



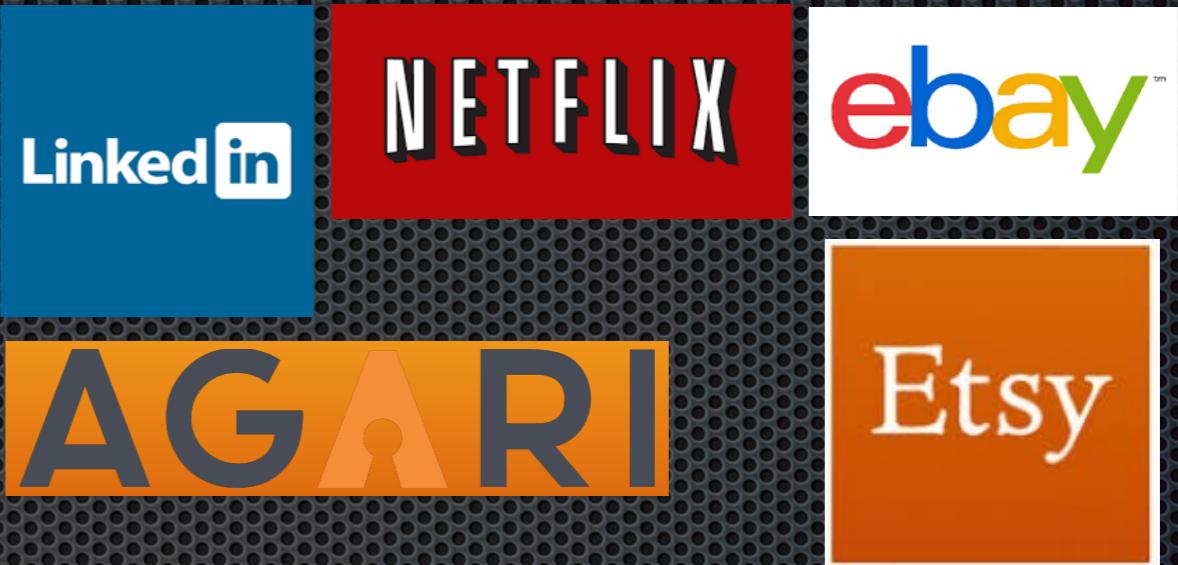
# Introducing Apache Airflow (Incubating)

Sid Anand (@r39132)  
Data Day Seattle 2016



# About Me

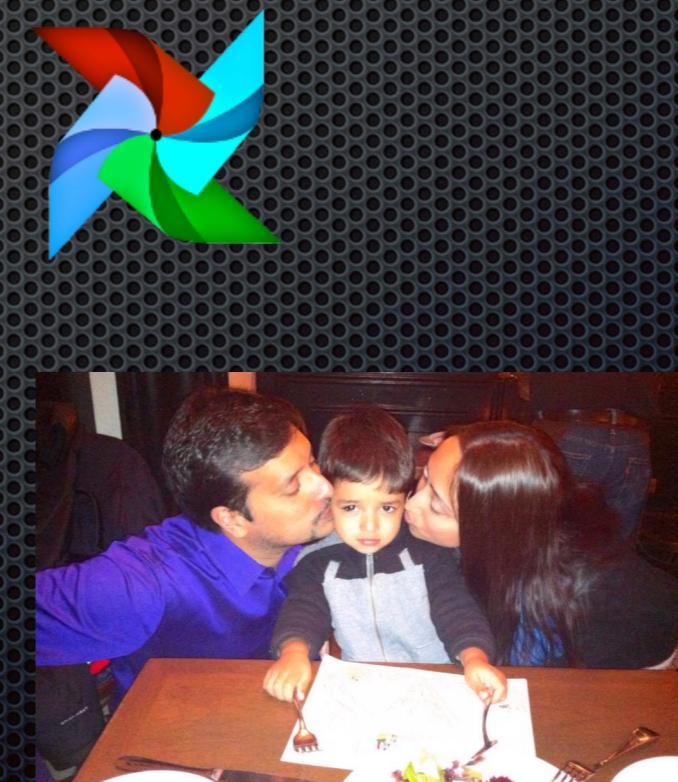
Work [ed | s] @



Co-Chair for



Maintainer on



Reports to

# Apache Airflow

What is it?



# Apache Airflow : What is it?

Airflow is a platform to programmatically author, schedule and monitor workflows (a.k.a. DAGs)

# Apache Airflow : What is it?

Airflow is a platform to programmatically author, schedule and monitor workflows (a.k.a. DAGs)

It ships with

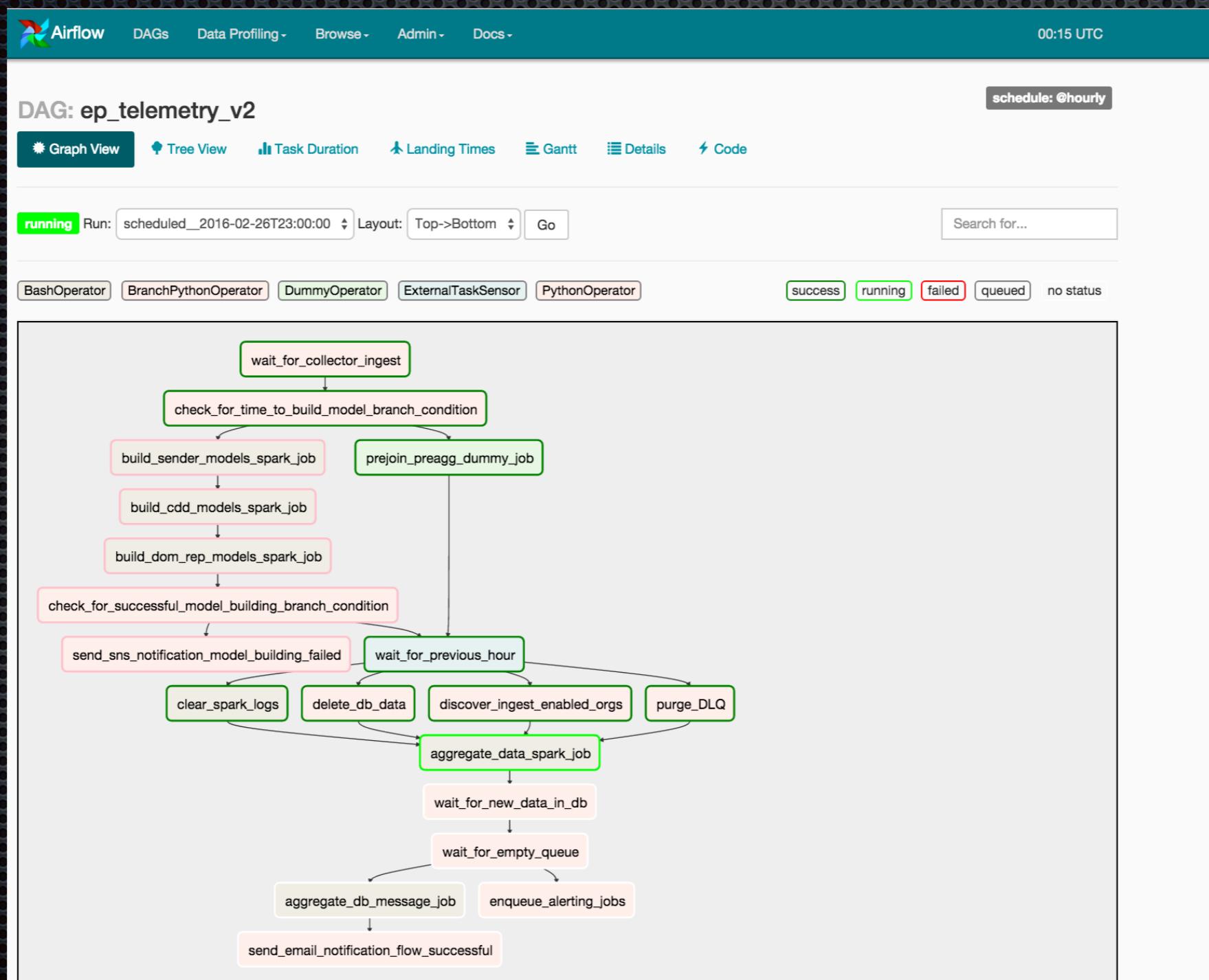
- DAG Scheduler
- Web application (UI)
- Powerful CLI

# Apache Airflow : What is it?

Airflow		DAGs	Data Profiling ▾	Browse ▾	Admin ▾	Docs ▾	21:04 UTC	Power
DAGs								
Show	10	entries	Search: <input type="text"/>					
	i	DAG	Schedule	Owner	Recent Statuses <small>i</small>		Links	
i	On	db_backup_v1	04***	aflury	3	○ ○ ○ ○ ○	✖ ✨ 📈 ⏱ ⚡ ⌂	
i	On	db_report_v4	*/10*****	scattaneo	1	○ ○ ○ ○ ○	✖ ✨ 📈 ⏱ ⚡ ⌂	
i	Off	emr_data_report	01***	kmandich	3	○ ○ ○ ○ ○	✖ ✨ 📈 ⏱ ⚡ ⌂	
i	On	emr_forwarders	08***	kmandich	3	○ ○ ○ ○ ○	✖ ✨ 📈 ⏱ ⚡ ⌂	
i	On	emr_model_building	01***	kmandich	5	○ ○ ○ ○ ○	✖ ✨ 📈 ⏱ ⚡ ⌂	
i	Off	ep_model_building_v1	01***	sanand	7	○ ○ ○ ○ ○	✖ ✨ 📈 ⏱ ⚡ ⌂	
i	Off	ep_reload_data	None	sanand	14	○ ○ ○ ○ ○	✖ ✨ 📈 ⏱ ⚡ ⌂	
i	Off	ep_summary_alert	@hourly	wforrester	1	○ ○ ○ ○ ○	✖ ✨ 📈 ⏱ ⚡ ⌂	
i	On	ep_telemetry_v2	@hourly	sanand	1	○ ○ ○ ○ ○	✖ ✨ 📈 ⏱ ⚡ ⌂	
i	Off	feedback_report_v1	1 day, 0:00:00	kmandich	○ ○ ○ ○ ○	○ ○ ○ ○ ○	✖ ✨ 📈 ⏱ ⚡ ⌂	
Showing 1 to 10 of 20 entries								
Previous				1	2	Next		

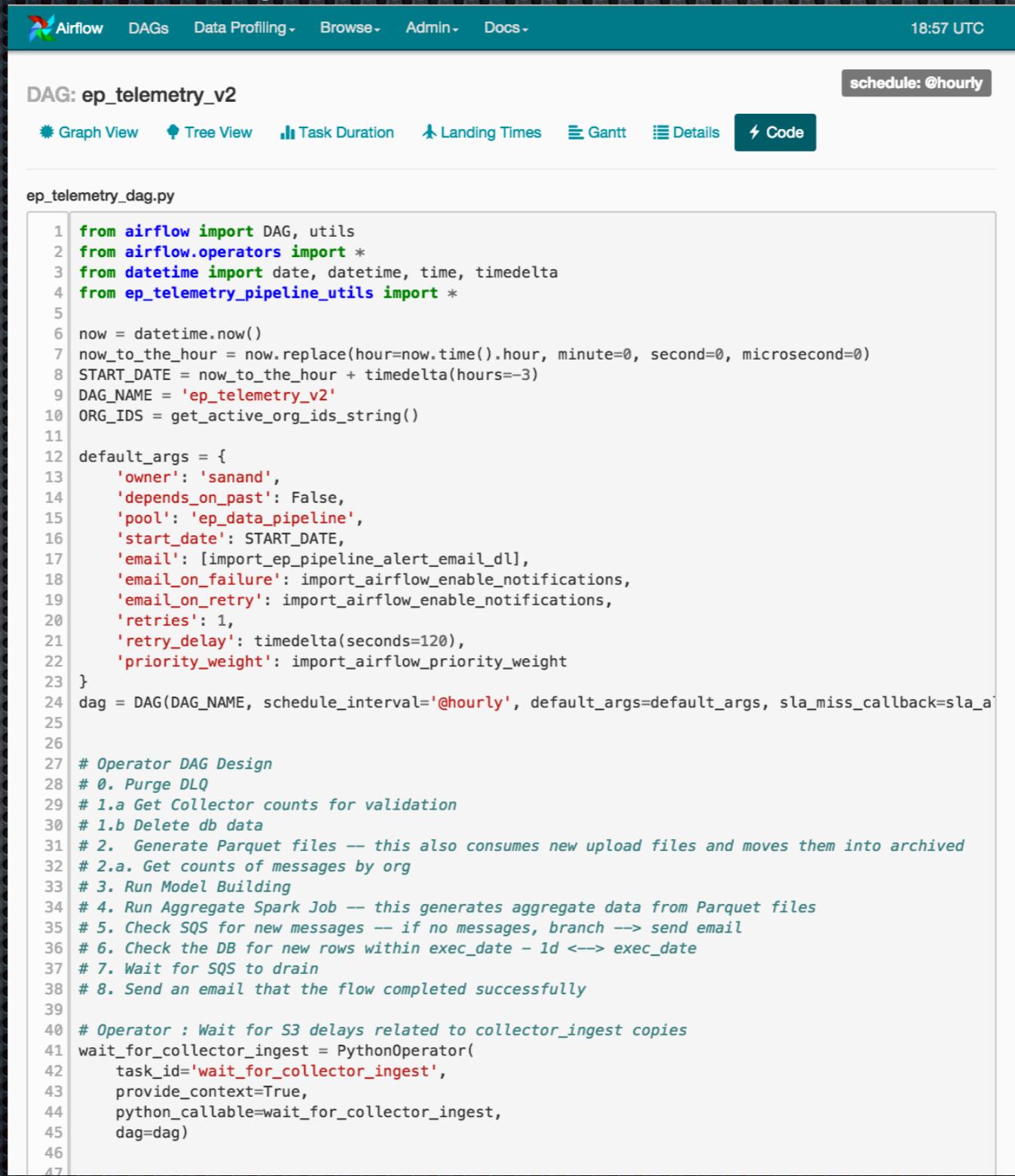
# Airflow - Authoring DAGs

## Airflow: Visualizing a DAG



# Airflow - Authoring DAGs

Airflow: Author DAGs in Python! No need to bundle many XML files!

