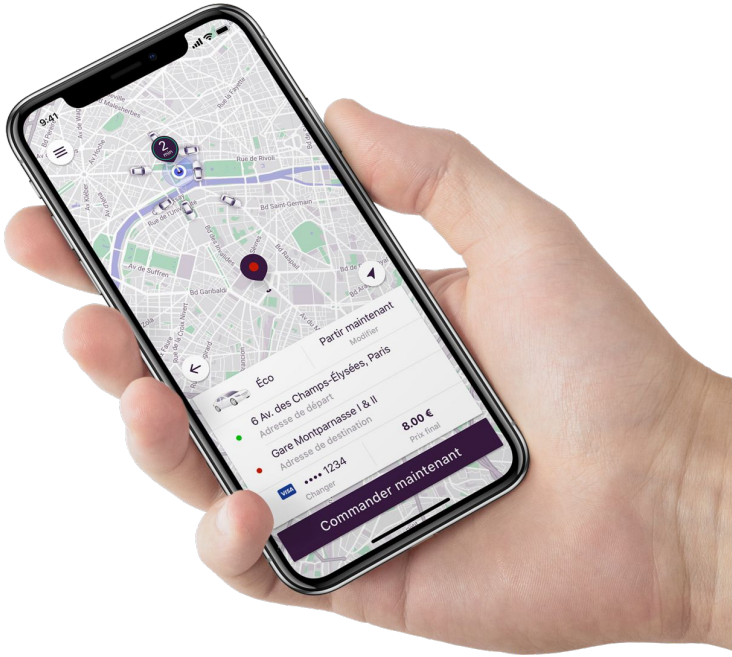
Meetup Airflow #1

How we built a datalake

Outline

1. CHAUFFEUR PRIVÉ
2. DATA @ CHAUFFEUR PRIVÉ
3. DATALAKE x AIRFLOW
4. QUESTIONS



Chauffeur Privé

Lancement
en Île-de-France

Lancement
sur la Côte d'Azur

Lancement
à Lyon

30ème place
Classement FrenchWeb des entreprises qui recrutent dans le numérique

FRENCHWEB.FR

Participation majoritaire
Le groupe Daimler AG prend une participation majoritaire au capital de Chauffeur Privé

DAIMLER

2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018

Gagnant
de la BFM Academie,
1er concours de créateurs d'entreprise

5M€ levés
30 collaborateurs
3 000 chauffeurs
300 000 utilisateurs

2ème start-up à la + forte croissance en France
Classement Deloitte des entreprises technologiques les + performantes

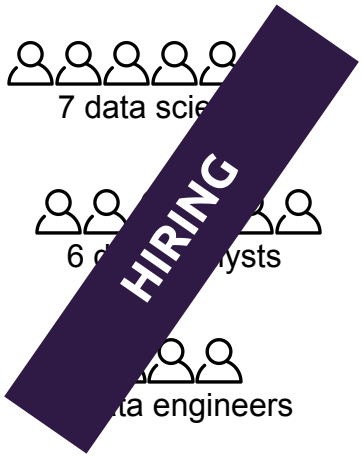
Deloitte. Technology Fast50

Lancement
à Lisbonne

DATA @ CHAUFFEUR PRIVÉ

4
TEAMS

16
PEOPLE



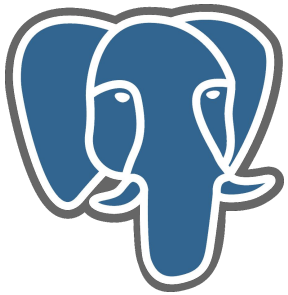
One datawarehouse on Amazon RDS

Actual issue: many timeouts (4h)

Scripts in bash + Makefile + cron

2To
DATA
VOLUME

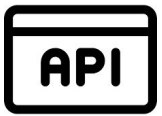
~400
TABLES IN DWH



95% of the static data

No schema

Microservice architecture > +**100** databases



Extract

Mirror each needed table, only needed fields

(sometimes with cleaning)

Example

`production.rides`
`production.users`

Load

SQL transformations on raw data to compute KPIs and business logic

Transform

SQL end users

External services (CRM, …)

