

Airflow

Kristo Raun

Data Engineering 2024 Fall



UNIVERSITY OF TARTU

What is Airflow?

- ◆ Airflow is a platform created by the community to programmatically author, schedule and monitor workflows.

Why Airflow?

Why Airflow?

- ◆ “ We need a new [*dashboard, external API, data warehouse, ML pipeline, ...*] ”

Why Airflow?

- ◆ “ We need a new [*dashboard, external API, data warehouse, ML pipeline, ...*] ”
- ◆ How will we make the job run automatically once per day?

Why Airflow?

- ◆ “ We need a new [*dashboard, external API, data warehouse, ML pipeline, ...*] ”
- ◆ How will we make the job run automatically once per day?
- ◆ Task A, takes ~ 1h

Why Airflow?

- ◆ “ We need a new [*dashboard, external API, data warehouse, ML pipeline, ...*] ”
- ◆ How will we make the job run automatically once per day?
- ◆ Task A, takes ~ 1h
- ◆ Task B, takes ~ 30 min

Why Airflow?

- ◆ “ We need a new [*dashboard, external API, data warehouse, ML pipeline, ...*] ”
- ◆ How will we make the job run automatically once per day?
- ◆ Task A, takes ~ 1h
- ◆ Task B, takes ~ 30 min
- ◆ Task C, takes ~ 30 min, should run only if Task A has finished successfully

Why Airflow?

- ◆ “ We need a new [*dashboard, external API, data warehouse, ML pipeline, ...*] ”
- ◆ How will we make the job run automatically once per day?
- ◆ Task A, takes ~ 1h
- ◆ Task B, takes ~ 30 min
- ◆ Task C, takes ~ 30 min, should run only if Task A has finished successfully
- ◆ Task D, takes ~ 20 min, should run after file “prevDay.csv” is uploaded to “/tmp/taskD”

Why Airflow?

- ◆ “ We need a new [*dashboard, external API, data warehouse, ML pipeline, ...*] ”
- ◆ How will we make the job run automatically once per day?
- ◆ Task A, takes ~ 1h
- ◆ Task B, takes ~ 30 min
- ◆ Task C, takes ~ 30 min, should run only if Task A has finished successfully
- ◆ Task D, takes ~ 20 min, should run after file “prevDay.csv” is uploaded to “/tmp/taskD”
- ◆ Can we still use our current scheduling tool?