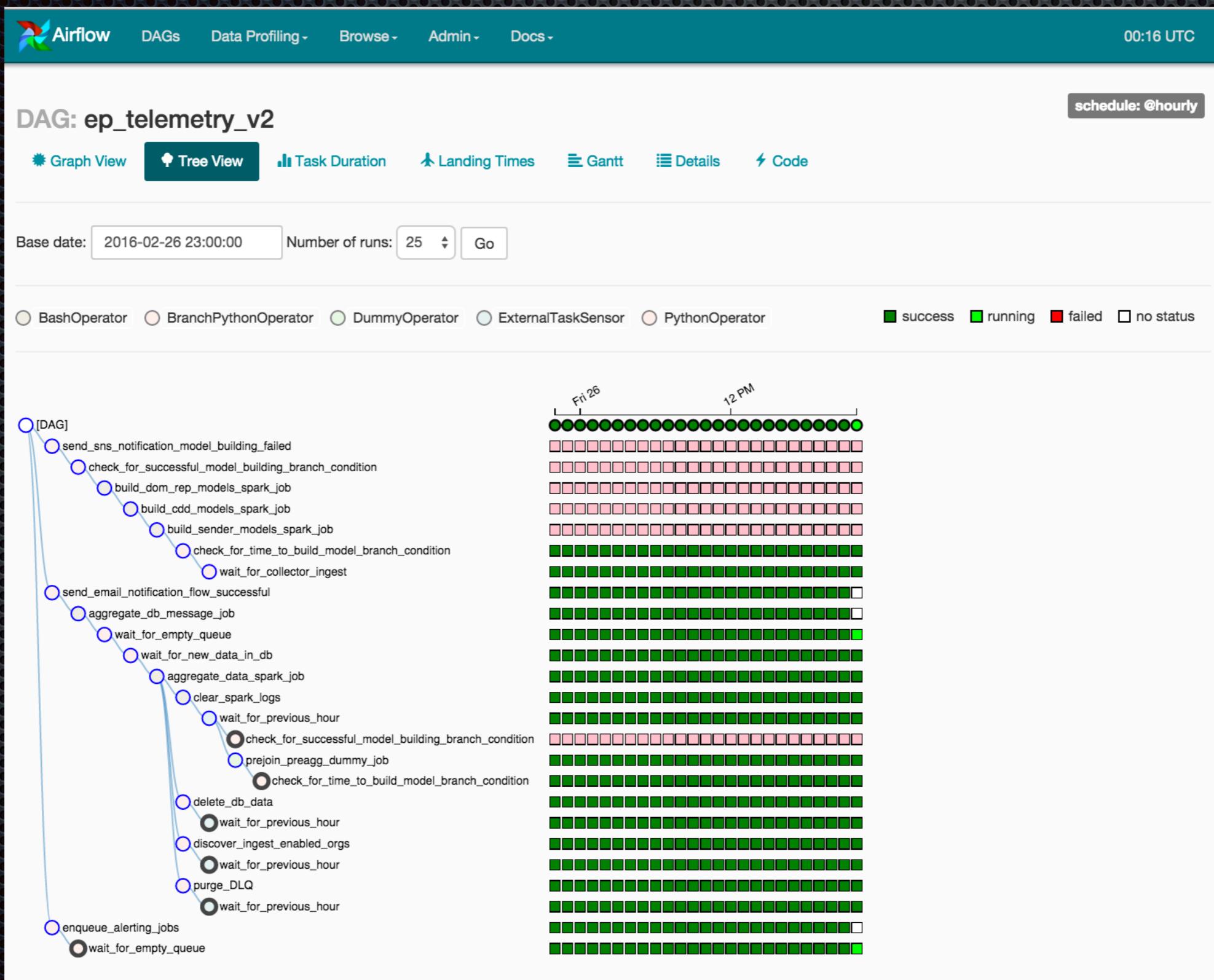


The screenshot shows the Airflow web interface with the title "DAG: ep\_telemetry\_v2". The top navigation bar includes links for Airflow, DAGs, Data Profiling, Browse, Admin, and Docs, along with the time "18:57 UTC". Below the title, there are several tabs: Graph View, Tree View, Task Duration, Landing Times, Gantt, Details, and a highlighted "Code" tab. A button labeled "schedule: @hourly" is visible. The main content area displays the Python code for the DAG:

```
1 from airflow import DAG, utils
2 from airflow.operators import *
3 from datetime import date, datetime, time, timedelta
4 from ep_telemetry_pipeline_utils import *
5
6 now = datetime.now()
7 now_to_the_hour = now.replace(hour=now.time().hour, minute=0, second=0, microsecond=0)
8 START_DATE = now_to_the_hour + timedelta(hours=-3)
9 DAG_NAME = 'ep_telemetry_v2'
10 ORG_IDS = get_active_org_ids_string()
11
12 default_args = {
13     'owner': 'sanand',
14     'depends_on_past': False,
15     'pool': 'ep_data_pipeline',
16     'start_date': START_DATE,
17     'email': [import_ep_pipeline_alert_email_dl],
18     'email_on_failure': import_airflow_enable_notifications,
19     'email_on_retry': import_airflow_enable_notifications,
20     'retries': 1,
21     'retry_delay': timedelta(seconds=120),
22     'priority_weight': import_airflow_priority_weight
23 }
24 dag = DAG(DAG_NAME, schedule_interval='@hourly', default_args=default_args, sla_miss_callback=sla_a
25
26
27 # Operator DAG Design
28 # 0. Purge DLQ
29 # 1.a Get Collector counts for validation
30 # 1.b Delete db data
31 # 2. Generate Parquet files — this also consumes new upload files and moves them into archived
32 # 2.a. Get counts of messages by org
33 # 3. Run Model Building
34 # 4. Run Aggregate Spark Job — this generates aggregate data from Parquet files
35 # 5. Check SQS for new messages -- if no messages, branch --> send email
36 # 6. Check the DB for new rows within exec_date - 1d <-> exec_date
37 # 7. Wait for SQS to drain
38 # 8. Send an email that the flow completed successfully
39
40 # Operator : Wait for S3 delays related to collector_ingest copies
41 wait_for_collector_ingest = PythonOperator(
42     task_id='wait_for_collector_ingest',
43     provide_context=True,
44     python_callable=wait_for_collector_ingest,
45     dag=dag)
46
47
```

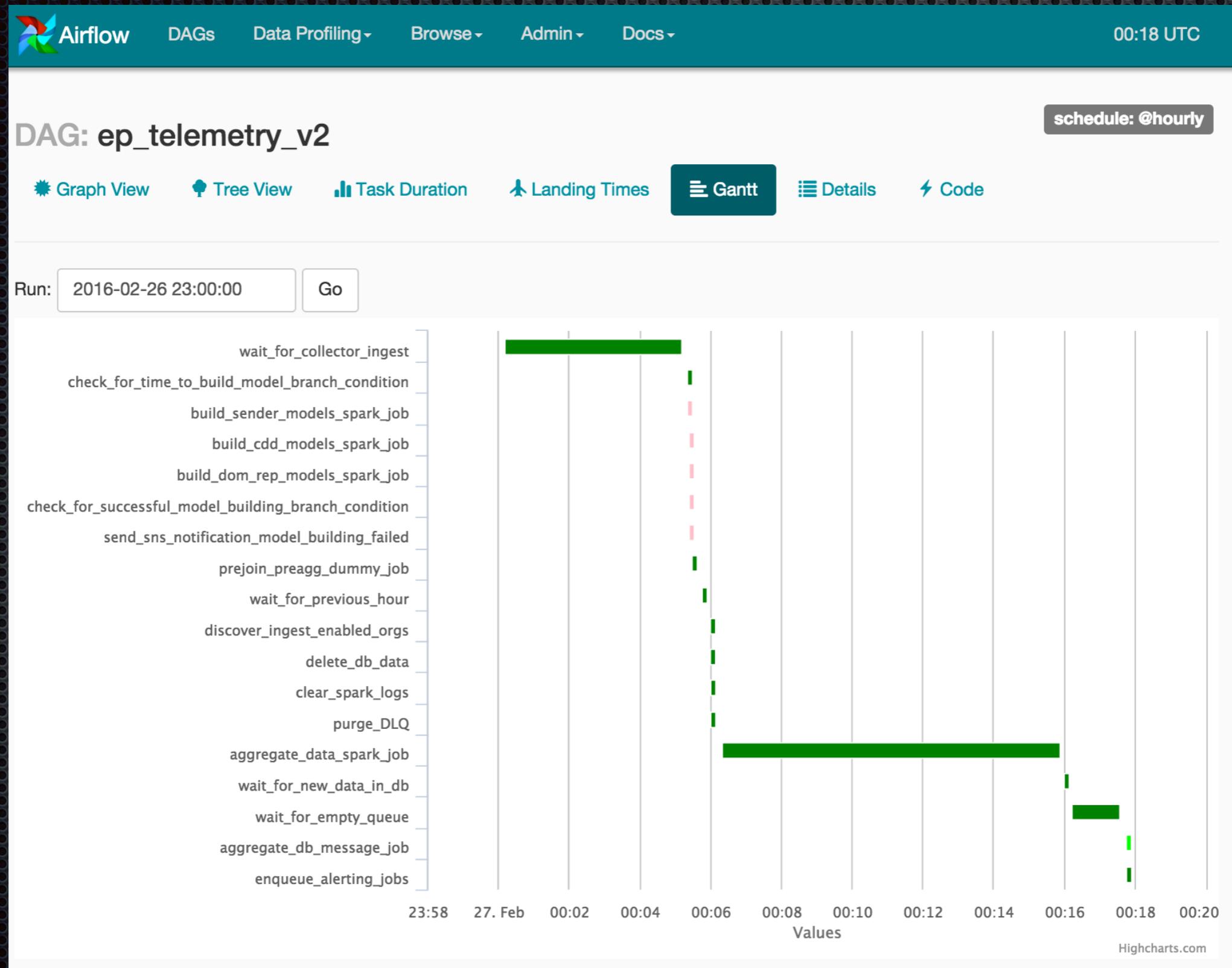
# Airflow - Authoring DAGs

Airflow: The Tree View offers a view of DAG Runs over time!



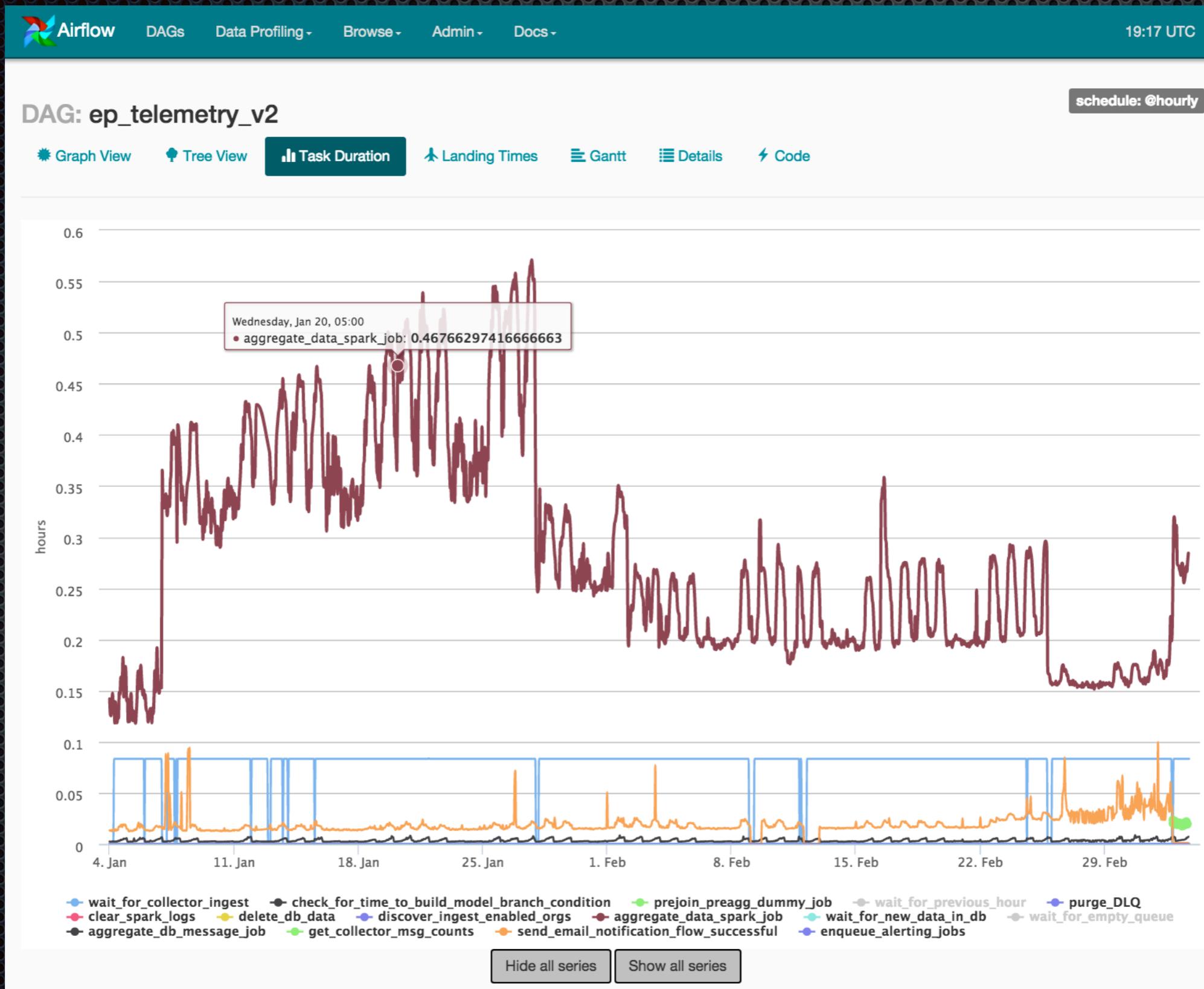
# Airflow - Performance Insights

Airflow: Gantt charts reveal the slowest tasks for a run!



# Airflow - Performance Insights

Airflow: ...And we can easily see performance trends over time



# Apache Airflow : What is it?

When would you use a Workflow Scheduler like Airflow?

- ETL Pipelines
- Machine Learning Pipelines
- Predictive Data Pipelines
  - Fraud Detection, Scoring/Ranking, Classification, Recommender System, etc...
- General Job Scheduling (e.g. Cron)
  - DB Back-ups, Scheduled code/config deployment

# Apache Airflow : What is it?

What should a Workflow Scheduler do well?

- Schedule a graph of dependencies
  - where Workflow = A DAG of Tasks
- Handle task failures
- Report / Alert on failures
- Monitor performance of tasks over time
- Enforce SLAs
  - E.g. Alerting if time or correctness SLAs are not met
- Scale

# Apache Airflow : What is it?

## What Does Apache Airflow Add?

- Configuration-as-code
- Usability - Stunning UI / UX
- Centralized configuration
- Resource Pooling
- Extensibility

# Apache Airflow

Incubating



# Apache Airflow : Incubating

## Timeline

- Airflow was created @ Airbnb in 2015 by Maxime Beauchemin
- Max launched it @ Hadoop Summit in Summer 2015
- On 3/31/2016, Airflow → Apache Incubator

## Today

- 166+ Contributors
- 300+ Users
- 40+ companies officially using it!
- 9 Committers/Maintainers <— We're growing here

# Agari

What We Do!

