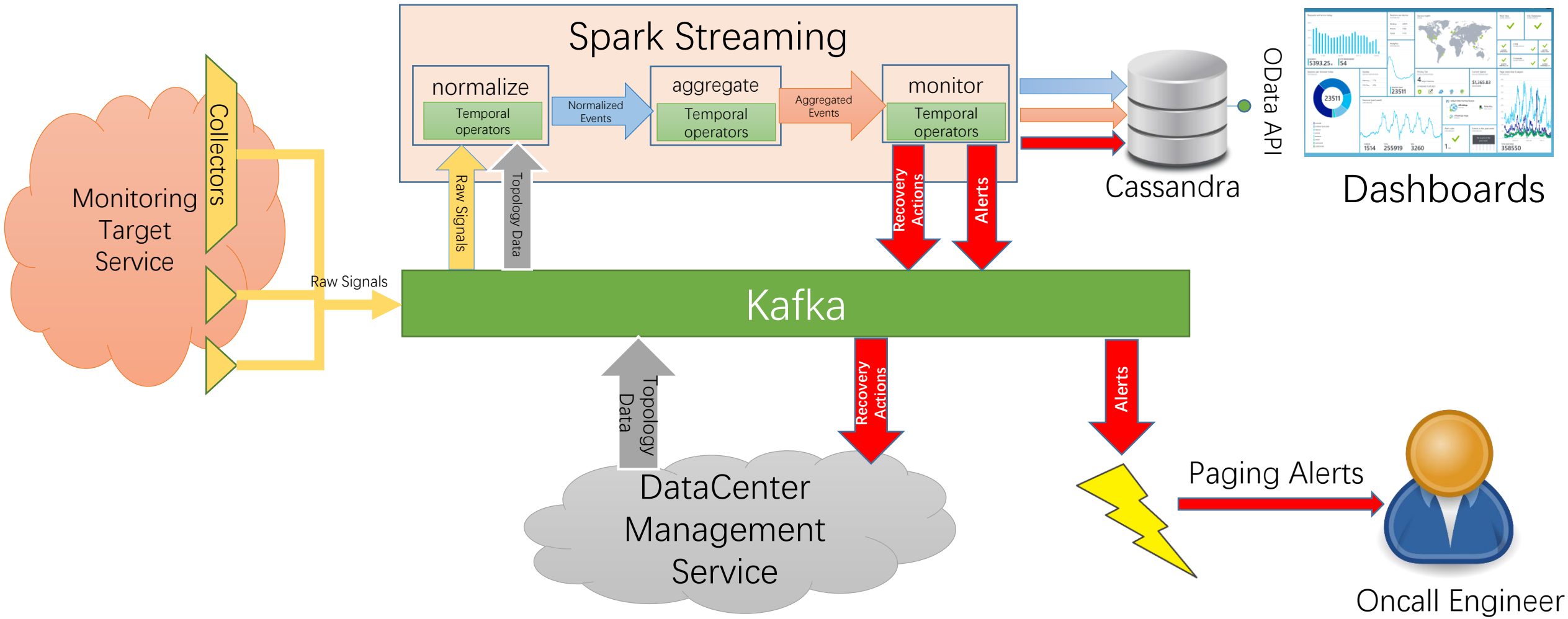


Component Errors



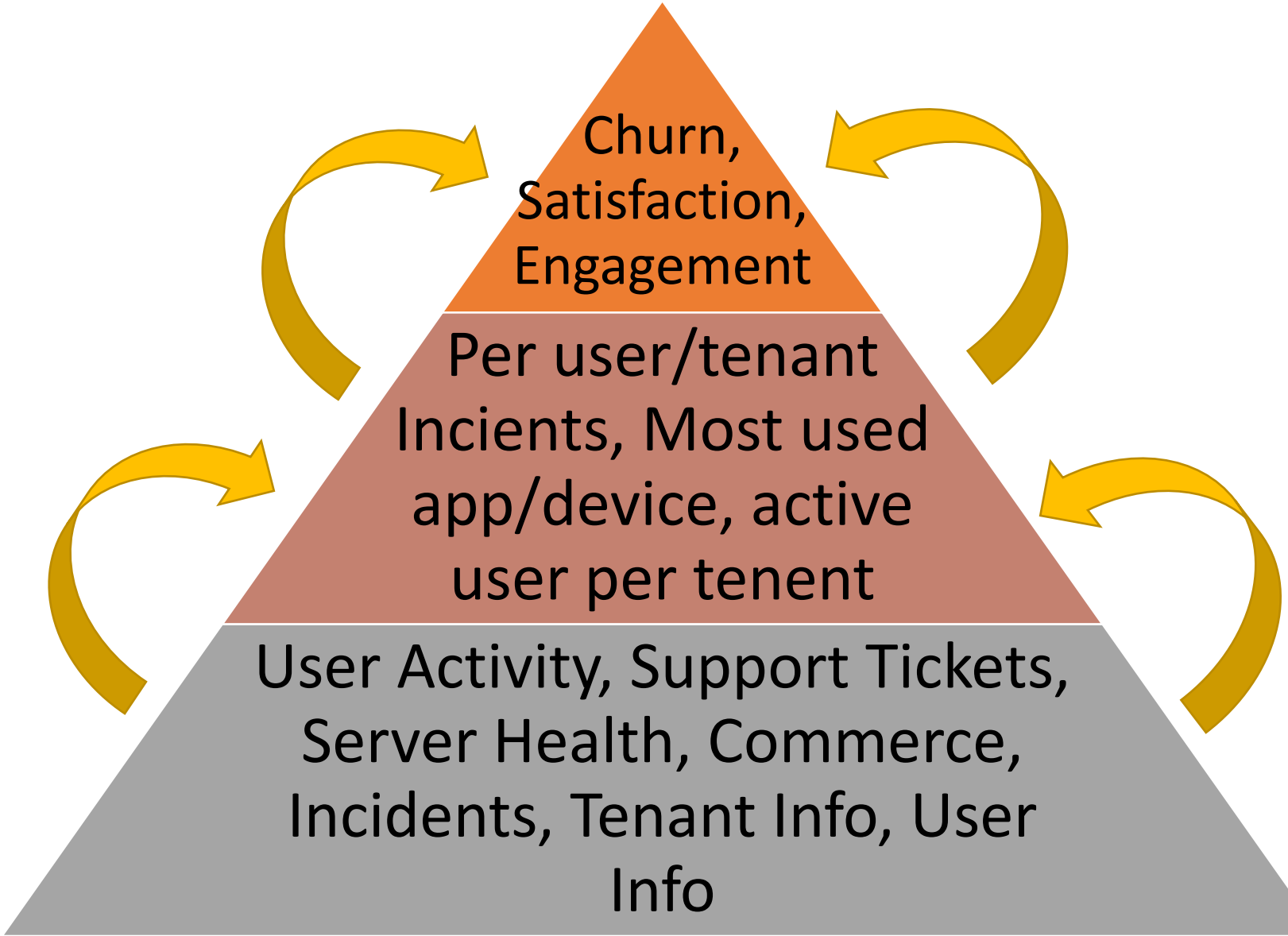
Machine Errors



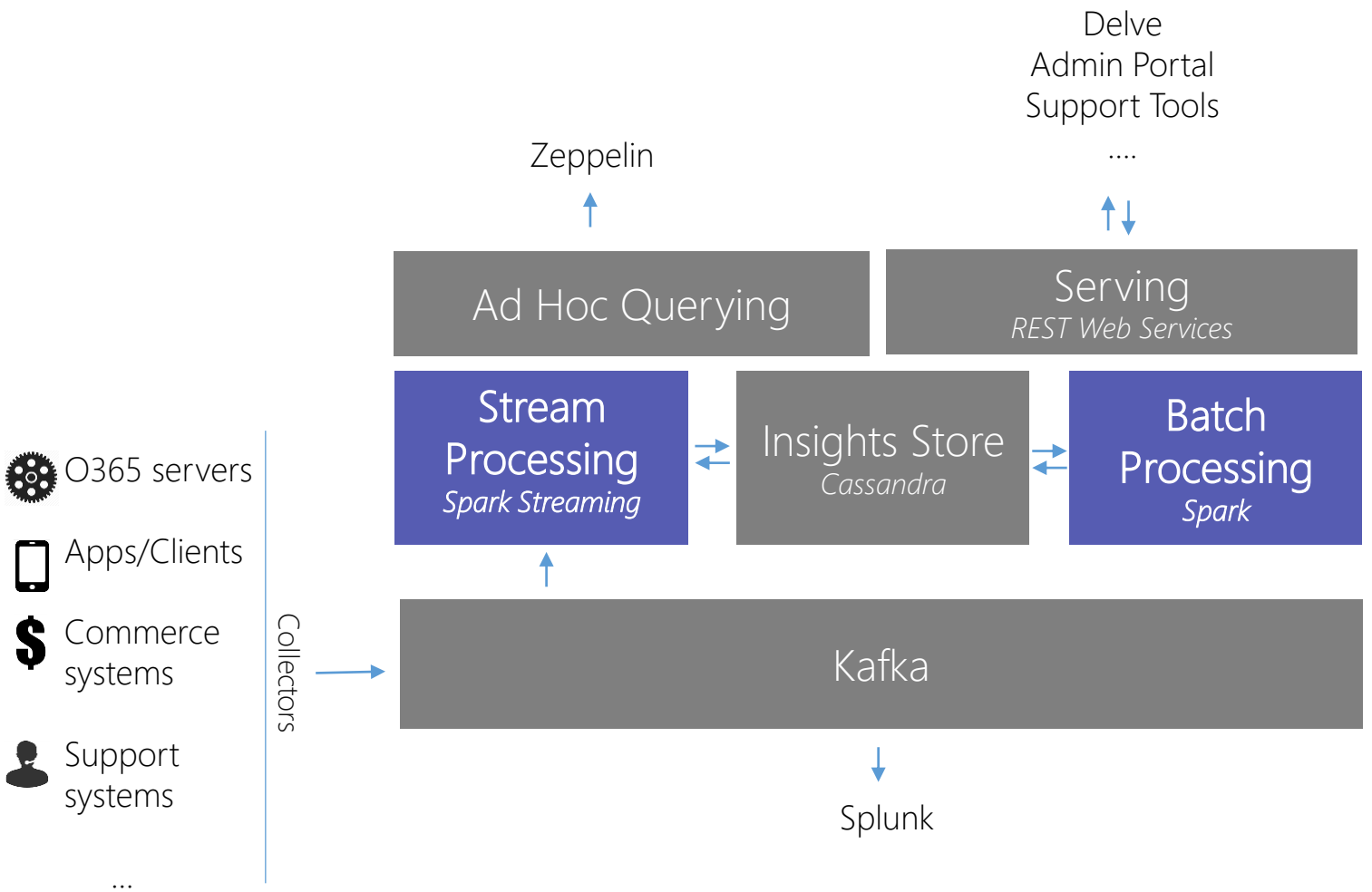

Spark Streaming
normalize
Temporal operators
Normalized Events
aggregate
Temporal operators
Aggregated Events
monitor
Temporal operators
Monitoring Target Service
Collectors
Raw Signals
Topology Data
Recovery Actions
Alerts
Kafka
Cassandra
OData API
Dashboards
DataCenter Management Service
Paging Alerts
Oncall Engineer

App 2: Customer Insights

Office365 Customer Insights



Architecture



Key facts

- Running in Azure
- Highly Scalable
- High ingestion rates
- Real time analytics
- Batch analytics
- Machine learning
- PII Compliance

App 3: Delve Analytics

Delve Analytics

Reinventing productivity through individual empowerment.

