Observability for Data Pipelines

Jiaqi Liu, Software Engineer at Button

Director, @WomenWhoCodeNYC

@jiaqicodes

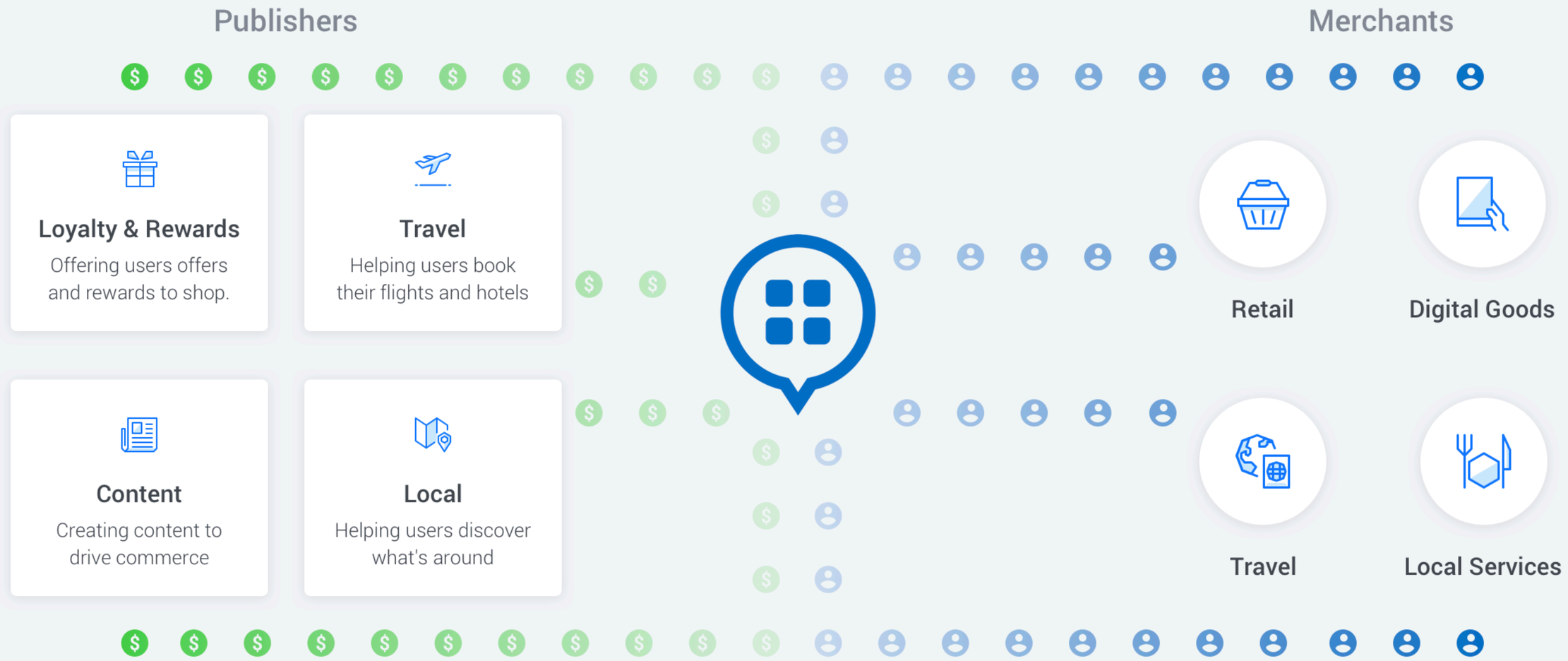
#OSCON

Button

usebutton.com

Our Marketplace

Button helps Publishers and Merchants build partnerships to earn revenue and acquire users through a mobile marketplace. The best brands in mobile use Button to enable and power mobile commerce partnerships.



Agenda

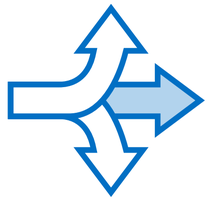
- Data Pipelines
- Challenges with Data Pipelines
- Designing Features:
  - Immutable Data
  - Dry Run Mode
  - Data Lineage
- Testing, Monitoring & Alerting

Data Pipeline

Button

ETL Pipeline

Extract data from a source, this could be scraping from a site, a large file, a realtime stream of data feeds.



Transform the data - this could be joining the data with additional information for an enhanced data set, running through a machine learning model, or aggregating the data in some way.

Load the data into a data warehouse or a User facing dashboard - wherever the end storage and display for data might be.

Data Pipeline Example

Button

In [174]:
```
import requests
from requests_html import HTMLSession
session = HTMLSession()
# get talks from oscon site
r = session.get('https://conferences.oreilly.com/oscon/oscon-or/public/schedule/speakers')
```

In [175]:
```
r.text
```

```
orry about anyone asking, &amp;quot;How long will that take?&amp;quot; Laura Bernardin Janusek explores how to use Agile, con
sensus-based estimation technique Planning Poker to generate thoughtful and data-backed estimations for any product b
uild.</p>\r\n          </div>\r\n          \r\n      </div>\r\n</div>\r\n\r\n<div id="speaker322954" class="speaker" dat
a-groups=\'["all", "Open@Amazon","Open Source","Open@Amazon ","Sponsored" ]\' data-id="322954" data-name="Zaheda Bhor
at" data-affiliation="Amazon Web Services" data-speaker-link="/oscon/oscon-or/public/schedule/speaker/322954" data-ph
oto-url="https://cdn.oreillystatic.com/en/assets/1/eventprovider/1/_%40user_322954.jpg" data-twitter="zahedab" >\r\n<
div class="speaker-bio hide"><p>Zaheda Bhorat is the head of open source strategy at <span class="caps">AWS</span>. A
computer scientist, Zaheda is a longtime active contributor to open source and open standards communities. Previousl
y, she shaped the first-ever open source program office at Google; launched successful programs, including Google Sum
mer of Code; and represented Google on many industry standards executive boards across multiple technologies. She als
o served as a senior technology advisor for the Office of the <span class="caps">CTO</span> at the UK Government Digi
tal Service, where she co-led the open standards policy, which is in use by the UK government on open document format
s. Zaheda was responsible for OpenOffice.org, and later NetBeans.org, at Sun Microsystems, where she built a thriving
global volunteer community and delivered the first user version, OpenOffice 1.0. Zaheda is passionate about technolog
y, education, open source, and the positive impact of collaboration for social good. She serves on the UK governmen
t's Open Standards Board, which determines the standards government should adopt. She also serves on the board of dir
ectors of the Mifos Initiative, an open source effort that is positioning financial institutions to become digitally
 connected providers of financial services to the poor. Zaheda speaks internationally on topics related to open sourc
e and social good.</p></div>\r\n<div class="polaroid-content" name="Zaheda Bhorat" affiliation="Amazon Web Services">
\r\n      <label tabindex=0 for="op"><img src="//cdn.oreillystatic.com/oreilly/images/transparent-1px.png" class="la
```

In [109]:
```
def parse_talk(talk):
    data = {
        'title': talk.find('a.proposal-title', first=True).text,
        'link': talk.links,
        'session_type': talk.find('span.session_type_tag', first=True).text,
        'description': talk.find('p', first=True).text
    }

    return data
```

In [110]:
```
parse_talk(talks[0])
```

Out[110]:
```
{'title': 'A decentralized reference architecture for cloud native applications (sponsored by Ballerina)',
 'link': {'/oscon/oscon-or/public/schedule/detail/77834'},
 'session_type': 'Session',
 'description': "Asanka Abeysinghe explores cell-based architecture, a self-contained composable unit of architectur
e. The cell is independently scalable. It's independently deployable. It's independently governed. It's part of an ec
osystem of cells. A cell-based architecture is a common pattern that any enterprise can connect architecture, impleme
ntation, and deployment by making autonomous development teams."}
```

In [172]:
```
def parse_speaker(speaker):
    data = {
        'talk' : parse_talk(speaker.find('div.proposal', first=True)),
        'bio' : speaker.find('div.speaker-bio', first=True).text,
        'name' : re.search('data-name=\"(.*)\" data-photo', speaker.html).group(1),
        'twitter' : re.search('data-twitter=\"(.*)\" id=', speaker.html).group(1),
    }

    return data
```

