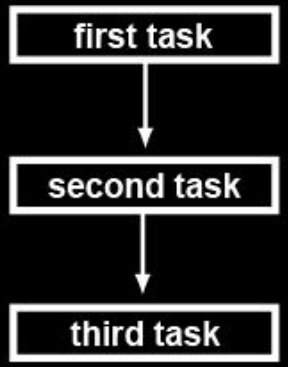
HeirFlow

Highly Available Workflow Orchestration with Apache Airflow

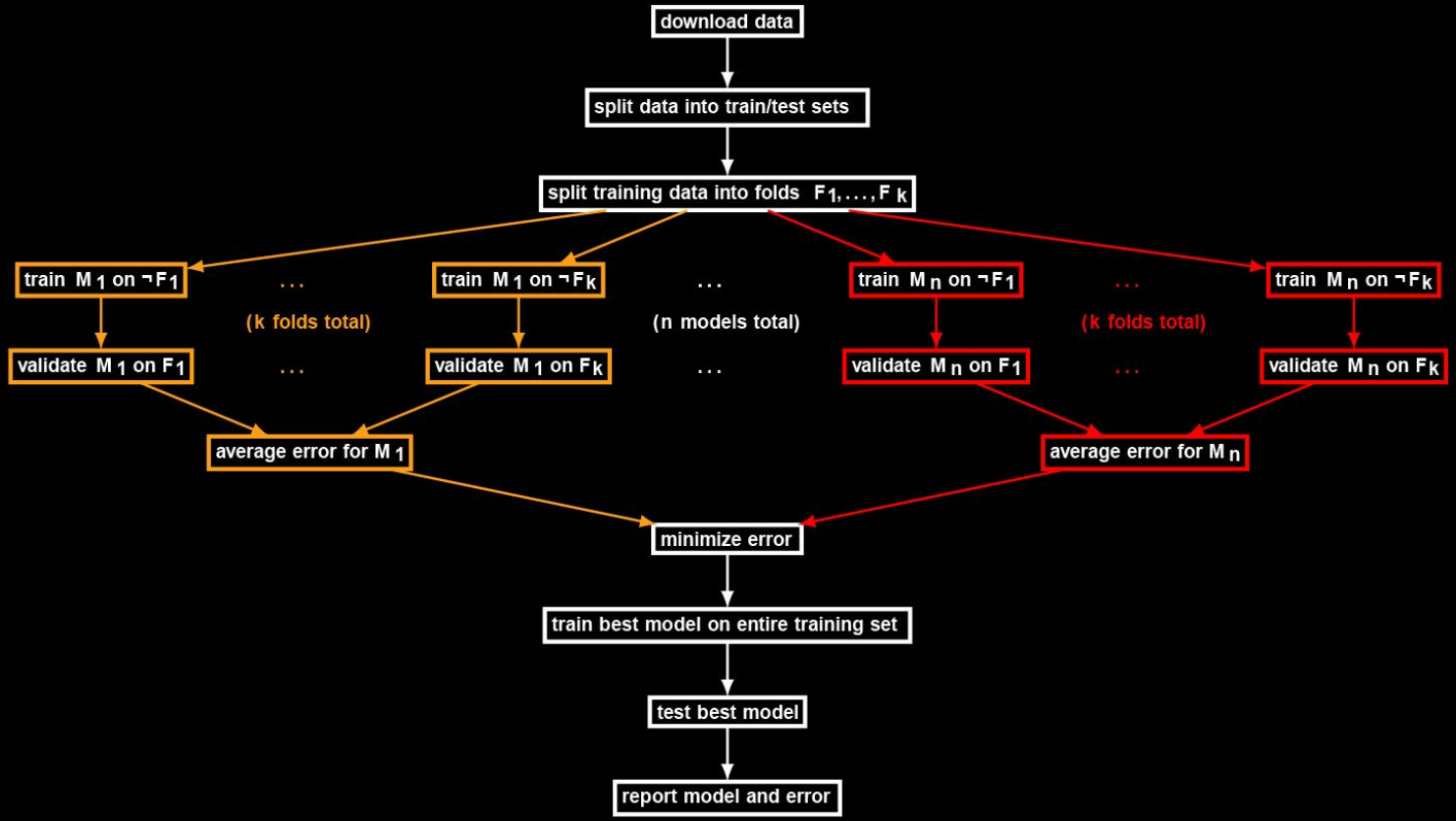
David Wiygul
Insight Data Engineering Fellowship, Los Angeles