Delve Analytics provides you with insights into two of the most important factors in personal productivity:

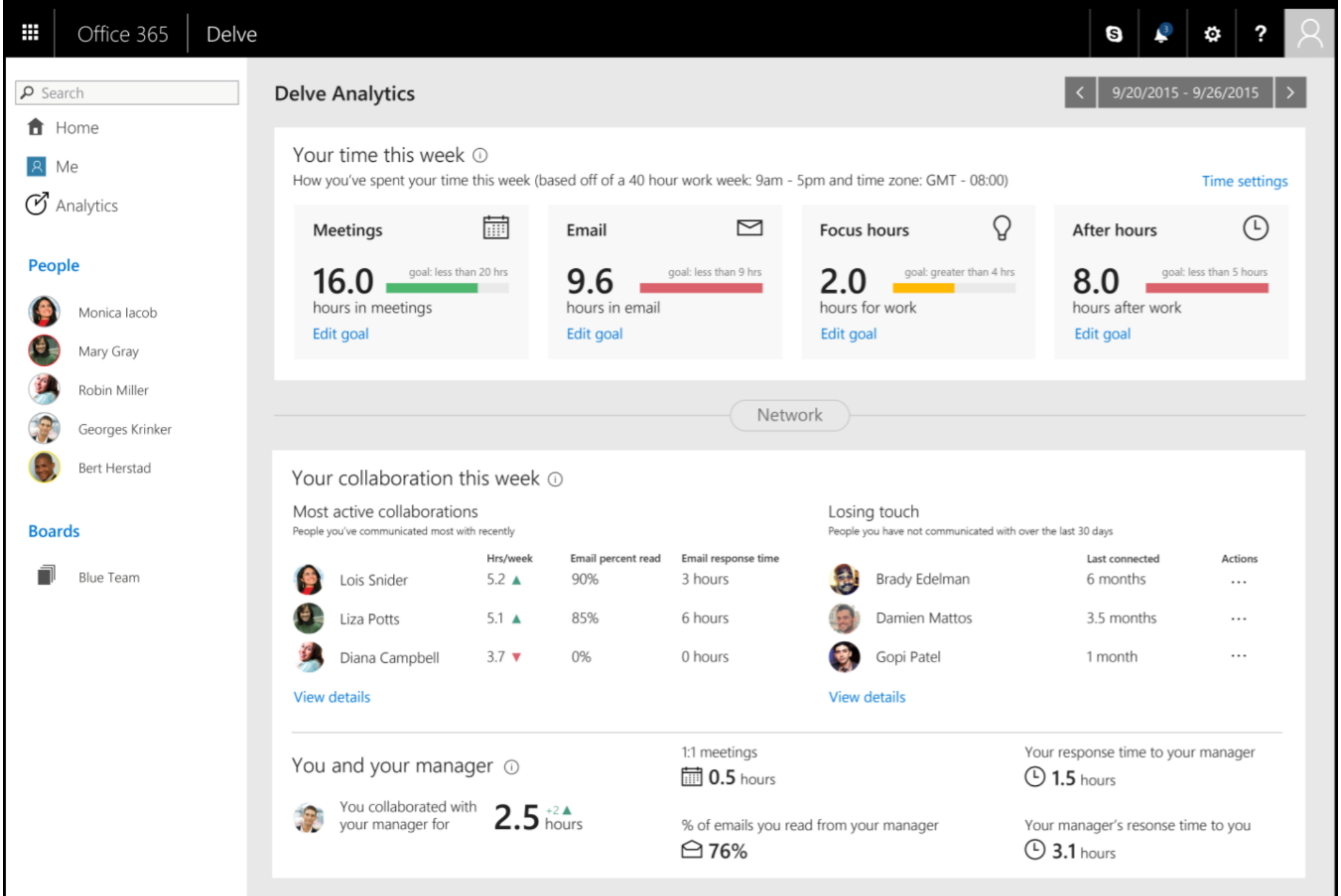
- How you spend your time
- Who you spend your time with

Delve Analytics helps you take back your time and achieve more.