In [173]:
```
parse_speaker(speakers[0])
```

Out[173]:
```
{'talk': {'title': 'A decentralized reference architecture for cloud native applications (sponsored by Ballerina)',
  'link': {'/oscon/oscon-or/public/schedule/detail/77834'},
  'session_type': 'Session',
  'description': "Asanka Abeysinghe explores cell-based architecture, a self-contained composable unit of architectur
e. The cell is independently scalable. It's independently deployable. It's independently governed. It's part of an ec
osystem of cells. A cell-based architecture is a common pattern that any enterprise can connect architecture, impleme
ntation, and deployment by making autonomous development teams."},
 'bio': 'Asanka Abeysinghe's goal is to connect humans and technology by helping organizations implement digital tran
sformation programs that result in consumer-driven digital applications. He drives efforts to create, refine, and enh
ance WSO2's corporate reference architecture and is responsible for spearheading a thought-leadership outlook that de
fines WSO2's corporate reference methodology for development, customer success, and implementation. Working closely w
ith customers, partners, and analysts, he evangelizes WSO2's technology vision. Asanka has over 20 years of industry
experience, which includes designing and implementing highly scalable distributed systems, SOA and microservice archi
tectures in the financial domain, mobile platforms, and various business integration solutions. He is also a committe
r of the Apache Software Foundation. Asanka is a regular speaker at numerous events, including ApacheCon, QCon, O'Rei
lly, API Days, API Strategy, LinuxFoundation, Gartner, WSO2Con, and many tech meetups.',
 'name': 'Asanka Abeysinghe',
 'twitter': 'asankama'}
```

```
oscon_2019_talks = []
for elem in speakers:
    oscon_2019_talks.append(parse_speaker(elem))
```

```
len(oscon_2019_talks)
```

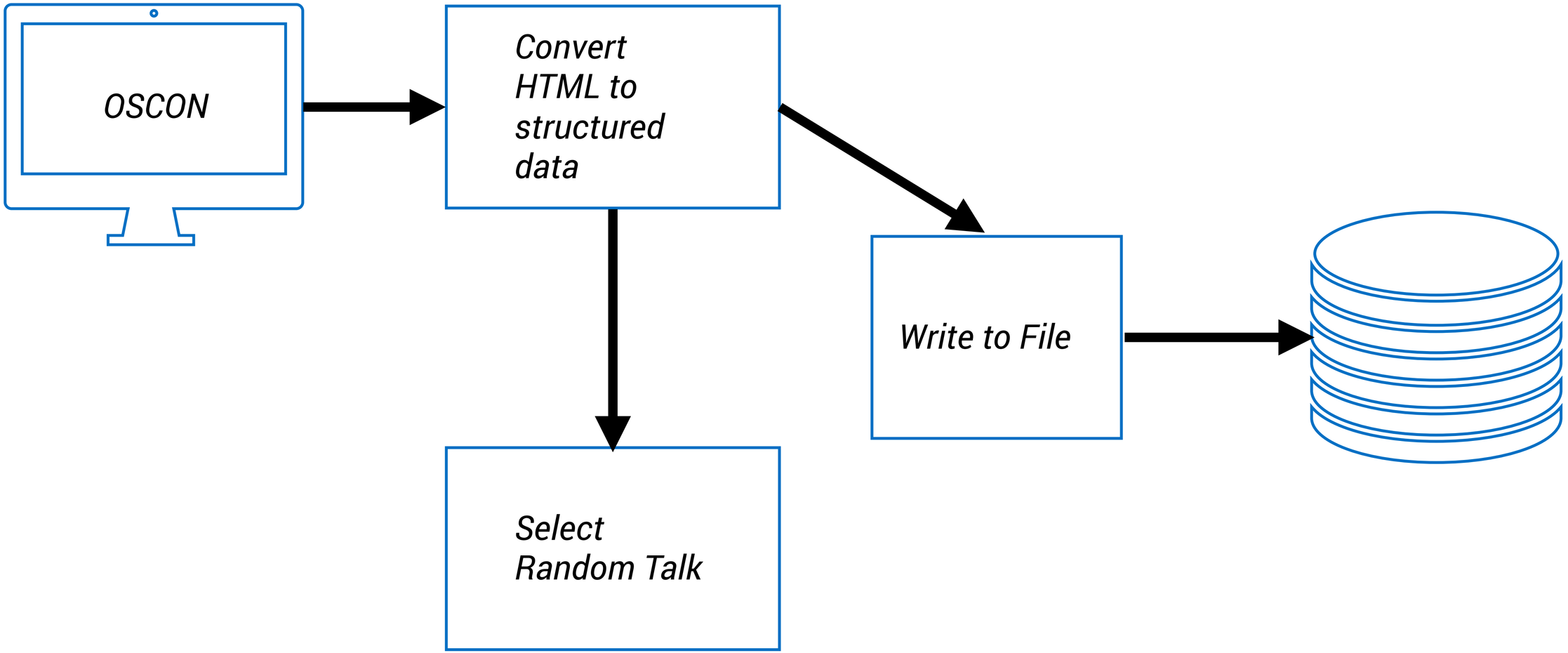
```
259
```

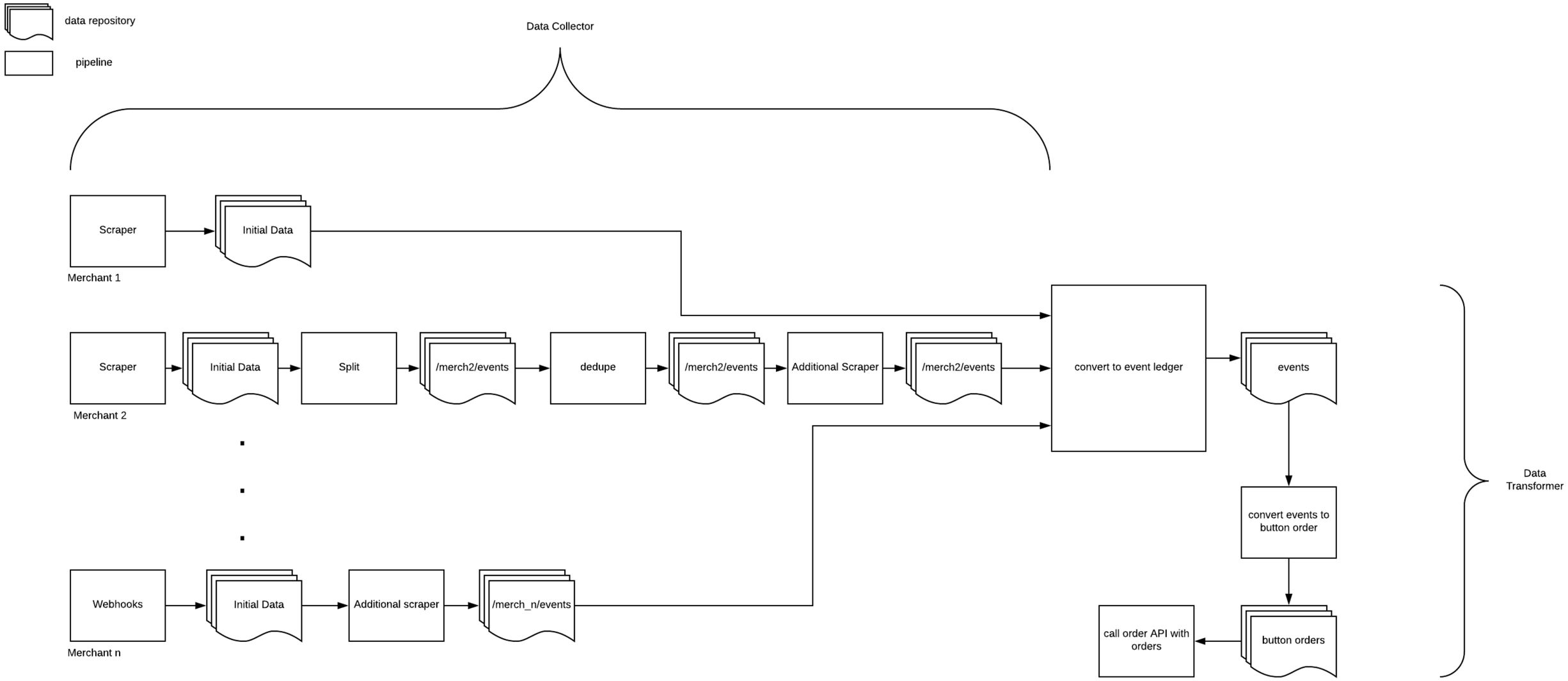
In [182]:
```
import random
import pprint
pp = pprint.PrettyPrinter(indent=4)
pp.pprint(random.choice(oscon_2019_talks))
```

```
{   'bio': 'Simon St.Laurent is a content manager at LinkedIn Learning, '
           'focusing primarily on the client side of the web. He’s a past '
           'cochair of the Fluent and OSCON conferences. He’s authored or '
           'coauthored books including Programming Crystal, Introducing '
           'Elixir, Introducing Erlang, Learning Rails 3, XML Pocket '
           'Reference, 3rd edition, XML: A Primer, and Cookies.',
    'name': 'Simon St.Laurent',
    'talk': {   'description': 'You program in Ruby but long for greater '
                               'concurrency. You dream of programs that run '
                               'instantly when called. You wish everything had '
                               'clearly defined types. Join Simon St.Laurent '
                               'to learn how Elixir and Crystal refine the '
                               'diverse approaches built into Ruby, modifying '
                               'the syntax and structures of this commonly '
                               'understood language to address their very '
                               'different priorities.',
                'link': {'/oscon/oscon-or/public/schedule/detail/75715'},
                'session_type': 'Session',
                'title': 'Children of Ruby: The different worlds of Elixir and '
                         'Crystal'},
    'twitter': 'simonstl'}
```

```
import csv
fieldnames = oscon_2019_talks[0].keys()

with open('oscon_2019_talks.csv', 'w') as csvfile:
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
    writer.writeheader()
    for tx in oscon_2019_talks:
        writer.writerow(tx)
```

