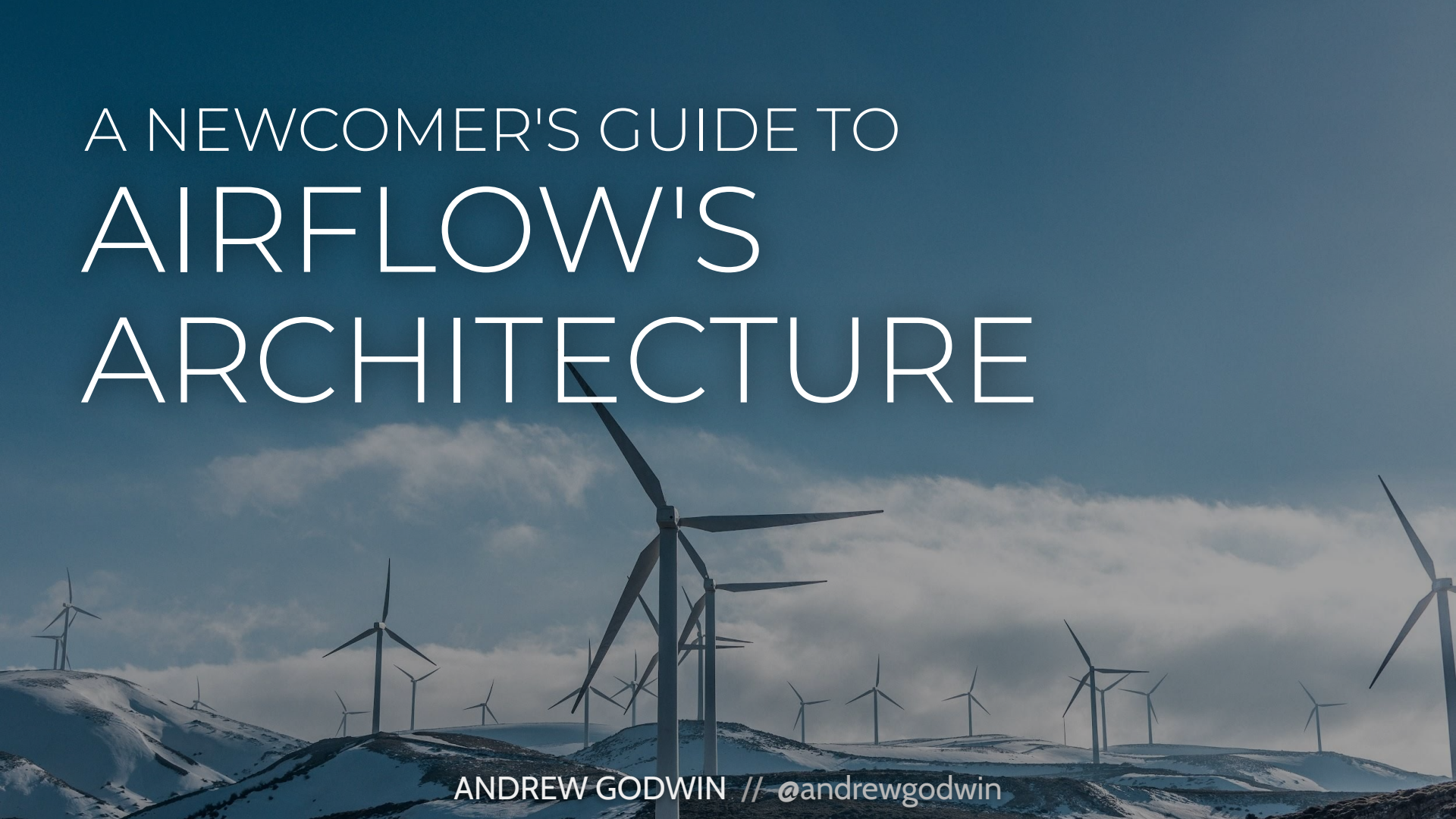
A NEWCOMER'S GUIDE TO AIRFLOW'S ARCHITECTURE



ANDREW GODWIN // @andrewgodwin

Hi, I'm

Andrew Godwin

- Principal Engineer at ASTRONOMER
- Also a Django core developer, ASGI author
- Using Airflow since 2021

High-Level Concepts

What exactly is going on?