Workflow

- Collection of interdependent tasks
- Represented as a directed acyclic graph (DAG)
- Vertices correspond to tasks
- Edges indicate dependence
- Often come with starting time and frequency
- Lifeblood of data engineering

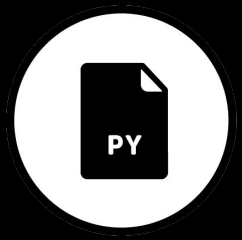


Example DAG (machine learning)



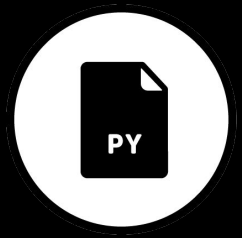
Airflow components

- DAGs (Python code)



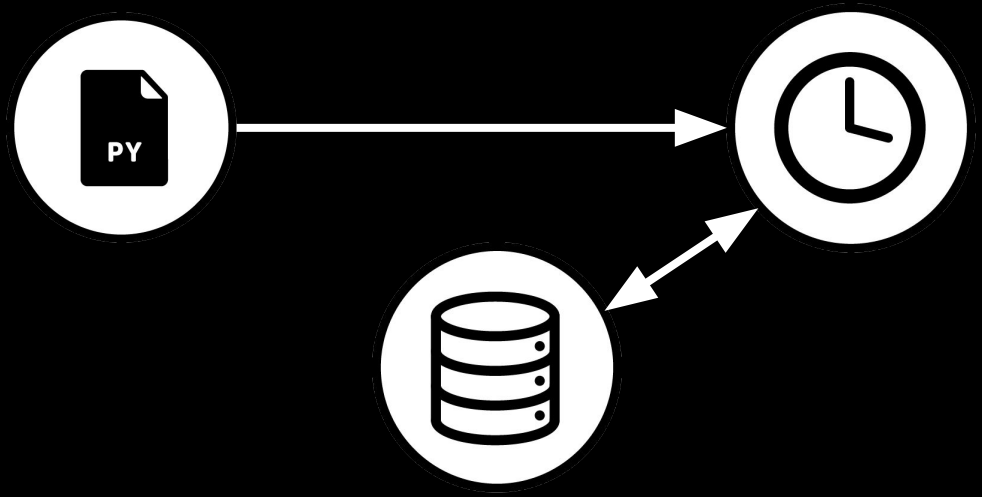
Airflow components

- DAGs (Python code)
- Metadata database



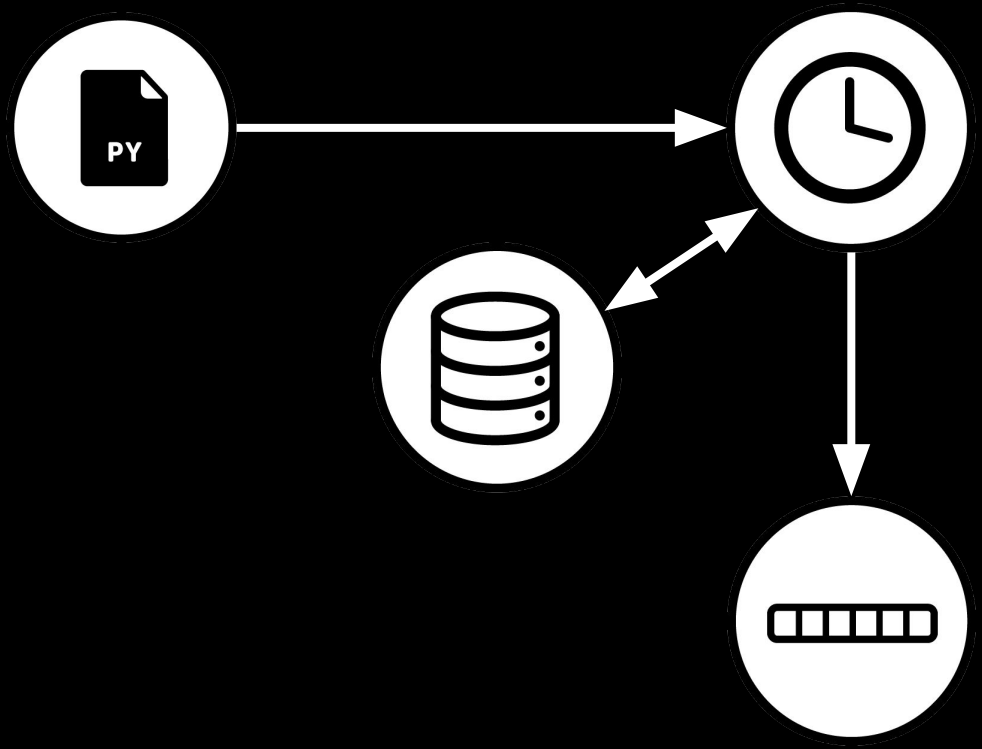
Airflow components

- DAGs (Python code)