Offered in the E5 SKU, and as an add-on to E1 or E3 subscriptions.

Delve Analytics inherits all Office 365 Security, Privacy and Compliance standards and commitments. Your insights are only available to you, otherwise service metadata is aggregated and anonymized and not personally identifiable.

- How many hours do I spend in Meetings?
- How many hours do I spend working after work?
- How many hours do I spend on email?
- How many hours to I spend on email compared to the rest of the organization?
- What are my most active collaborations?


Office 365
Delve
Search
Home
Me
Analytics
People
Monica Iacob
Mary Gray
Robin Miller
Georges Krinker
Bert Herstad
Boards
Blue Team
Delve Analytics
9/20/2015 - 9/26/2015
Your time this week
How you've spent your time this week (based off of a 40 hour work week: 9am - 5pm and time zone: GMT - 08:00)
Time settings
Meetings
16.0
goal: less than 20 hrs
hours in meetings
Edit goal
Email
9.6
goal: less than 9 hrs
hours in email
Edit goal
Focus hours
2.0
goal: greater than 4 hrs
hours for work
Edit goal
After hours
8.0
goal: less than 5 hours
hours after work
Edit goal
Network
Your collaboration this week
Most active collaborations
People you've communicated most with recently
Hrs/week
Email percent read
Email response time
Lois Snider 5.2 ▲ 90% 3 hours
Liza Potts 5.1 ▲ 85% 6 hours
Diana Campbell 3.7 ▼ 0% 0 hours
View details
Losing touch
People you have not communicated with over the last 30 days
Last connected
Actions
Brady Edelman 6 months ...
Damien Mattos 3.5 months ...
Gopi Patel 1 month ...
View details
You and your manager
You collaborated with your manager for 2.5 +2 ▲ hours
1:1 meetings
0.5 hours
% of emails you read from your manager
76%
Your response time to your manager
1.5 hours
Your manager's resonse time to you
3.1 hours

Email hours

9.6 +2▲ hrs

12% less than org average

4.4 hrs writing emails
5.2 hrs reading emails

Sent and received email

Percent read by others

Sent to an individual (To/CC)

Sent to a group

Percent read by you

Sent to you (To/CC)

Sent from a group

Response time to you
2.0 -0.5▼ hours

Your response time to others
3.0 +2▲ hours

Want to know how many people read a specific email? Learn more about Delve Analytics in Outlook

