

Scaling Data Infrastructure @ Spotify

matti@spotify.com

kalvans@spotify.com





Mārtiņš Kalvāns
kalvans@spotify.com



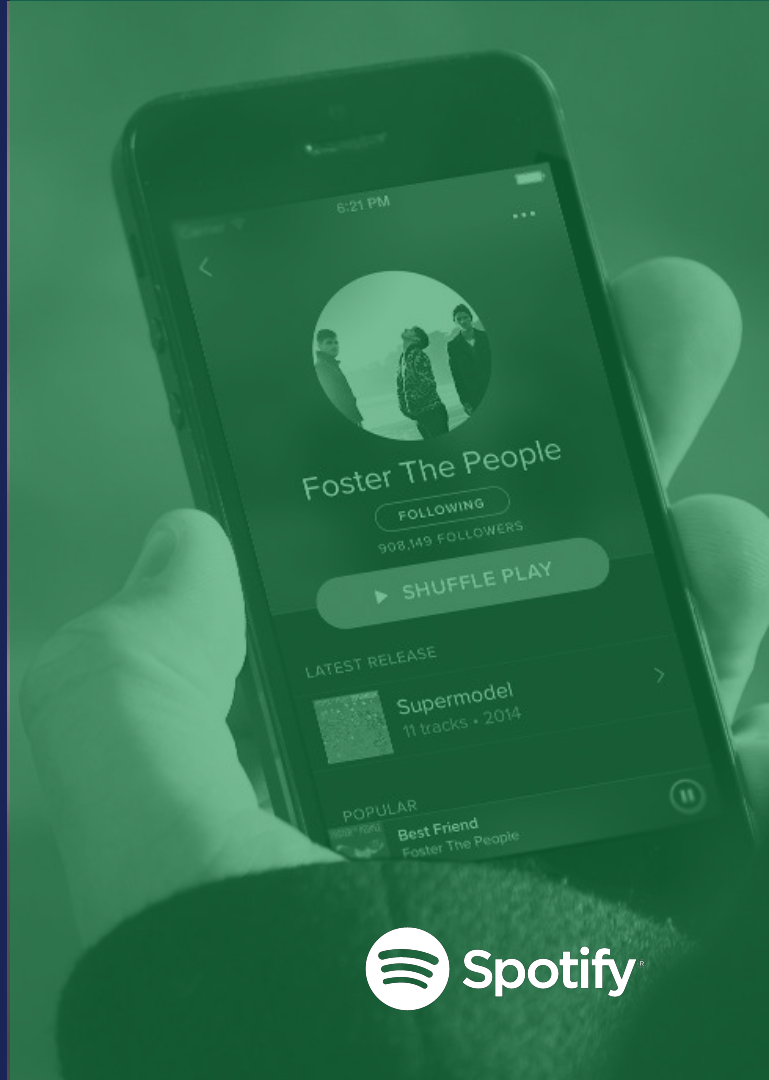
Matti Pehrs
matti@spotify.com

Agenda

1. **Data at Spotify**
2. **Summer of 2015**
3. **Challenges & Victory**
 - Datamon
 - Styx
 - GABO

Spotify big-data context

- Over 100 million monthly active users
- Over 30 million song
- Over 2 billion playlists
- Active in 60 markets



Data is at the heart of Spotify

In 2007

- Monthly Royalty Report

In 2016

- Monthly Royalty Report
- Weekly Billboard
- Daily reports to partners
- ...

- AB-Testing
- Discover weekly
- Daily Mix
- ...