	A	B	C	D
1	talk	bio	name	twitter
2	{'title': 'A decentralized reference architecture for cloud	Asanka Abeysinghe,Äôs goal i	Asanka Abeysinghe	asankama
3	{'title': 'Level up your web apps with WebAssembly', 'lin	Robert Aboukhalil is a bioinfo	Robert Aboukhalil	RobAboukha
4	{'title': 'Enabling community-driven development', 'link':	Aaron Aldrich is a community	Aaron Aldrich	crayzeigh
5	{'title': 'If only production incidents could speak', 'link': {'	Subbu Allamaraju is a vice pr	Subbu Allamaraju	sallamar
6	{'title': 'Measuring embedded machine learning', 'link': {	Alasdair Allan is a director at	Alasdair Allan	aallan
7	{'title': 'Practical DevOps for the busy data scientist: Alic	Tania Allard (she/her) is a clo	Tania Allard	
8	{'title': 'Open source citizenship', 'link': {'/oscon/oscon-o	Cat Allman is an open source	Cat Allman	catallman
9	{'title': 'Optimizing analytical queries on Cassandra by 1(	Shradha Ambekar is a staff sc	Shradha Ambekar	shradha151
10	{'title': 'Introduction to Kubeflow Pipelines', 'link': {'/osc	Dan Anghel is a strategic clou	Dan Anghel	
11	{'title': 'Getting traction for digital transformation', 'link'	Mike Angstadt is the Director	Michael Angstadt	
12	{'title': 'Keeping open source open', 'link': {'/oscon/oscor	Matt Asay is the head of deve	Matt Asay	mjasay
13	{'title': 'A community of communities: Empowering mai	Ben Balter is a senior manag	Ben Balter	
14	{'title': 'Building modern APIs with GraphQL', 'link': {'/os	Alex Banks is a software engi	Alex Banks	moontahoe


OSCON
Convert HTML to structured data
Write to File
Select Random Talk


data repository
pipeline
Data Collector
Scraper
Initial Data
Merchant 1
Scraper
Initial Data
Split
/merch2/events
dedupe
/merch2/events
Additional Scraper
/merch2/events
Merchant 2
Webhooks
Initial Data
Additional scraper
/merch_n/events
Merchant n
convert to event ledger
events
convert events to button order
button orders
call order API with orders
Data Transformer

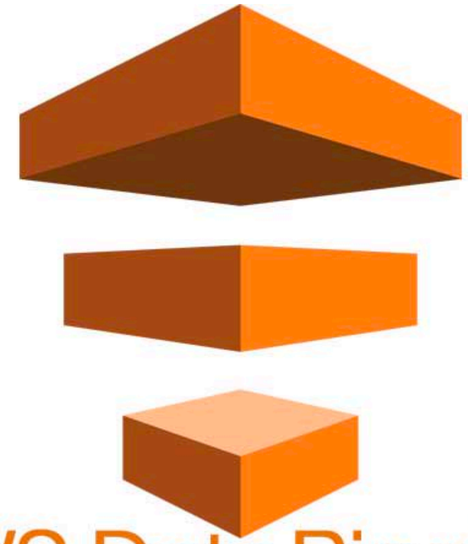
Batch

Periodic Process that reads data in bulk (typically from a filesystem or a database)