AIRFLOW x DATALAKE

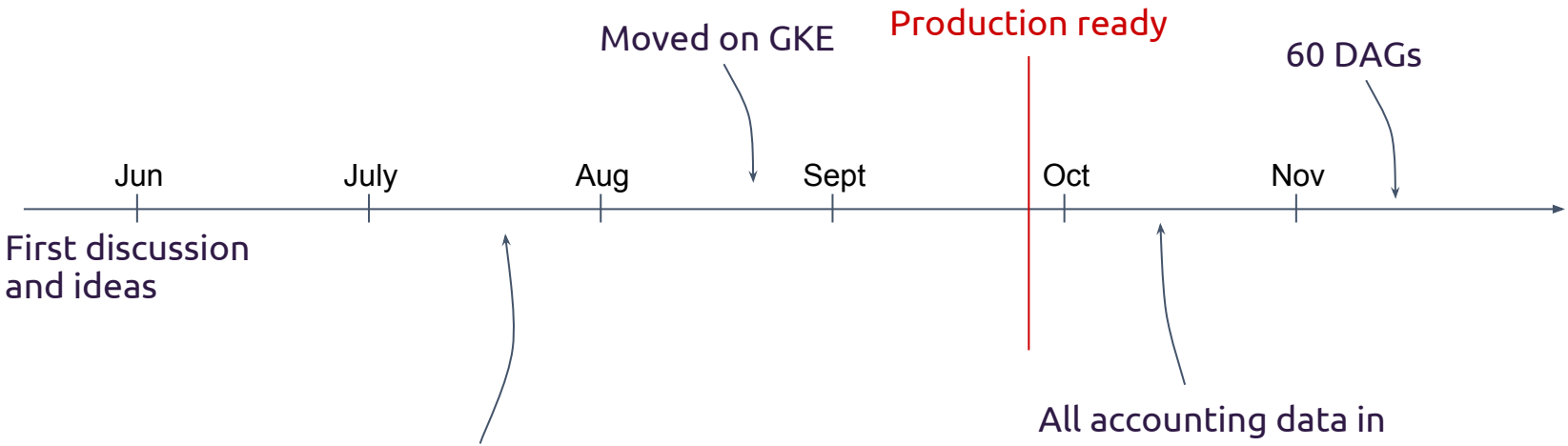
DAG > Operator > TaskInstance

Idempotency

Connections

Pools

SLA



Cloud
Composer

"A fully managed workflow orchestration service built on Apache Airflow on Google Cloud Platform"