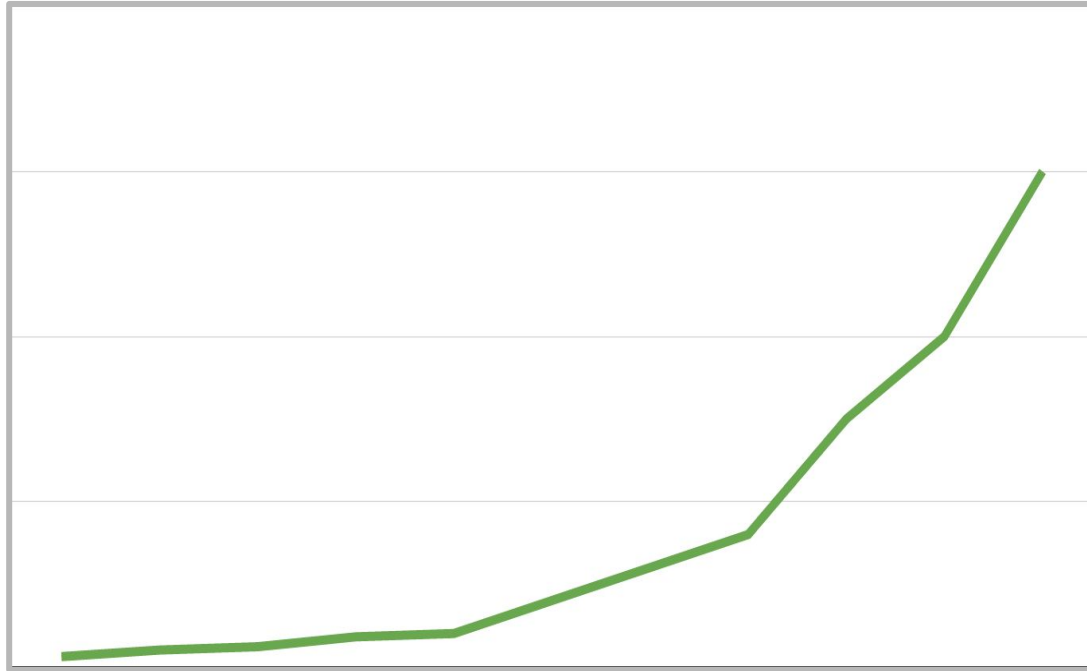
Our growth in Data



Users



+50 TB/day
+100M Users

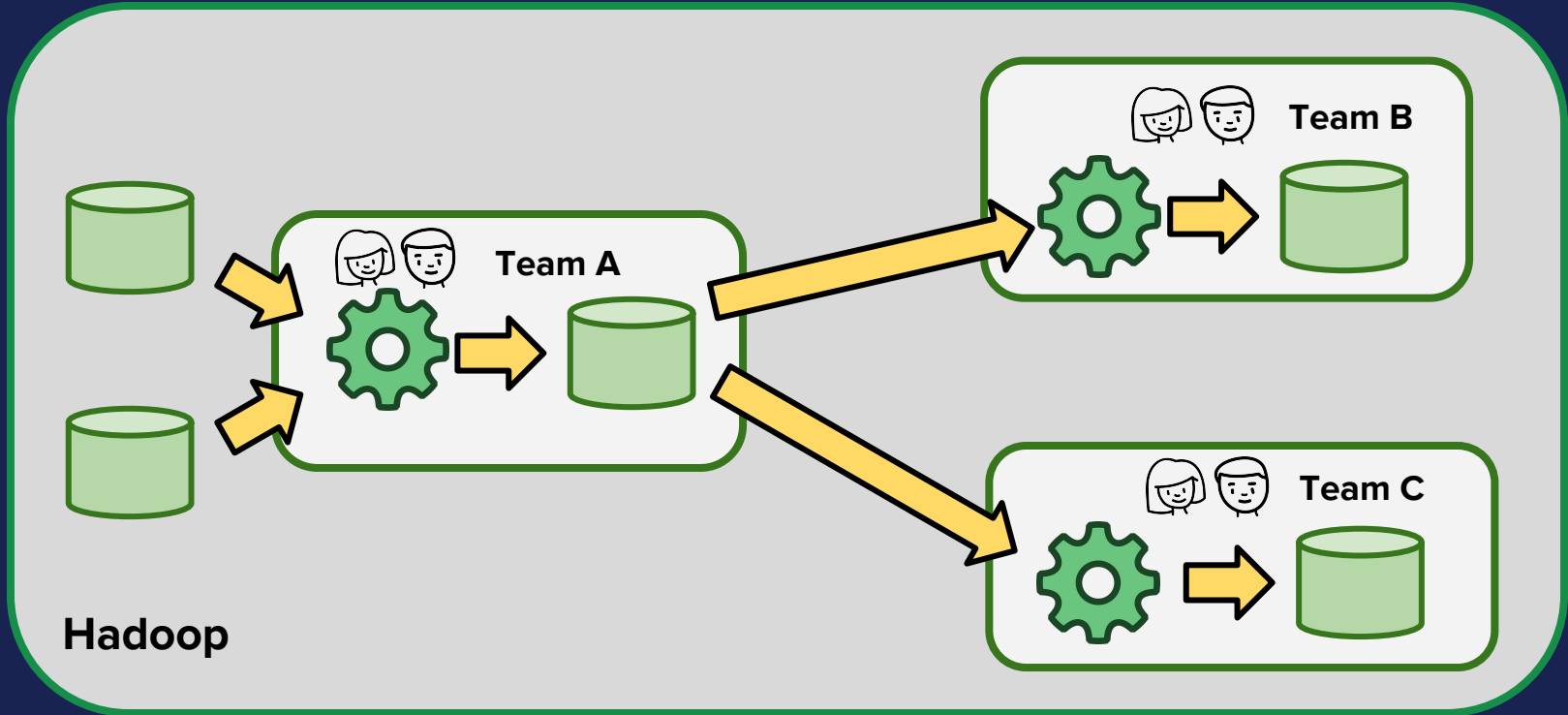


Developers



+60 TB/day
+10k M/R jobs

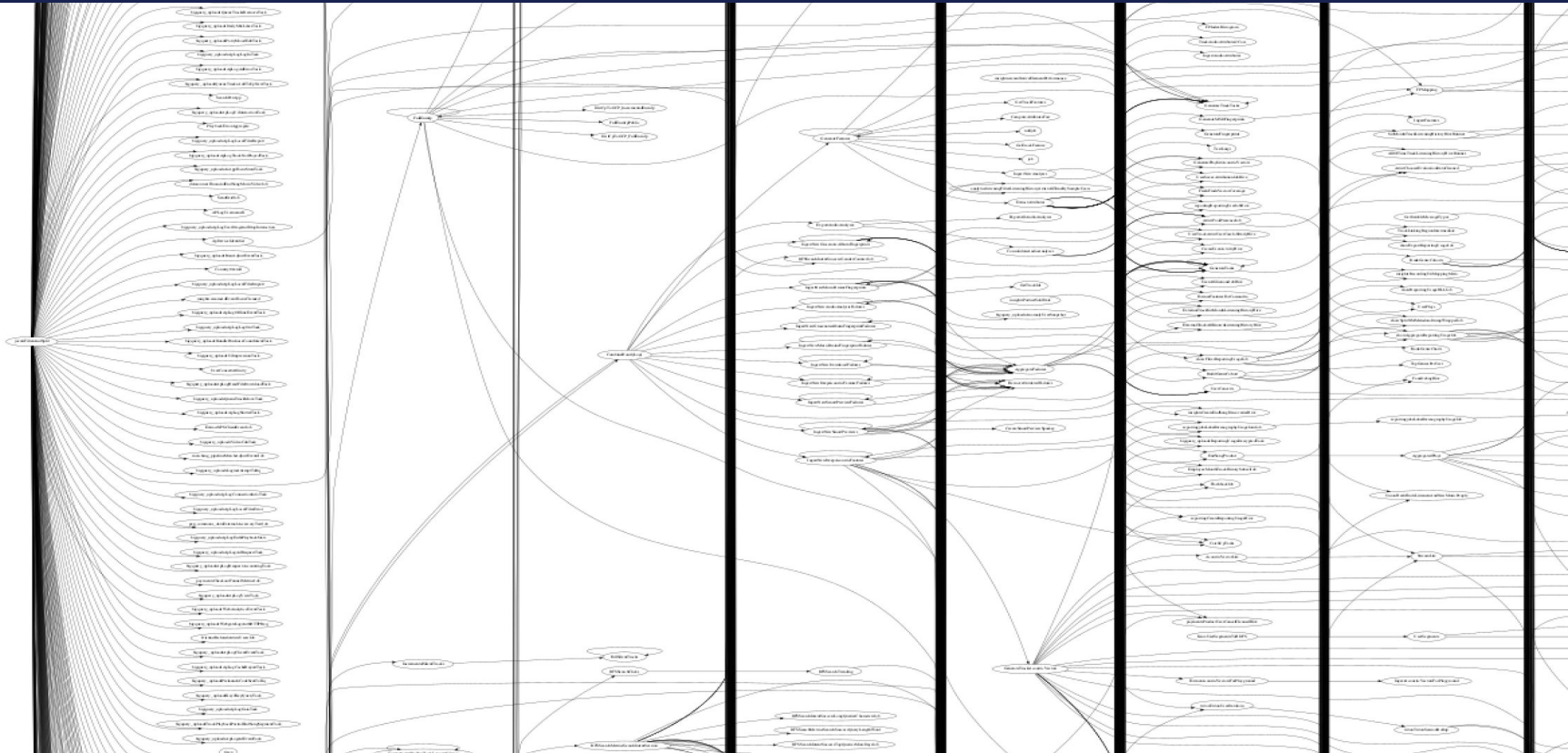
Autonomy & Dependencies



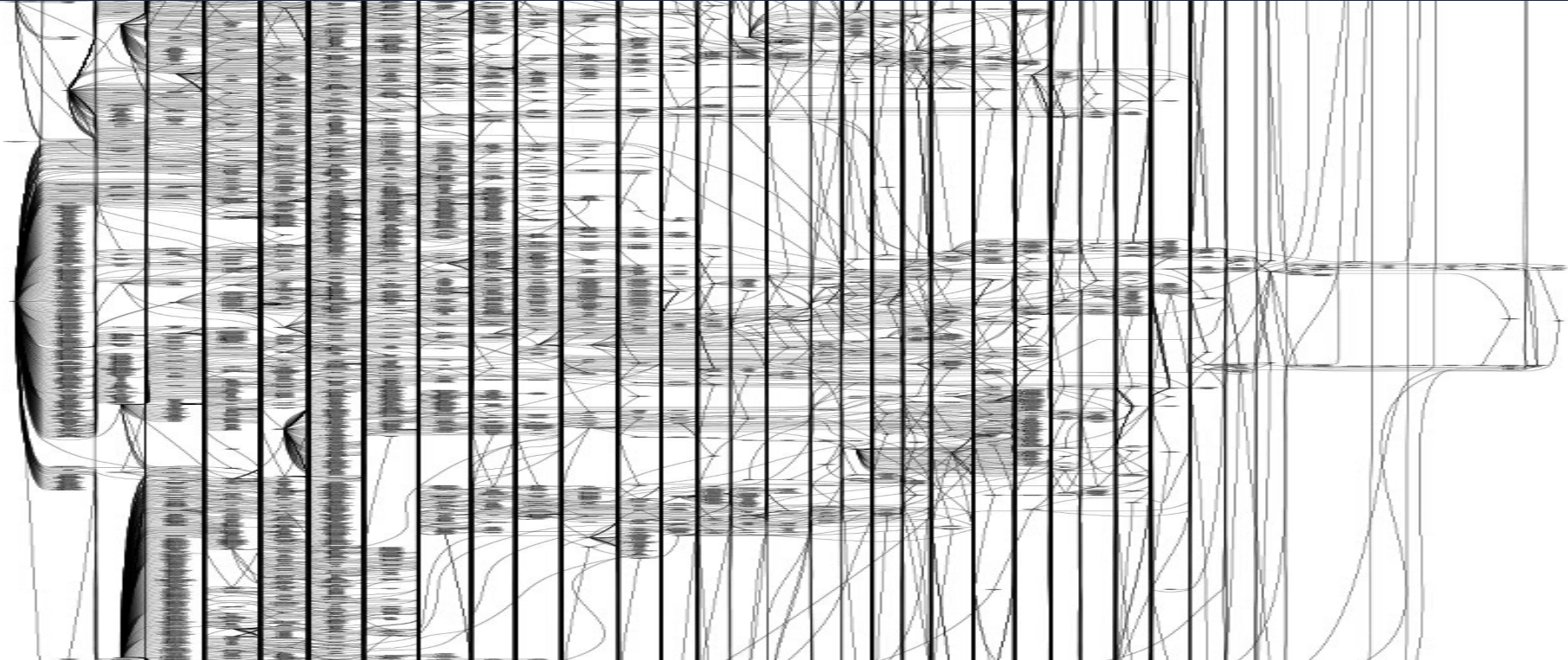
Autonomy & Dependencies