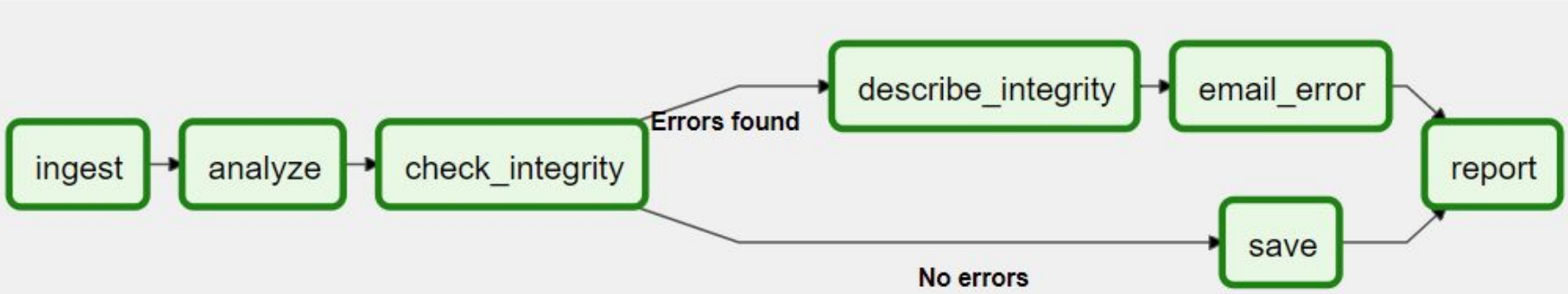
The Good and the Bad

Or, How I Learned To Stop Worrying And Love The Scheduler

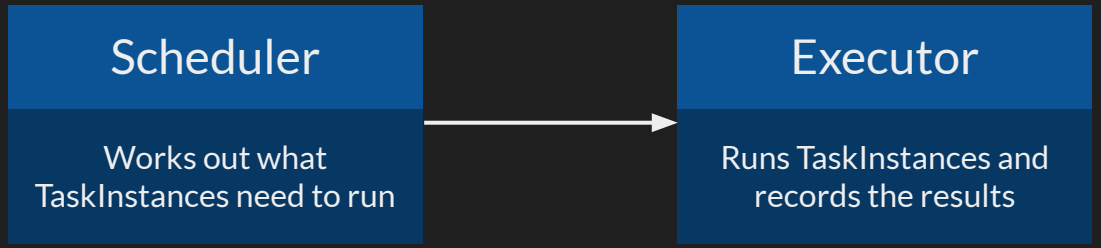
Problems, Fixes & The Future

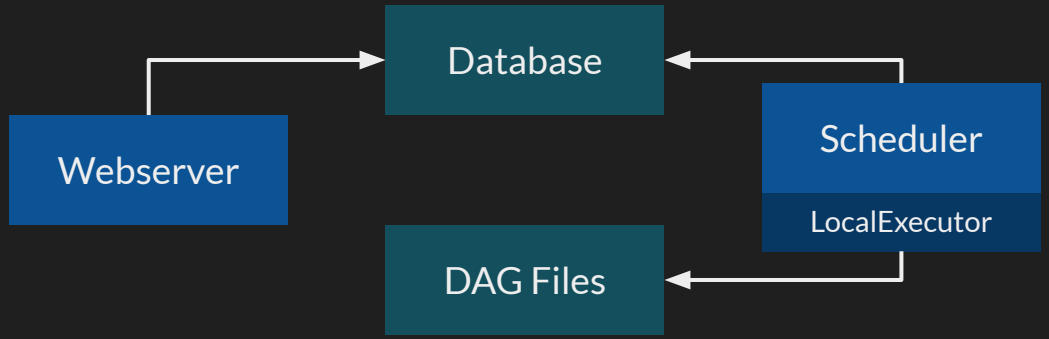
Where we go from here

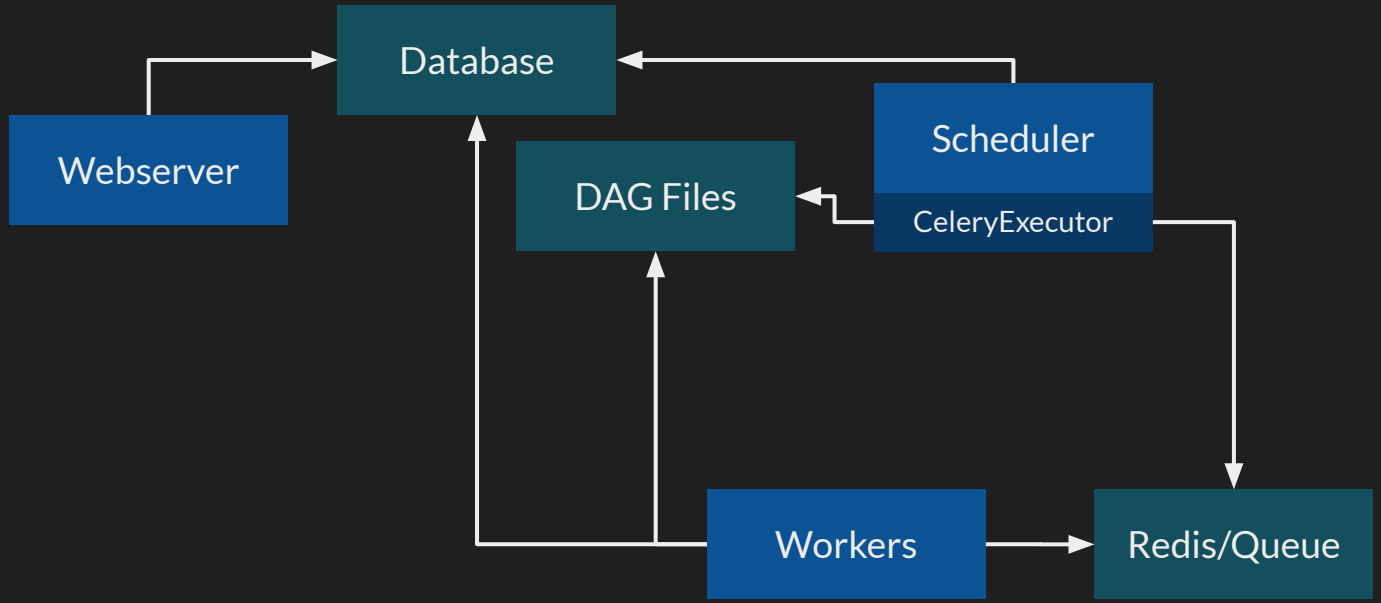

Airflow grew organically
It started off as an internal ETL tool


ingest
analyze
check_integrity
Errors found
describe_integrity
email_error
report
save
No errors


DAG ➡ DagRun
One per scheduled run, as the run starts
Operator ➡ Task
When you call an operator in a DAG
Task ➡ TaskInstance
When a Task needs to run as part of a DagRun


Scheduler