- Metadata database
- Scheduler



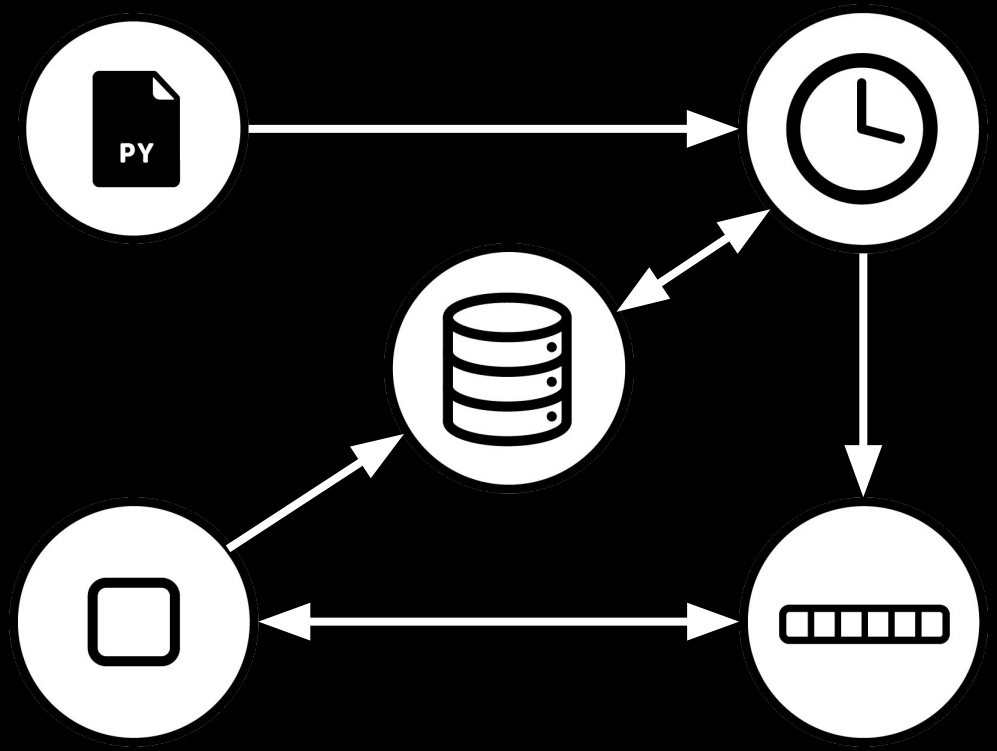
Airflow components

- DAGs (Python code)
- Metadata database
- Scheduler
- Task queue



Airflow components

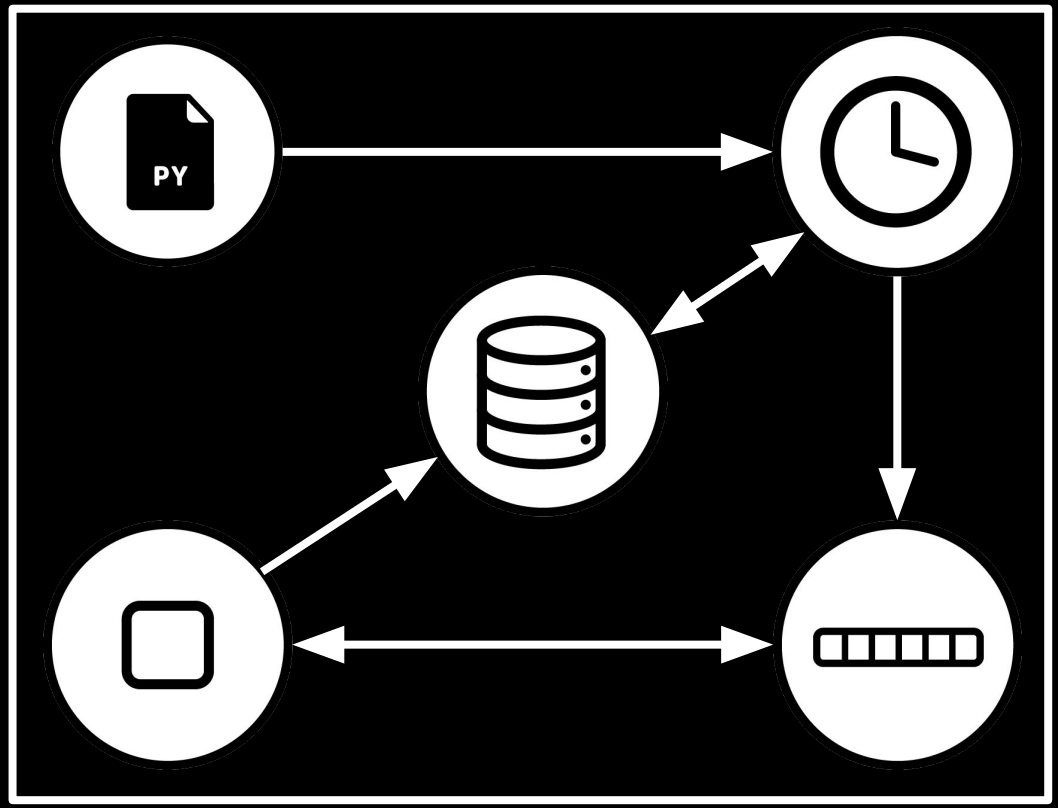
- DAGs (Python code)
- Metadata database
- Scheduler
- Task queue
- Worker