+ Pros

Easy to have a fresh Airflow

SSO and HTTPS

On Kubernetes

- Cons

Python 2.7 (3 in Beta atm) and Airflow 1.9

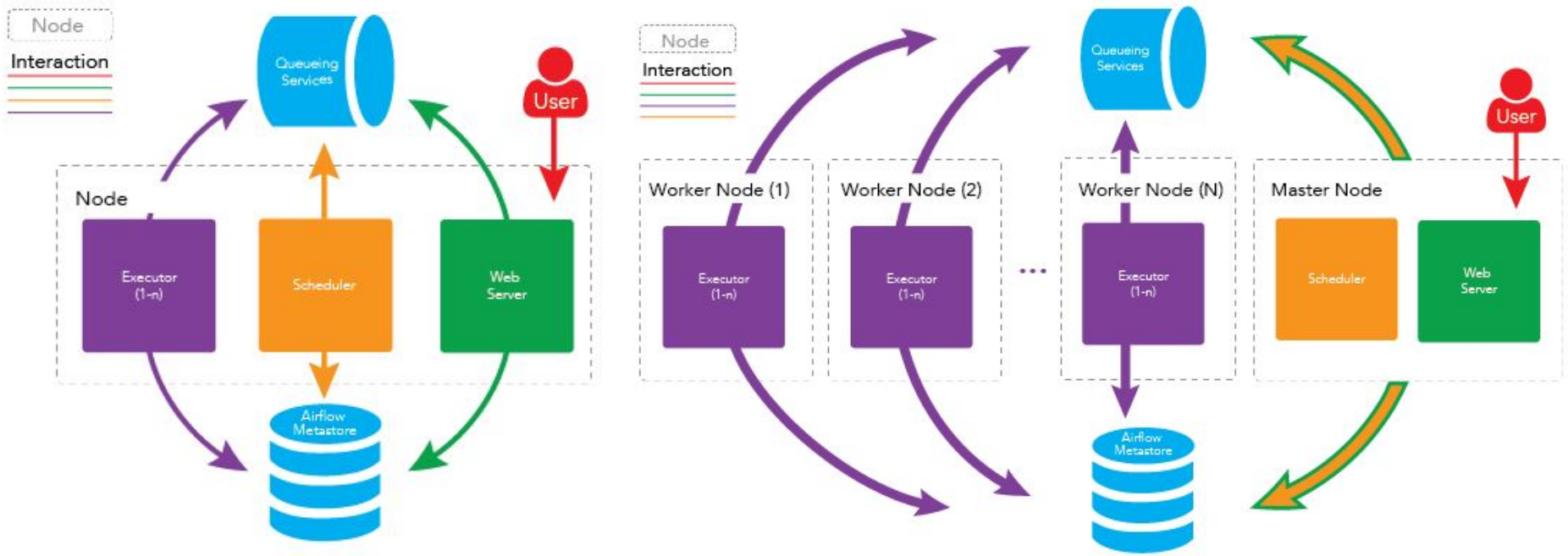
Webserver managed in another project

Process to deploy 👎

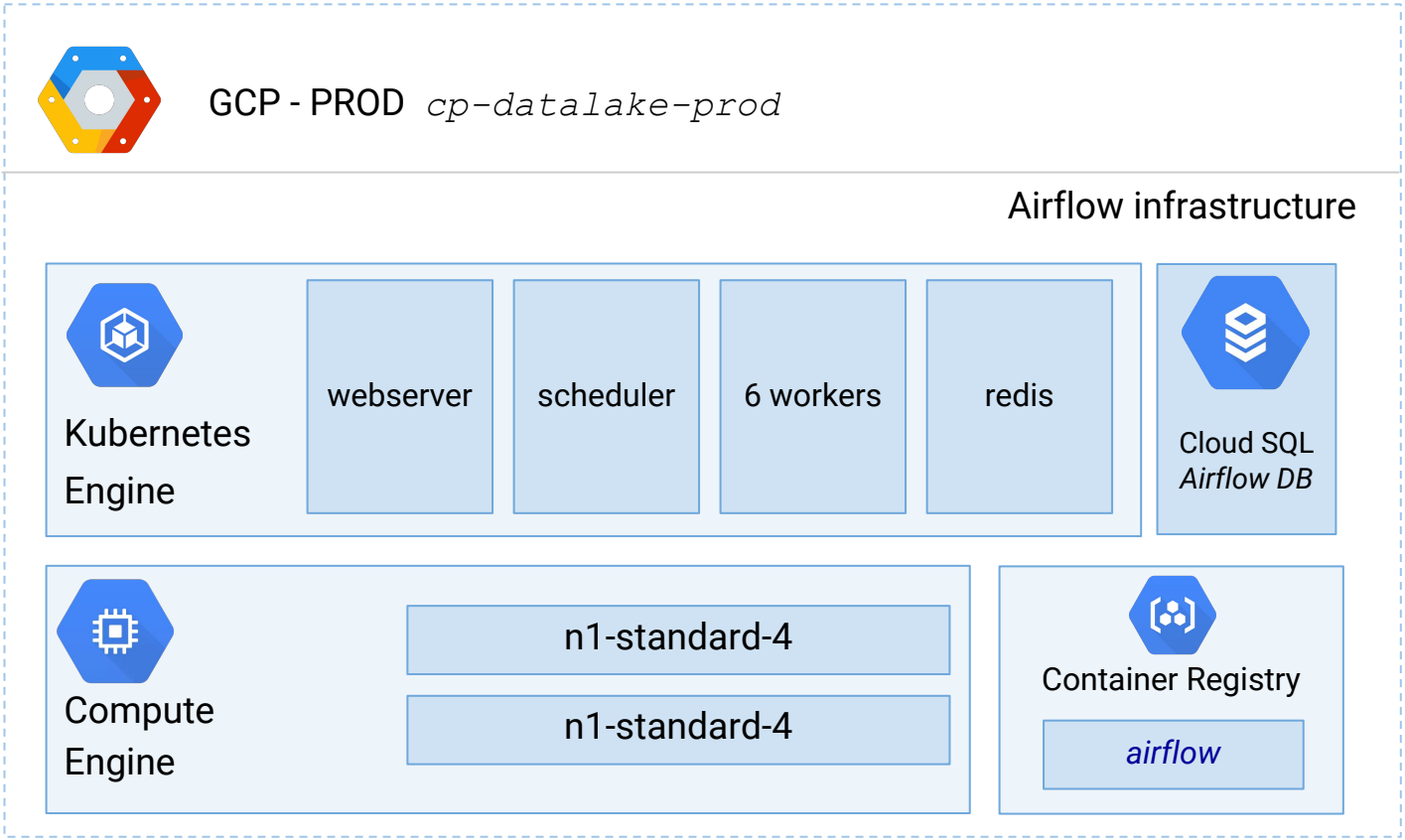
Cost more money

https://cloud.google.com/composer/

Airflow x Datalake - Self managed Airflow