Works out what TaskInstances need to run
Executor
Runs TaskInstances and records the results

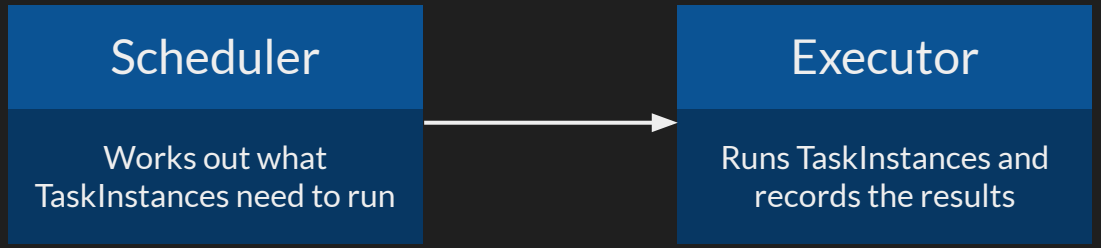

Database
Webserver
Scheduler
LocalExecutor
DAG Files

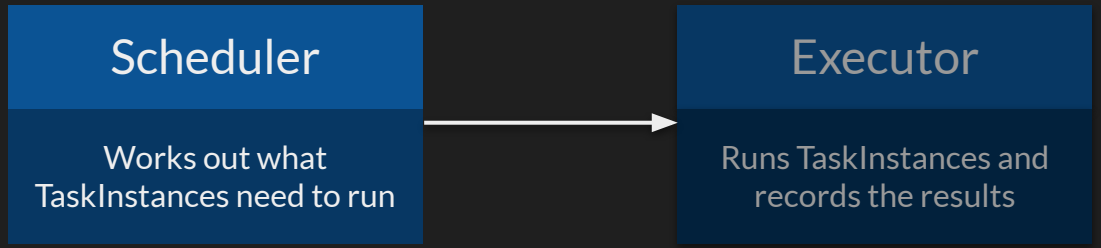

Database
Webserver
Scheduler
DAG Files
CeleryExecutor
Workers
Redis/Queue


The Executor runs inside the Scheduler
Its logic, at least, and the tasks too for local ones


Everything talks to the database
It's the single central point of coordination


Scheduler, Workers, Webserver
All can be run in a high-availability pattern


Scheduler
Works out what TaskInstances need to run
Executor
Runs TaskInstances and records the results