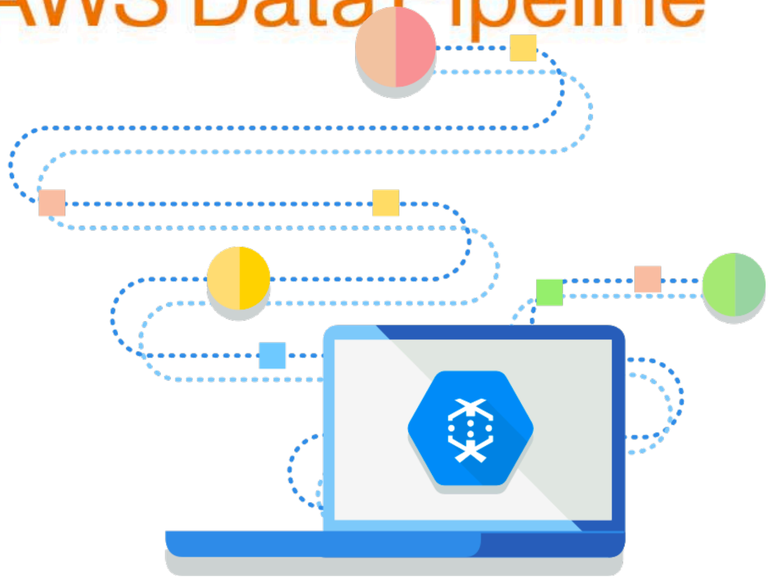
Stream

High throughput, low latency system that reads data from a stream or a queue

Luigi



Apache Airflow



Google Dataflow

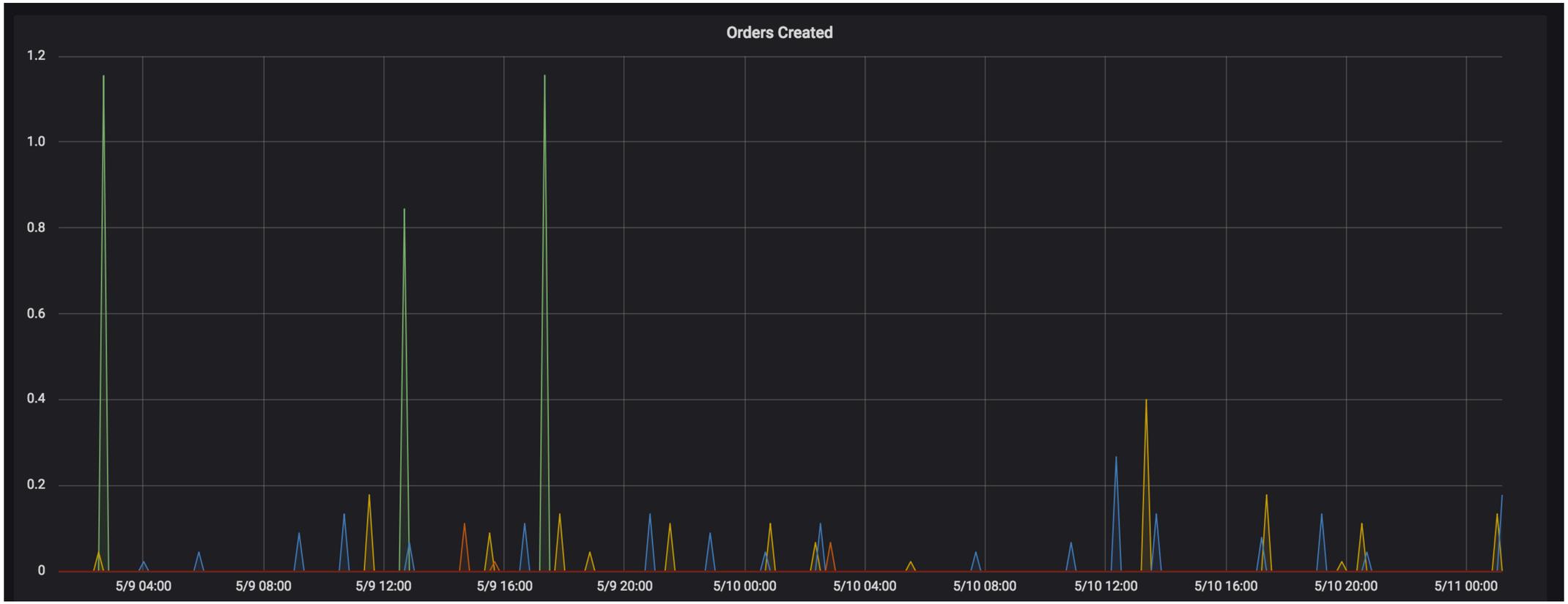
What Could Go Wrong?

Button

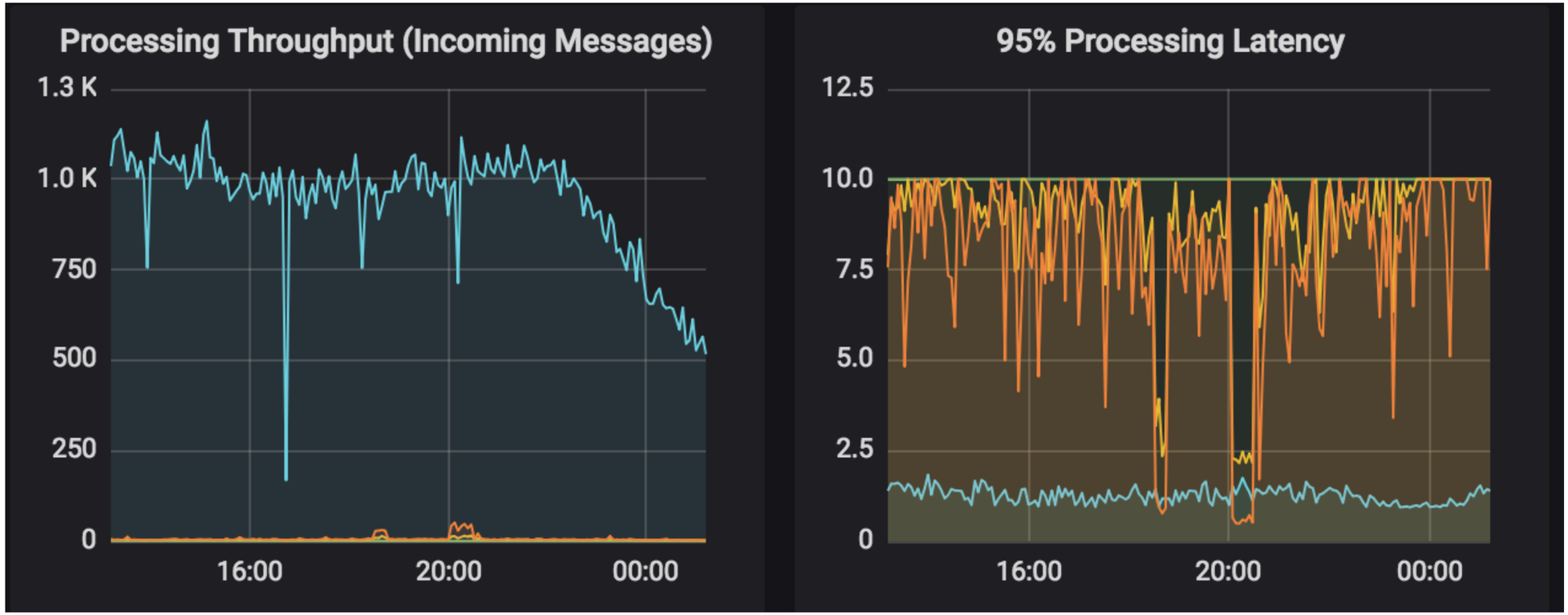
Problems

- Batch Job is never scheduled
- Batch Job takes too long to run
- Data is malformed or corrupt
- Data is lost
- Stream is backed-up, Stream data is lost
- Non-deterministic models

Batch Jobs


Orders Created
1.2
1.0
0.8
0.6
0.4
0.2
0
5/9 04:00
5/9 08:00
5/9 12:00
5/9 16:00
5/9 20:00
5/10 00:00
5/10 04:00
5/10 08:00
5/10 12:00
5/10 16:00
5/10 20:00
5/11 00:00

Stream Jobs


Processing Throughput (Incoming Messages)
1.3 K
1.0 K
750
500
250
0
16:00
20:00
00:00
95% Processing Latency
12.5
10.0
7.5
5.0
2.5
0
16:00
20:00
00:00

Data Pipeline Concerns

Delayed Processing

Speed in Data Processing could be Core to Business

Data Integrity

Data is exposed or lost or malformed. A statistical model is producing highly inaccurate results

It’s not enough to know that the pipeline is healthy, you also have to know that the data being processed is **accurate**.

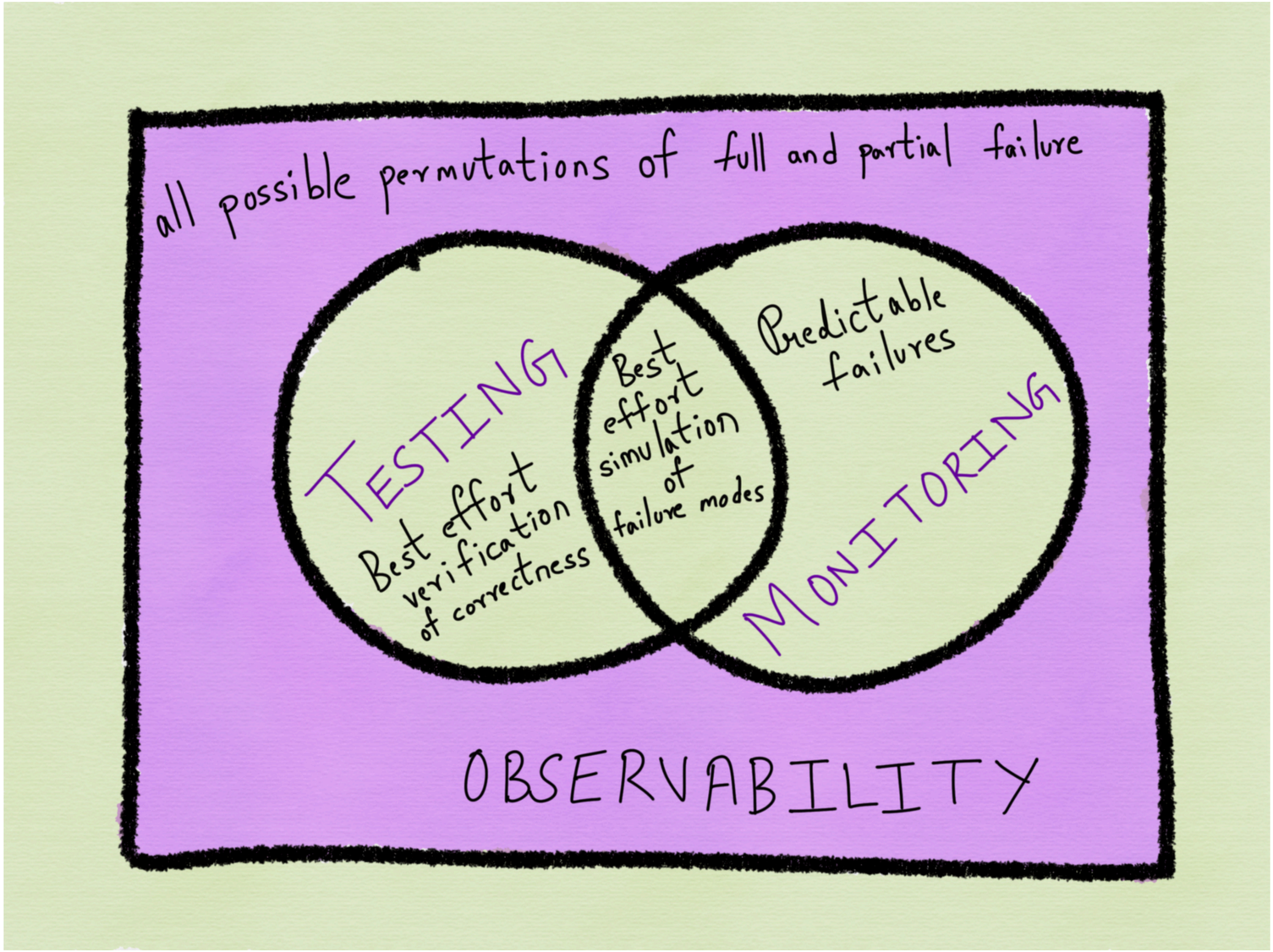
Build data pipelines that support interpretability and observability