More

Meeting hours

16.0 +2▲ hrs

10% more than org average

8.5 hrs you scheduled
7.5 hrs others scheduled

Focus hours

2.0 +2▲ hrs

10% less than org average

After hours

8.0 -2▼ hrs

8% less than org average

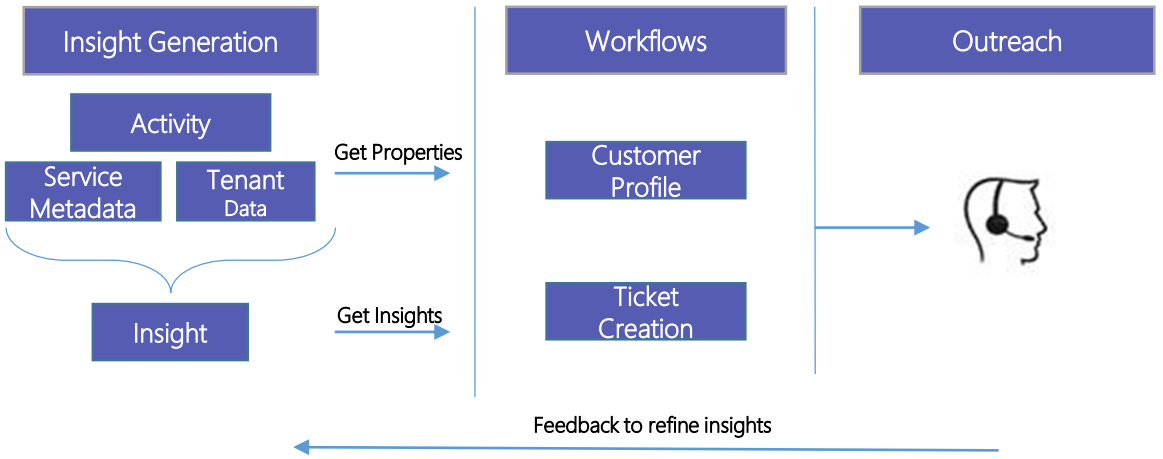
Spark Usage Patterns

Spark Batch Use Case

Support Scenario - Prevent customers who are actively using our service from getting disabled due to expired subscriptions (Dunning).

We decided to win on *satisfaction* with these customers by proactive outreach and helping customers renew the service on time.

Using spark batch analytics we flagged customers who were about to be dunned and automatically created support tickets for our support agents to act on. We also generated customer profiles so that our agents are empowered with targeted information.

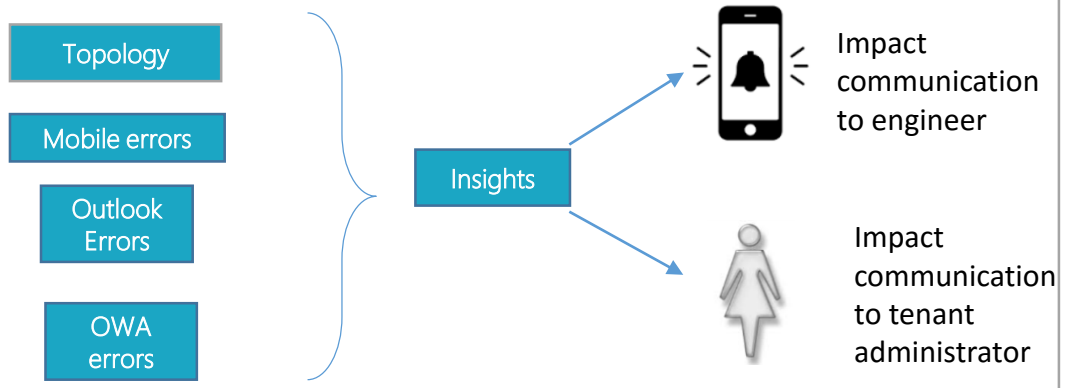


Spark Streaming Use Case

Service Scenario – Detect impact of a service incident in real time and narrowcast status to customers.

The reality of the service world is that it is subject to incidents which impact the user experience. The key is to handle them proactively and in a timely manner: alert before the service availability dips below a threshold, investigate the issue in real time and narrowcast communications to the specific set of impacted users.

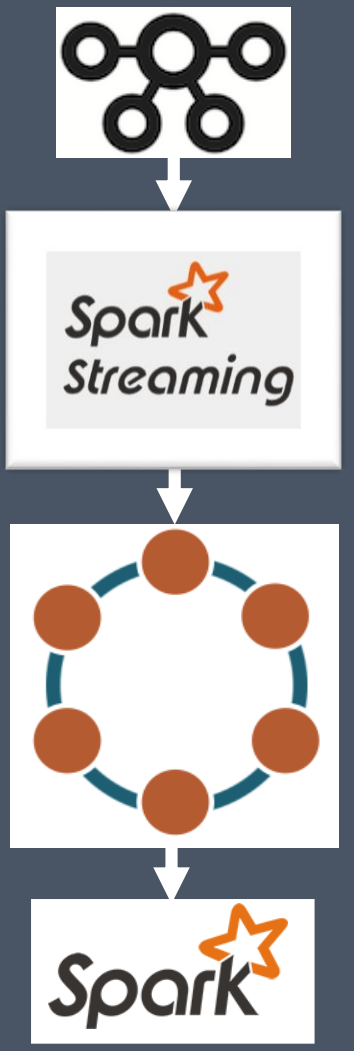
Using spark streaming we correlate the error signals with our topology to determine those who were impacted and proactively communicate with them.