http://site.clairvoyantsoft.com/setting-apache-airflow-cluster/



1. Something fully automatic

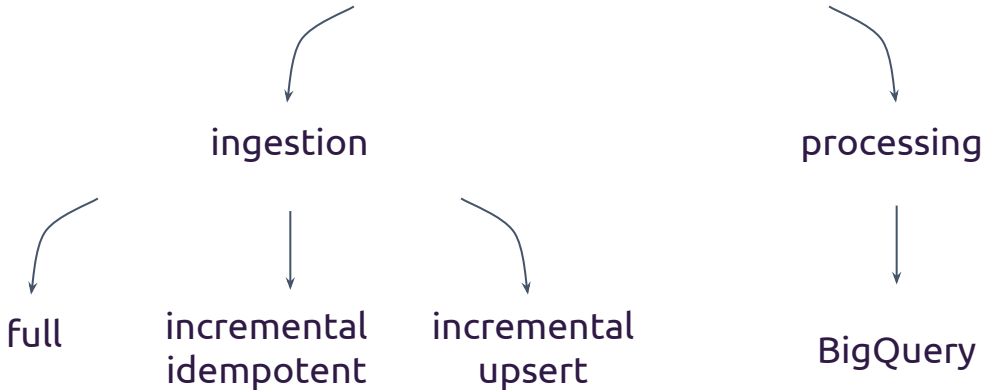
2. Easy for data people to implement new pipelines

3. Faster than the DWH

> So we built a framework above Airflow <

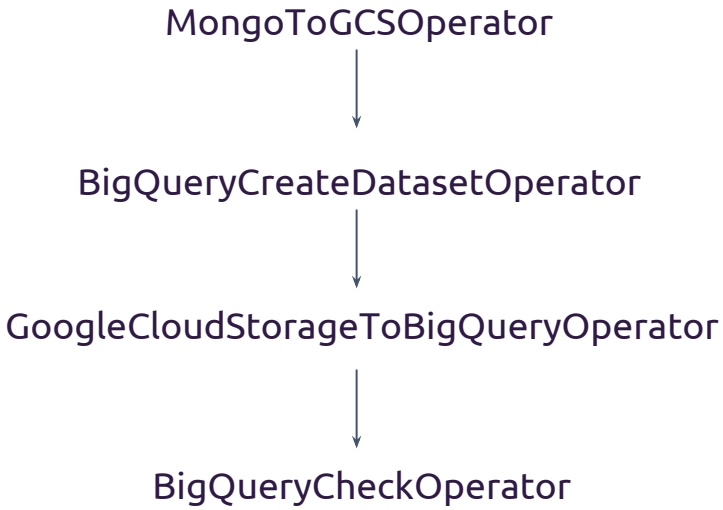
2 types of DAGs in our Airflow framework