About

- ◆ Running a workflow in Airflow is represented as a DAG

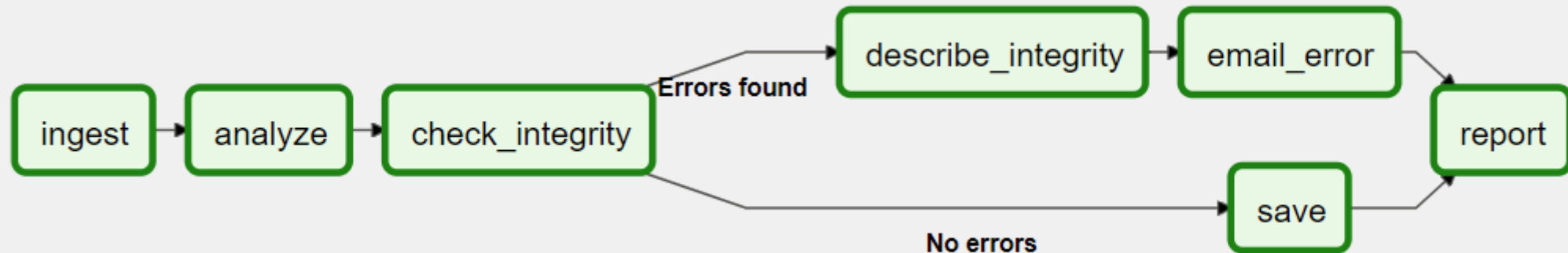
Directed Acyclic Graph



About

- ◇ Running a workflow in Airflow is represented as a DAG

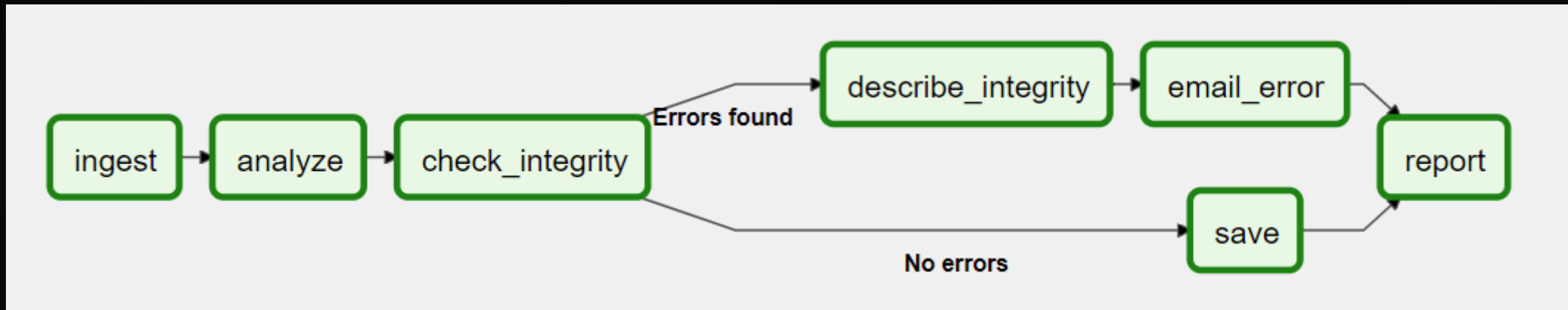
Directed Acyclic Graph



About

- Running a workflow in Airflow is represented as a DAG

Directed Acyclic Graph

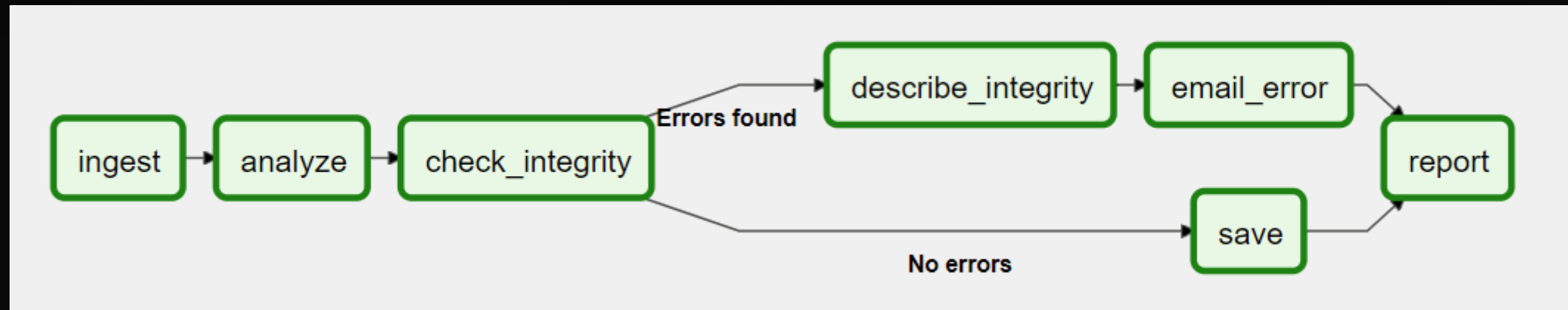


- A DAG specifies the dependencies between Tasks, and the order in which to execute them and run retries;

About

- Running a workflow in Airflow is represented as a DAG

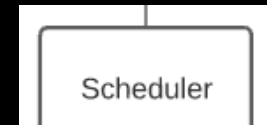
Directed Acyclic Graph



- A DAG specifies the dependencies between Tasks, and the order in which to execute them and run retries;
- The Tasks themselves describe what to do, be it fetching data, running analysis, triggering other systems, or more.

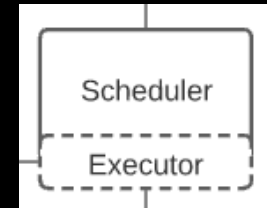
Architecture

- ◇ Scheduler
 - ◇ triggering scheduled workflows
 - ◇ submitting Tasks to the executor to run



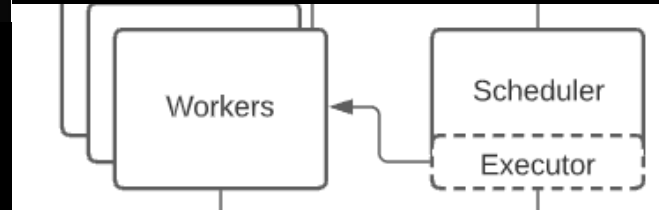
Architecture

- ◇ Scheduler
 - ◇ triggering scheduled workflows
 - ◇ submitting Tasks to the executor to run
- ◇ Executor
 - ◇ handling the running of tasks