Interpretability

Not just understand what a model predicted but also **why**.

Allows for **debugging** and **auditing** machine learning models.

Because Button is a marketplace, we see the side effects of user behavior in our data and have to decipher what assumptions are safe to make.



Cindy Sridharan - https://medium.com/@copyconstruct/monitoring-and-observability-8417d1952e1c

Mystery Free Prod

Pipeline Features

Build feature to support Interpretability and observability

Button

Features to Include

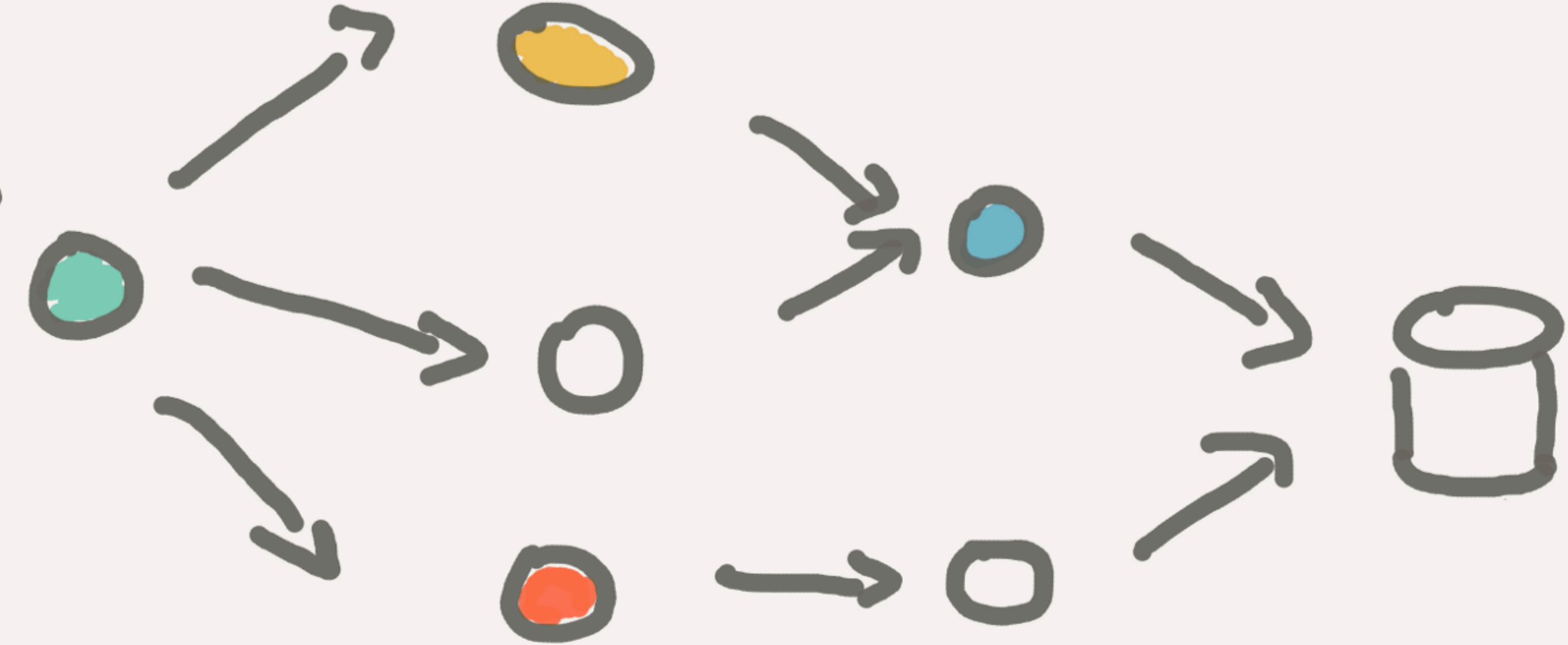
- Immutable Data
- Data Lineage
- Having a Test Run Feature

Immutable Data

Reproducible Outcomes

transaction_id	created_dt	account_id	transaction_type	amount
1	2019-01-15 01:00:24.032473	12345	credit	100
2	2019-01-30 10:01:07.683552	12345	debit	-5
3	2019-02-01 11:01:28.153952	12345	debit	-10

Data Lineage → Diagnostics



Tag Records With Metadata (version of code, source of data)

Log to a distributed tracer (use consistent unique identifiers to track)

Test Run

Validate Assumptions

What assumptions did you make about your data?

Schema

```
schema = {
    'bio': string
    'name': string
    'talk': {
        'description': string
        'link': string
        'session_type': string, // enum
        'title': string
        },
    'twitter': string}
}
```

Protocol buffers are a language-neutral, platform-neutral extensible mechanism for serializing structured data.

```
message Talk {
    required string description = 1;
    required string link = 2;

    enum SessionType {
        SESSION = 0;
        EVENT = 1;
        KEYNOTE = 2;
    }

    required SessionType type = 3 [default = SESSION];
    required string title  = 4;
}
```

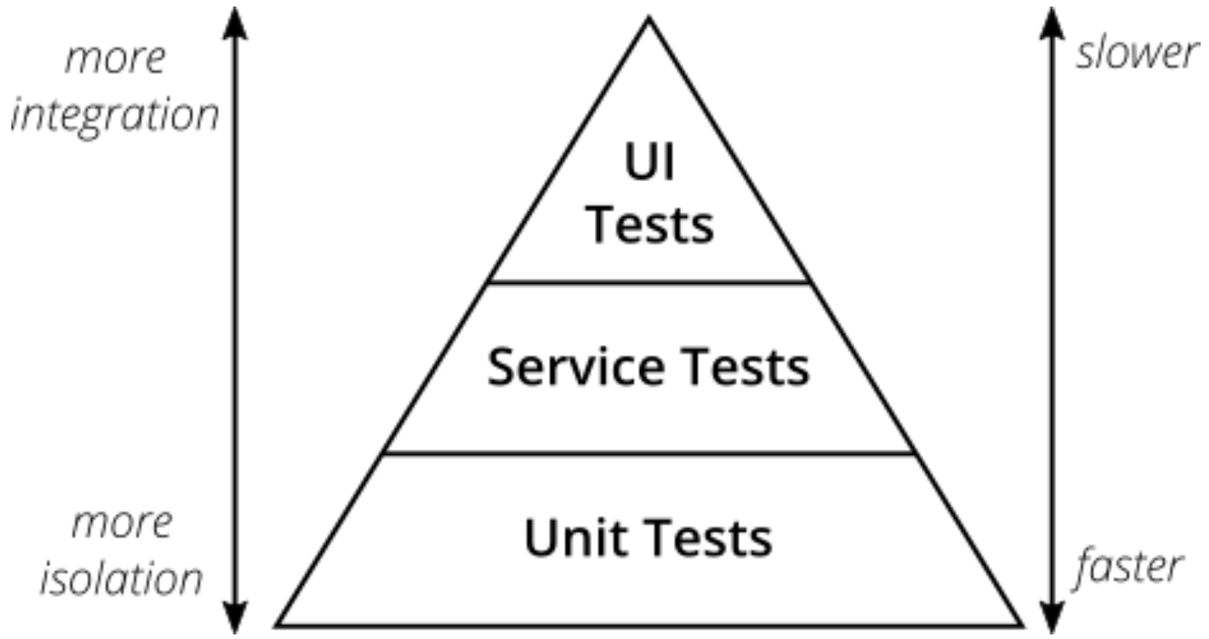
```
message Speaker {
    required string bio = 1;
    required string name = 2;
    required string twitter = 3;
    repeated Talk talk = 4;
}
```