# Agari : What We Do



# Agari : What We Do



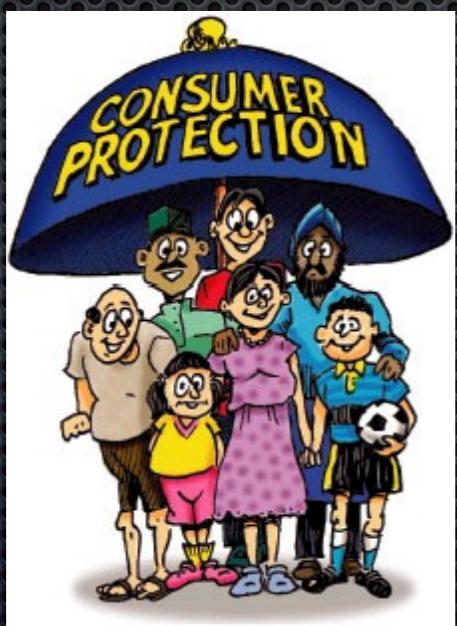
# Agari : What We Do



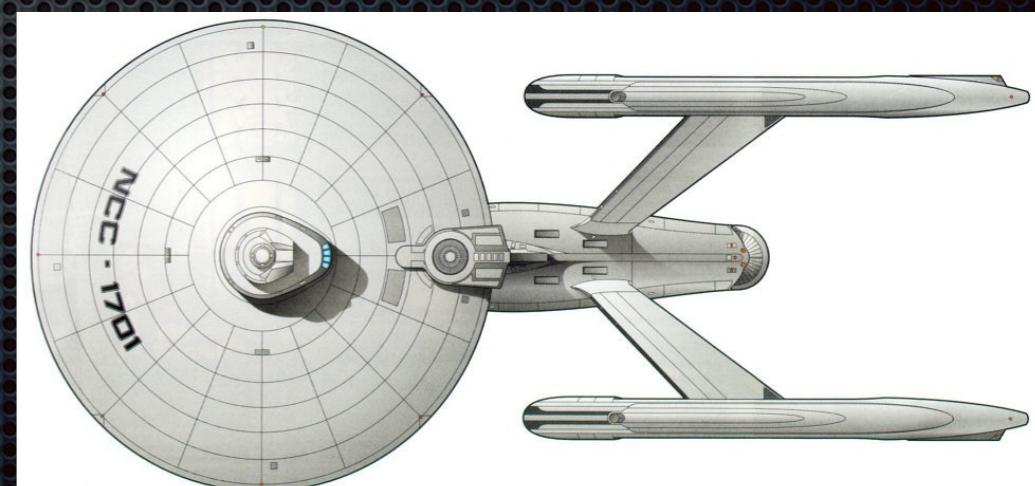
# Agari : What We Do



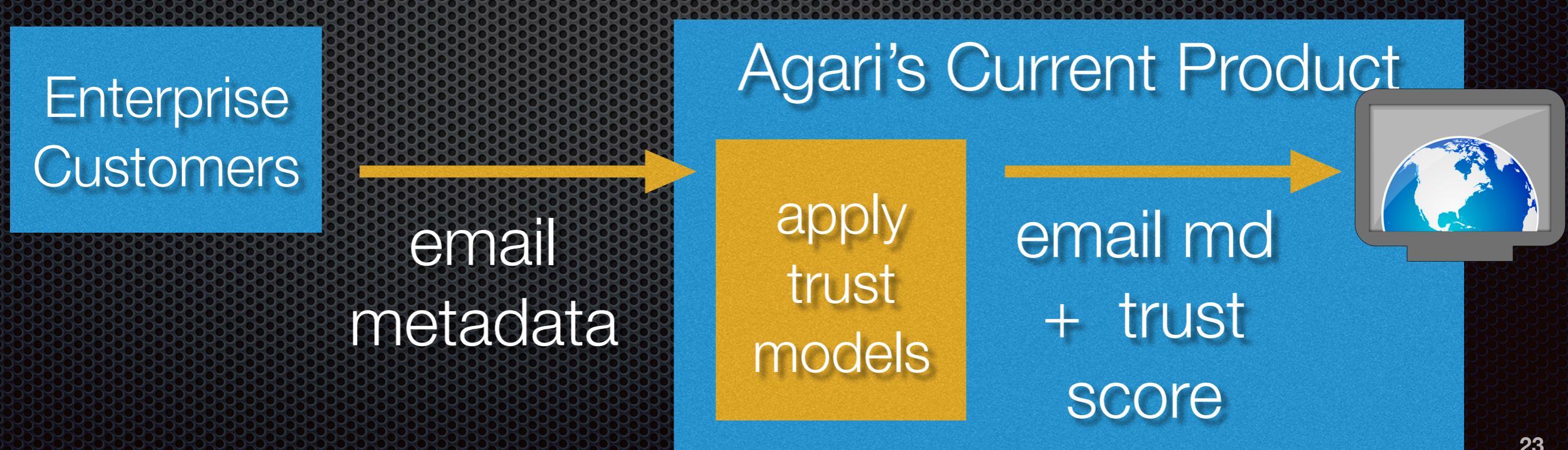
AGARI



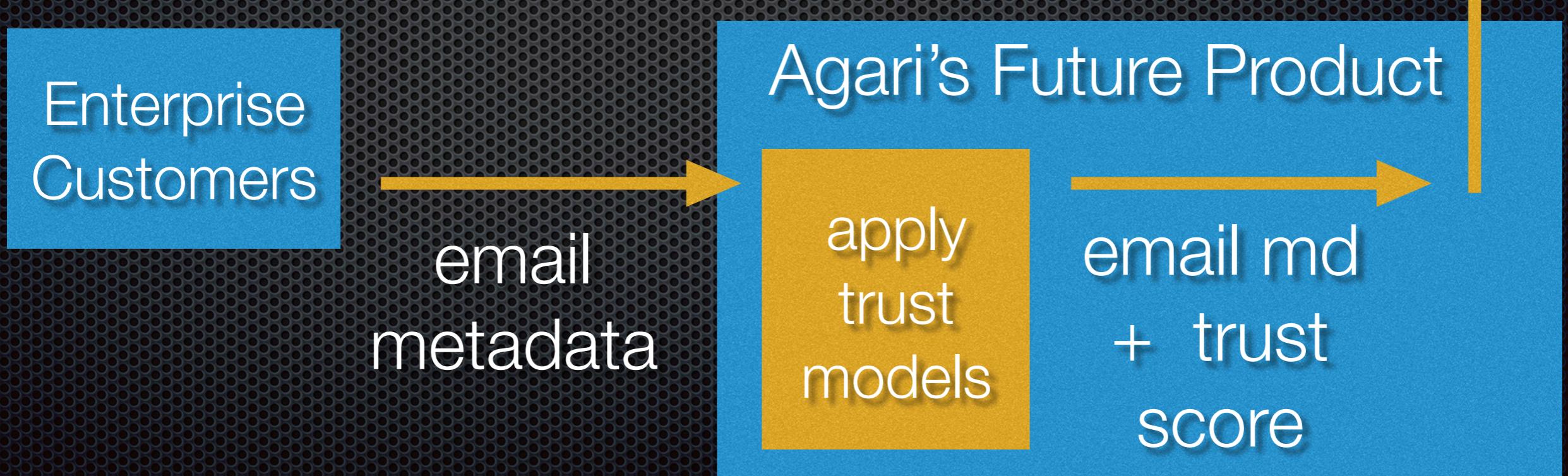
# Agari : What We Do



# Agari : What We Do



# Agari : What We Do



# Apache Airflow @ Agari

## How Do We Use It?

# Classes of Orchestration

**New Product**  
(Enterprise Protect)

build trust  
models

apply trust  
models  
(message  
scoring)

**Operational  
Automation**

cron++  
(general  
job  
scheduler)

**BI / ETL**

N / A

# Classes of Orchestration

## New Product

(Enterprise Protect)

build trust  
models

apply trust  
models  
(message  
scoring)

## Operational Automation

cron++  
(general  
job  
scheduler)