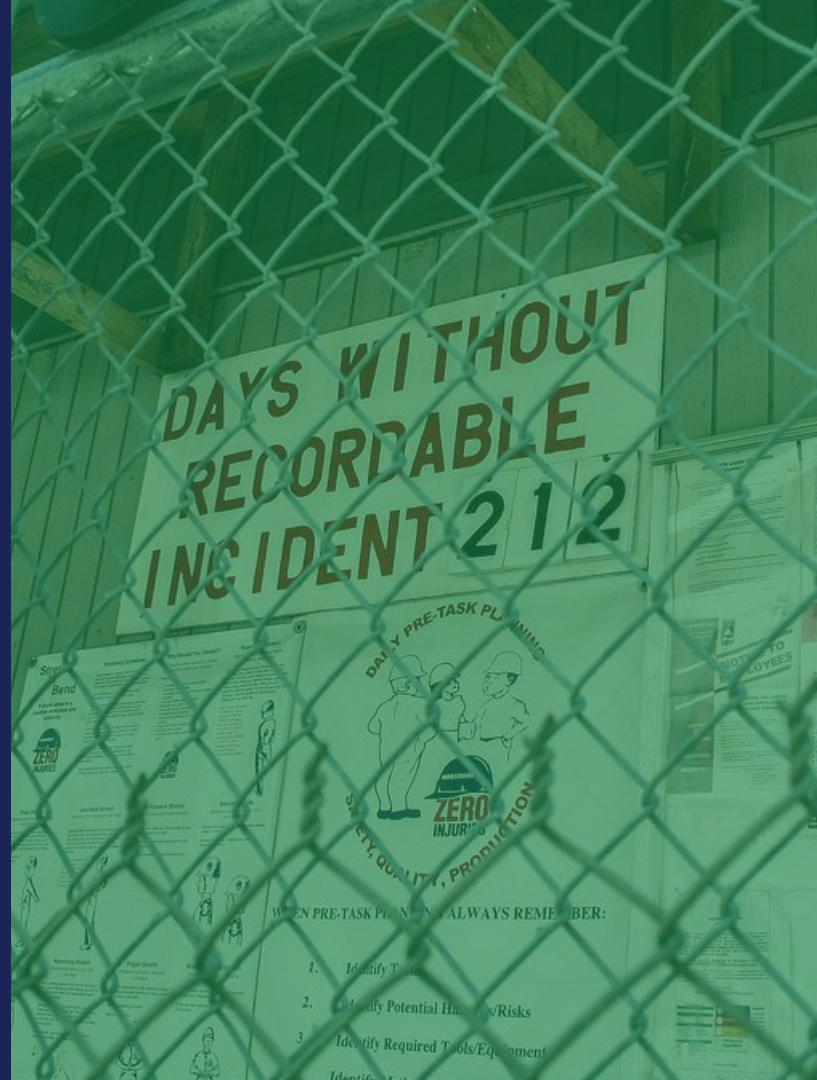
Autonomy & Dependencies



Autonomy & Dependencies

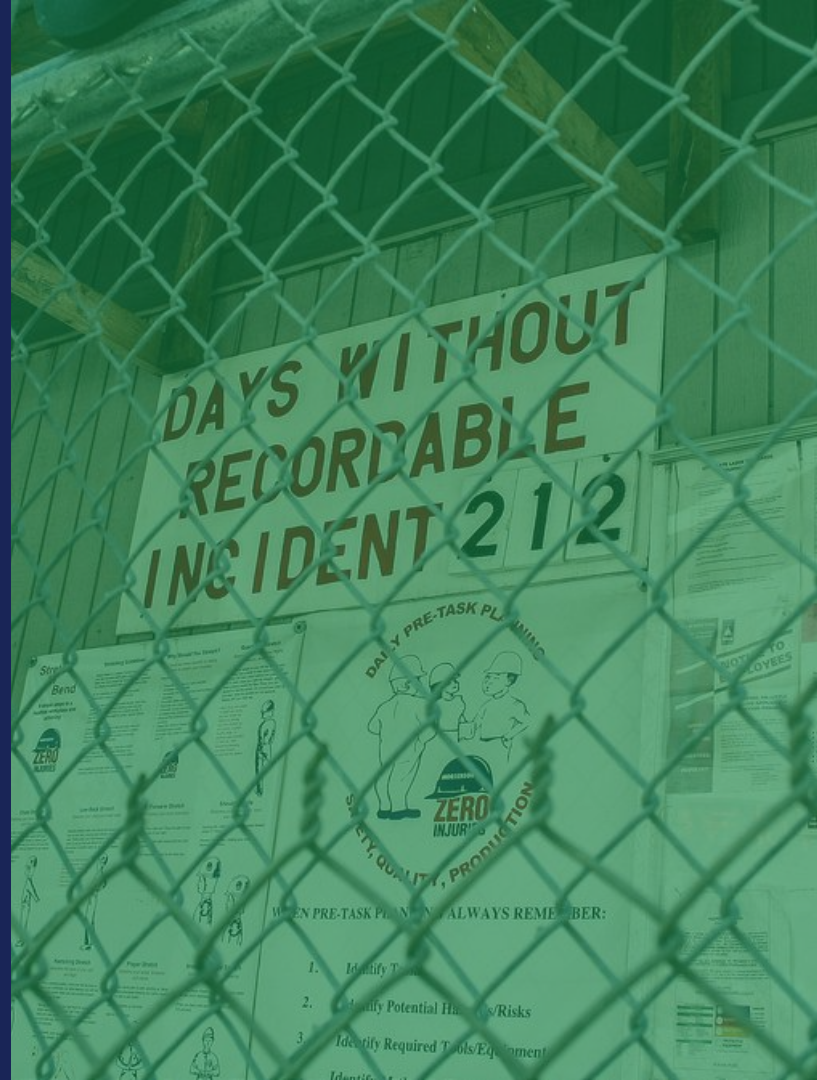


Summer of Incidents



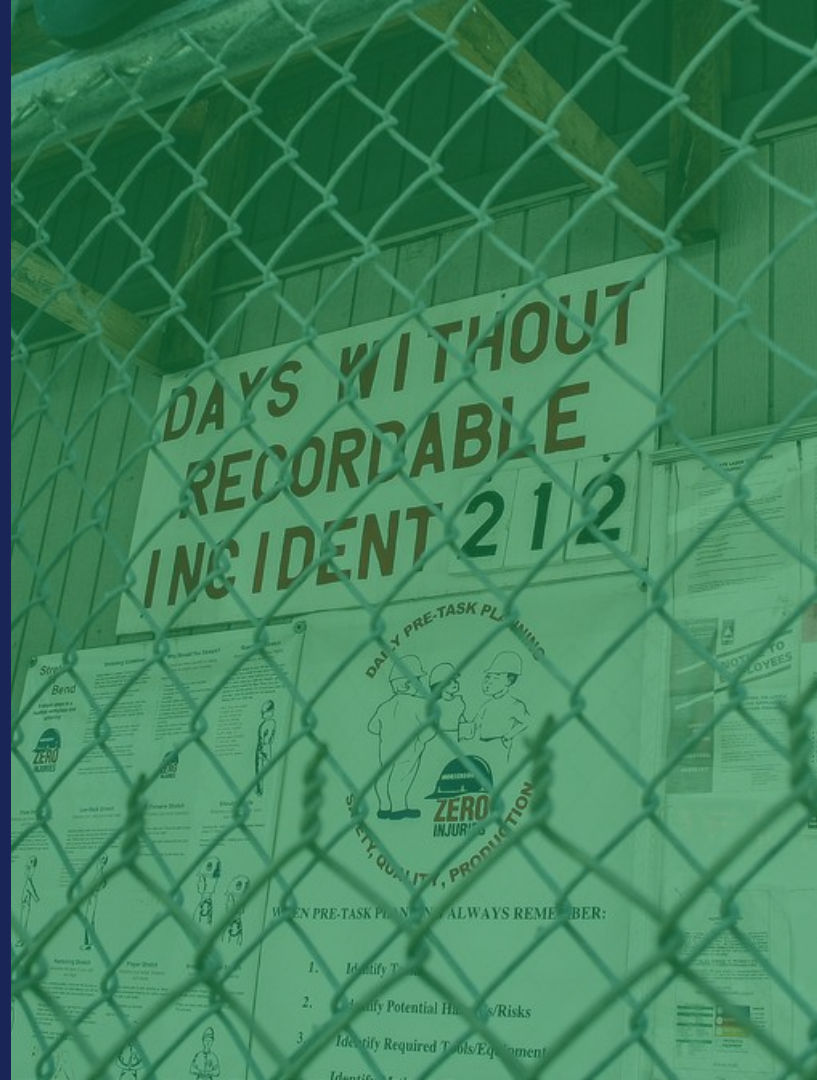
Summer of Incidents

- A strain of incidents



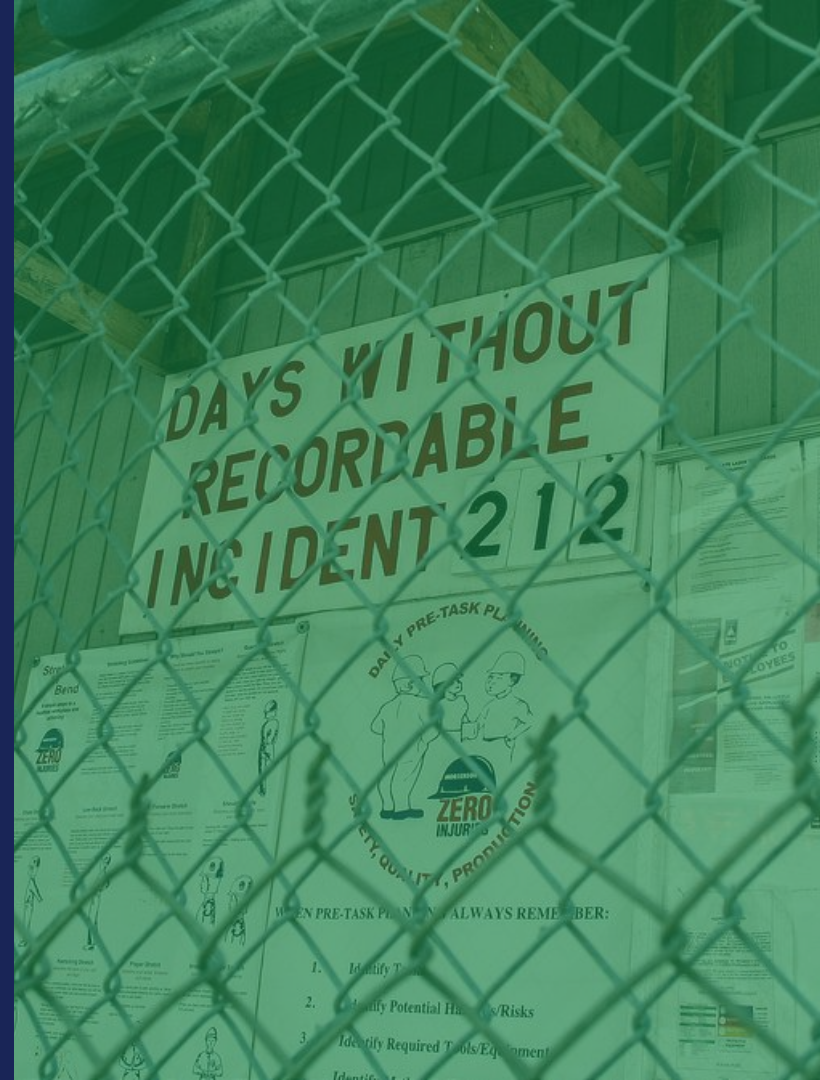
Summer of Incidents

- A strain of incidents