Airflow components

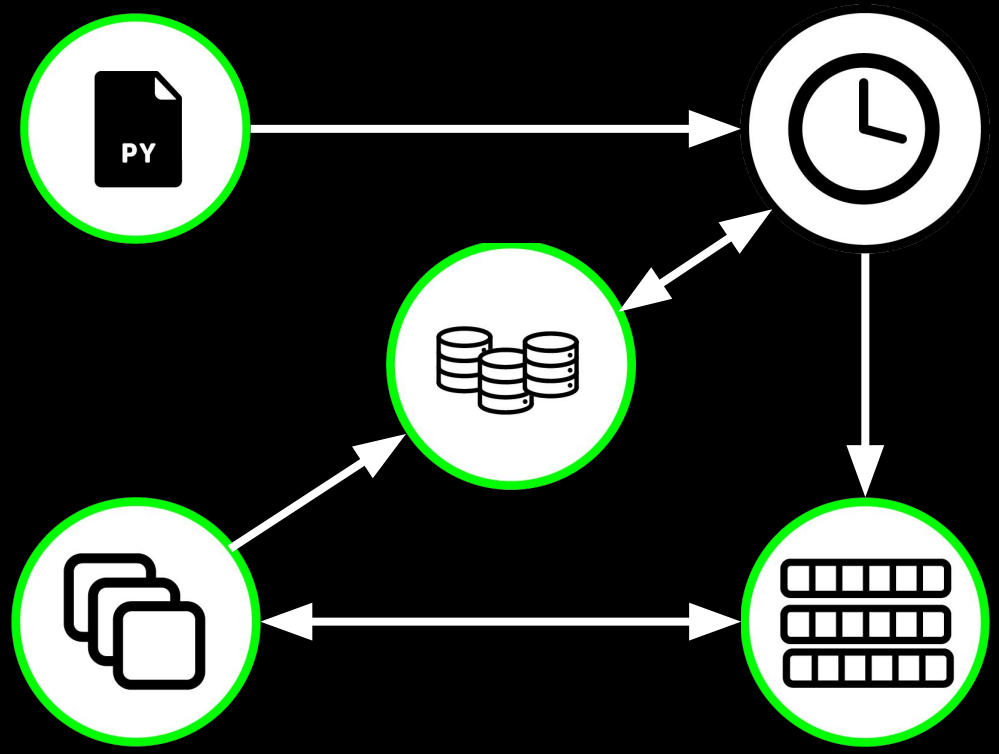
- DAGs (Python code)
- Metadata database
- Scheduler
- Task queue
- Worker

Default config: all on one machine



Horizontal scaling for availability and performance

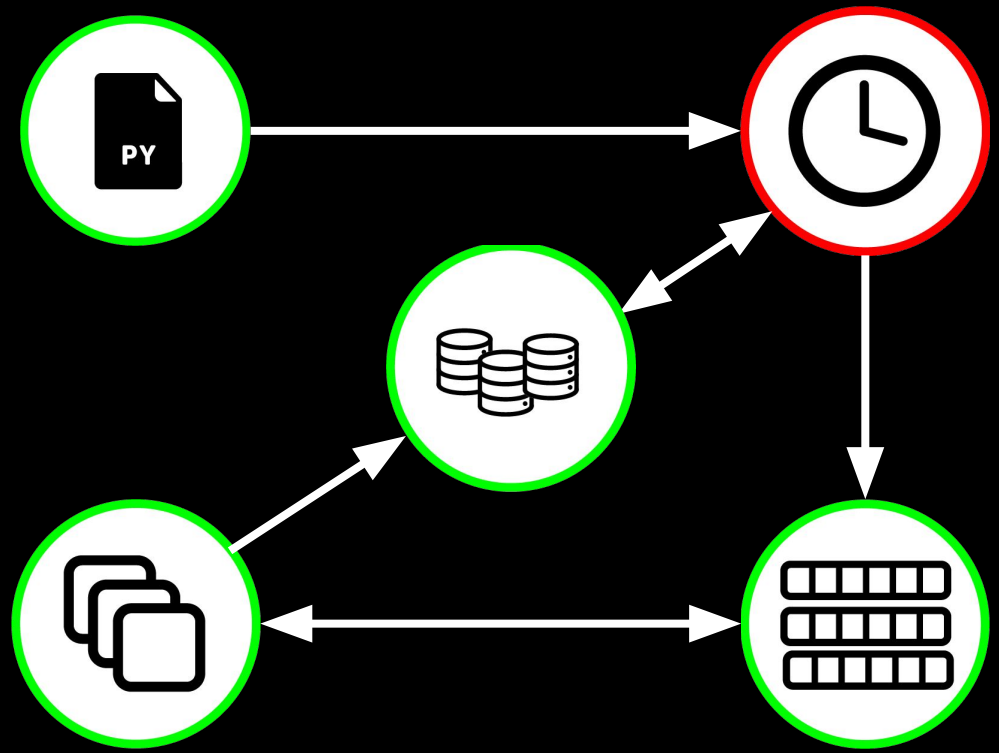
- Downtime sucks.
- Interrupted workflows really suck.
- Parallelism rocks.
- Most components are straightforward to scale out.