Ability to test output of data transformation before committing to database

Testing

Button

Test Pyramid



https://martinfowler.com/articles/practical-test-pyramid.html

Different Types of Tests

Unit Test

Functional

Regression

System

Unit Tests

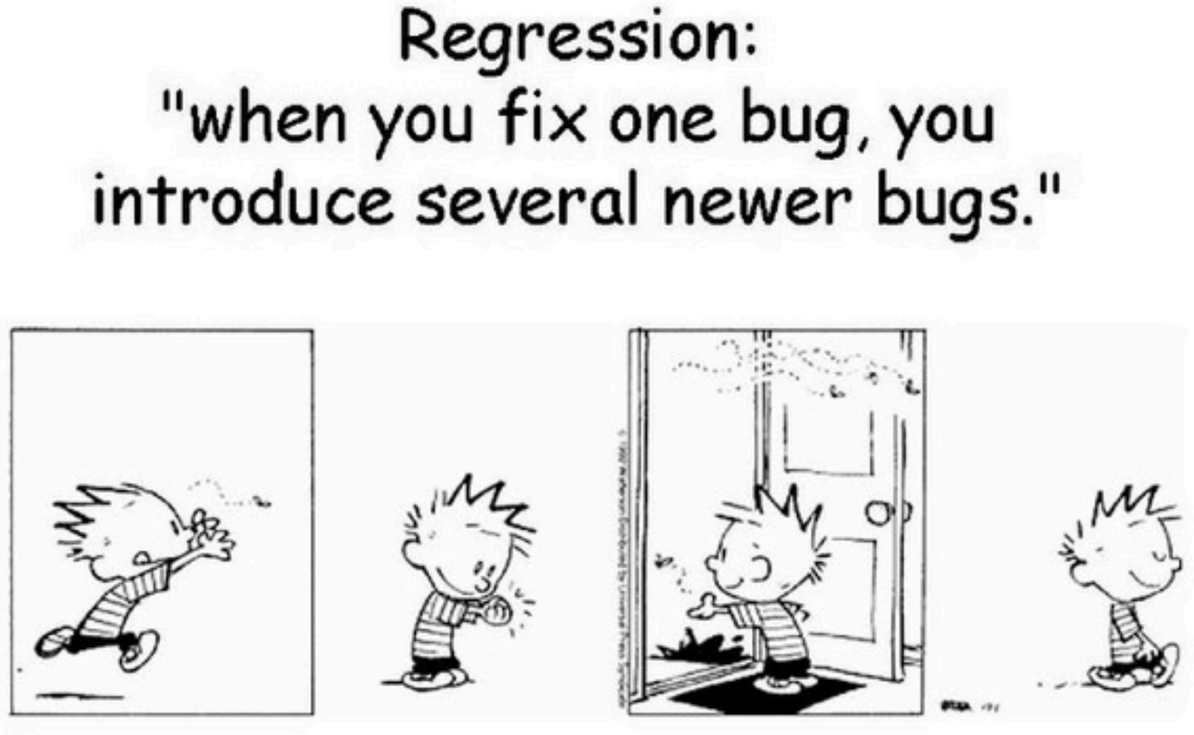
```python
import unittest
import mock

from common.decorators import retry


class ExponentialRetryTestCase(unittest.TestCase):
    def test_when_does_not_fail(self):
        def f():
            return "result"

        f = retry()(f)
        res = f()
        self.assertEqual("result", res)

    def test_retries_when_fails(self):
        m = mock.MagicMock(side_effect=Exception())
        f = retry(max_retries=3, delay=0.01)(m)
        self.assertRaises(Exception, f)
        self.assertEqual(3, m.call_count)
```

Functional Tests

- Can also be known as Integration Tests
- In the case of data, it’s the golden tests
  - Sets the gold standard for data in and data out
  - Doesn’t need to be logic specific like Unit Tests are
  - Build a Golden Test framework and define fixtures (expected input, and expected output)

Failed Gold Test

Regression Tests



https://www.ibeta.com/regression-testing-nutshell/

Champion/Challenger Model

93%

Model A Precision

95%

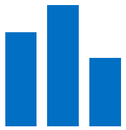
Model B Precision

System Tests

Black Box

Measures Health of Whole System

Continuous



Relies on Other Signals

Monitoring & Testing

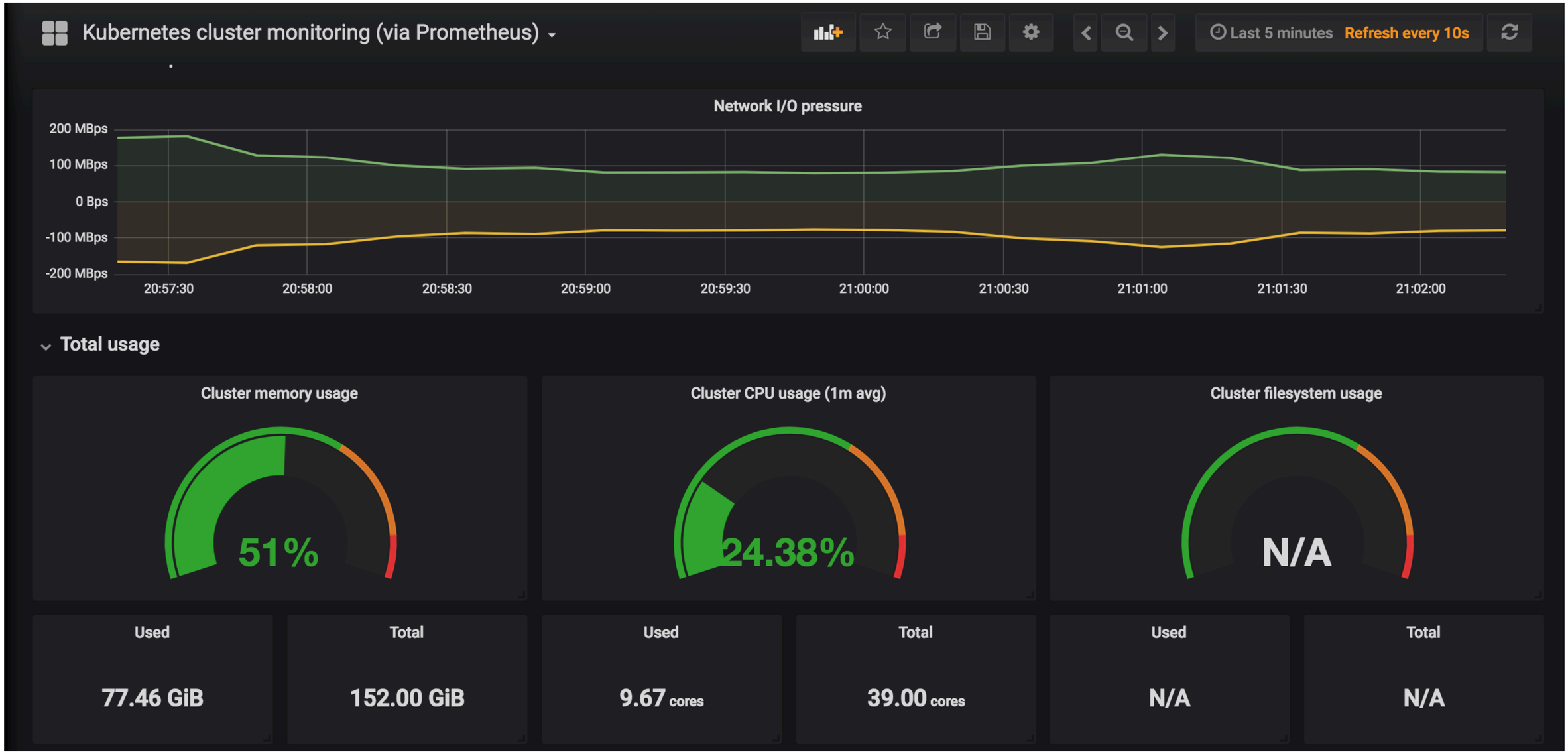
	Web Service	Data Pipeline
Health Check	Have some kind of health check endpoint and check that when you ping `/healthcheck` you get a 200 status code	Check that a job has succeeded
Integration Test	`POST` to one endpoint and expect to get the correct data from a corresponding `GET` endpoint	Verify some fake data made its way through the data transformation *This can be hard to replicate if there's no easy way to feed fake data into the data pipeline
Latency	Measure the average response time of an API	Measure time it takes for data pipeline to complete

