

Beyond Jupyter Notebooks



Build your own data science platform with Python & Docker

github.com/jgoerner/beyond-jupyter

"SEXIEST JOB OF 21TH CENTURY"



IRREPRODUCIBLE, SPAGHETTI CODE

I DON'T LIKE NOTEBOOKS

Joel Grus (@joelgrus)

#JupyterCon 2018

{ audience booing }

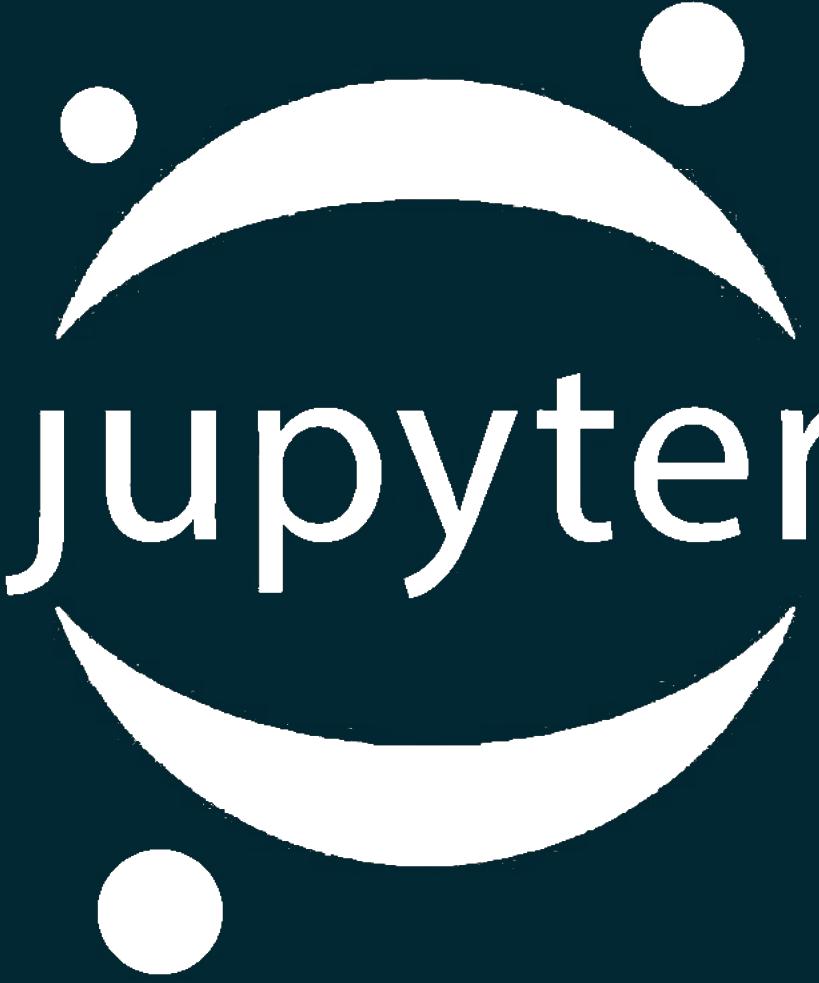
The ugly truth

Static
Visualisations

Difficult
Model Retraining

Isolated
Flat Data Files

Challenging
Model Exposure



jupyter 101

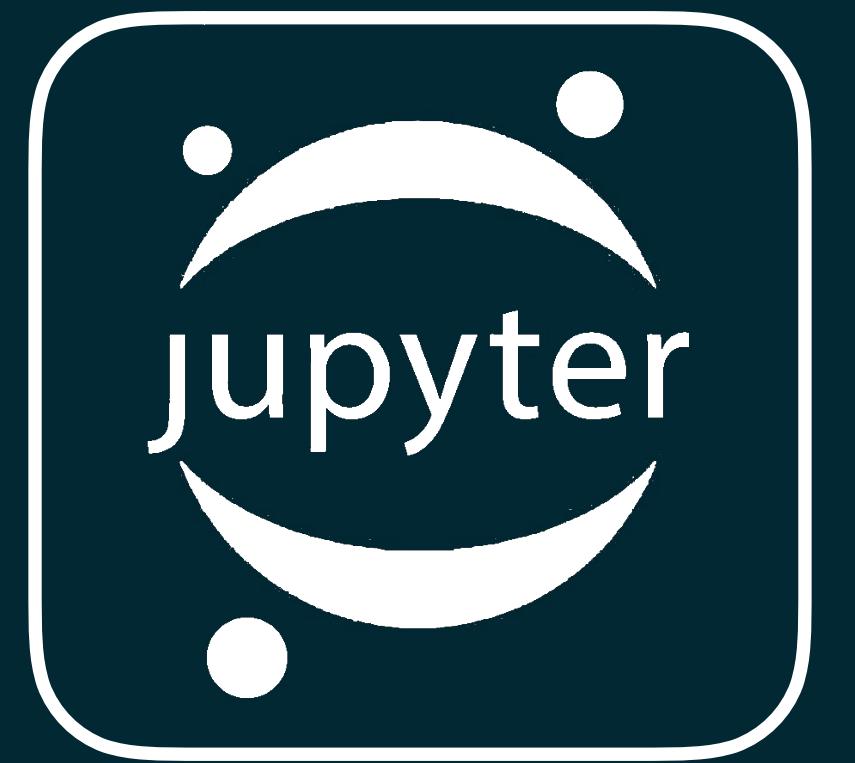
Jupyter: Live demo

<http://localhost:8888>

Docker Command

```
run in background          image name  
  |  
docker run -d -p 8888:8888 jupyter/scipy-notebook  
  |  
run container      map ports
```