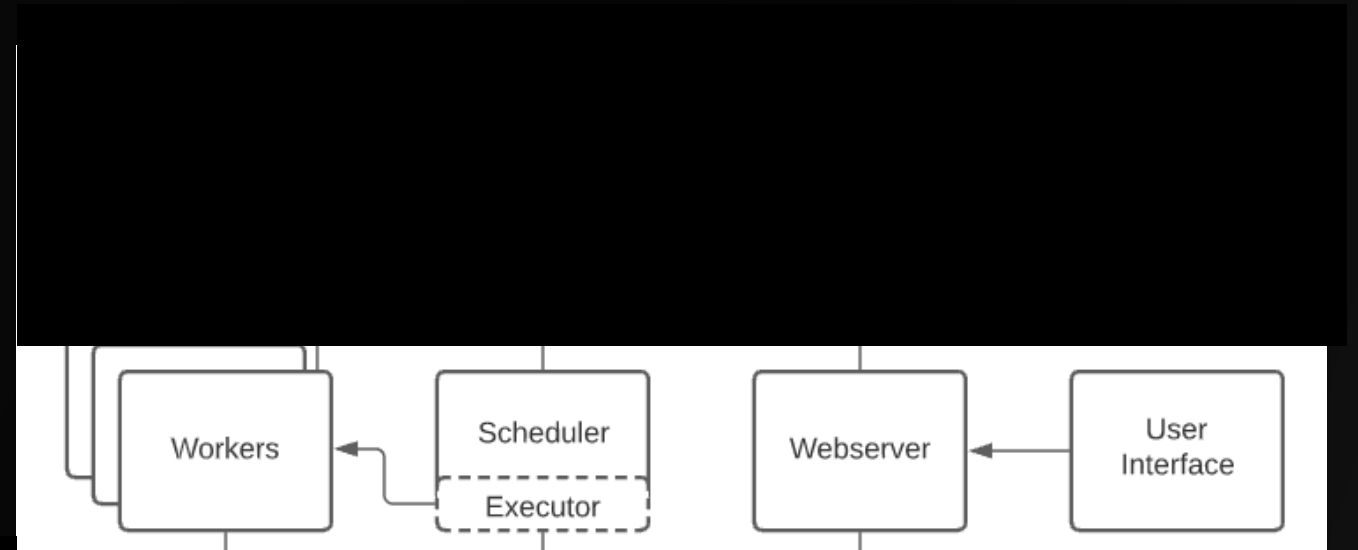
Architecture

- ◇ Scheduler
 - ◇ triggering scheduled workflows
 - ◇ submitting Tasks to the executor to run
- ◇ Executor
 - ◇ handling the running of tasks