Spark Usage Patterns

We have three Spark usage patterns:

- Near Real-Time Processing
- Batch Processing
- Ad-hoc Querying



Usage Pattern #1: Near Real-Time Processing

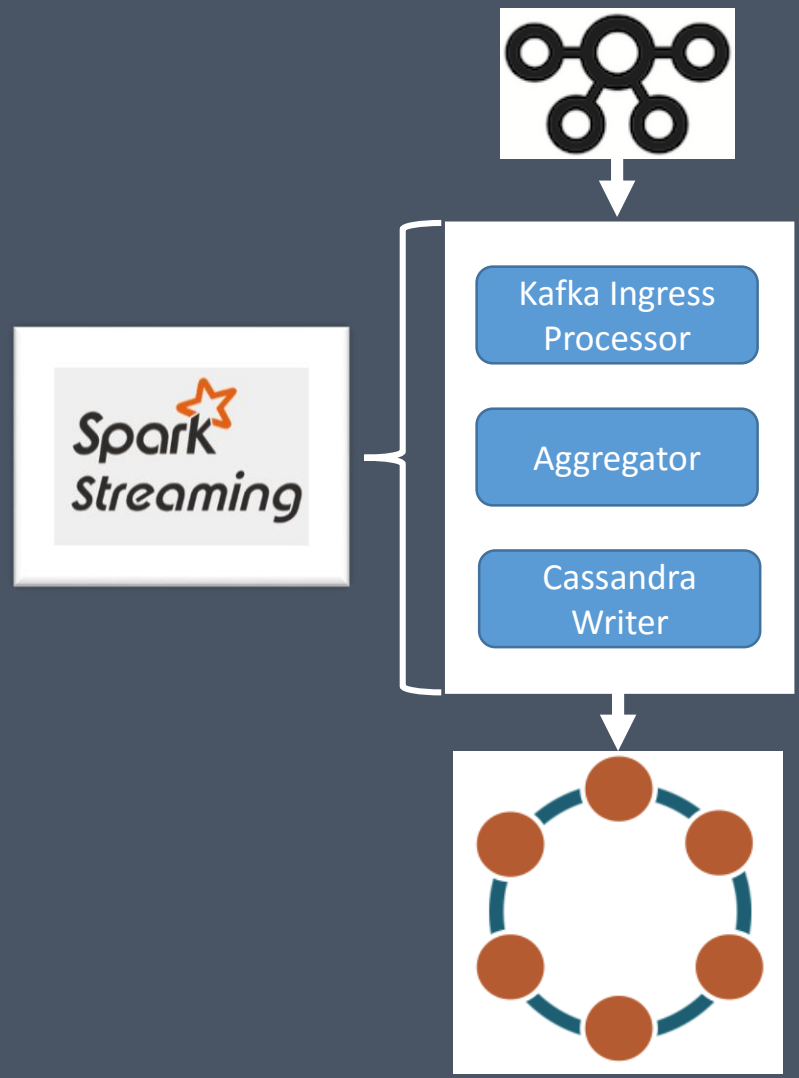
Spark Streaming jobs pipe data from one stage to another in real time.

When do we use this?

- Scenario needs to be completed near real-time
- Event disorders, late events or event drops are accepted
- Don’t have a big look back window

Pros: Less data stored in Cassandra; Near Real-time;

Cons: If system is unhealthy, since the buffering window is small, there is no easy way to recover the data.