> Python configuration



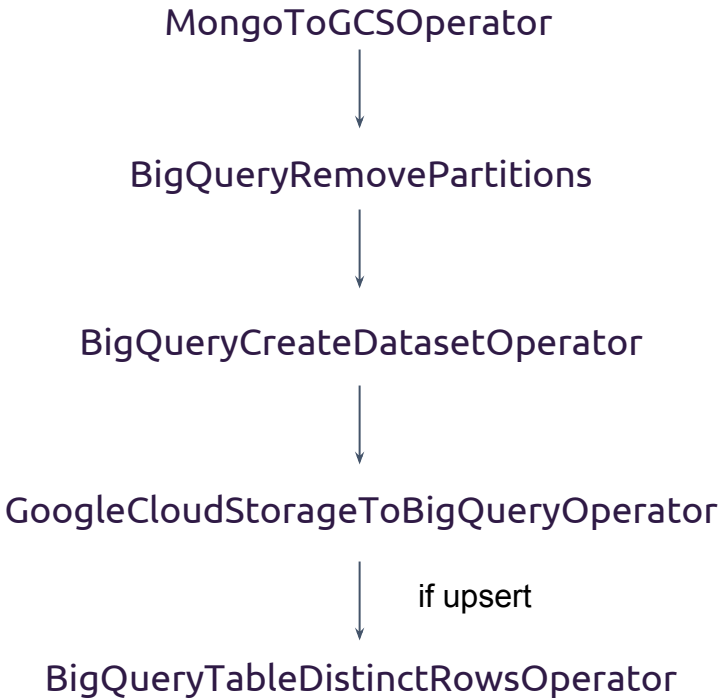
ingestion / production / users.py

```
config = {
  'type': 'full',
  'source': 'mongo',  # or postgres | api
  'connection': 'production',
  'table': 'users',
  'description': 'Riders table',

  'default_args': {
    'owner': 'core',
    'schedule_interval': '0 0 * * *',
  },

  'rules': [
    {'field': 'age', 'to': int},
  ],
}
```



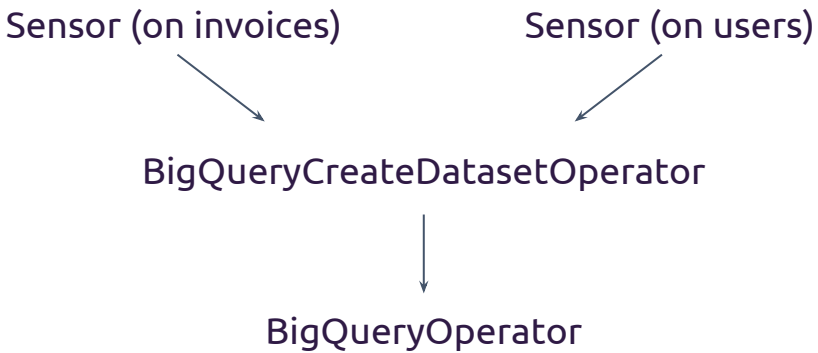
ingestion / accounting / invoices.py

```
config = {
  'type': 'incremental',
  'source': 'mongo',
  'connection': 'accounting',
  'table': 'invoices',
  'description': 'Accounting table',

  'default_args': {
    'start_date': '2018-01-01',
    'catchup': True,
  },
}
```



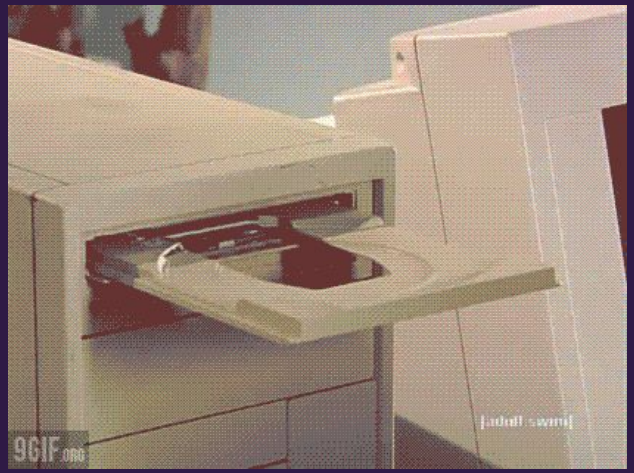
processing / finance / incomes.py

```
from accounting import invoices
from production import users


config = {
  'type': 'processing',
  'dataset': 'finance',
  'table': 'incomes',
  'query': 'finance__incomes.sql',
  'dependencies': [
    invoices,
    users,
  ],
  'sla': timedelta(hours=1, minutes=45),
}
```



- Specific Airflow configuration
  - 12 CPUs / 32 GB RAM
  - SSO + HTTPS
  - 6 (workers) * 16 (concurrency) = 96 tasks simultaneously
  - 2 pools
- 8 custom operators
- 1 custom view
- 93% of codecoverage
- ~300 $ / month

DEMO



Next steps

- More type of processing (Python, Spark ?, etc.)
- Finish the DWH migration
- Open source some of BigQuery operators, maybe parts of the GKE configuration (?)

Questions?

Thank you!