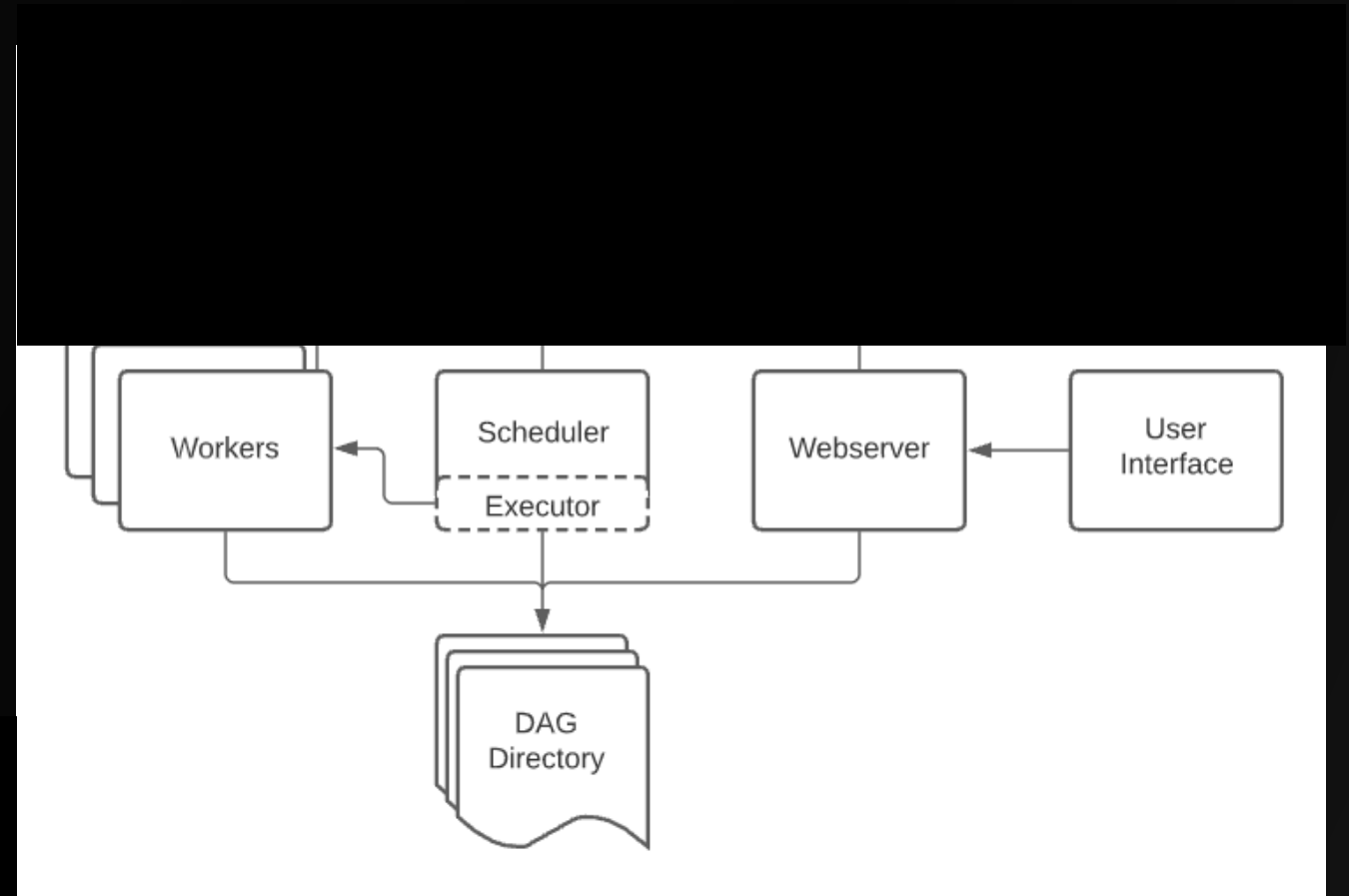
Architecture

- ◇ Scheduler
 - ◇ triggering scheduled workflows
 - ◇ submitting Tasks to the executor to run
- ◇ Executor
 - ◇ handling the running of tasks
- ◇ Webserver
 - ◇ UI to inspect, trigger and debug the behaviour of DAGs and tasks