- War-room



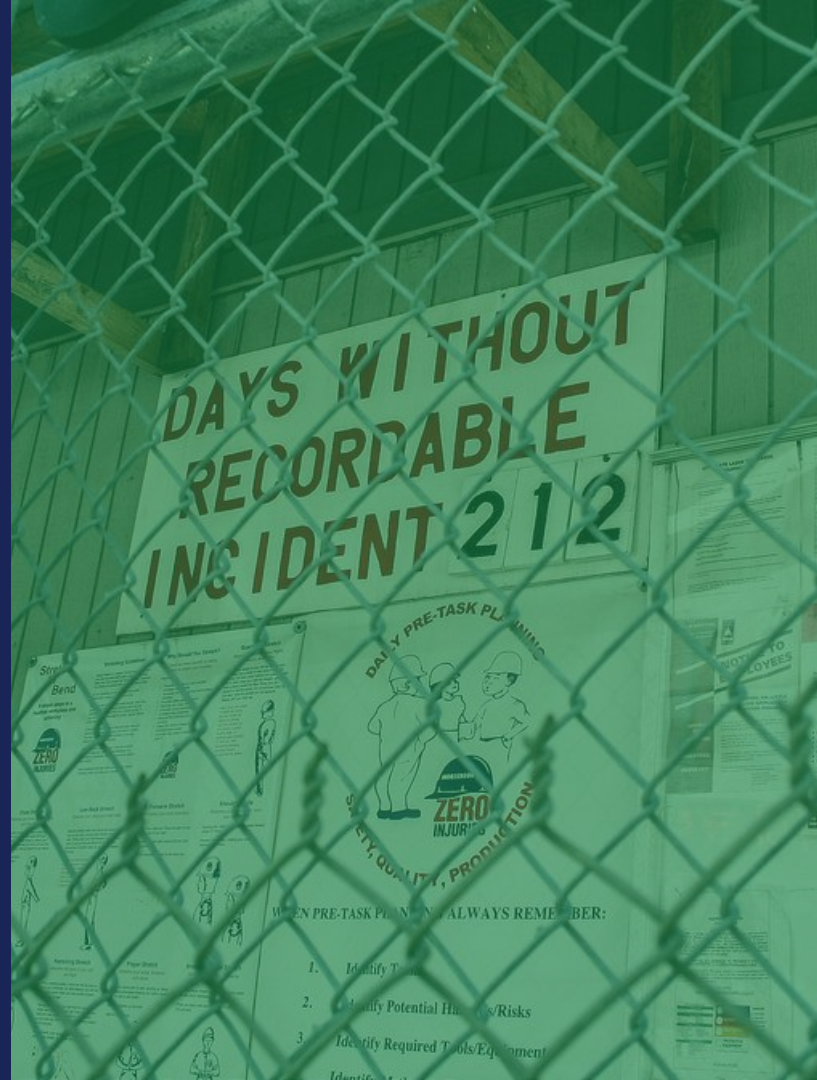
Summer of Incidents

- A strain of incidents
- War-room
- Hadoop on it's knees



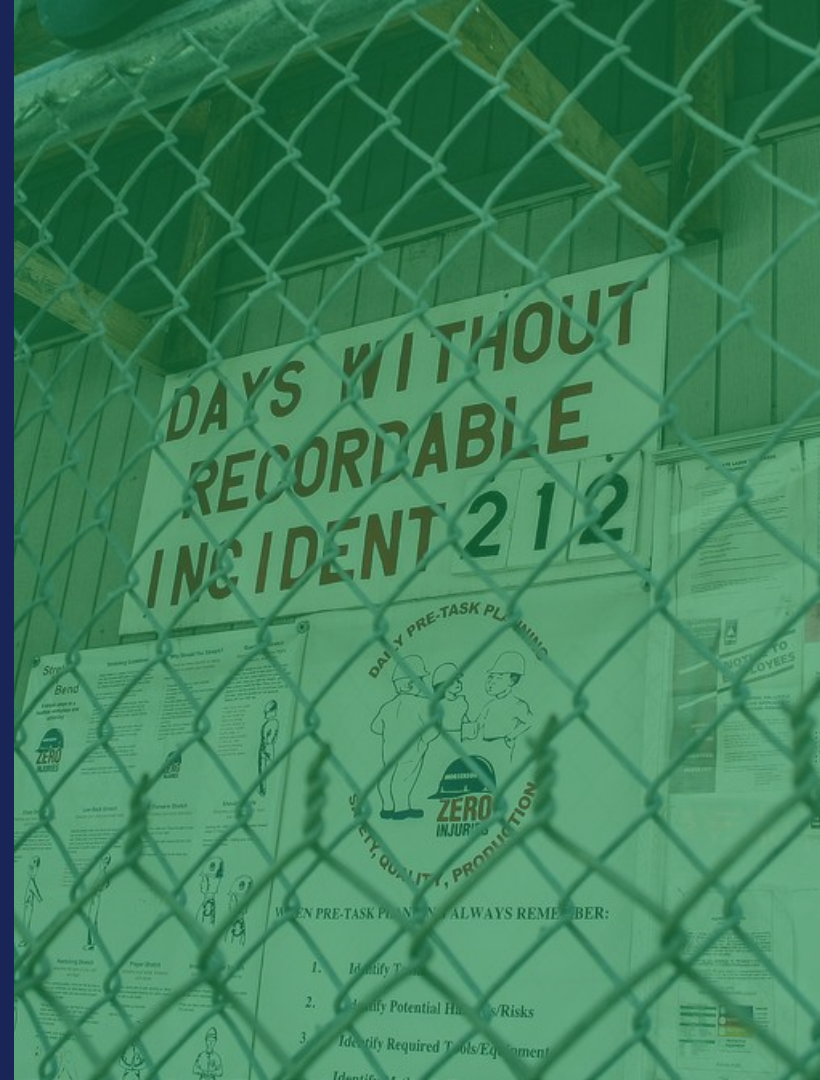
Summer of Incidents

- A strain of incidents
- War-room
- Hadoop on it's knees
- Event Delivery Catch up



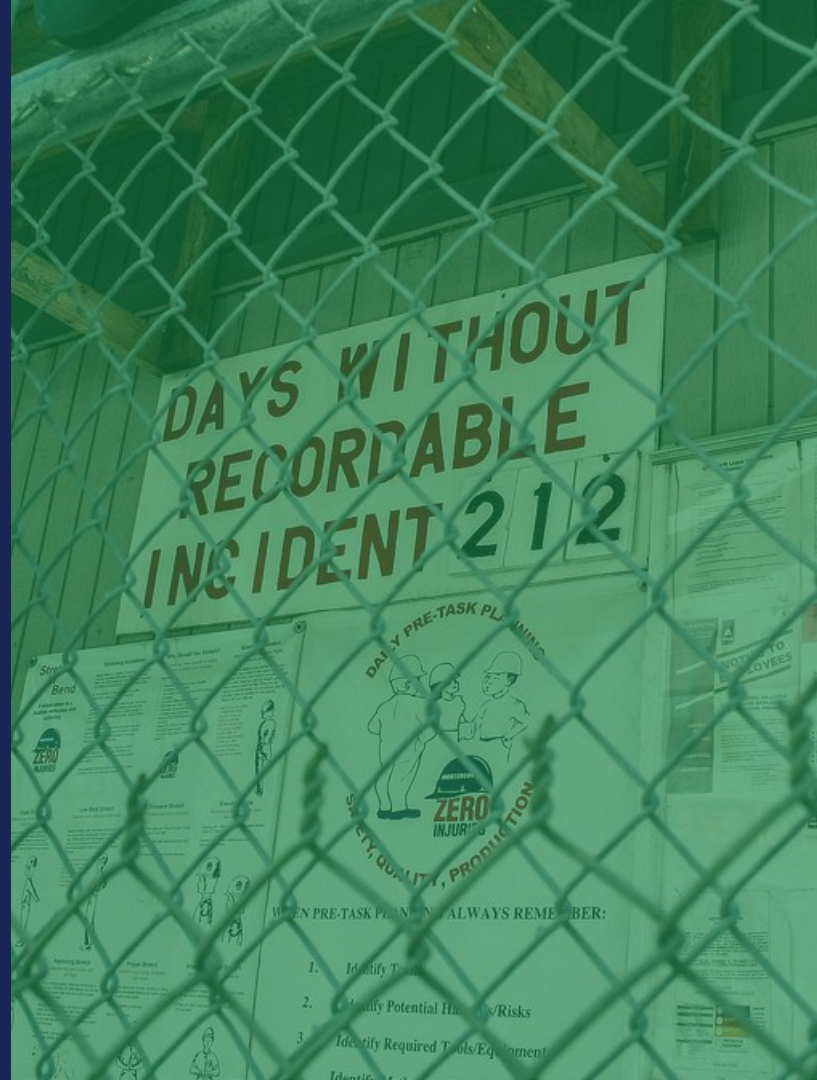
Summer of Incidents

- A strain of incidents
- War-room
- Hadoop on it's knees
- Event Delivery Catch up
- Reprocessing of data



Summer of Incidents

- A strain of incidents
- War-room
- Hadoop on it's knees
- Event Delivery Catch up
- Reprocessing of data
- Hard to debug data issues



Challenges and the path to victory...