Usage Pattern #2: Batch Processing

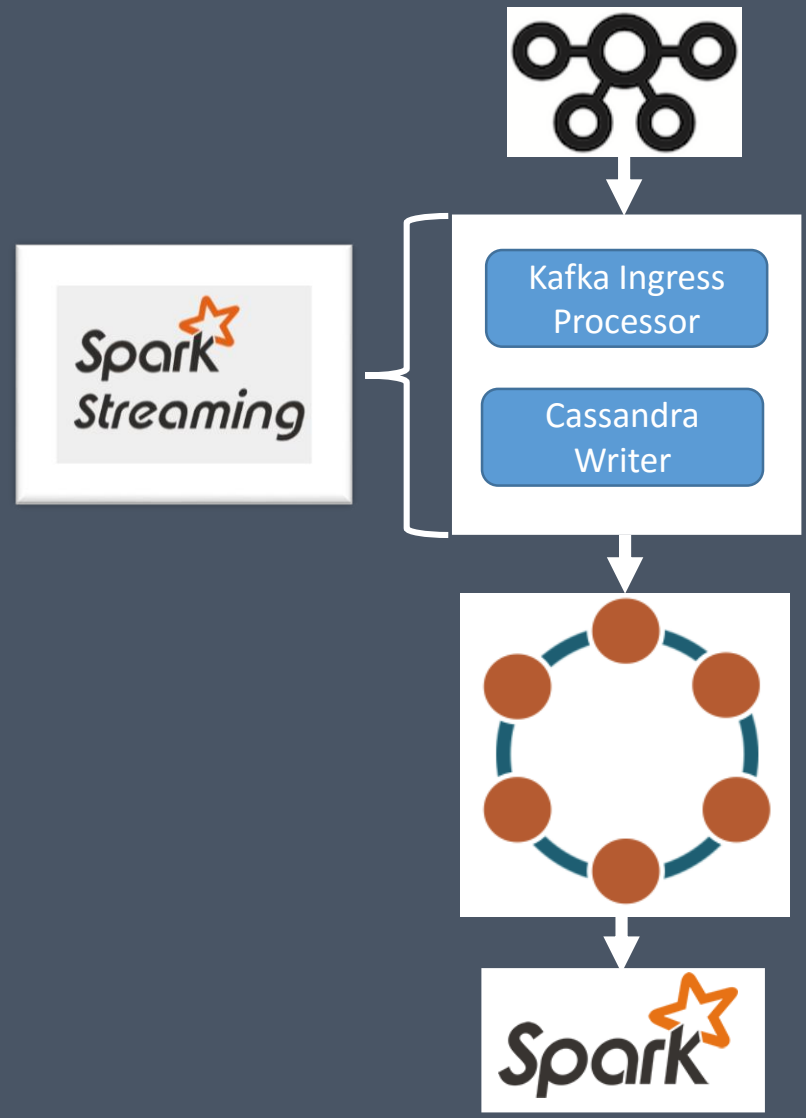
Spark Streaming jobs move the raw data from Kafka, do simple data conversion and output processed raw data to Cassandra.

Spark Batch Jobs do further aggregations and analysis.

When do we use this?

- Event accuracy and order is very important to the stream
- Need to look back a few days / weeks / months of data for trends
- Provide a common datasets for other jobs to leverage
- Complicated joins with multiple datasets to produce rich insights

Pros: High data accuracy; Can easily recover from issues; Complicated analytics like TopN become feasible; Allows other jobs to reuse the common curated datasets



Usage Pattern #3: Ad-hoc Querying

Query data through Zeppelin which supports spark interpreters.

When do we use this?

- Explore valuable existing insights for planning
- Validate data to ensure accuracy
- Ad-hoc data access. Dream up a query and run it!

Pros: Flexible; Democratizes access to rich insights;

Thank You