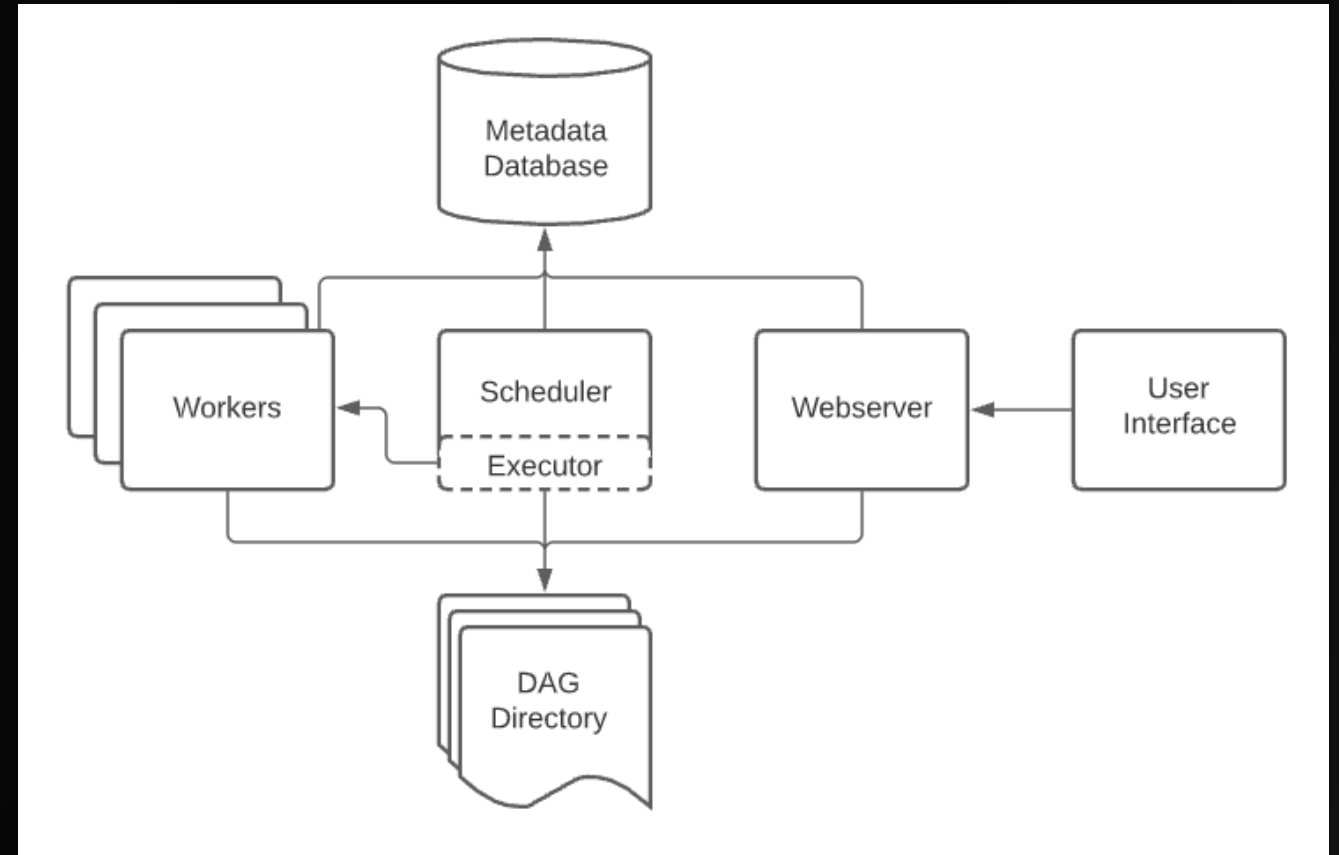
Architecture

- ◇ Scheduler
 - ◇ triggering scheduled workflows
 - ◇ submitting Tasks to the executor to run
- ◇ Executor
 - ◇ handling the running of tasks
- ◇ Webserver
 - ◇ UI to inspect, trigger and debug the behaviour of DAGs and tasks
- ◇ A folder of DAG files
 - ◇ read by the scheduler and executor (and any workers the executor has)



Architecture

- ◇ Scheduler
 - ◇ triggering scheduled workflows
 - ◇ submitting Tasks to the executor to run
- ◇ Executor
 - ◇ handling the running of tasks
- ◇ Webserver
 - ◇ UI to inspect, trigger and debug the behaviour of DAGs and tasks
- ◇ A folder of DAG files
 - ◇ read by the scheduler and executor (and any workers the executor has)
- ◇ A metadata database
 - ◇ used by the scheduler, executor and webserver to store state.



Airflow dictionary

- ◇ DAG

- ◇ “workflow”

Airflow dictionary

- ◇ DAG
 - ◇ “workflow”
- ◇ Task
 - ◇ A basic unit of execution

Airflow dictionary

- ◊ DAG
 - ◊ “workflow”
- ◊ Task
 - ◊ A basic unit of execution
- ◊ Operator
 - ◊ A template for a task. *PostgresOperator, PythonOperator, HiveOperator, ...*

Airflow dictionary

- ◊ DAG
 - ◊ “workflow”
- ◊ Task
 - ◊ A basic unit of execution
- ◊ Operator
 - ◊ A template for a task. *PostgresOperator, PythonOperator, HiveOperator, ...*
- ◊ Sensor
 - ◊ An operator for waiting for something. *FileSensor, SqlSensor, ExternalTaskSensor*

Final words...

Final words...

- ◆ Airflow is great for organizing complex workflows



Final words...

- ◆ Airflow is great for organizing complex workflows
- ◆ Airflow is not a classical ETL tool