## BI / ETL

N / A

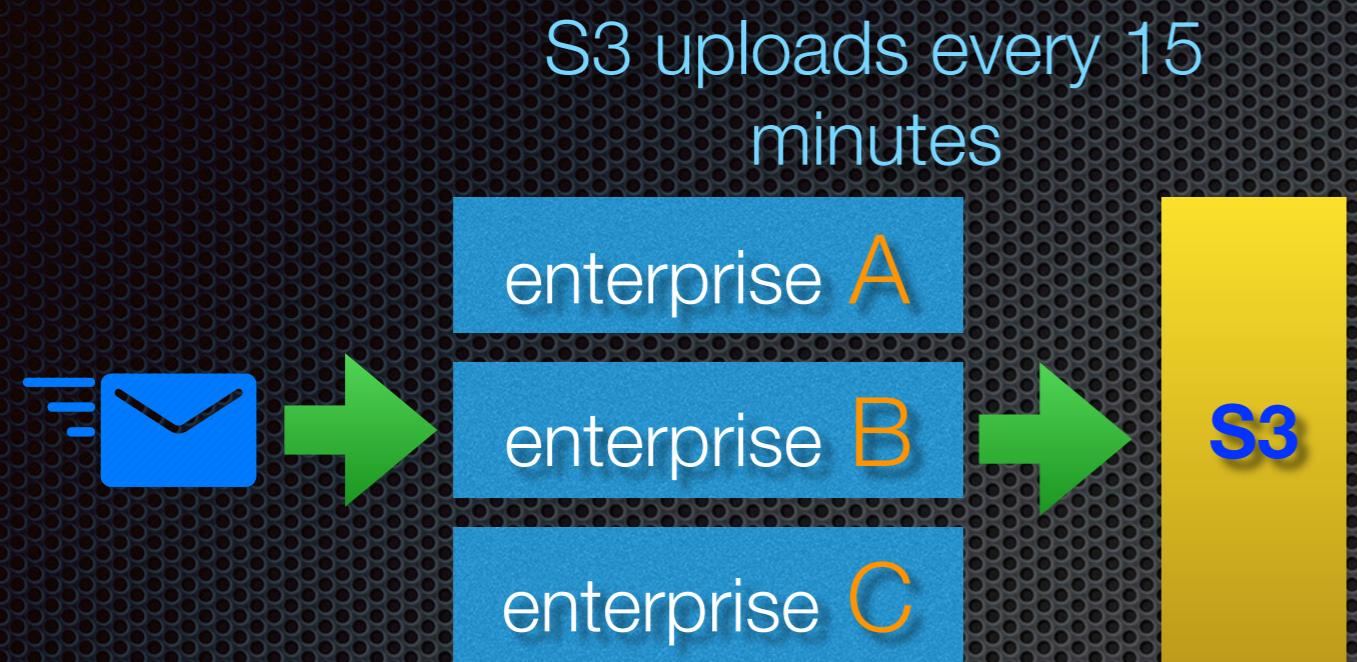


This Talk

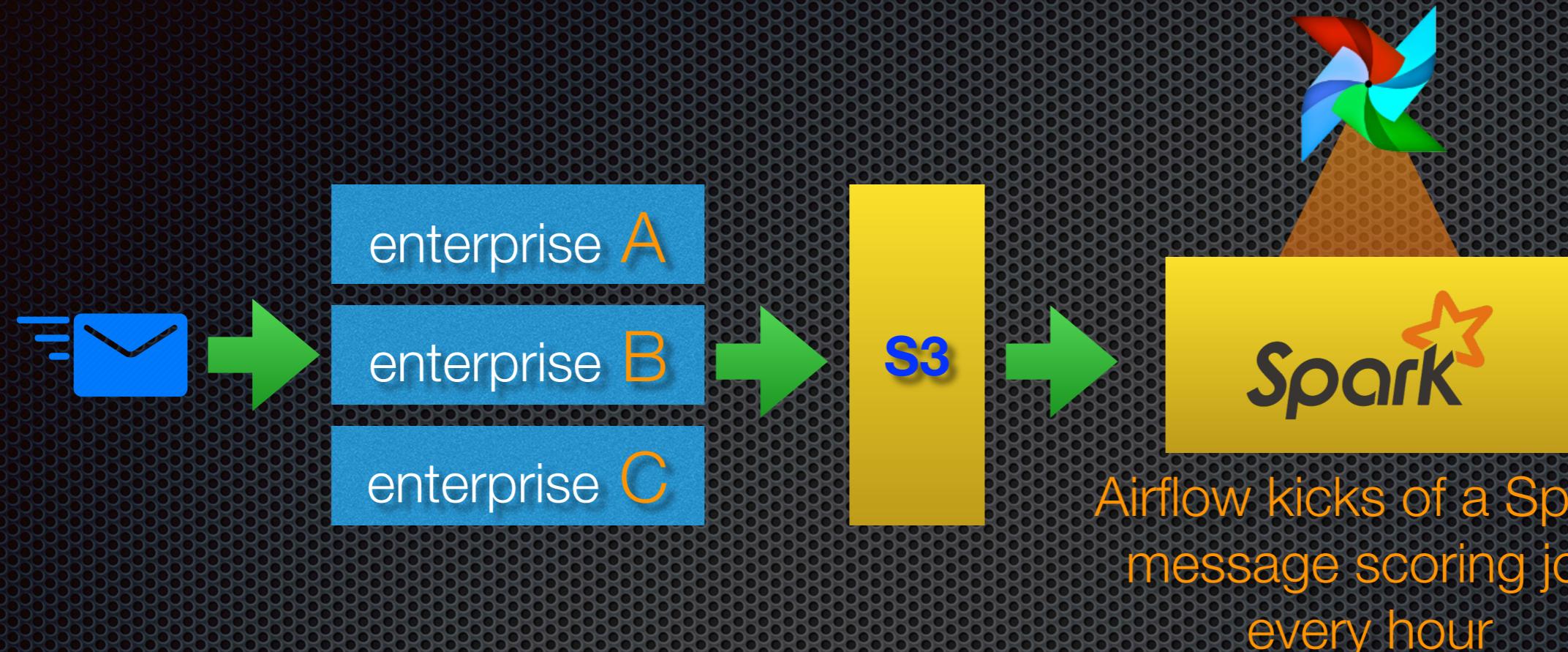
# Use-Case : Message Scoring

## Batch Pipeline Architecture

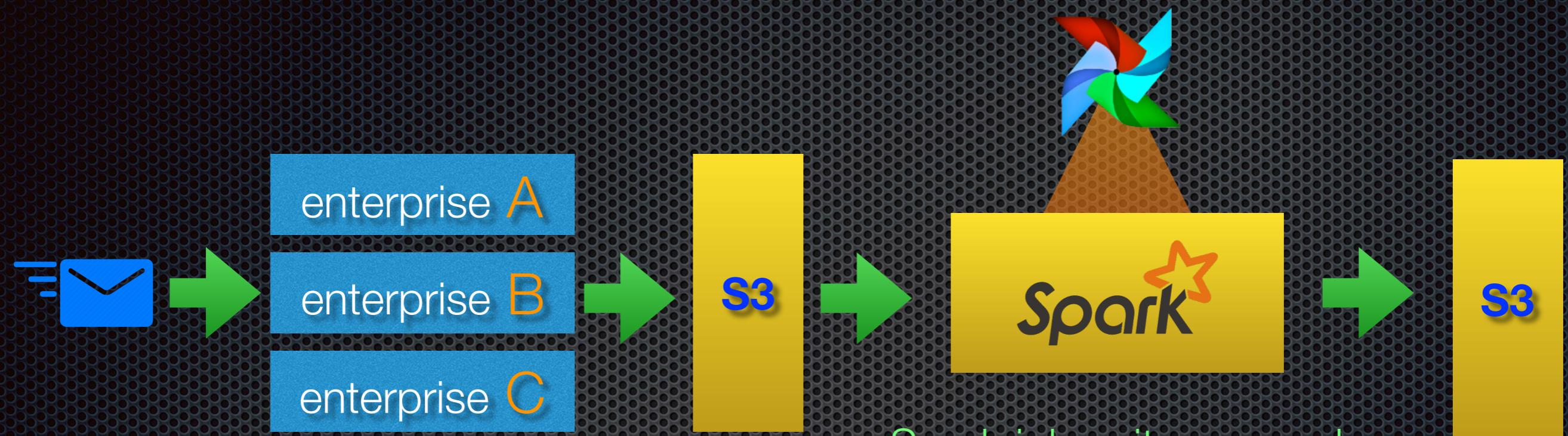
# Use-Case : Message Scoring



# Use-Case : Message Scoring

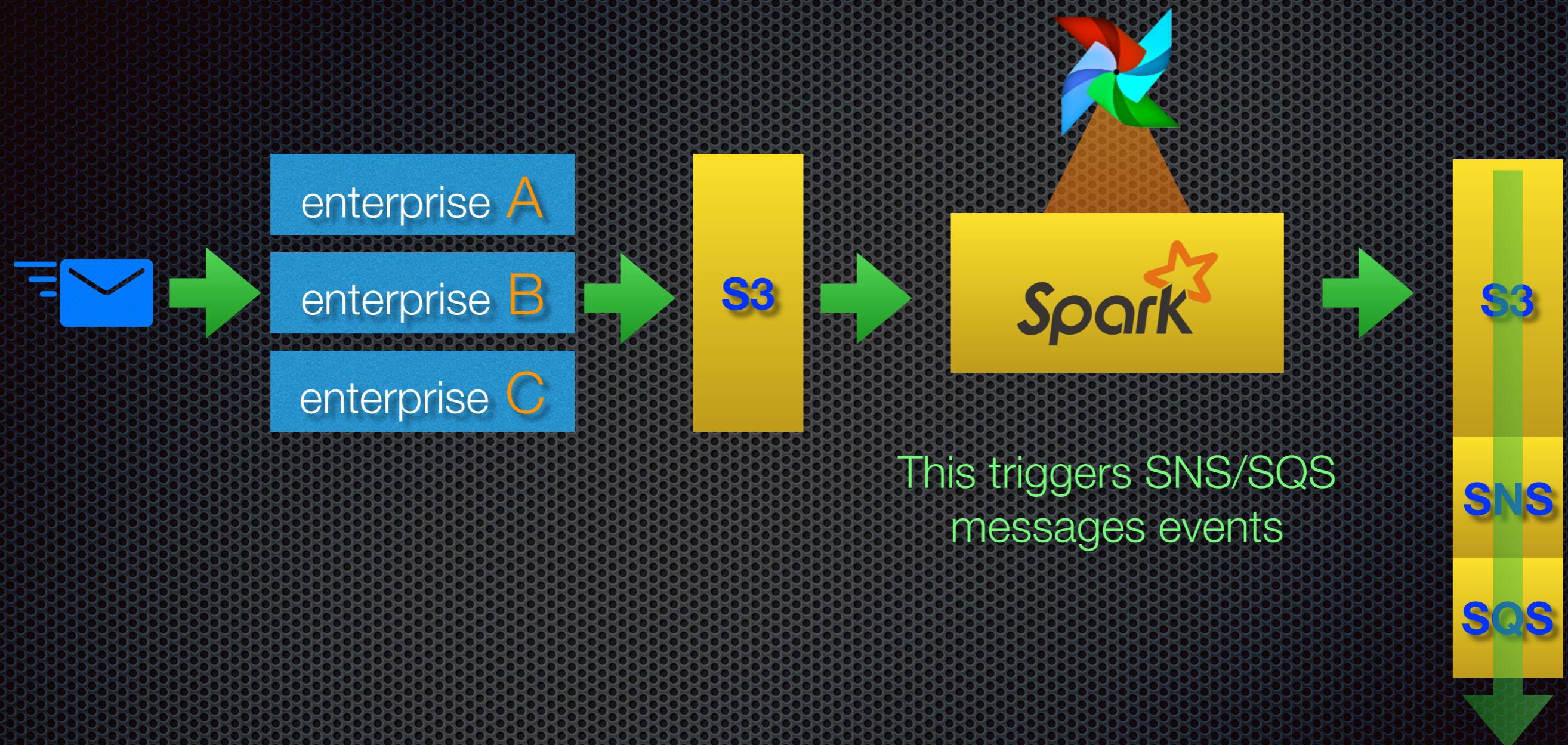


# Use-Case : Message Scoring

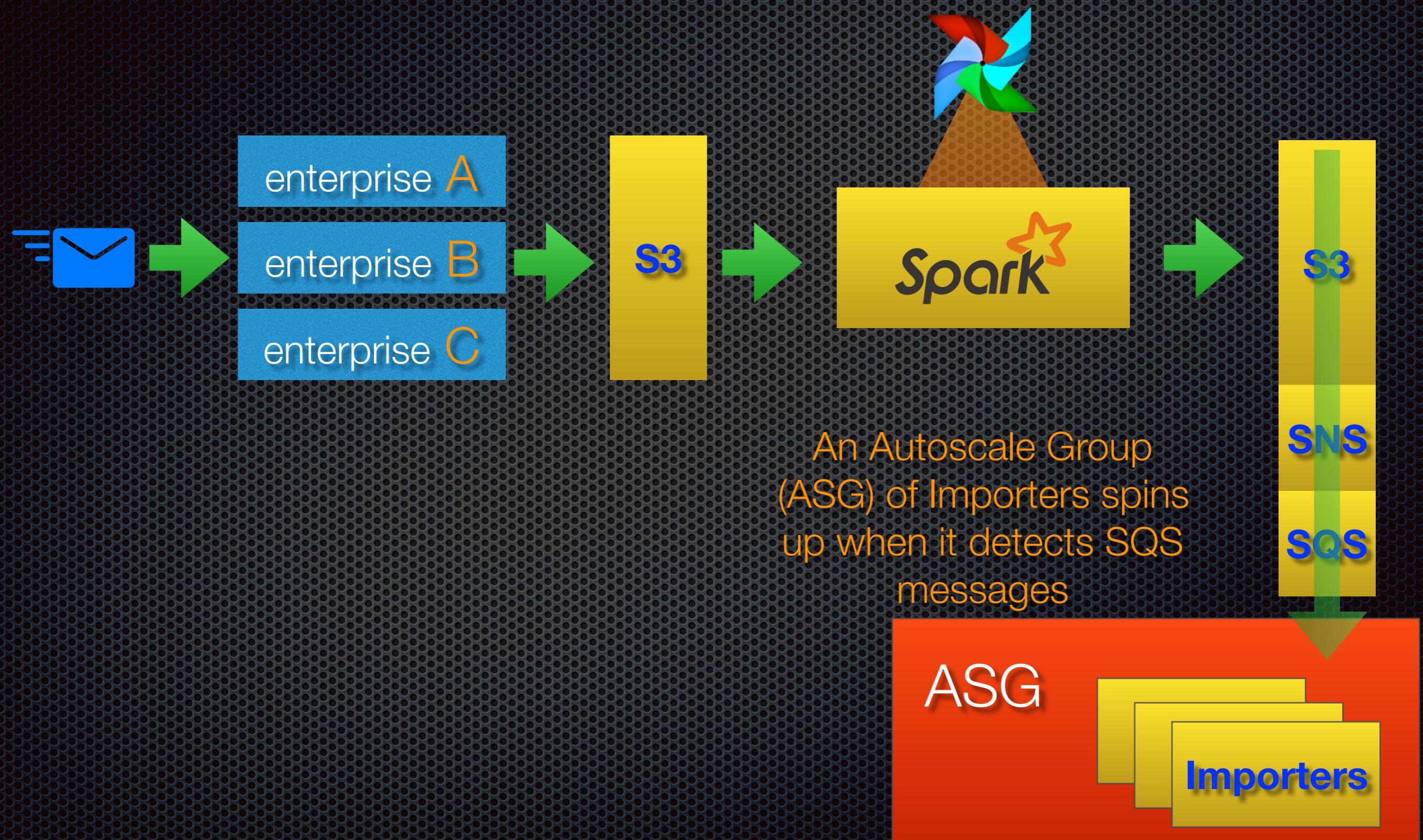


Spark job writes scored  
messages and stats to  
another S3 bucket

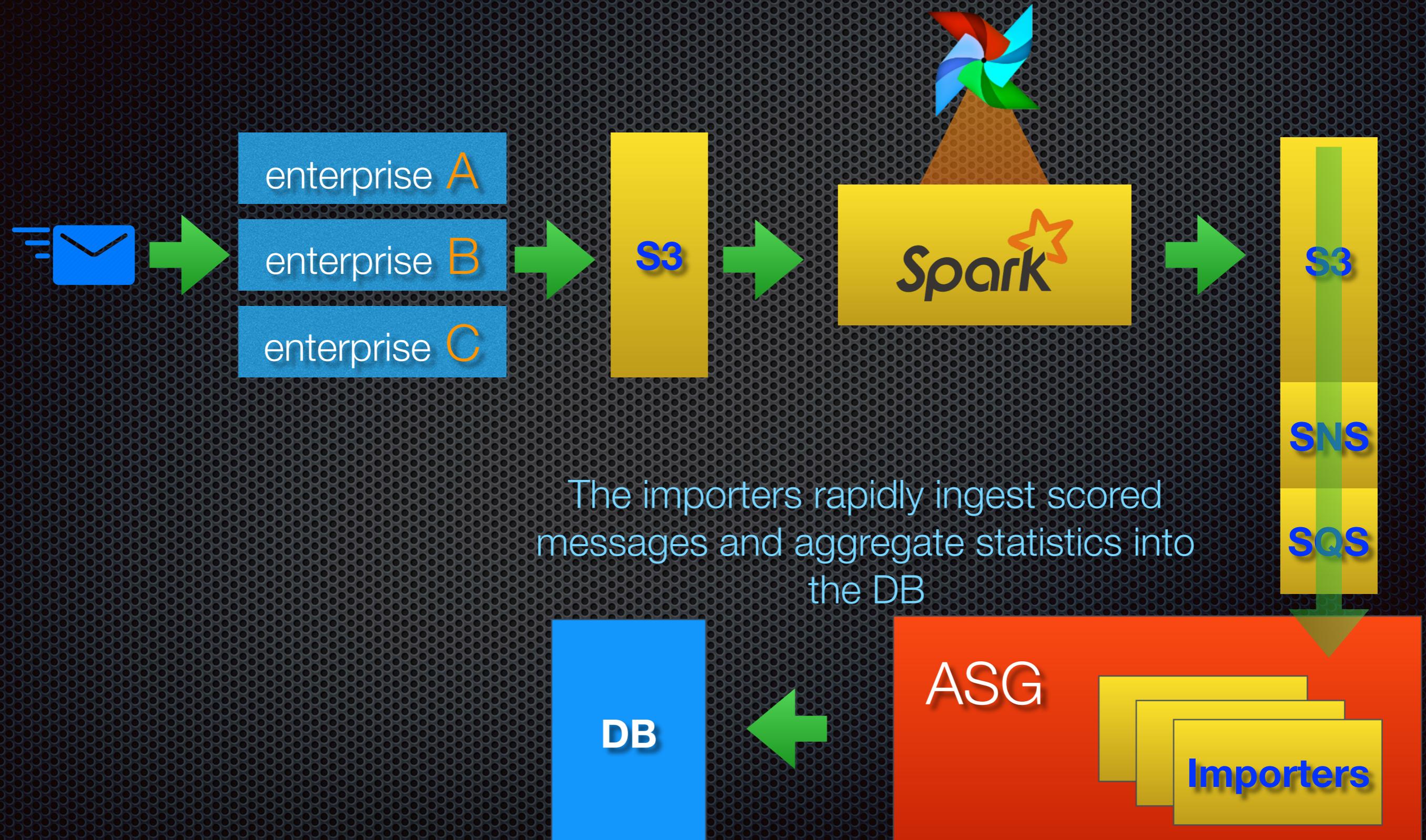
# Use-Case : Message Scoring



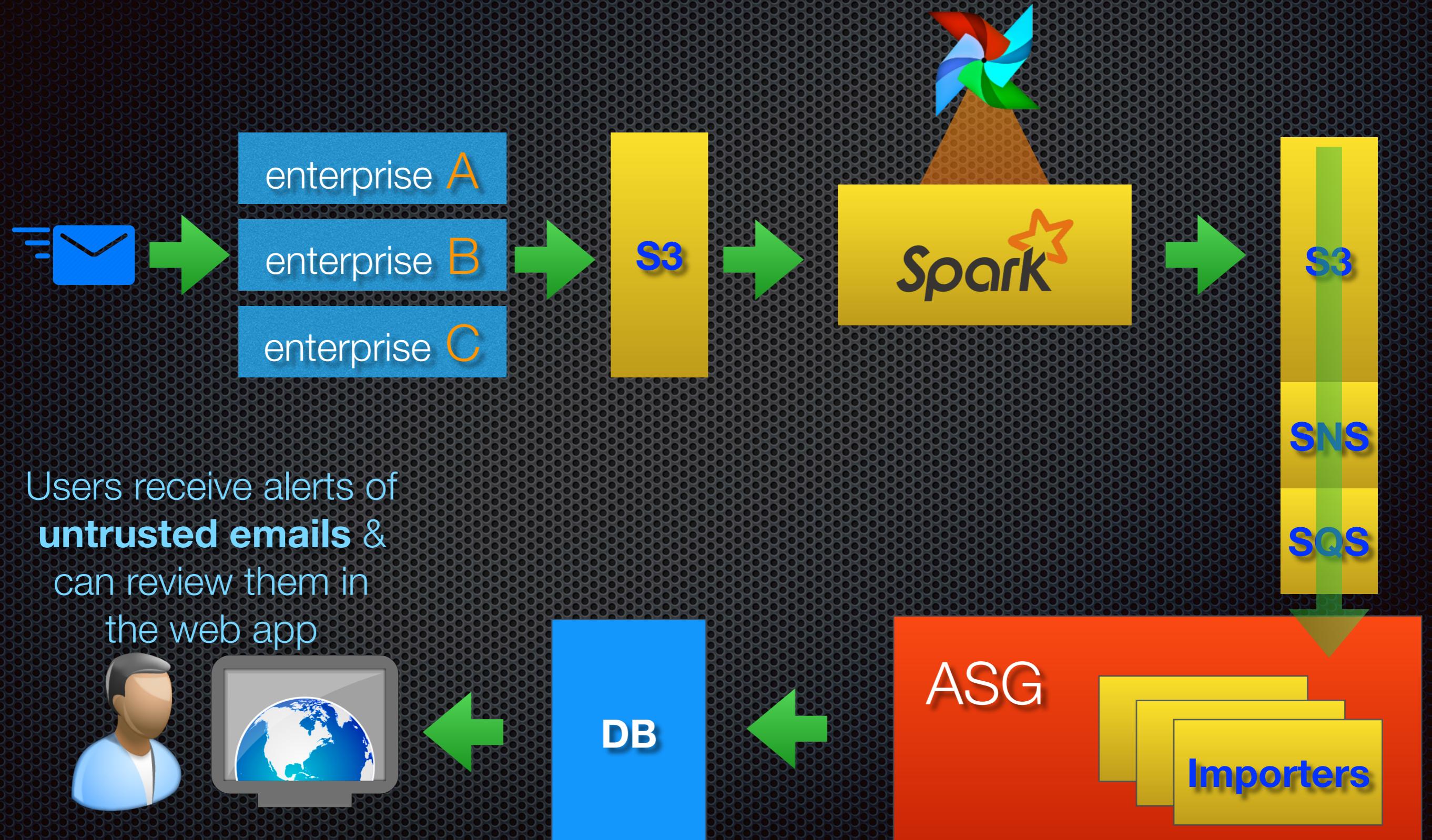
# Use-Case : Message Scoring



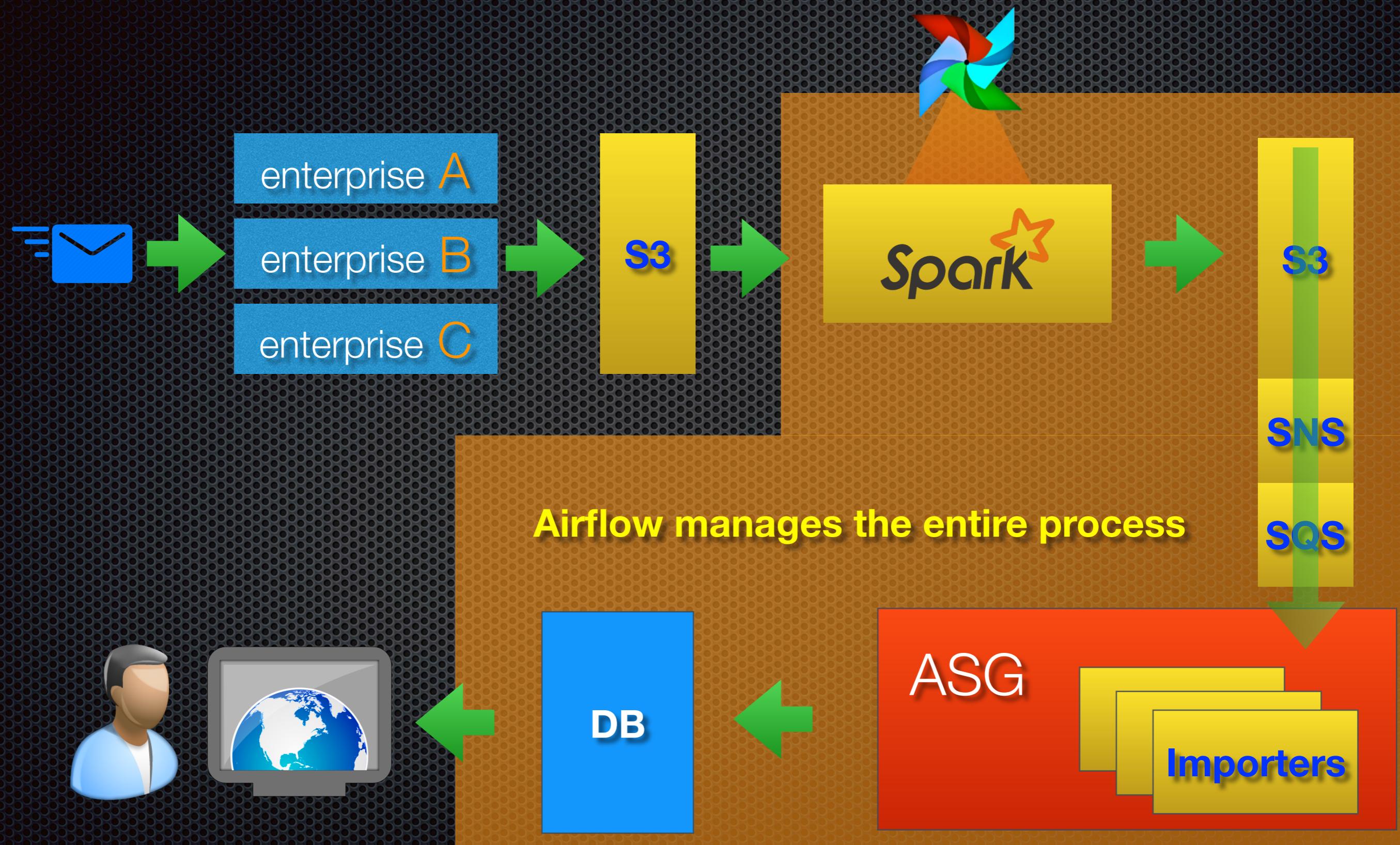
# Use-Case : Message Scoring



# Use-Case : Message Scoring



# Use-Case : Message Scoring

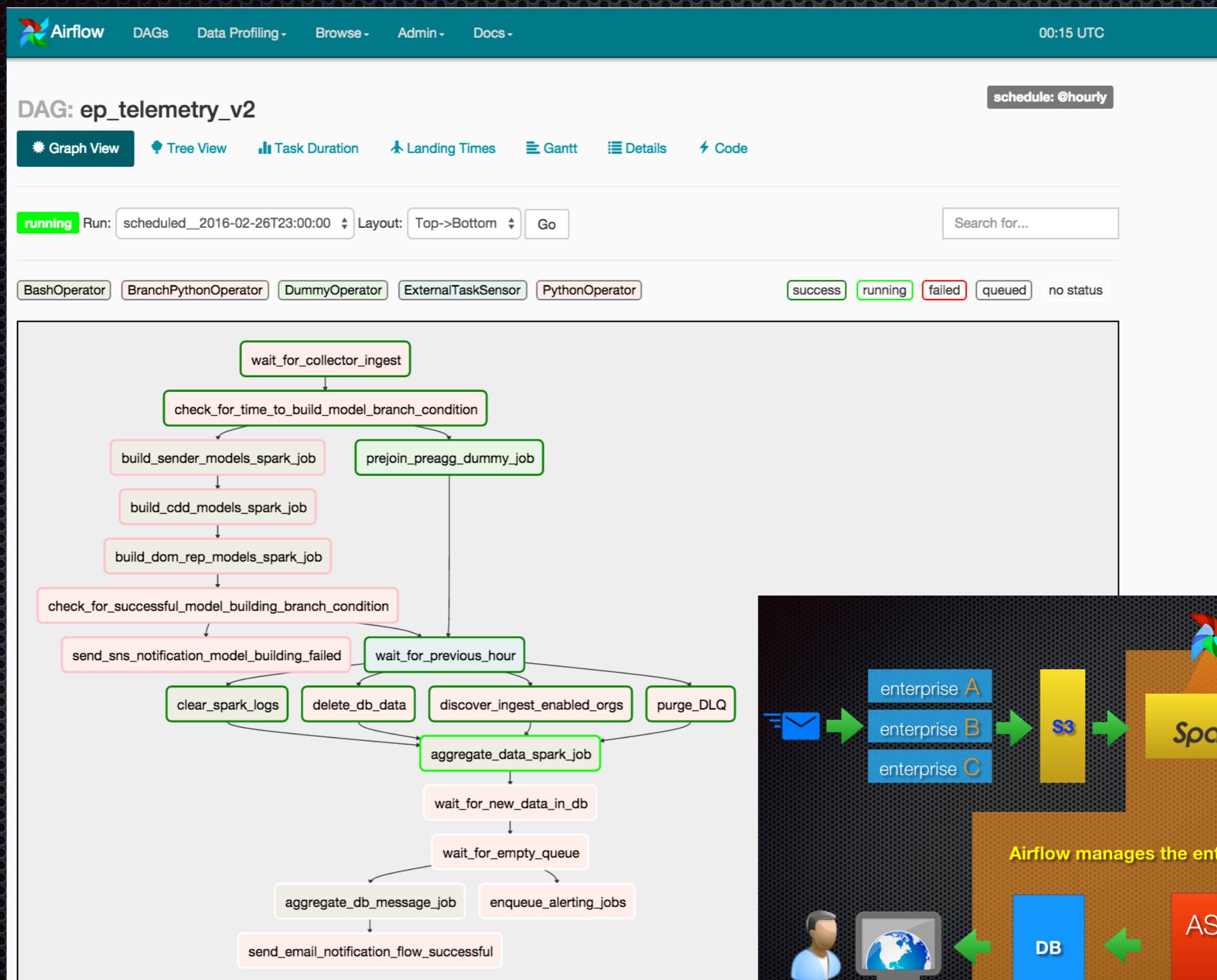


# Use-Case : Message Scoring

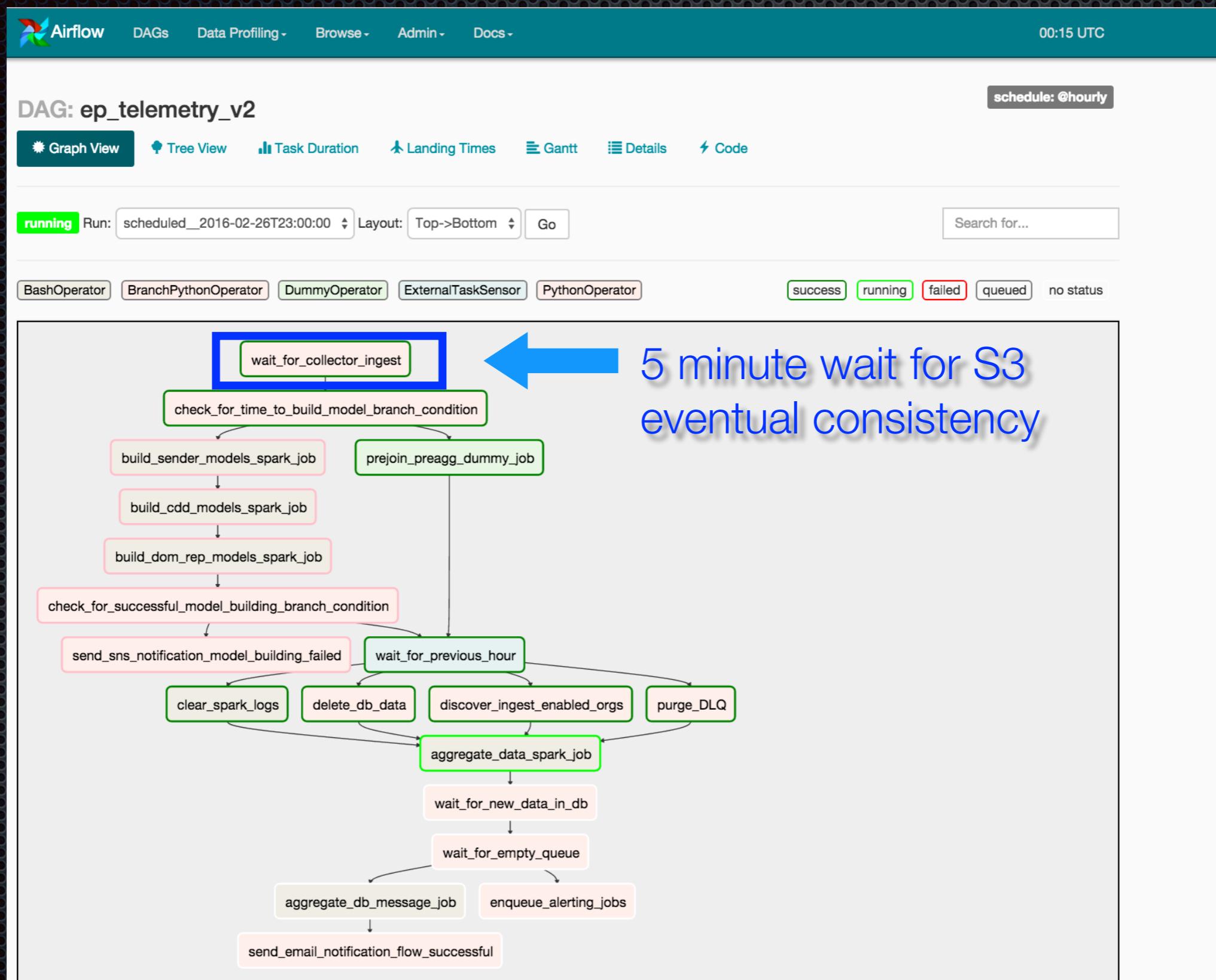
Airflow DAG



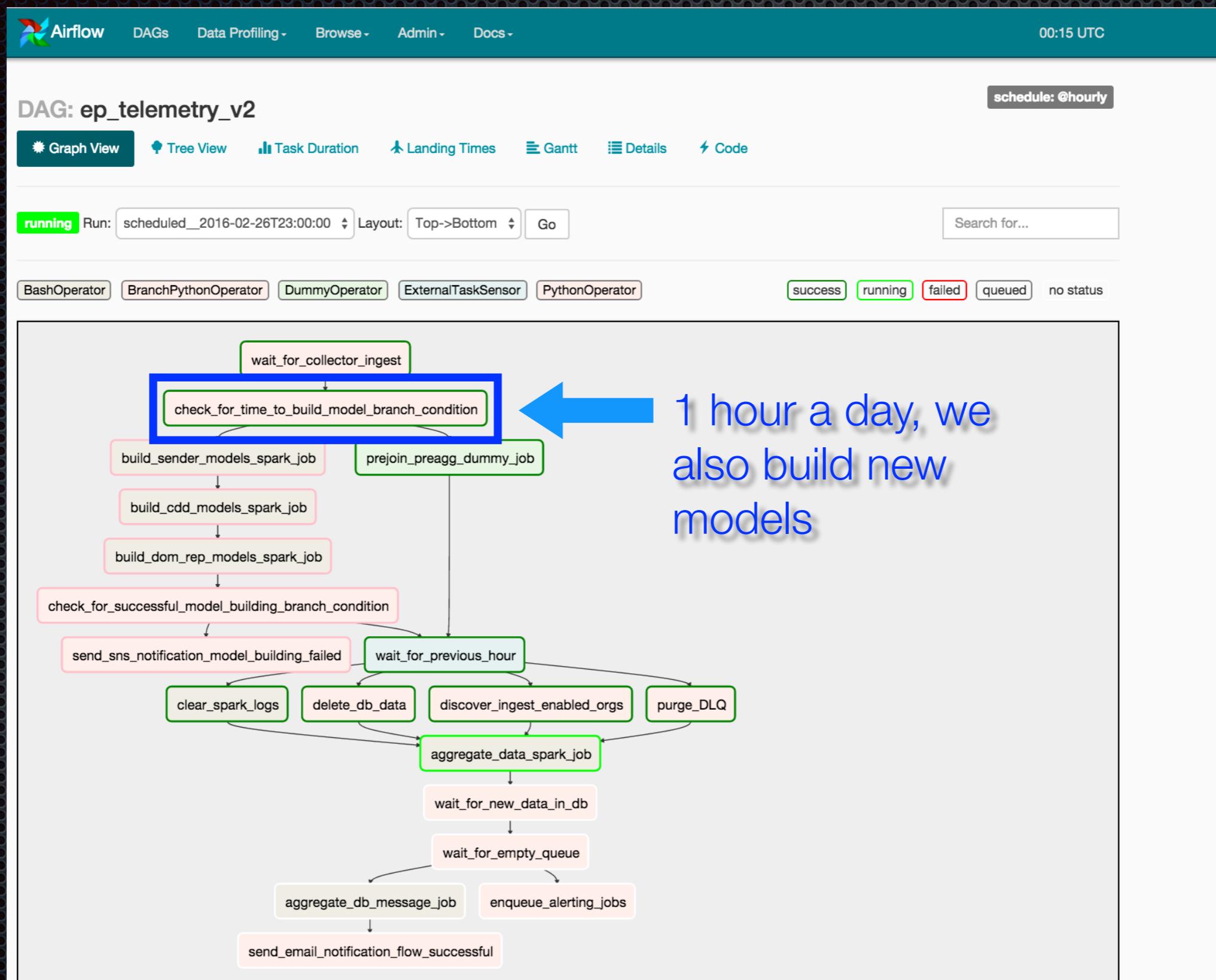
# Airflow DAG



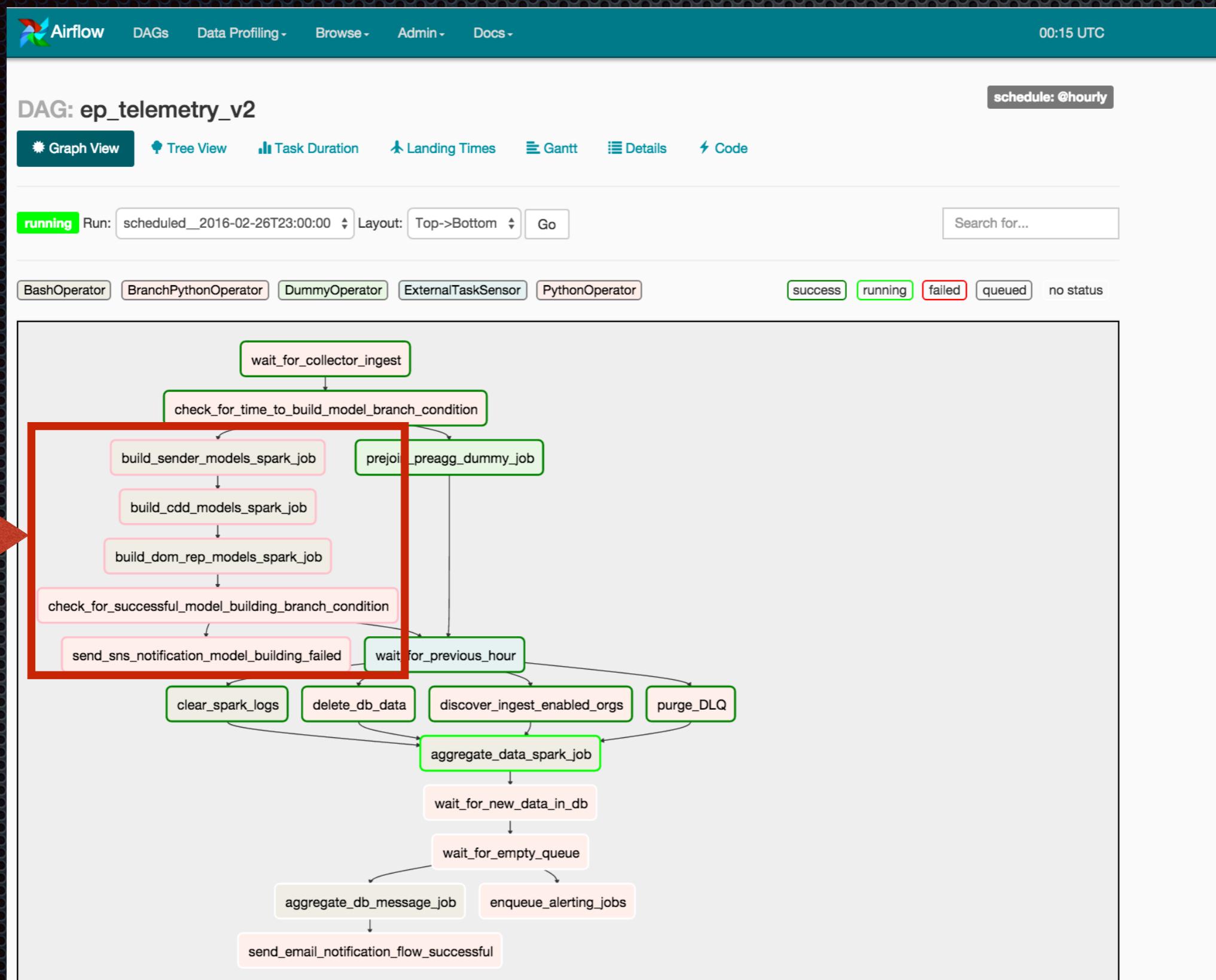
# Airflow DAG



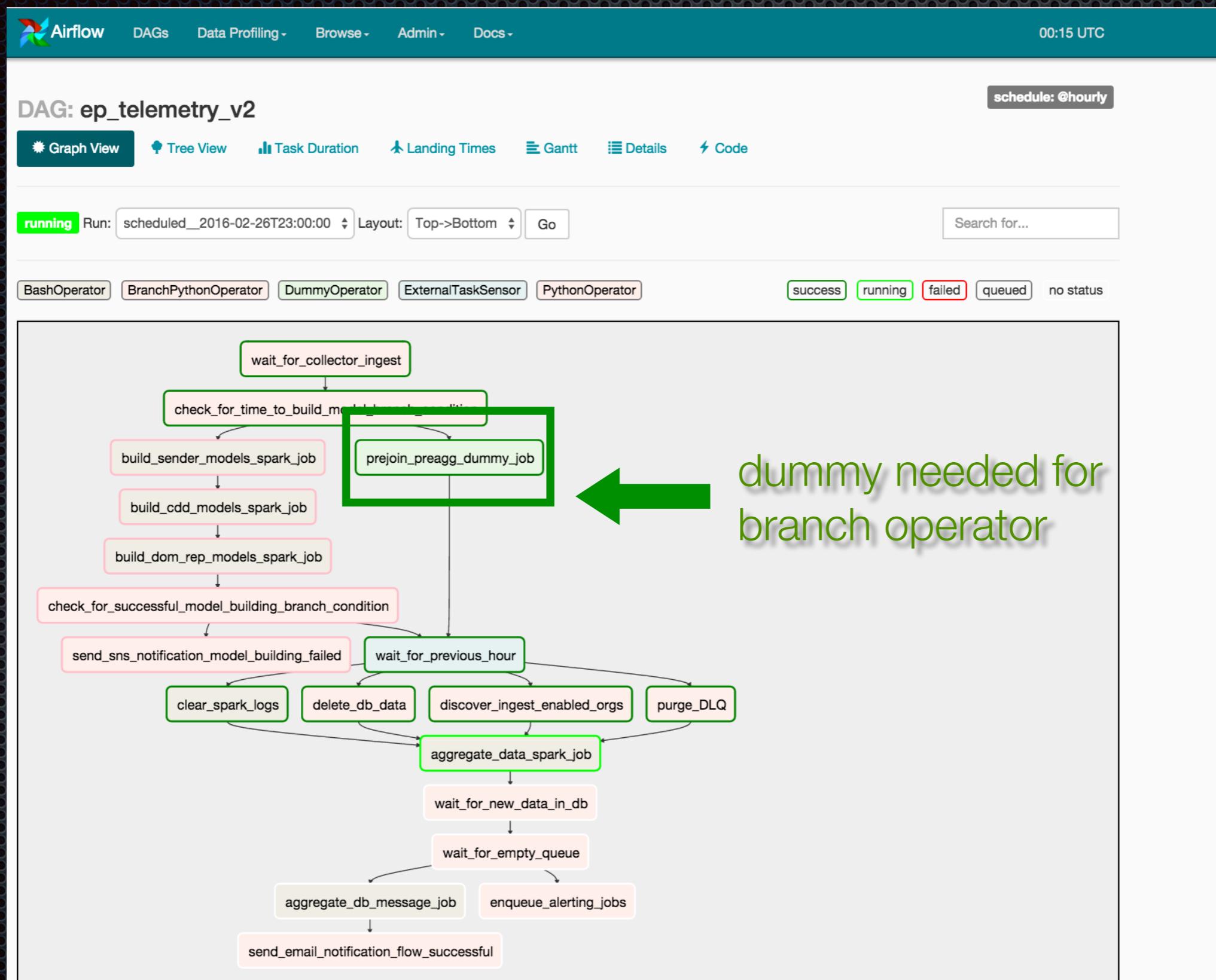
# Airflow DAG



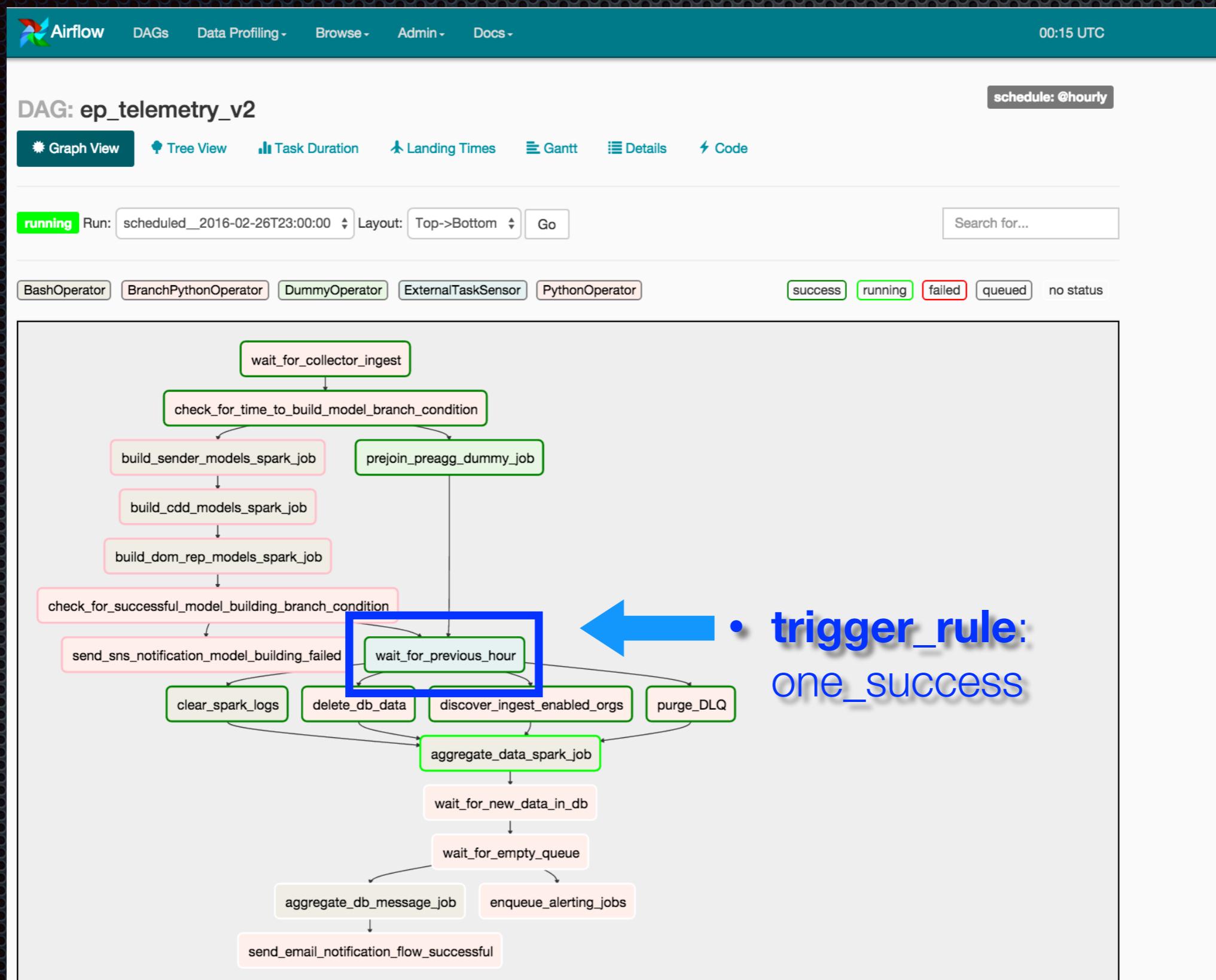
# Airflow DAG



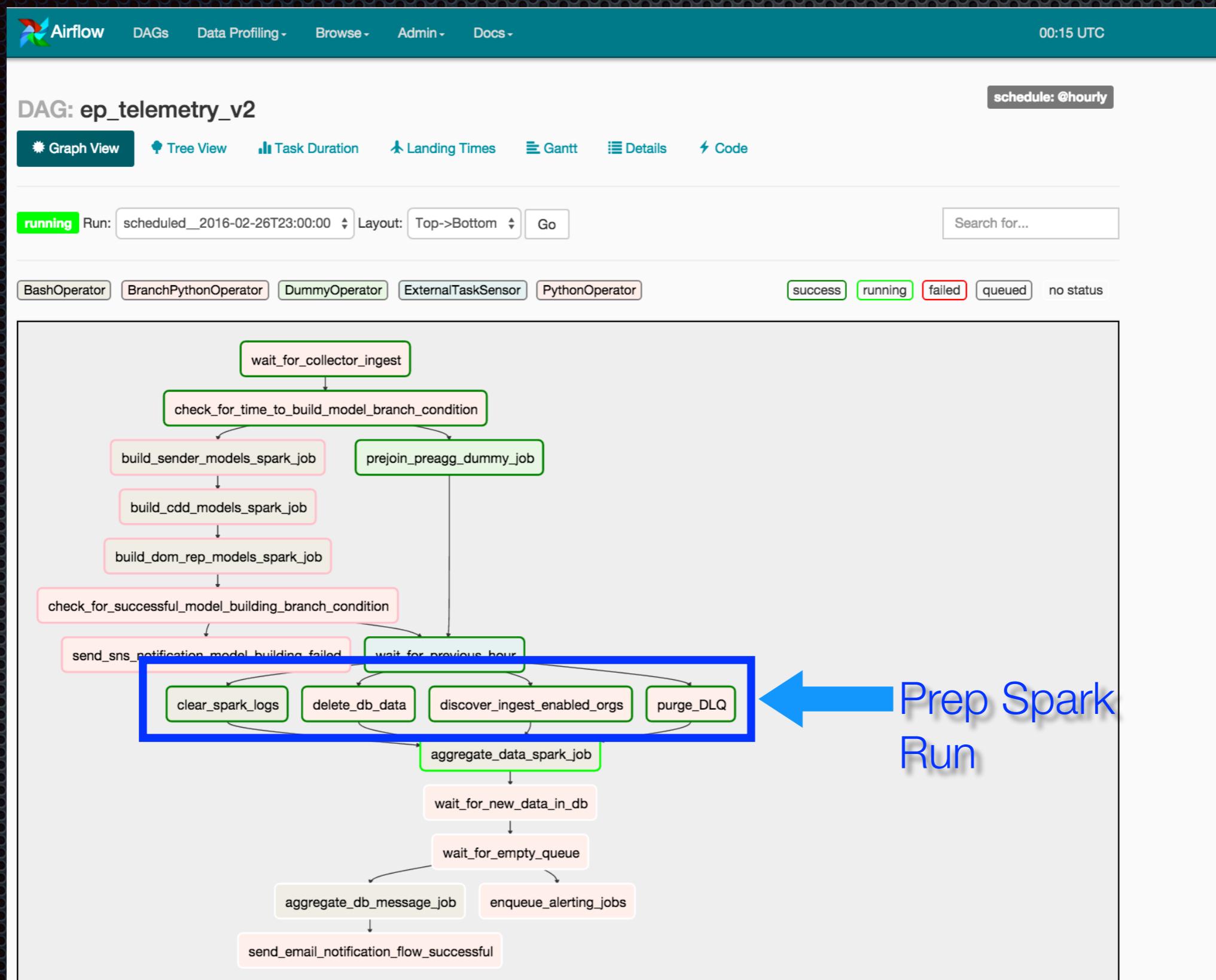
# Airflow DAG



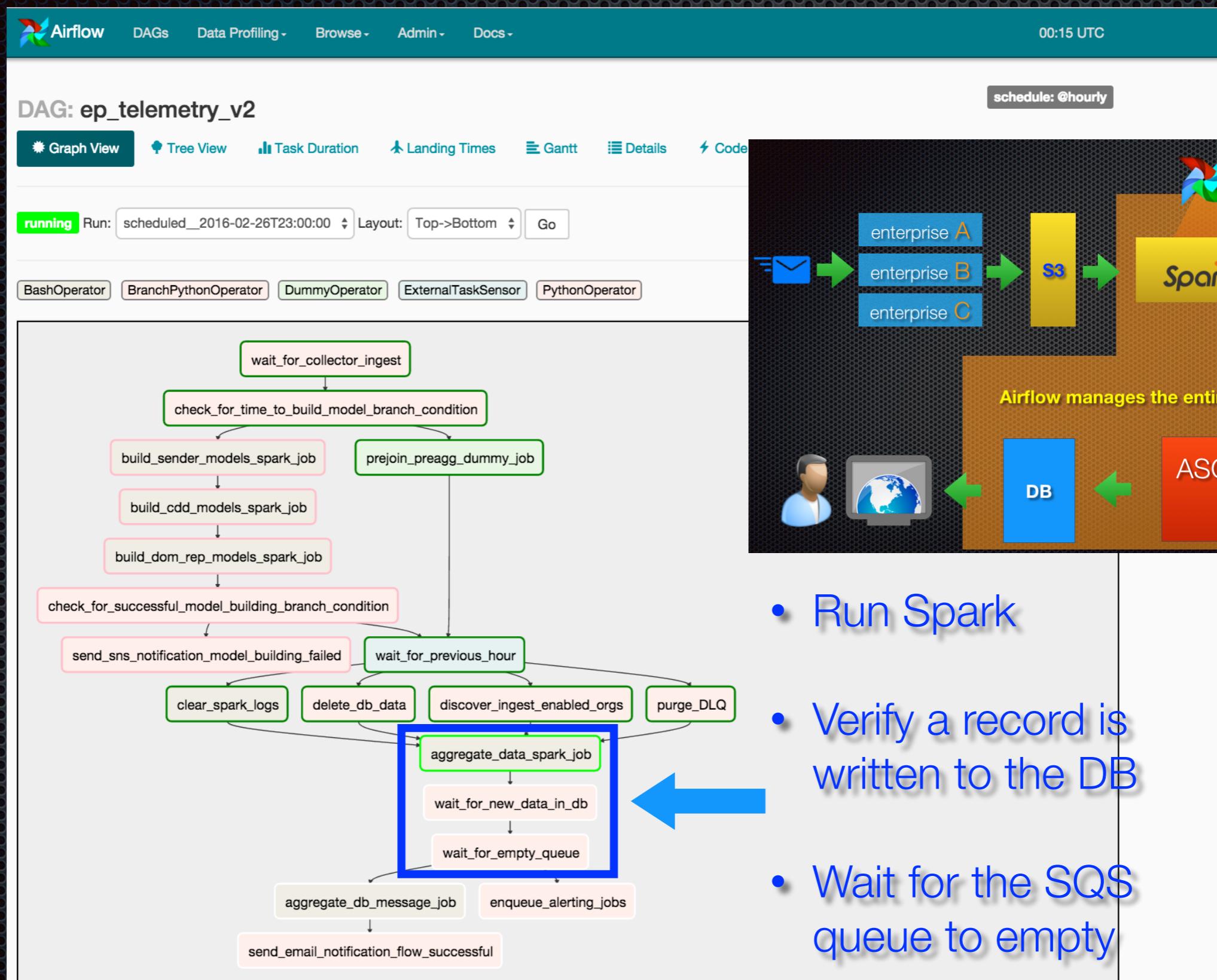
# Airflow DAG



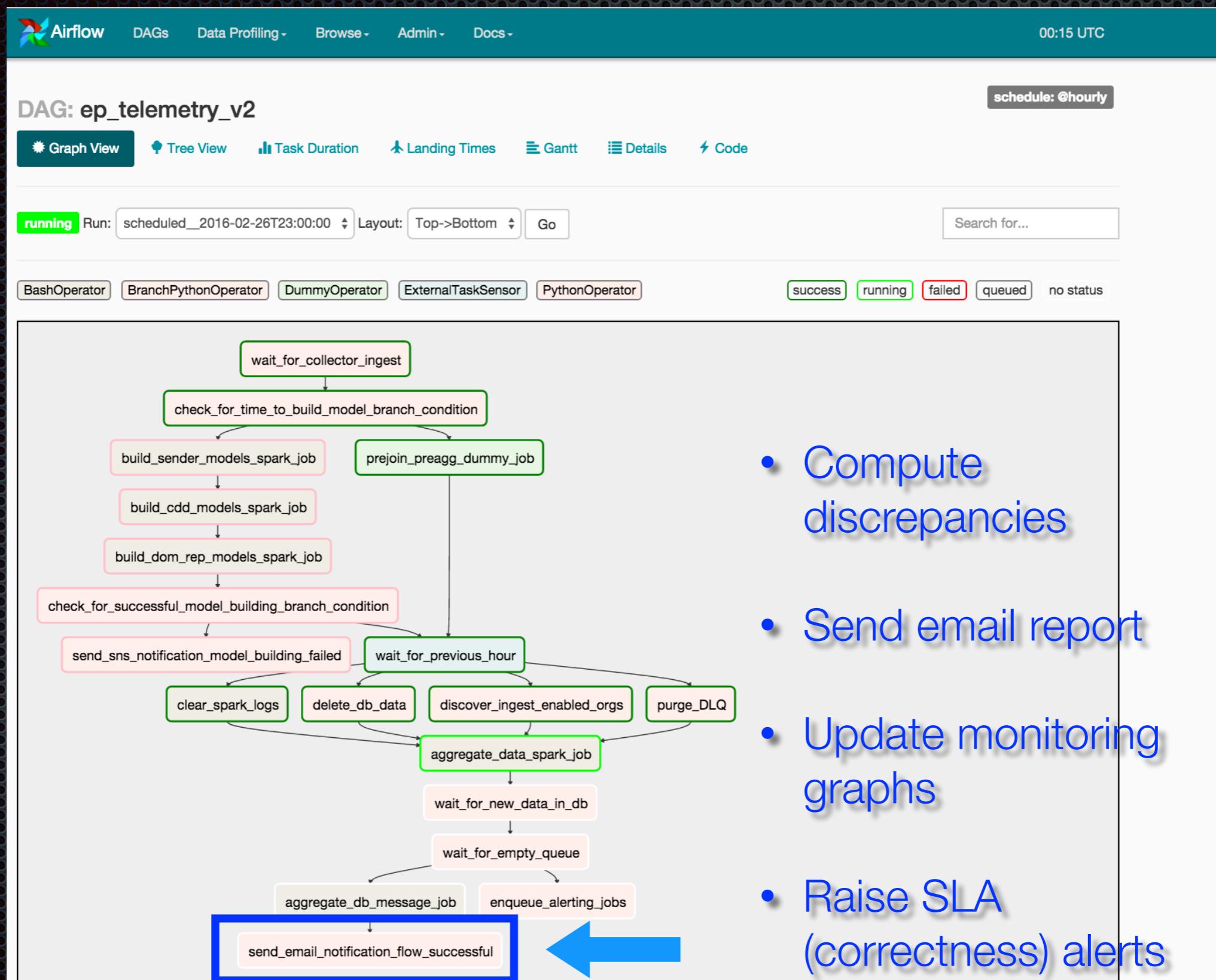
# Airflow DAG



# Airflow DAG



# Airflow DAG



# SLAs & Insights



# Desirable Qualities of a Resilient Data Pipeline

## Correctness

- Data Integrity (no loss, etc...)