Challenges and the path to victory...

1. Early Warning

Datamon - Data monitoring

Challenges and the path to victory...

1. Early Warning

Datamon - Data monitoring

2. Debuggability & Control

Styx - Scheduling and control

Challenges and the path to victory...

1. **Early Warning**

Datamon - Data monitoring

2. **Debuggability & Control**

Styx - Scheduling and control

3. **Automate Capacity**

GABO - Event Delivery

Challenges and the path to victory...

1. Early Warning

Datamon - Data monitoring

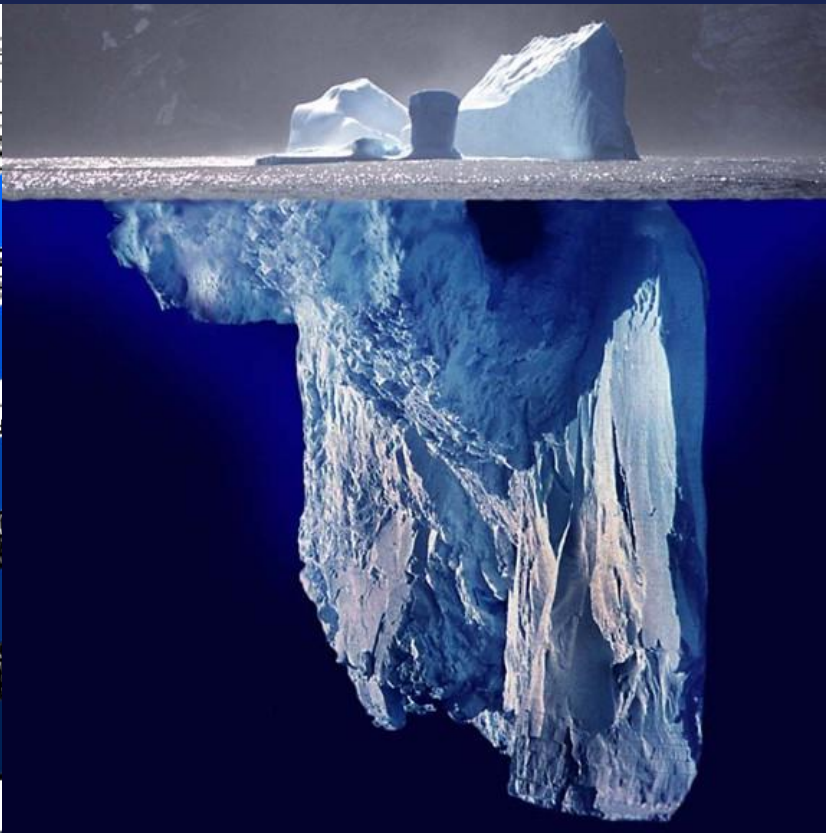
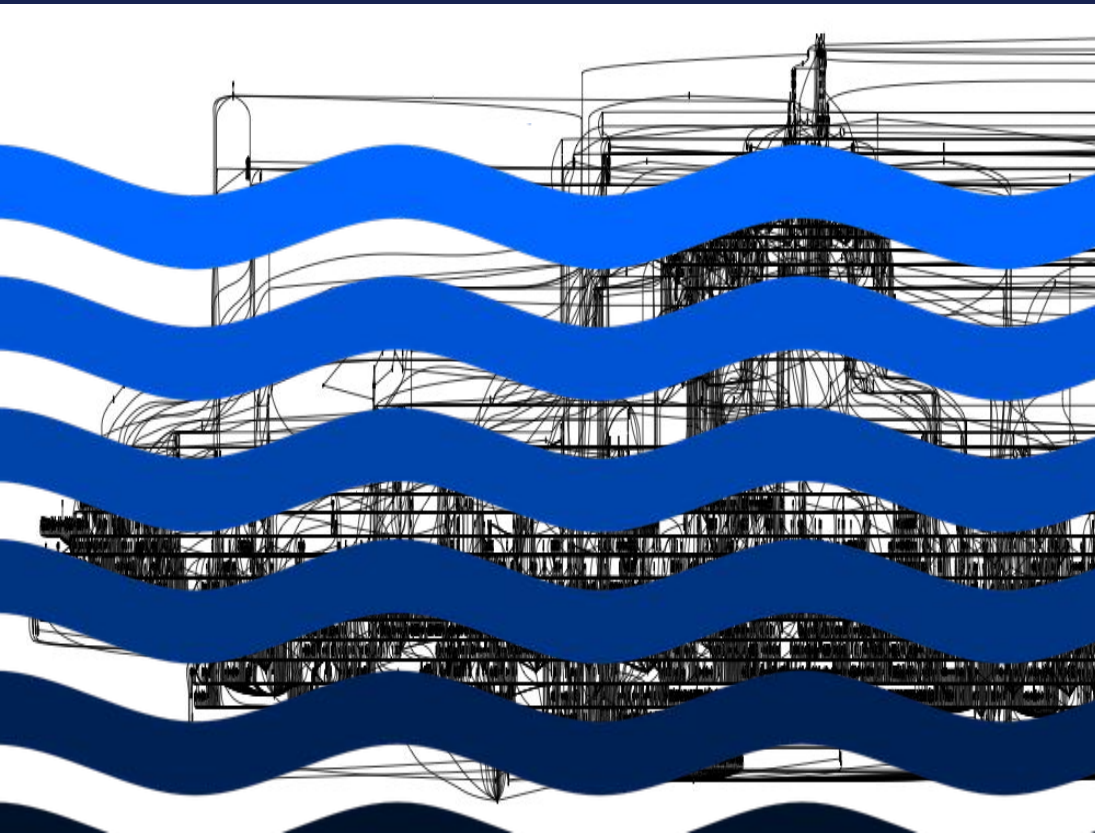
2. Debuggability & Control

Styx - Scheduling and control

3. Automate Capacity

GABO - Event Delivery

Early Warning - Datamon

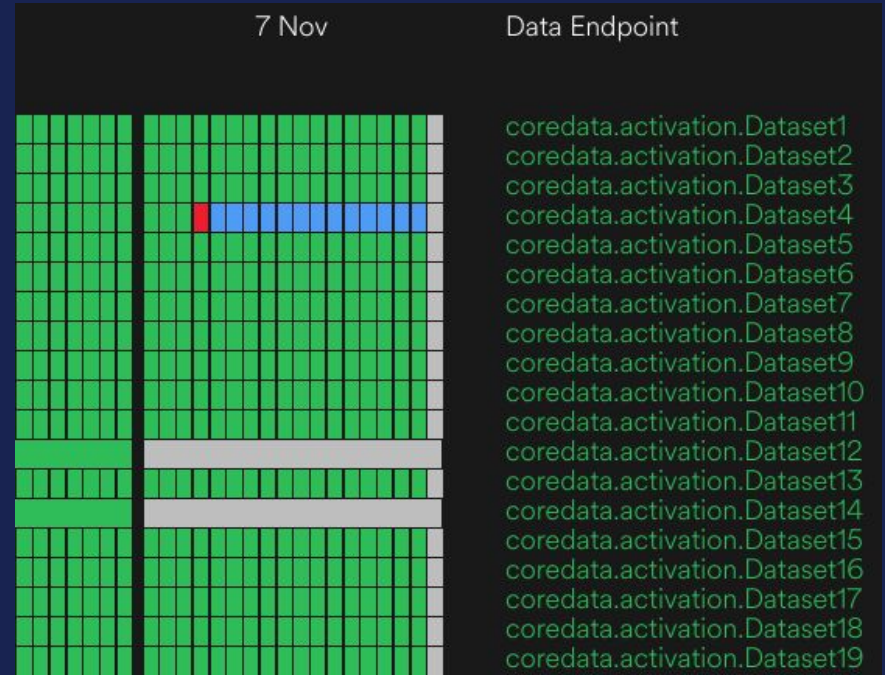


Early Warning - Datamon

- **Unified view**
 - Alignment between teams
- **Ownership**
 - Clear ownership of data
- **SLA**
 - Alert on late data

Early Warning - Datamon

- Define terminology
- Provide metadata language
- Implement a Datamon service



Challenges and the path to victory...