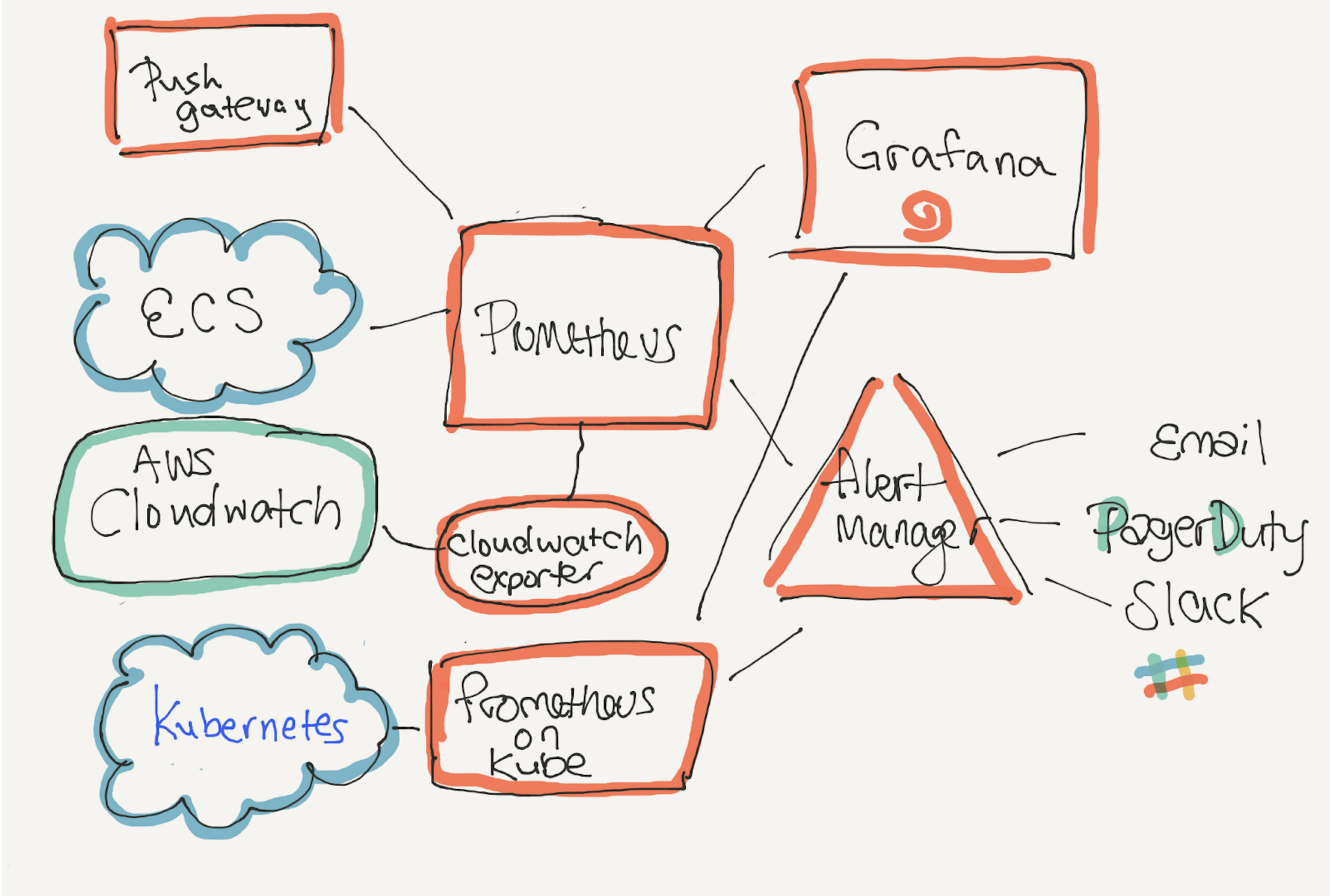
Confidential

Monitoring

Button

Monitoring Tools


Kubernetes cluster monitoring (via Prometheus)
Last 5 minutes
Refresh every 10s
Network I/O pressure
200 MBps
100 MBps
0 Bps
-100 MBps
-200 MBps
20:57:30
20:58:00
20:58:30
20:59:00
20:59:30
21:00:00
21:00:30
21:01:00
21:01:30
21:02:00
Total usage
Cluster memory usage
51%
Cluster CPU usage (1m avg)
24.38%
Cluster filesystem usage
N/A
Used
77.46 GiB
Total
152.00 GiB
Used
9.67 cores
Total
39.00 cores
Used
N/A
Total
N/A


Push gateway
ECS
AWS Cloudwatch
Cloudwatch exporter
Kubernetes
Prometheus on kube
Prometheus
Grafana
Alert Manager
Email
PagerDuty
Slack

Metric Types

- Counter - single monotonically increasing counter
- Gauge - single numerical value that can go up or down (think temperature)
- Histogram - samples observations and counts them in buckets
- Summary - samples observations and calculates quantiles over a sliding window

Time Series Metrics

Metrics to be scraped by prometheus:

```
job_last_success_unixtime = prometheus_client.Gauge('job_last_success_unixtime',
    'Time of last successful batch job')
job_duration_seconds = prometheus_client.Summary('job_duration_seconds',
    'Duration of batch job run in seconds')
```

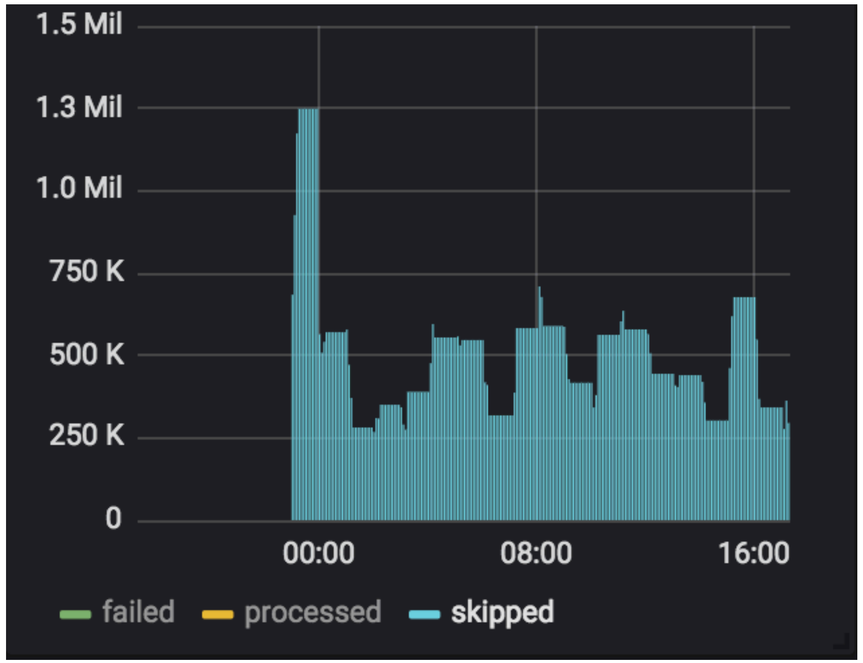
Metrics are calculated at the end of the pipeline as such:

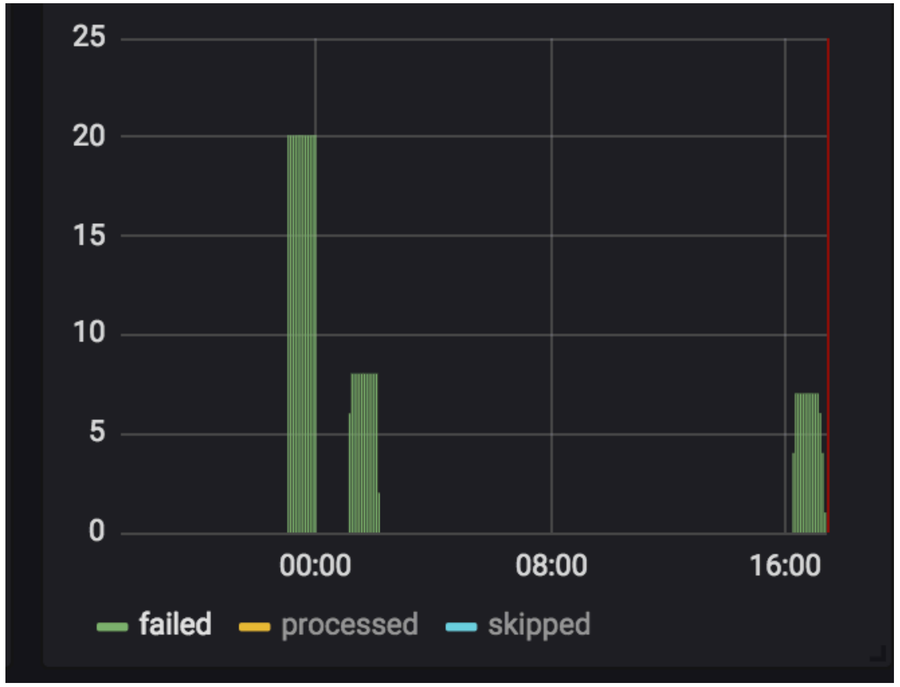
```
with job_duration_seconds.time():
  run_pipeline()
  time_now = int(time.time())
  job_last_success_unixtime.set(time_now)
```

Alerting

```
ALERT BatchJobFailed
    IF time() - job_last_success_unixtime > (3 * 60 * 60)
    FOR 15m
    LABELS { severity="high", owner="data-platform" }
    ANNOTATIONS {
      summary = "The batch job has failed.",
      description = "The {{ $labels.job }} has not succeeded in over 3 hours",
      playbook = "https://button.phacility.com/w/engineering/alerts/#batchjobfailure"
    }
```

Monitoring with Time Series Data





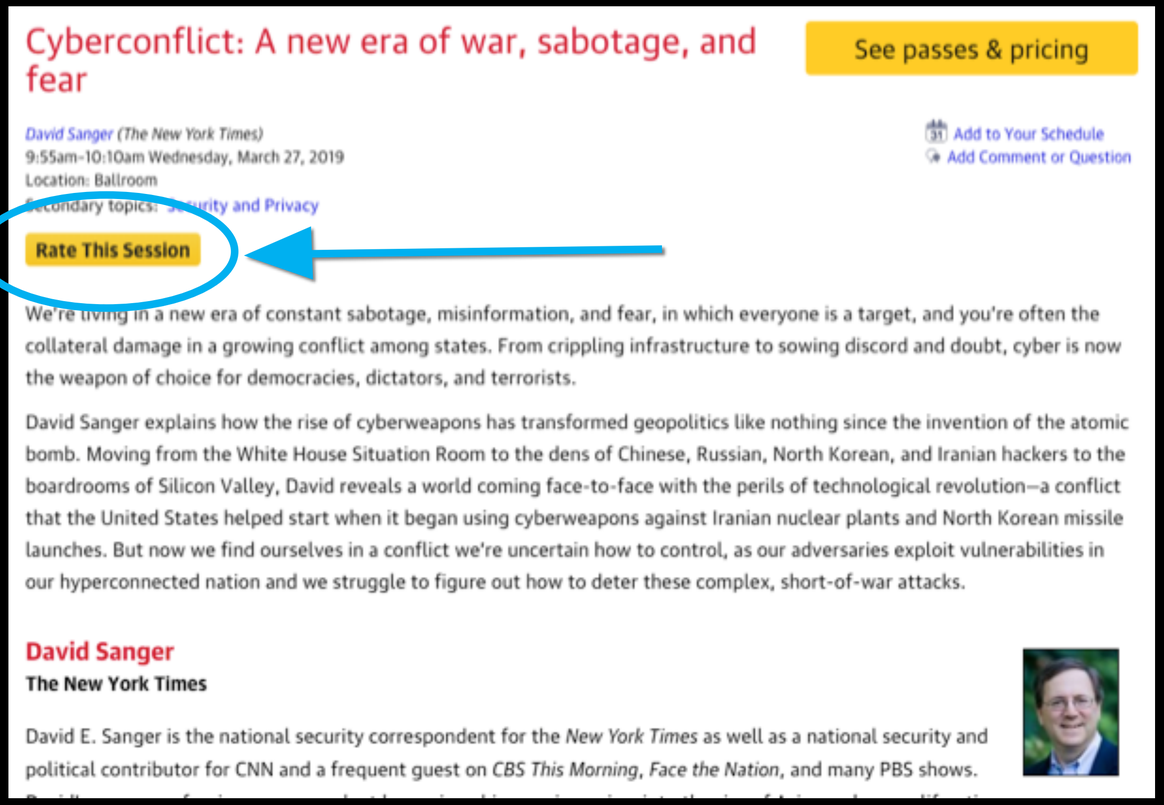
Set a threshold that works for you. Establish a baseline and go from there.

Page on symptoms not root causes. Create trail of causes for diagnostics

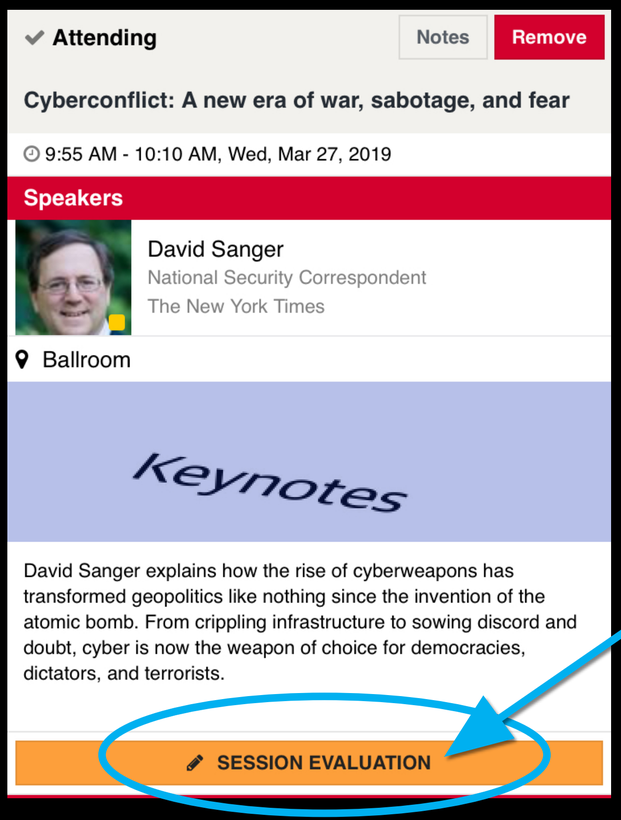
Data Lineage, Immutable Data, Test Run -> Ease of development when working with evolving data

Monitoring & Alerting -> Overall Pipeline Observable

Rate today’s session



Session page on conference website



O’Reilly Events App

Thanks! We’re hiring!

www.usebutton.com

jiaqi@usebutton.com

Button