- Expected data distributions

## Operability

- Fine-grained Monitoring & Alerting of Correctness & Timeliness SLAs
- Quick Recoverability

## Timeliness

- All output within time-bound SLAs (e.g. 1 hour)

## Scalable/Available

- ASGs, SQS, SNS, S3

# Desirable Qualities of a Resilient Data Pipeline

Correctness	<b>SLA</b>	Operability
<ul style="list-style-type: none"><li>• Data Integrity (no loss, etc...)</li><li>• Expected data distributions</li></ul>		<ul style="list-style-type: none"><li>• Fine-grained Monitoring &amp; Alerting of Correctness &amp; Timeliness SLAs</li><li>• Quick Recoverability</li></ul>
Timeliness	<b>SLA</b>	Scalable/Available
<ul style="list-style-type: none"><li>• All output within time-bound SLAs (e.g. 1 hour)</li></ul>		<ul style="list-style-type: none"><li>• ASGs, SQS, SNS, S3</li></ul>

# Correctness : Email Reporting

airflow@agari.com

to AirflowJobComp.

6:20 PM (16 hours ago) 

Hi EP Folks! The EP Data Pipeline (Airflow) Loaded Data in the EP\_STAGE environment for day: 2016-04-28 12:00:00 GMT/UTC

[Airflow](#)

Day	RDAs	Messages
2016-04-28 00:00:00	22058	261254

## Message Counts by Org and by Pipeline Stage Output

Org_ID	Collector Msgs	Agg Msgs	DB Msgs	Discrepancy	Discr %	NRT DB Msgs	NRT Discrepancy	NRT Discr %	Skipped	Notes
1	340	266	266	0	0.0	340	-74	-21.8	74	
5	43115	43115	43115	0	0.0	358	42757	99.2	0	