1. Early Warning

Datamon - Data monitoring

2. Debuggability & Control

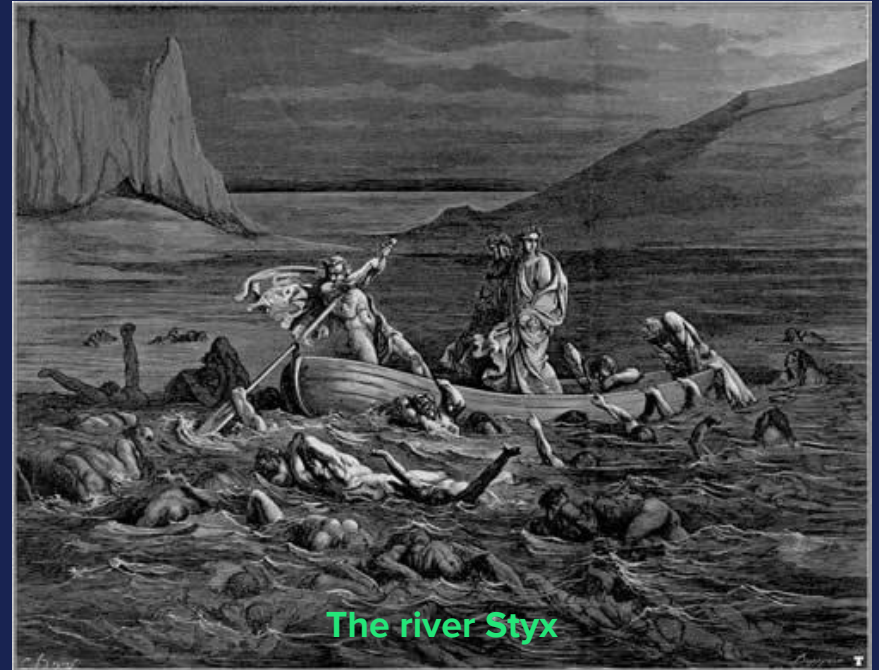
Styx - Scheduling and control

3. Automate Capacity

GABO - Event Delivery

Debuggability & Control - Styx

- **Execution control**
 - Self service for data users
- **Execution information**
 - Expose debug information
- **Execution isolation**
 - Docker for data jobs



The river Styx

Debuggability & Control - Styx

- Execution control
 - Centralized execution API

DATA ENDPOINTS ?		
coredata.activation.Dataset1	hourly	<input checked="" type="checkbox"/>
coredata.activation.Dataset2	hourly	<input checked="" type="checkbox"/>
coredata.activation.Dataset3	hourly	<input checked="" type="checkbox"/>
coredata.activation.Dataset4	hourly	<input checked="" type="checkbox"/>
coredata.activation.Dataset5	hourly	<input checked="" type="checkbox"/>
coredata.activation.Dataset6	hourly	<input checked="" type="checkbox"/>

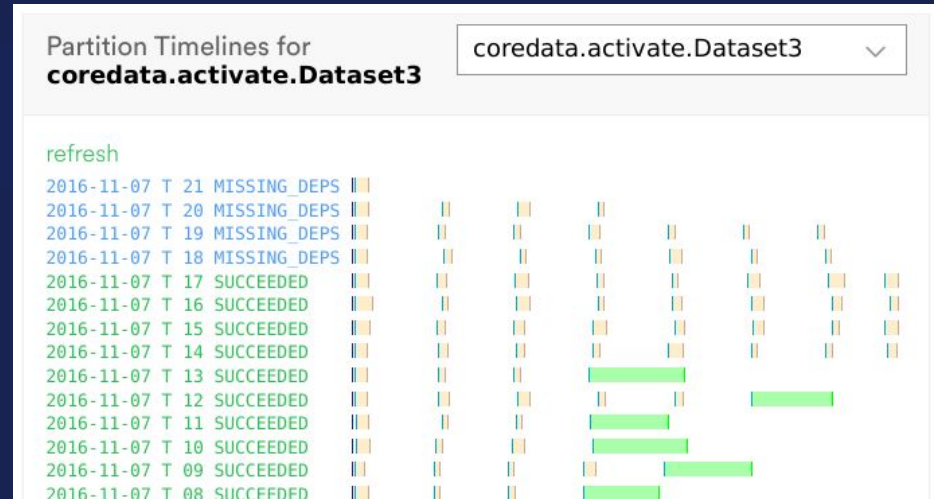
Debuggability & Control - Styx

- Execution control
 - Centralized execution API
 - Backfilling and reprocessing

```
Terminal - matti@matti-ThinkPad-T540p: /opt/matti/src/tmp
File Edit View Terminal Tabs Help
2016-10-28T12:08:38 submit Execution description: ExecutionDescription{dockerImage=registry.spotify.net/spotify
/some-job:bc25f14-1477653357111, dockerArgs=[wrap-luigi, --hydra, --module, tst, matti.SampleJob, --datehour, {}, --local-scheduler], se
cret=Optional[Secret{name=pipeline-core-secret, mountPath=/etc/gcp-keys}], commitSha=Optional[bc25f14e0e9ff20ea1f953cc8c91ad797f54d5]}
2016-10-28T12:08:38 submitted Execution id: styx-run-yggd7
2016-10-28T12:08:39 started
2016-10-28T12:08:55 terminate Exit code: 0
2016-10-28T12:08:55 success
2016-10-28T15:41:42 triggerExecution Trigger id: ad-hoc-cli-1477669302657-07468
2016-10-28T15:41:42 submit Execution description: ExecutionDescription{dockerImage=registry.spotify.net/spotify
/some-job:2fa3469-1477668749515, dockerArgs=[wrap-luigi, --hydra, --module, tst, matti.SampleJob, --datehour, {}, --local-scheduler], se
cret=Optional[Secret{name=pipeline-core-secret, mountPath=/etc/gcp-keys}], commitSha=Optional[2fa346978abbb1e948c5b7a9590e517083a630]}
2016-10-28T15:41:42 submitted Execution id: styx-run-oevh9
2016-10-28T15:41:44 started
2016-10-28T15:43:03 terminate Exit code: 35
2016-10-28T15:43:03 retryAfter Delay (seconds): 180
2016-10-28T15:46:04 retry
2016-10-28T15:46:04 submit Execution description: ExecutionDescription{dockerImage=registry.spotify.net/spotify
/some-job:2fa3469-1477668749515, dockerArgs=[wrap-luigi, --hydra, --module, tst, matti.SampleJob, --datehour, {}, --local-scheduler], se
cret=Optional[Secret{name=pipeline-core-secret, mountPath=/etc/gcp-keys}], commitSha=Optional[2fa346978abbb1e948c5b7a9590e517083a630]}
2016-10-28T15:46:04 submitted Execution id: styx-run-lcasd
2016-10-28T15:46:06 started
2016-10-28T15:47:27 terminate Exit code: 35
2016-10-28T15:47:27 retryAfter Delay (seconds): 180
2016-10-28T15:50:28 retry
2016-10-28T15:50:28 submit Execution description: ExecutionDescription{dockerImage=registry.spotify.net/spotify
/some-job:2fa3469-1477668749515, dockerArgs=[wrap-luigi, --hydra, --module, tst, matti.SampleJob, --datehour, {}, --local-scheduler], se
cret=Optional[Secret{name=pipeline-core-secret, mountPath=/etc/gcp-keys}], commitSha=Optional[2fa346978abbb1e948c5b7a9590e517083a630]}
2016-10-28T15:50:28 submitted Execution id: styx-run-fvj15
2016-10-28T15:50:39 started
2016-10-28T15:51:19 terminate Exit code: 35
2016-10-28T15:51:19 retryAfter Delay (seconds): 180
2016-10-28T15:54:20 retry
2016-10-28T15:54:20 submit Execution description: ExecutionDescription{dockerImage=registry.spotify.net/spotify
/some-job:8a9242f7c09f5dea8027d59f4426364aa1a98bb}
2016-10-28T15:54:20 submitted Execution id: styx-run-llm2w
2016-10-28T15:54:27 started
2016-10-28T16:33:43 terminate Exit code: 0
2016-10-28T16:33:44 success
$ styx retry pipeline-core matti.SampleJob GCP 2016-10-27T00
```