Horizontal scaling for availability and performance

- Downtime sucks.
- Interrupted workflows really suck.
- Parallelism rocks.
- Most components are straightforward to scale out.

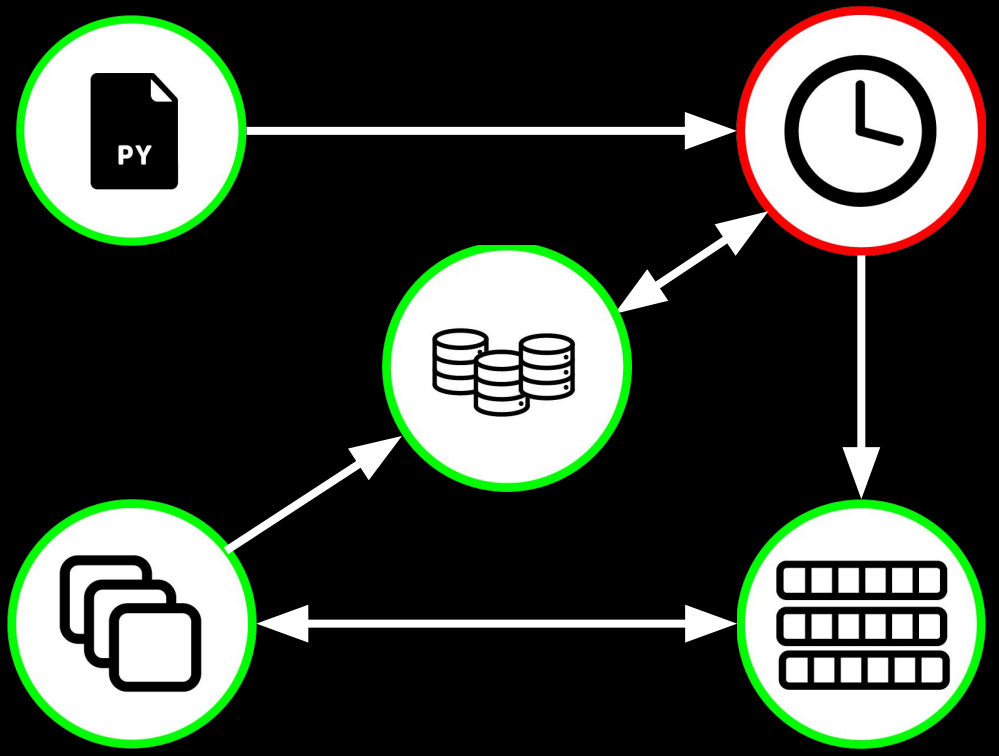
But not the scheduler!



Scheduler redundancy

Schedulers running at the same time might schedule the same task repeatedly.

- ✖ Risky
- ✖ Wasteful



One solution

Use a cluster orchestration tool like Apache Zookeeper (or Consul) to ensure one scheduler is in action at one time