6	156	153	153	0	0.0	0	153	98.1	3	
7	73	68	68	0	0.0	0	68	93.2	5	
12	315	111	111	0	0.0	0	111	35.2	204	
14	35648	28731	28731	0	0.0	0	28731	80.6	6917	
16	35383	34010	34010	0	0.0	0	34010	96.1	1373	
25	20530	20530	20530	0	0.0	0	20530	100.0	0	
54	30003	26190	26190	0	0.0	0	26190	87.3	3813	
56	108084	108080	108080	0	0.0	0	108080	100.0	4	

↑  
orgs



# Correctness : Email Reporting

airflow@agari.com

to AirflowJobComp.

6:20 PM (16 hours ago) 



Hi EP Folks! The EP Data Pipeline (Airflow) Loaded Data in the EP\_STAGE environment for day: 2016-04-28 12:00:00 GMT/UTC  
[Airflow](#)

Day	RDAs	Messages
2016-04-28 00:00:00	22058	261254

## Message Counts by Org and by Pipeline Stage Output

Org_ID	Collector Msgs	Agg Msgs	DB Msgs	Discrepancy	Discr %	NRT DB Msgs	NRT Discrepancy	NRT Discr %	Skipped	Notes
1	340	266	266	0	0.0	340	-74	-21.8	74	
5	43115	43115	43115	0	0.0	358	42757	99.2	0	
6	156	153	153	0	0.0	0	153	98.1	3	
7	73	68	68	0	0.0	0	68	93.2	5	
12	315	111	111	0	0.0	0	111	35.2	204	
14	35648	28731	28731	0	0.0	0	28731	80.6	6917	
16	35383	34010	34010	0	0.0	0	34010	96.1	1373	
25	20530	20530	20530	0	0.0	0	20530	100.0	0	
54	30003	26190	26190	0	0.0	0	26190	87.3	3813	
56	108084	108080	108080	0	0.0	0	108080	100.0	4	

orgs



For each org, we check for duplicate or missing data as a count & percentage

# Correctness : Email Reporting

airflow@agari.com

to AirflowJobComp.