Debuggability & Control - Styx

- Execution control
- Execution information
 - Timeline



Debuggability & Control - Styx

- Execution control
- Execution information
 - Timeline
 - Google Cloud Logging

The screenshot displays the Google Cloud Platform interface for monitoring a job. At the top, it shows 'Partition Timelines for coredata.activate.Dataset3'. Below this, a 'refresh' button and a timeline view are visible, showing dates from 2016-11-07. The main part of the interface is the 'Logs' section, which displays a detailed log of events for the job. The logs include various messages such as 'Shutdown finished.', 'File system hdfsClient client doesn't support atomic mv.', 'Running file existence check', 'INFO: [pid 1] Worker Worker(salt=437169980, workers=1, host=styx-run-c208ba8-0446-4bc6-a85c-78aba11b3360, username=root, pid=1) done coredata.CreateUserMappingIntermediate_TEST14.', 'INFO: Informed scheduler that task coredata.CreateUserMappingIntermediate_TEST_2016_11_04T08_51F07364 has status DONE', and 'INFO: Done'. The logs also show the completion of the job and the shutdown of the worker.

Debuggability & Control - Styx

- Execution control
- Execution information
- **Execution isolation**
 - Docker



kubernetes



docker

Challenges and the path to victory...

1. Early Warning

Datamon - Data monitoring

2. Debuggability & Control

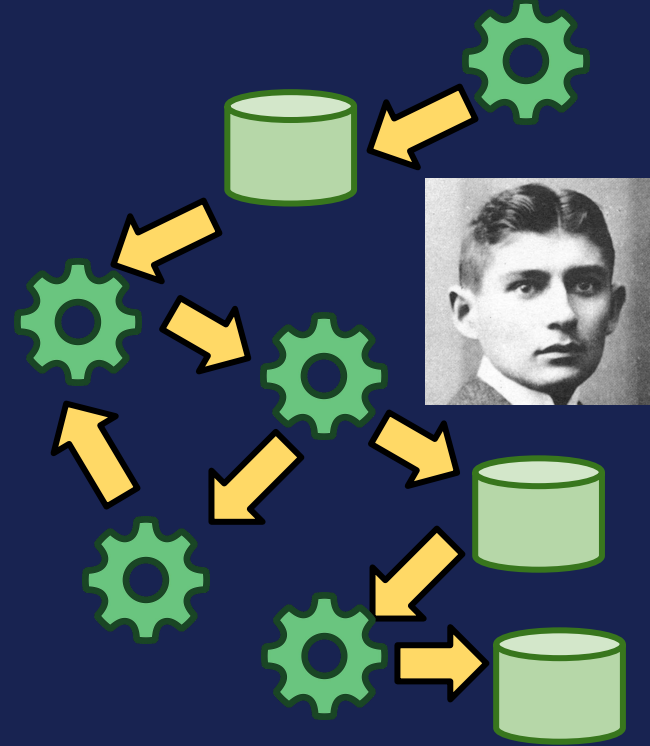
Styx - Scheduling and control

3. Automate Capacity

GABO - Event Delivery

Automate Capacity - GABO/Event Delivery

- Complex and manual config



Automate Capacity - GABO/Event Delivery

- Complex and manual config
- Pubsub & Dataflow streaming



Automate Capacity - GABO/Event Delivery

- **Complex and manual config**
- **Pubsub & Dataflow streaming**
- **Pubsubs at scale**



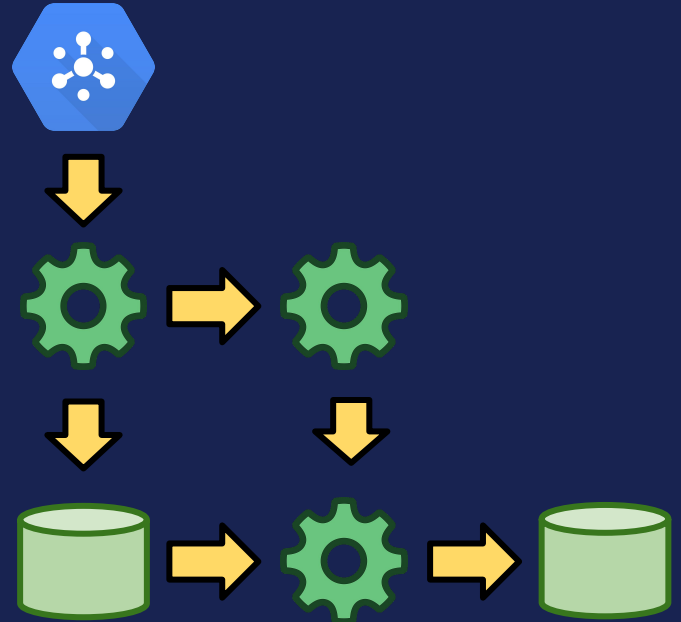
Automate Capacity - GABO/Event Delivery

- **Complex and manual config**
- **Pubsub & Dataflow streaming**
- **Pubsubs at scale**
- **Dataflow streaming**



Automate Capacity - GABO/Event Delivery

- Complex and manual config
- Pubsub & Dataflow streaming
- Pubsubs at scale
- Dataflow streaming :-(
● 2 micro services + 1 Map/Reduce job



Automate Capacity - GABO/Event Delivery

- **Complex and manual config**
- **Pubsub & Dataflow streaming**
- **Pubsubs at scale**
- **Dataflow streaming :-)**
- **2 micro services + 1 Map/Reduce job**
- **Autoscaling & The Stuffer**



GABO - WIP

- Handles at least 10x our load
- Darkloading
- Autoscale everything
- Self service



Summary

- **Make sure you have the right tools to deal with data incidents**
 - Make sure you have time to implement the tools you need
- **Remember that your capacity model can fail at larger scale**
 - Keep track of your scale and Automate, automate, automate...

Thank you!

kalvans@spotify.com

matti@spotify.com

Want to join the band?

<http://spoti.fi/jobs>