Scheduler
Works out what TaskInstances need to run
Executor
Runs TaskInstances and records the results

Timing

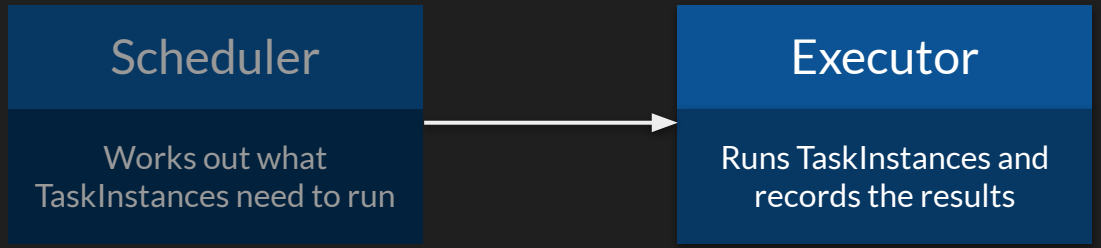
Dependencies

Retries

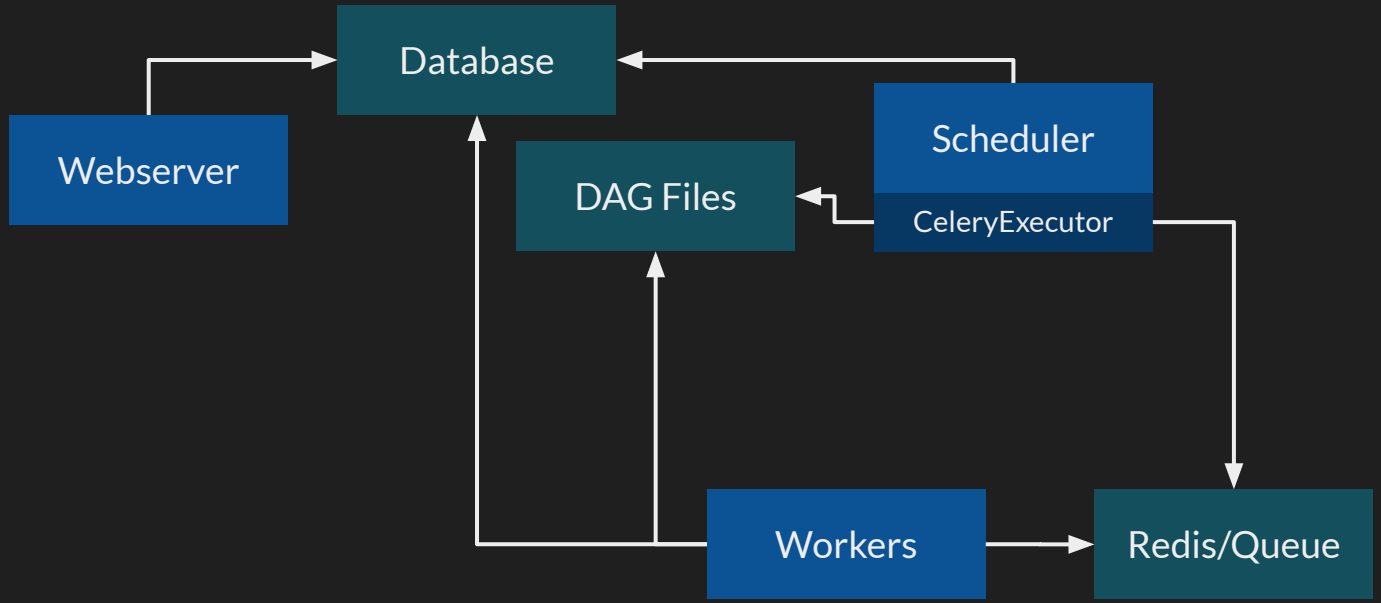
Concurrency

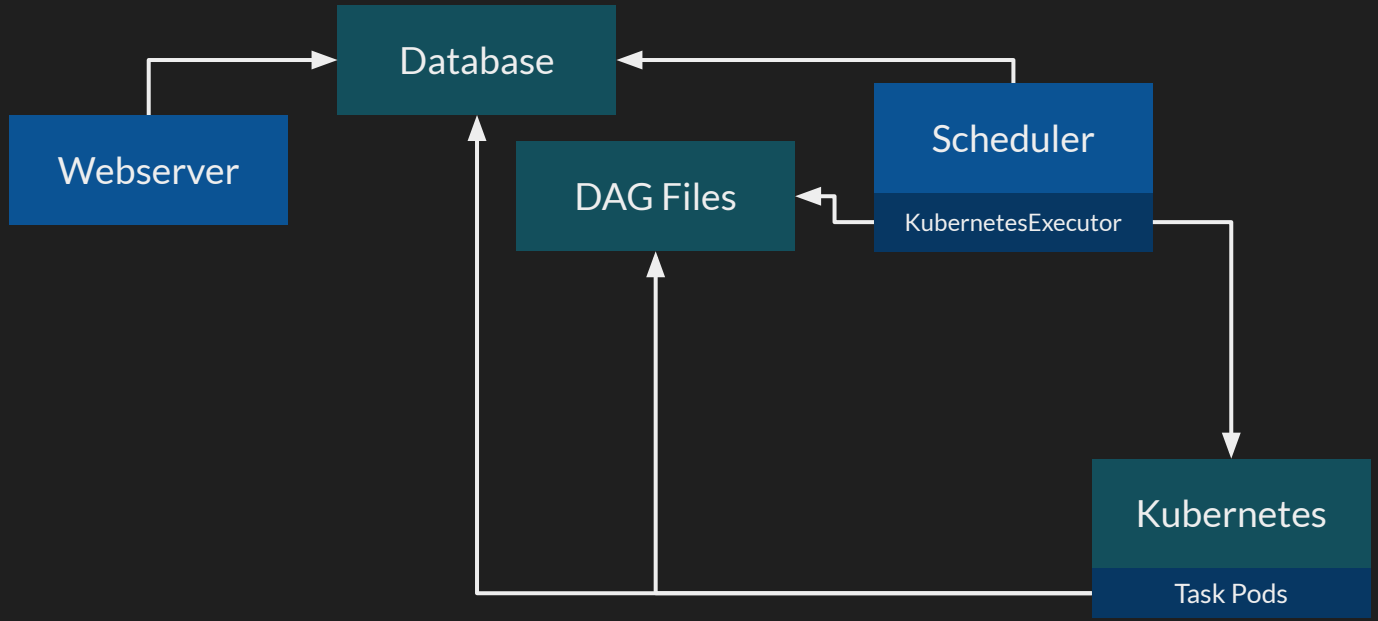
Callbacks

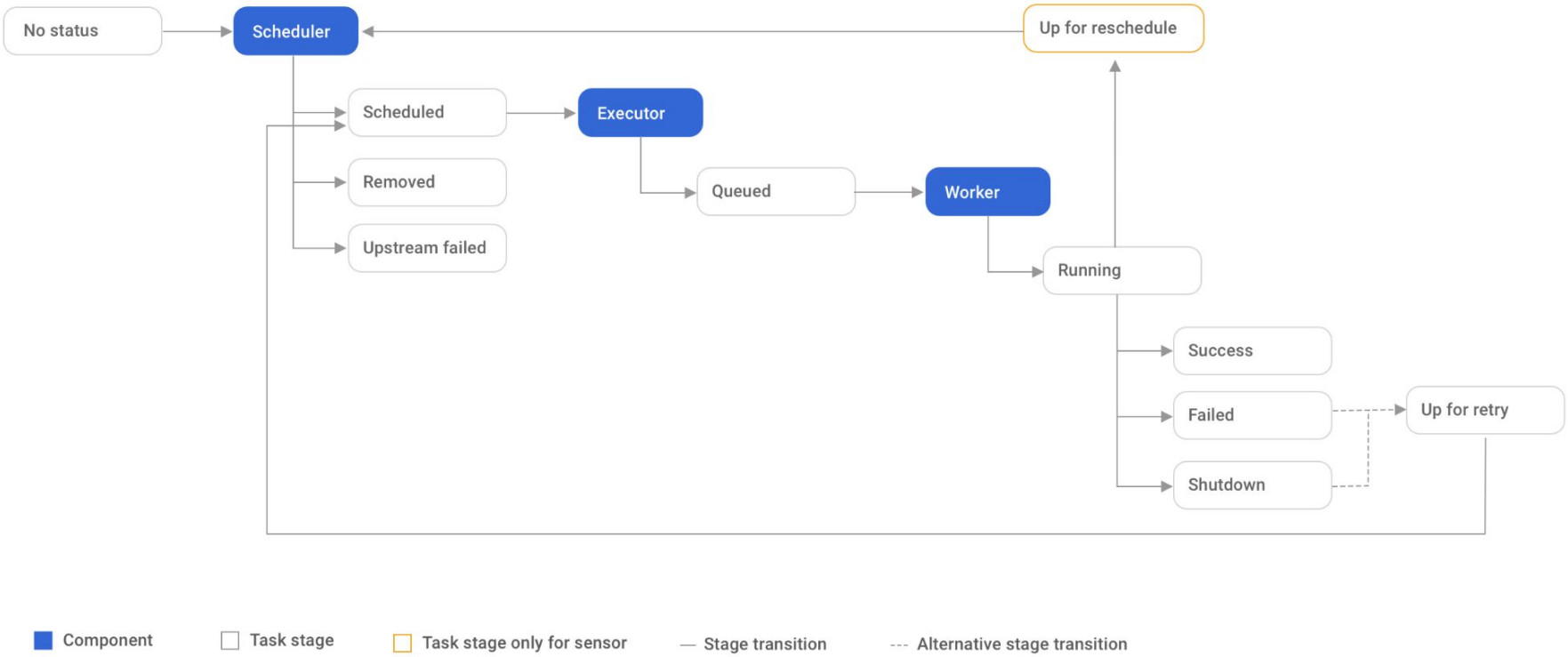
...