6:20 PM (16 hours ago) 

Hi EP Folks! The EP Data Pipeline (Airflow) Loaded Data in the EP\_STAGE environment for day: 2016-04-28 12:00:00 GMT/UTC

[Airflow](#)

Day	RDAs	Messages
2016-04-28 00:00:00	22058	261254

Message Counts by Org and by Pipeline Stage Output

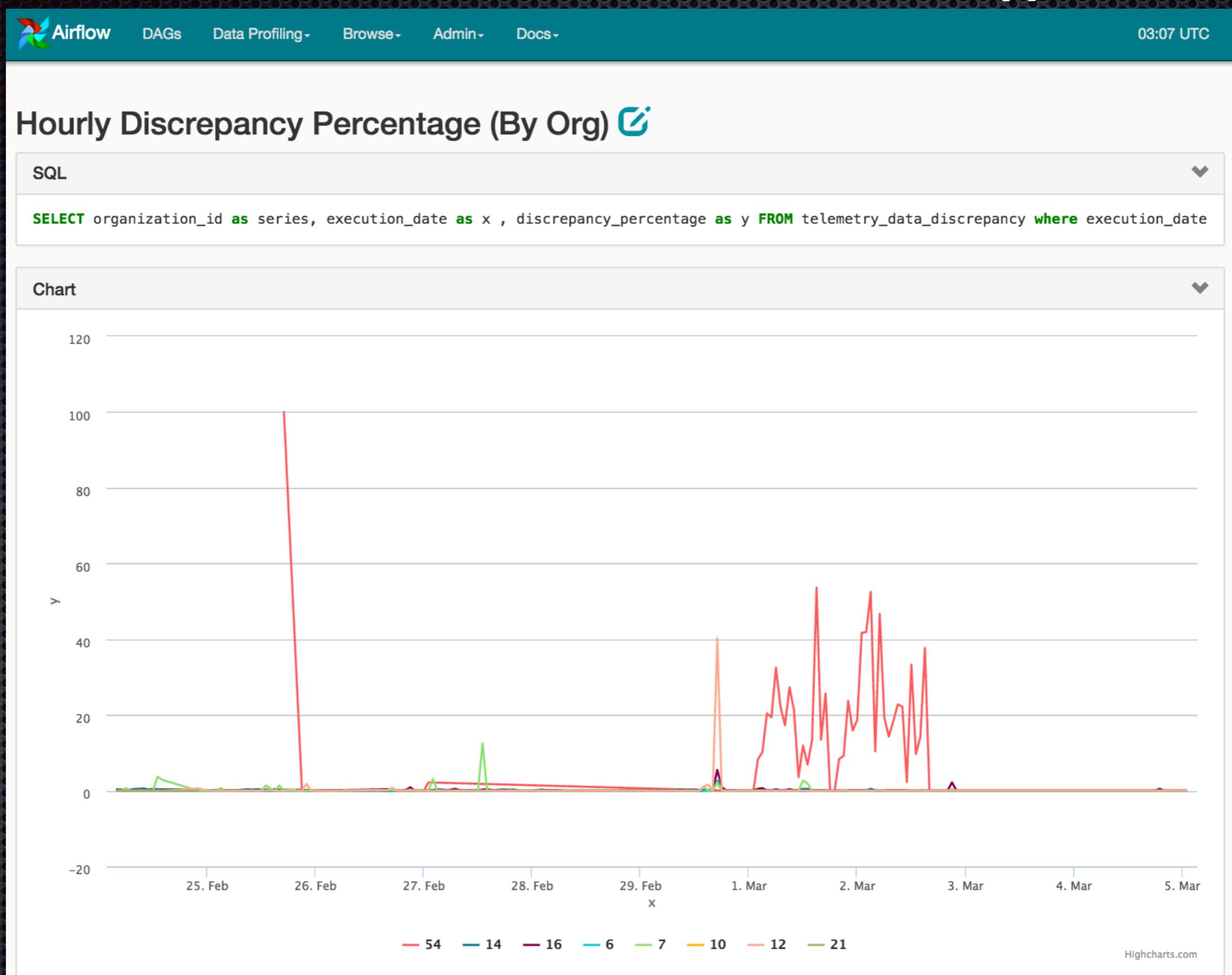
Org_ID	Collector Msgs	Agg Msgs	DB Msgs	Discrepancy	Discr %	NRT DB Msgs	NRT Discrepancy	NRT Discr %	Skipped	Notes
1	340	266	266	0	0.0	340	-74	-21.8	74	
5	43115	43115	43115	0	0.0	358	42757	99.2	0	
6	156	153	153	0	0.0	0	153	98.1	3	
7	73	68	68	0	0.0	0	68	93.2	5	
12	315	111	111	0	0.0	0	111	35.2	204	
14	35648	28731	28731	0	0.0	0	28731	80.6	6917	
16	35383	34010	34010	0	0.0	0	34010	96.1	1373	
25	20530	20530	20530	0	0.0	0	20530	100.0	0	
54	30003	26190	26190	0	0.0	0	26190	87.3	3813	
56	108084	108080	108080	0	0.0	0	108080	100.0	4	

orgs



These are the 3 stages of the pipeline. We can detect where a discrepancy is coming from - often related to a code push!

# Correctness : Monitoring



# Correctness & Timeliness : Alerting

The screenshot shows a Slack interface with the following details:

- Left Sidebar:** Shows the team "Agari" and the user "sid". It also lists channels: # STARRED, # airflow, # analysis, # eng, # eng-ep, # ep-ops (which is highlighted), # ep-real-time-alerting, and 3 users: scotfree, kevin, wforr...  
A red box highlights the "# ep-ops" channel.
- Header:** The channel is "#ep-ops" with a star icon. A description says "A channel with info that can help you resolve VictorOps alerts". There are 13 members, a search bar, and a "Today" tab.
- Messages:**
  - Airflow BOT 10:00 AM:** EP\_STAGE - ep\_telemetry\_v2 SLA Miss for task [send\\_email\\_notification\\_flow\\_successful on 2016-02-26T18:00:00](#)
  - sid 10:02 AM:** So, we need to catch up here.. any reason we don't want to just mark success for the many hours it is behind?
  - Airflow BOT 11:00 AM:** EP\_STAGE - ep\_telemetry\_v2 SLA Miss for task [send\\_email\\_notification\\_flow\\_successful on 2016-02-26T19:00:00](#)
  - Airflow BOT 5:27 PM:** ep\_telemetry\_v2 on [etl-00.ep-old.prod.agari.com](#) completed [2016-02-03 00:00:00](#) with High Discrepancies
  - Airflow BOT 8:48 PM ☆:** ep\_telemetry\_v2 on [workflow-00.ep.stage.agari.com](#) completed [2016-02-26 03:00:00](#) with 1 DLQs : Sample Exception
- Timestamps:** The messages are timestamped with "Today" and "February 3rd".

# Correctness & Timeliness : Alerting

Agari

sid

STARRED

# airflow

# analysis

# eng

# eng-ep

# ep-ops

# ep-real-time-alerting

3 scotfree, karen, wforr

Timeliness SLA miss

Slack

#ep-ops A channel with info that can help you resolve VictorOps alerts

13 Search

Today

Airflow BOT 10:00 AM EP\_STAGE - ep\_telemetry\_v2 SLA Miss for task [send\\_email\\_notification\\_flow\\_successful on 2016-02-26T18:00:00](#)

sid 10:02 AM So, we need to catch up here.. any reason we don't want to just mark success for the many hours it is behind?

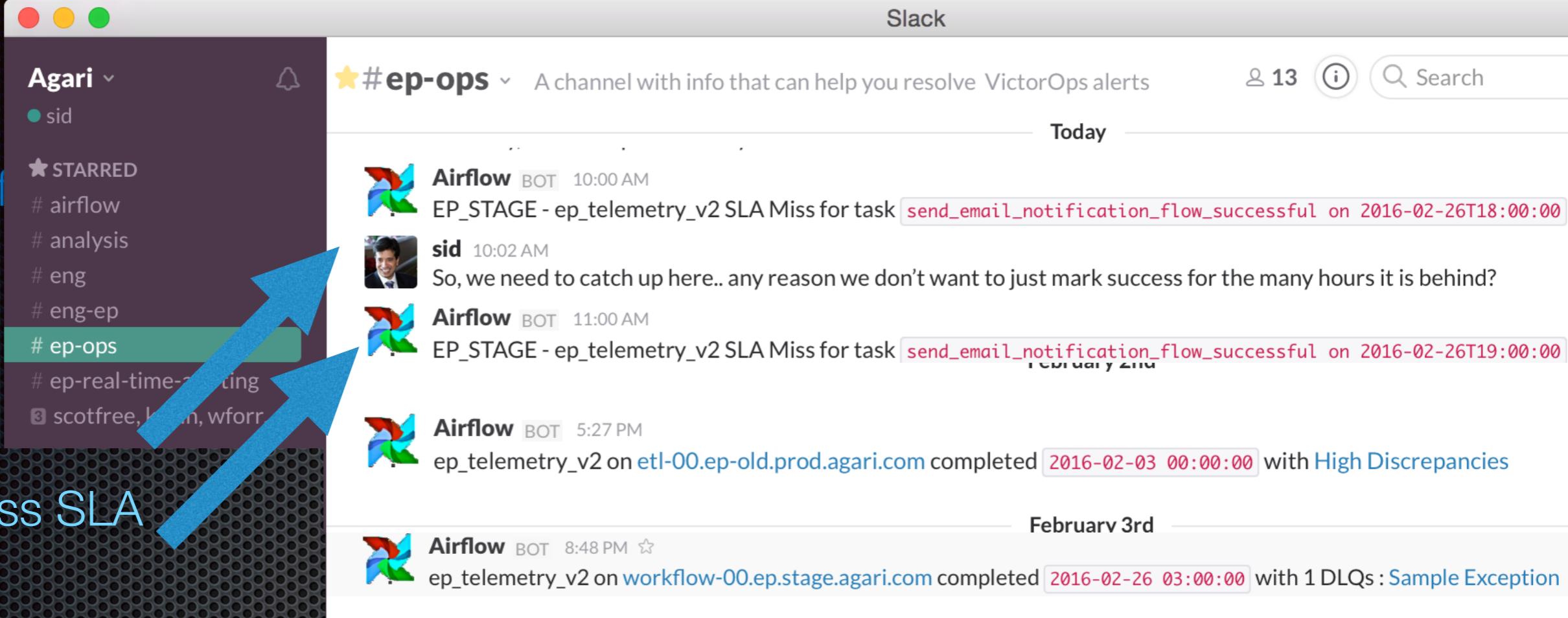
Airflow BOT 11:00 AM EP\_STAGE - ep\_telemetry\_v2 SLA Miss for task [send\\_email\\_notification\\_flow\\_successful on 2016-02-26T19:00:00](#)

Airflow BOT 5:27 PM ep\_telemetry\_v2 on [etl-00.ep-old.prod.agari.com](#) completed [2016-02-03 00:00:00](#) with High Discrepancies

Airflow BOT 8:48 PM ep\_telemetry\_v2 on [workflow-00.ep.stage.agari.com](#) completed [2016-02-26 03:00:00](#) with 1 DLQs : Sample Exception

February 3rd

# Correctness & Timeliness : Alerting



Agari sid

STARRED

- # airflow
- # analysis
- # eng
- # eng-ep
- # ep-ops**
- # ep-real-time-alerting
- 3 scotfree, kiran, wforr

Timeliness SLA miss

Slack

#ep-ops A channel with info that can help you resolve VictorOps alerts 13 Search

Today

Airflow BOT 10:00 AM EP\_STAGE - ep\_telemetry\_v2 SLA Miss for task [send\\_email\\_notification\\_flow\\_successful on 2016-02-26T18:00:00](#)

sid 10:02 AM So, we need to catch up here.. any reason we don't want to just mark success for the many hours it is behind?

Airflow BOT 11:00 AM EP\_STAGE - ep\_telemetry\_v2 SLA Miss for task [send\\_email\\_notification\\_flow\\_successful on 2016-02-26T19:00:00](#)

Airflow BOT 5:27 PM ep\_telemetry\_v2 on [etl-00.ep-old.prod.agari.com](#) completed [2016-02-03 00:00:00](#) with High Discrepancies

Airflow BOT 8:48 PM ep\_telemetry\_v2 on [workflow-00.ep.stage.agari.com](#) completed [2016-02-26 03:00:00](#) with 1 DLQs : Sample Exception

February 3rd

```
dag = DAG(DAG_NAME,  
          schedule_interval='@hourly',  
          default_args=default_args,  
          sla_miss_callback=sla_alert_func)
```

# Correctness & Timeliness : Alerting

Agari

sid

STARRED

# airflow

# analysis

# eng

# eng-ep

# ep-ops

# ep-real-time-alerting

3 scotfree, karen, wforr

Timeliness SLA miss

Slack

#ep-ops A channel with info that can help you resolve VictorOps alerts

13 Search

Today

Airflow BOT 10:00 AM EP\_STAGE - ep\_telemetry\_v2 SLA Miss for task [send\\_email\\_notification\\_flow\\_successful on 2016-02-26T18:00:00](#)

sid 10:02 AM So, we need to catch up here.. any reason we don't want to just mark success for the many hours it is behind?

Airflow BOT 11:00 AM EP\_STAGE - ep\_telemetry\_v2 SLA Miss for task [send\\_email\\_notification\\_flow\\_successful on 2016-02-26T19:00:00](#)

Airflow BOT 5:27 PM ep\_telemetry\_v2 on [etl-00.ep-old.prod.agari.com](#) completed [2016-02-03 00:00:00](#) with High Discrepancies

Airflow BOT 8:48 PM ep\_telemetry\_v2 on [workflow-00.ep.stage.agari.com](#) completed [2016-02-26 03:00:00](#) with 1 DLQs : Sample Exception

February 3rd

Correctness  
SLA miss

# Correctness & Timeliness : Alerting

The screenshot displays two interfaces: Slack and Agari. On the left, the Agari interface shows a timeline of alerts. On the right, the Slack interface shows the same alerts being communicated. A large pink arrow points from the Agari timeline to the Slack channel, illustrating the flow of information.

**Agari Timeline:**

- Feb 26, 2016 17:00:02 PST: Triggered: 0
- Feb 26, 2016 16:00:04 PST: Acked: 0
- Feb 26, 2016 15:41:40 PST: Resolved

**Slack #ep-ops Channel:**

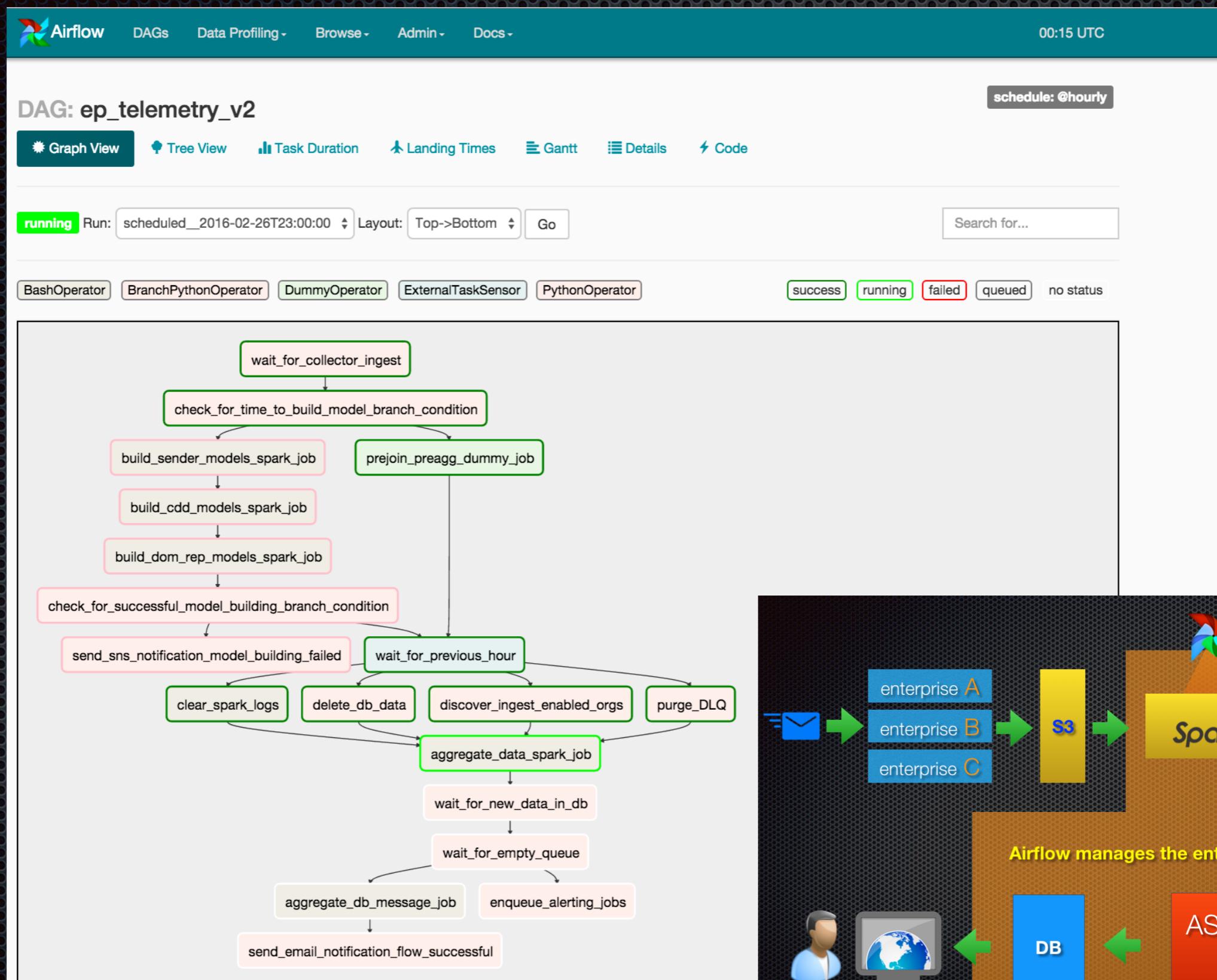
- Airflow BOT 10:00 AM:** EP\_STAGE - ep\_telemetry\_v2 SLA Miss for task `send_email_notification_flow_successful` on `2016-02-26T18:00:00`
- sid 10:02 AM:** So, we need to catch up here.. any reason we don't want to just mark success for the many hours it is behind?
- Airflow BOT 11:00 AM:** EP\_STAGE - ep\_telemetry\_v2 SLA Miss for task `send_email_notification_flow_successful` on `2016-02-26T19:00:00`
- Airflow BOT 5:27 PM:** ep\_telemetry\_v2 on `etl-00.ep-old.prod.agari.com` completed `2016-02-03 00:00:00` with High Discrepancies
- Airflow BOT 8:48 PM:** ep\_telemetry\_v2 on `workflow-00.ep.stage.agari.com` completed `2016-02-26 03:00:00` with 1 DLQs : Sample Exception