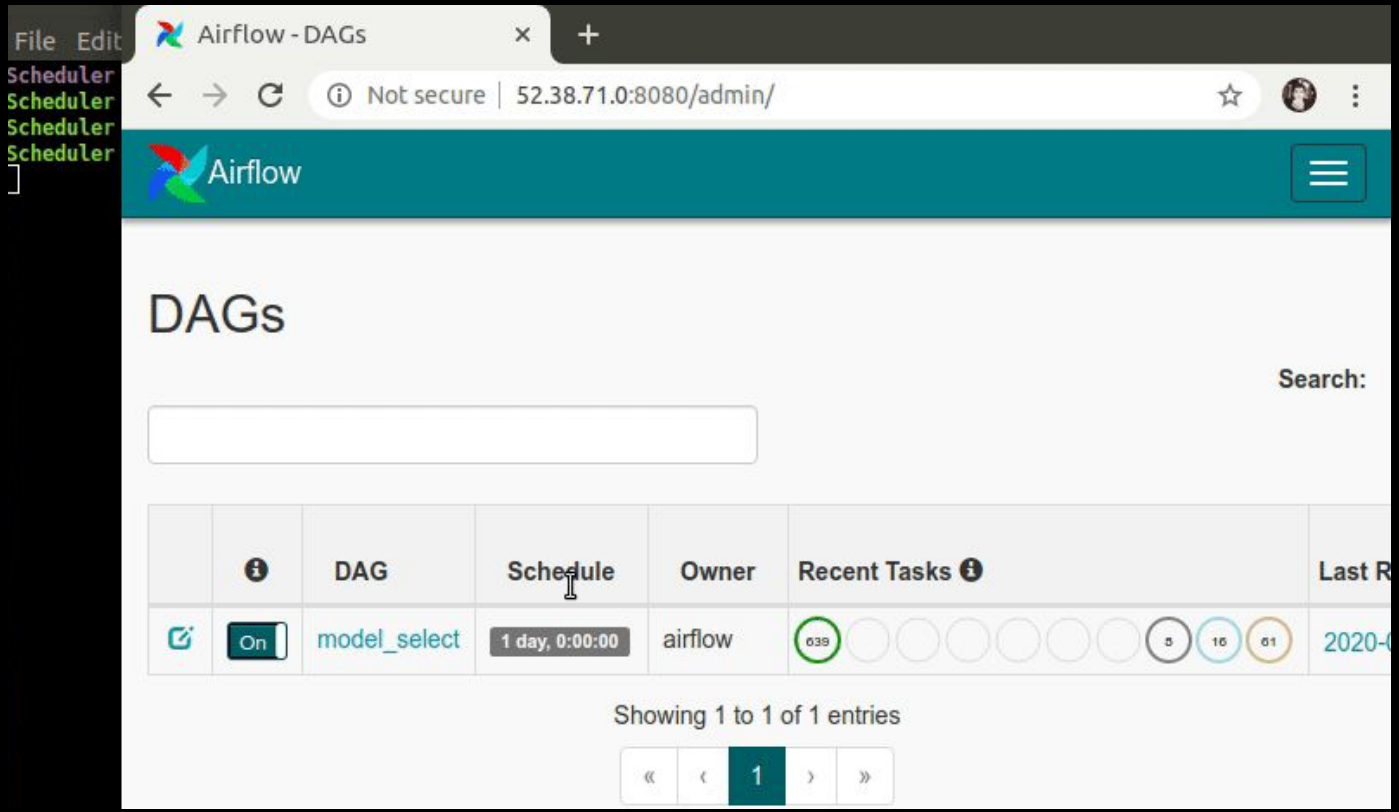
- ★ Convenient when already in place
- ✖ Otherwise requires additional resources and maintenance

Engineering challenge: scheduler redundancy

To ensure exactly one scheduler is running at a given time:

- Add a table (SSOT) to the already existing metadata DB.
- Wrap each scheduler in a Python script.
- The script checks in periodically with the DB.
- The oldest healthy scheduler takes charge.
- There is always a clear heir!