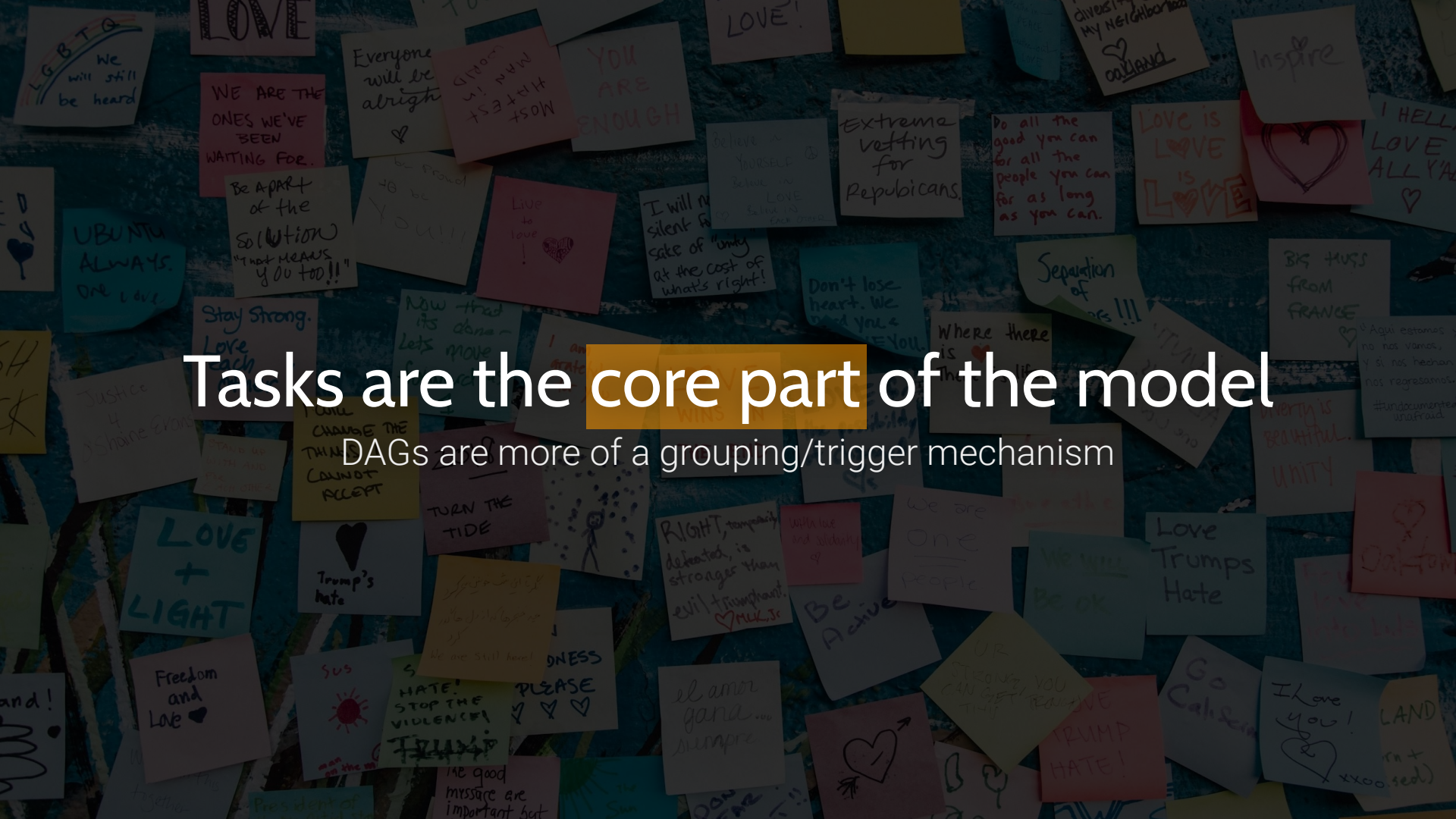

Scheduler
Works out what TaskInstances need to run
Executor
Runs TaskInstances and records the results