Timeliness &  
Correctness SLA  
misses sent to  
PagerDuty/VictorOps

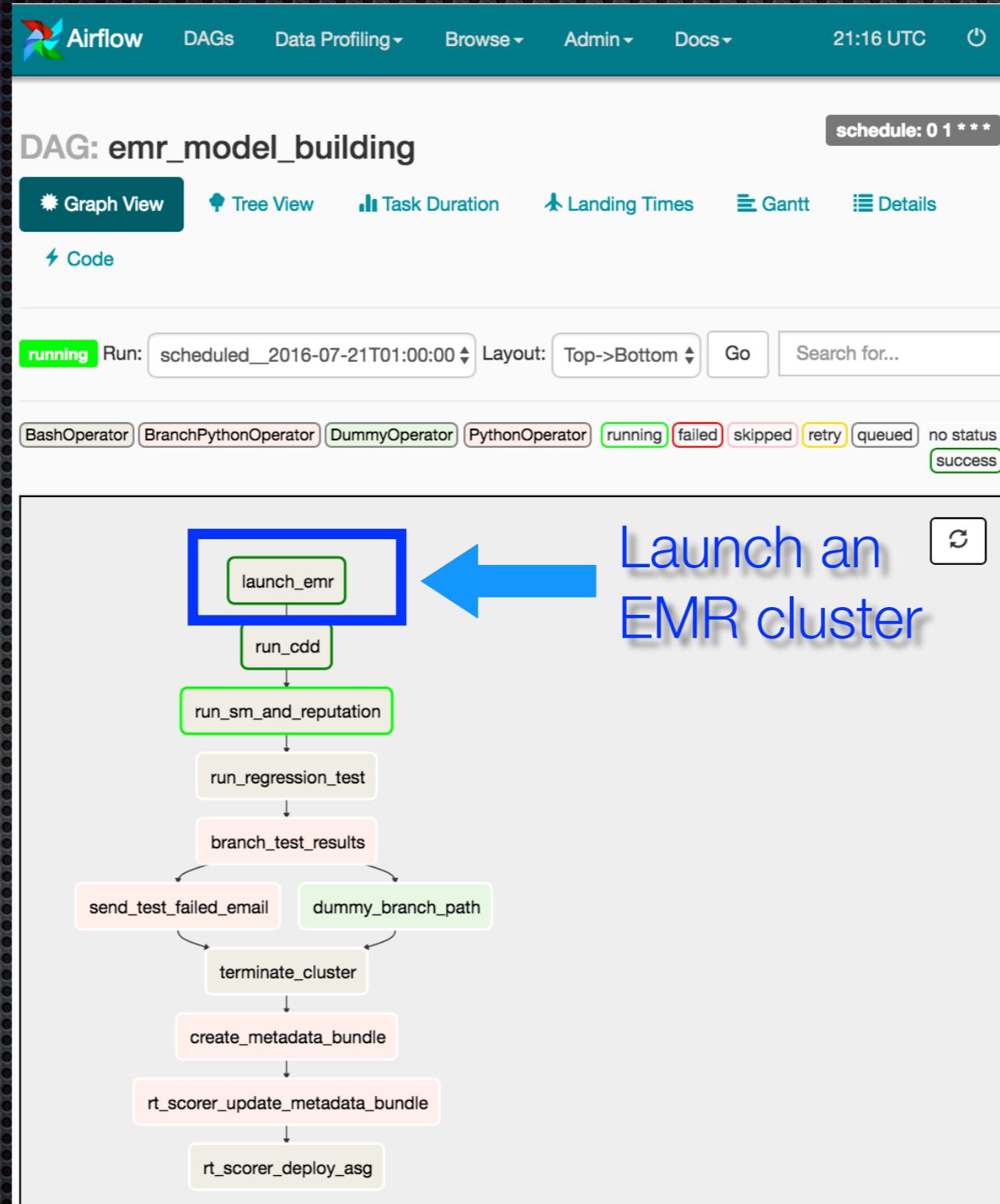
# Use-Case : Model Building v2

For Both Batch & Near Realtime Scoring  
Pipelines

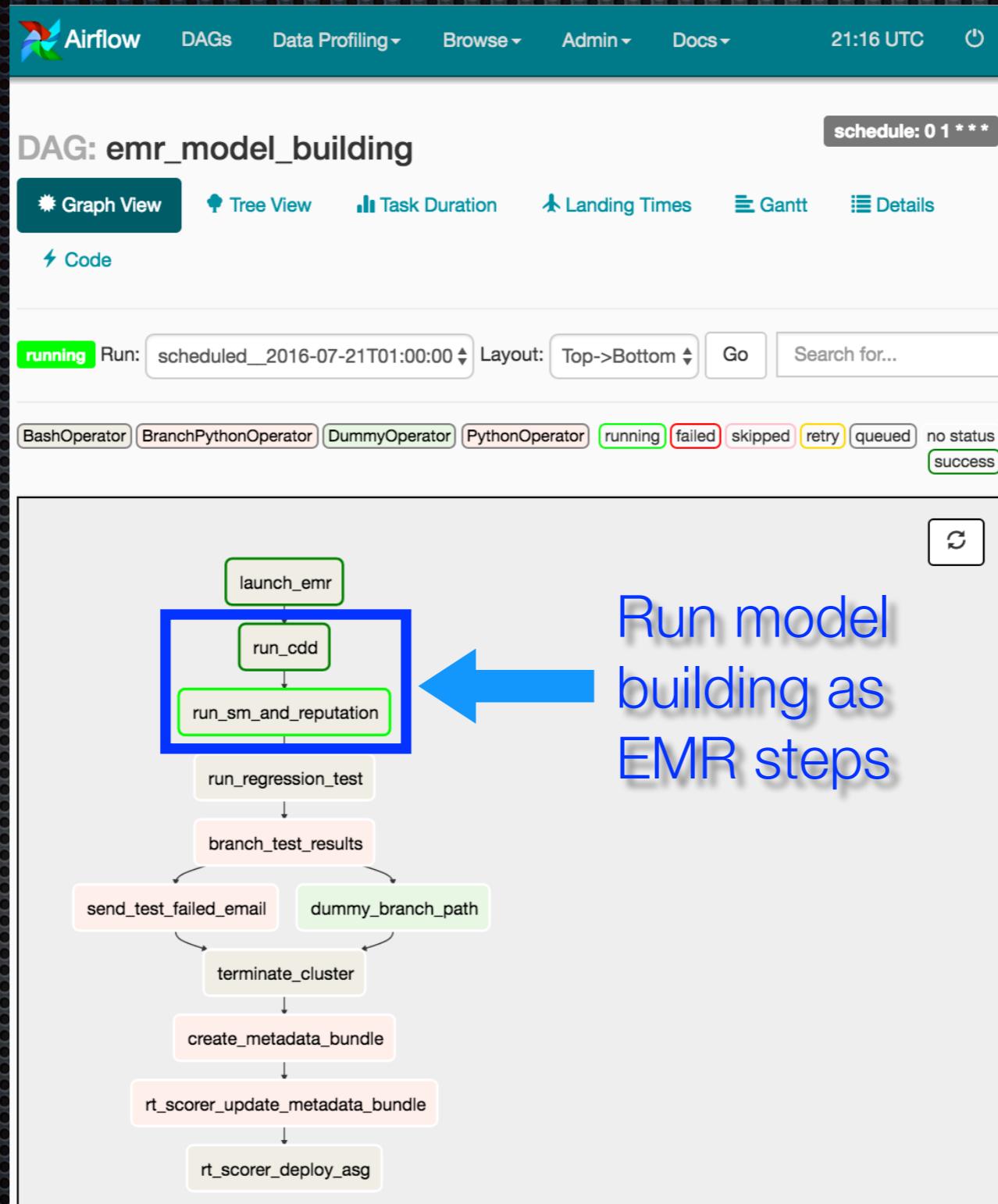
# Airflow DAG



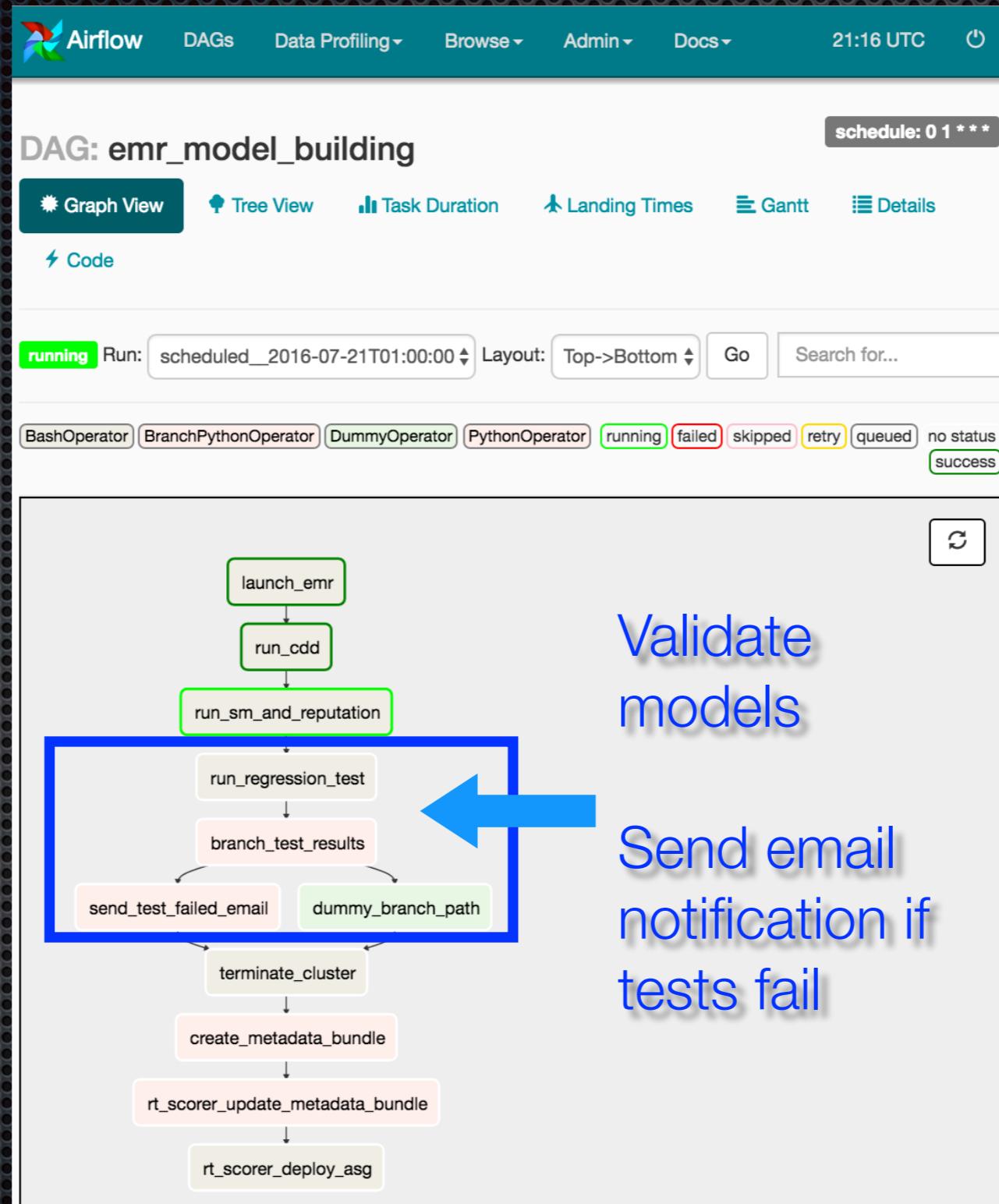
# Model Building DAG



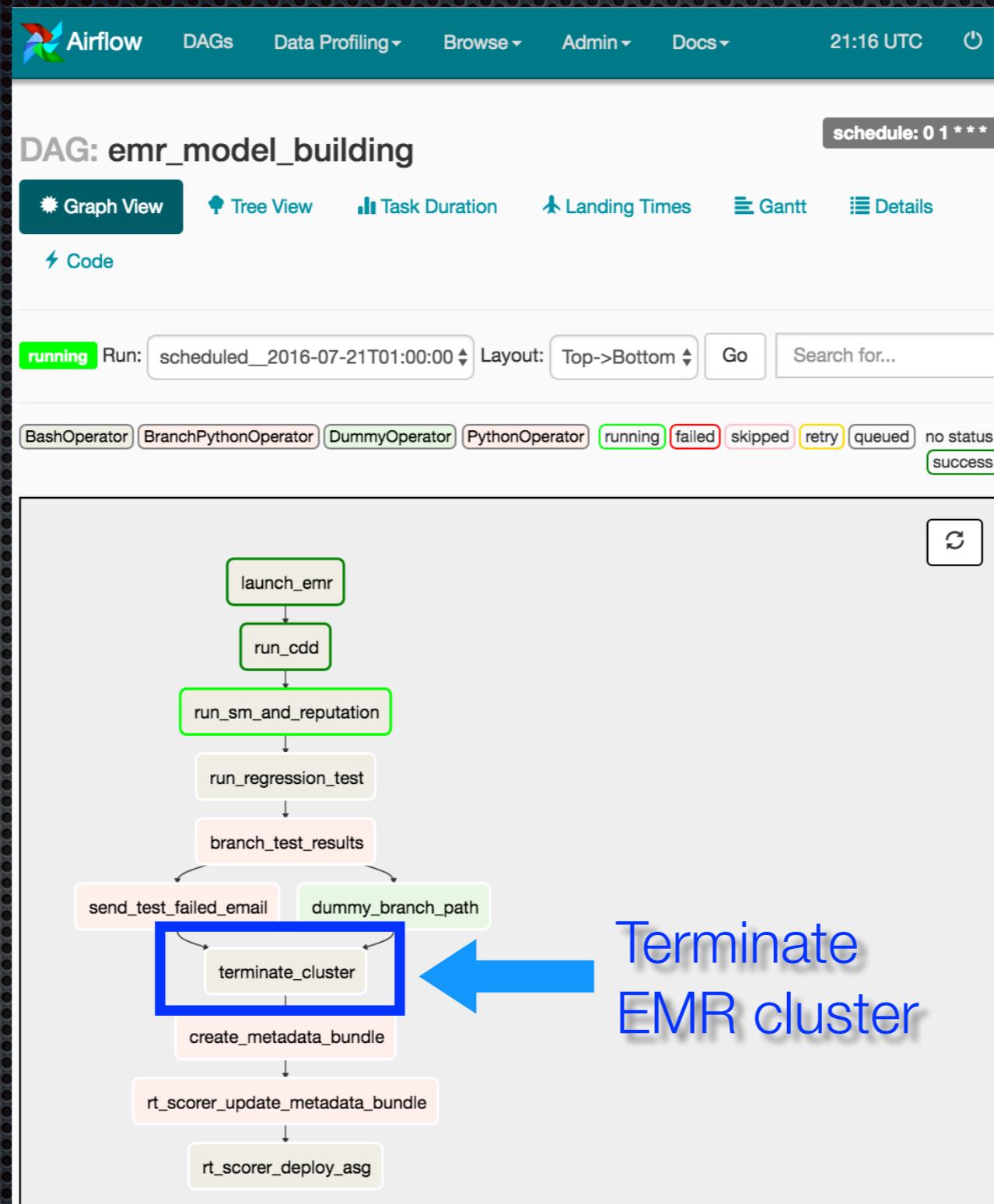
# Model Building DAG



# Model Building DAG



# Model Building DAG



# Apache Airflow Next Steps

## Areas for Improvement



# Apache Airflow Next Steps

## Improvement Areas

- Security
- API (though we do have a CLI)
- Deployment / Versioning
- Execution Scale Out
- On-demand Execution

# Acknowledgments

None of this work would be possible without the contributions of the strong team below

- Vidur Apparao
- Stephen Cattaneo
- Jon Chase
- Andrew Flury
- William Forrester
- Chris Haag
- Mike Jones
- Scot Kennedy
- Thede Loder
- Paul Lorence
- Kevin Mandich
- Gabriel Ortiz
- Jacob Rideout
- Josh Yang
- Julian Mehnle

# Questions? (@r39132)



Why am I in the Water?  
And what the .... is THAT?