HeirFlow in action

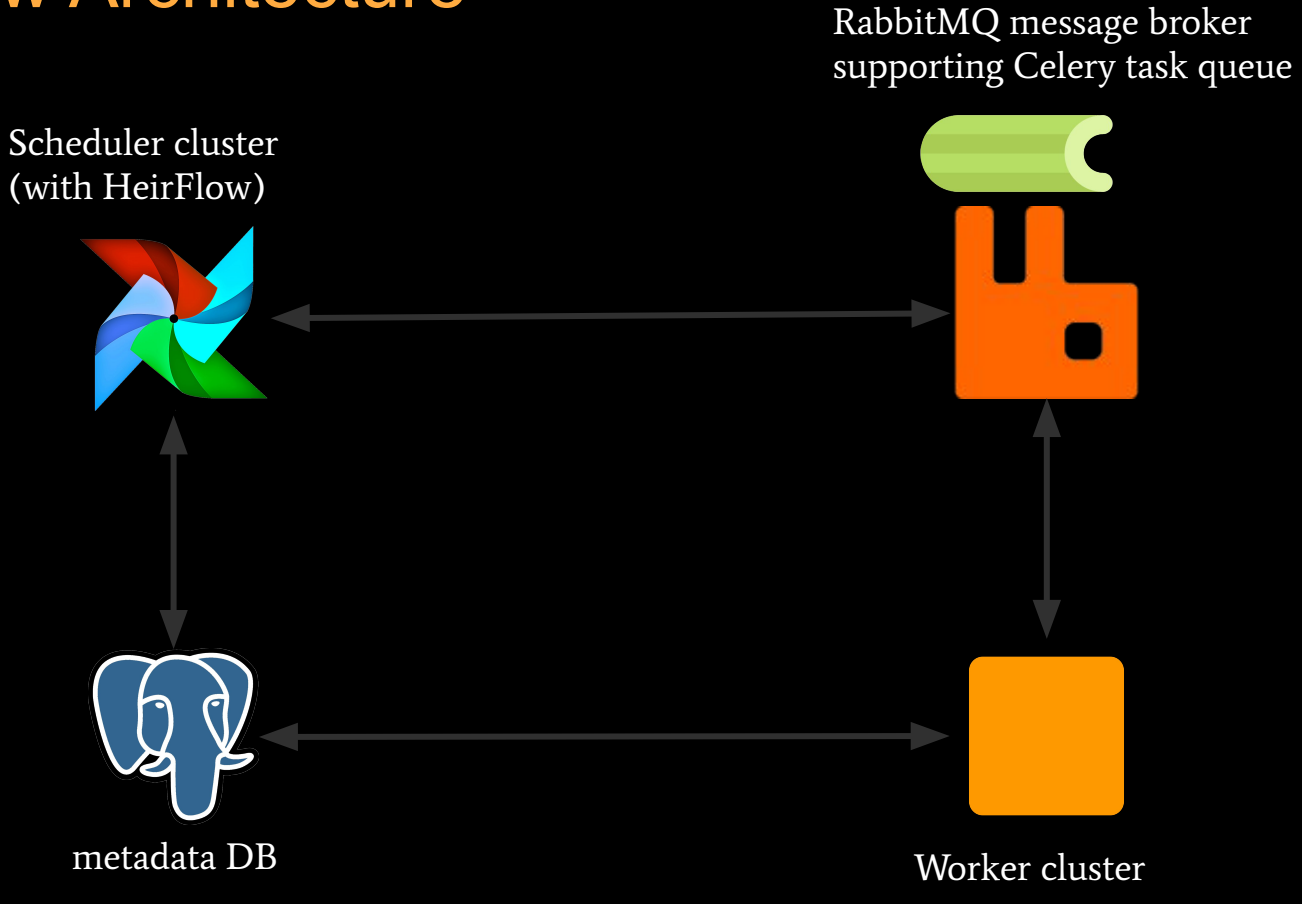


GAME OF THRONES™
It's important to have a clear heir.

Pros and cons

- ★ Availability
- ★ Lean setup
- ★ Flexibility
- ✖ Additional load on schedulers and (especially DB)

Airflow Architecture



About me: David Wiygul

- Before Insight I was a mathematician working in differential geometry and general relativity.
- Theme: complex systems obeying simple rules.
- As a data engineer I get to work with systems that offer (almost) instant feedback!
- I'm also proud dog-dad (see mutually recursive figure)...
- and an expecting human dad (yikes!)