Celery or Kubernetes
Our two main options, currently


Database
Webserver
Scheduler
DAG Files
CeleryExecutor
Workers
Redis/Queue


Database
Webserver
DAG Files
Scheduler
KubernetesExecutor
Kubernetes
Task Pods


No status
Scheduler
Up for reschedule
Scheduled
Executor
Removed
Queued
Worker
Upstream failed
Running
Success
Failed
Up for retry
Shutdown
Component
Task stage
Task stage only for sensor
Stage transition
Alternative stage transition


Tasks are the core part of the model
DAGs are more of a grouping/trigger mechanism

Very flexible runtime environments

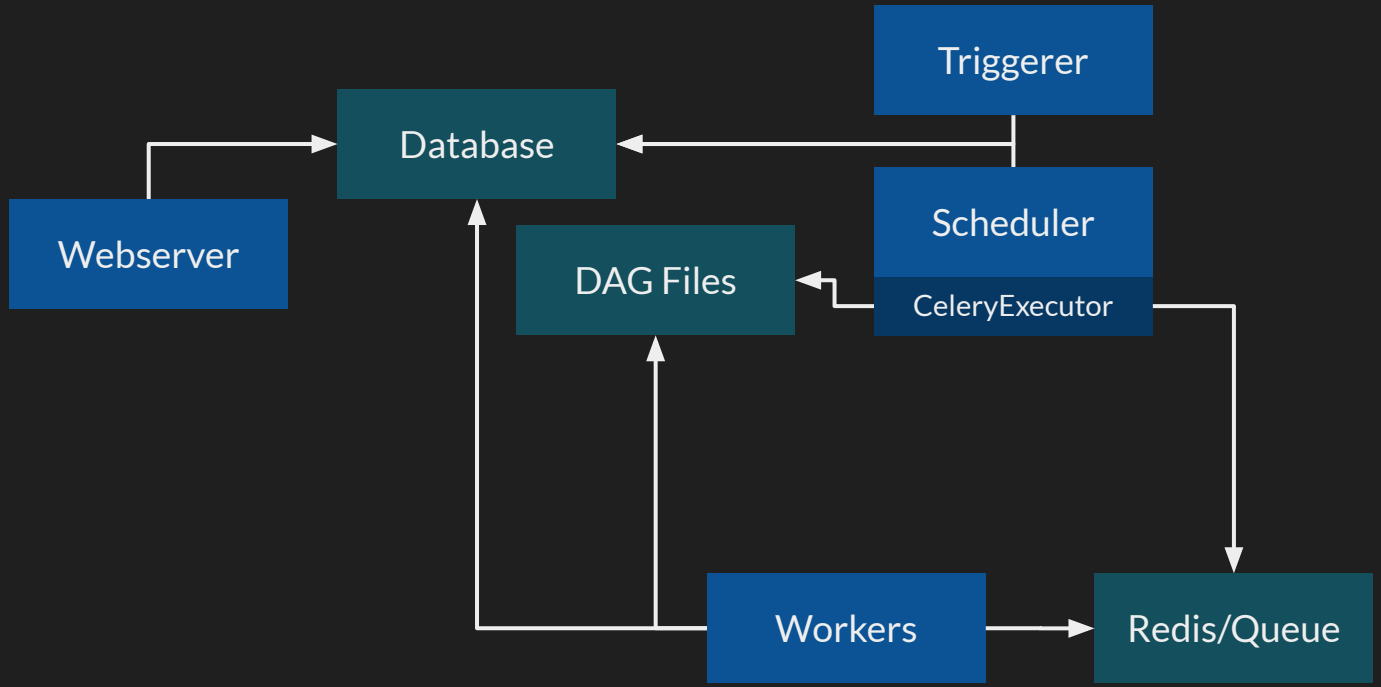
Airflow's strength, and its weakness


Airflow doesn't know what you're running
Though this is also kind of an advantage.


What can we improve?
Let's talk about The Future

More Async & Eventing

Anything that involves waiting!

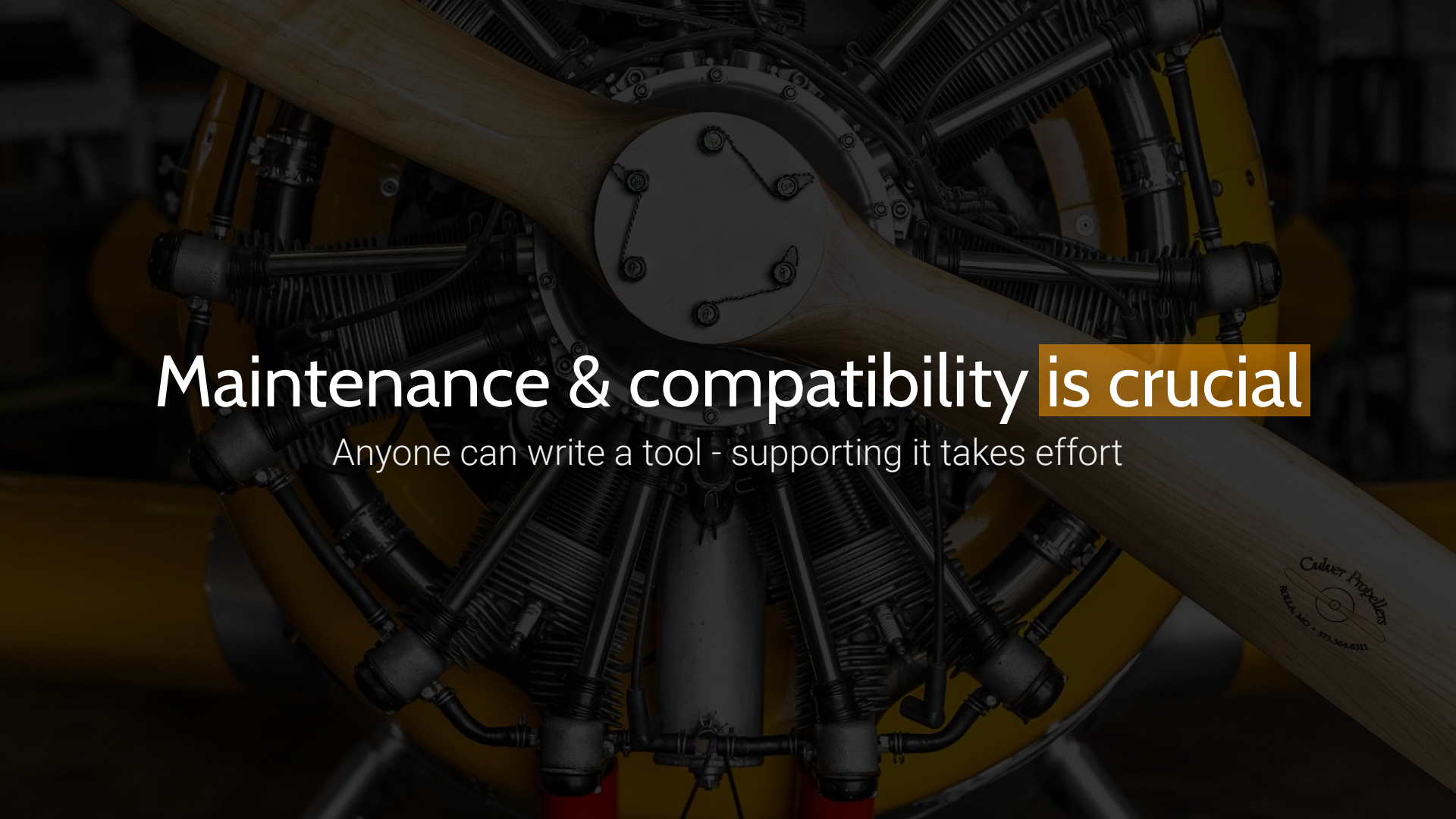

Triggerer
Database
Webserver
Scheduler
DAG Files
CeleryExecutor
Workers
Redis/Queue

Removing Database Connections

APIs scale a lot better!


I do like the database, though
There's a lot of benefit in proven technology


Software Engineering is not just coding
Any large-scale project needs documentation, architecture, and coordination



Maintenance & compatibility is crucial

Anyone can write a tool - supporting it takes effort


Airflow is forged by people like you.
Coding, documentation, triage, QA, support - it all needs doing.

Thanks.

Andrew Godwin
@andrewgodwin
andrew.godwin@astronomer.io