Architecture



Computation

The ugly truth

Static
Visualisations

Difficult
Model Retraining

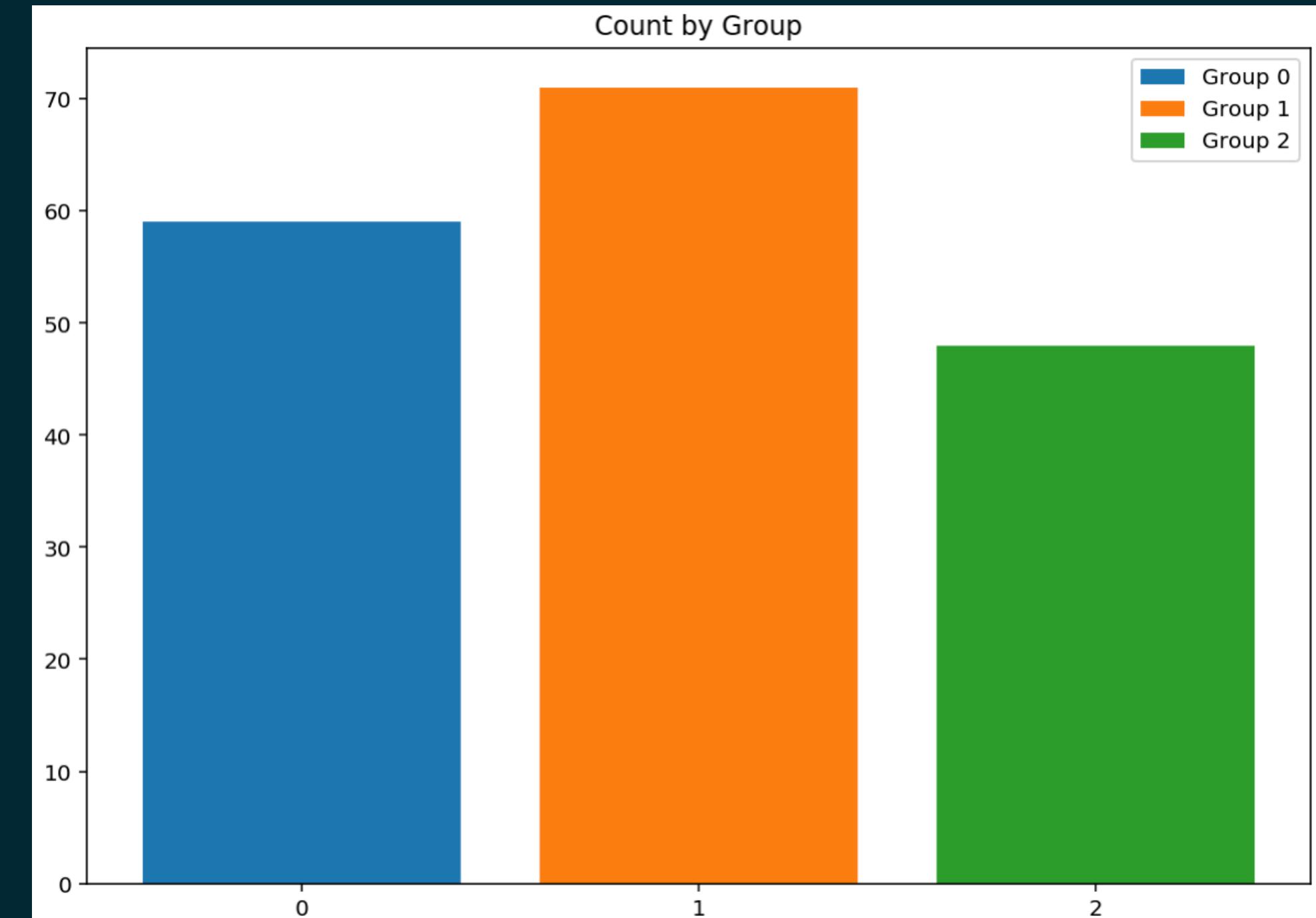
Isolated
Flat Data Files

Challenging
Model Exposure

Effort vs. Outcome

```
1 # calculate classes and counts
2 classes, counts = np.unique(
3     df_wine["target"],
4     return_counts=True,
5 )
6
7 # plot frequency per bar
8 for (cls, cnt) in zip(classes, counts):
9     plt.bar(
10         x=cls,
11         height=cnt,
12         label="Group {}".format(**locals()))
13
14 # general aesthetics
15 plt.legend(loc=0)
16 plt.xticks([0, 1, 2])
17 plt.title("Count by Group")
```

Input



Output



101

Superset

Superset: Live demo

<http://localhost:8088>

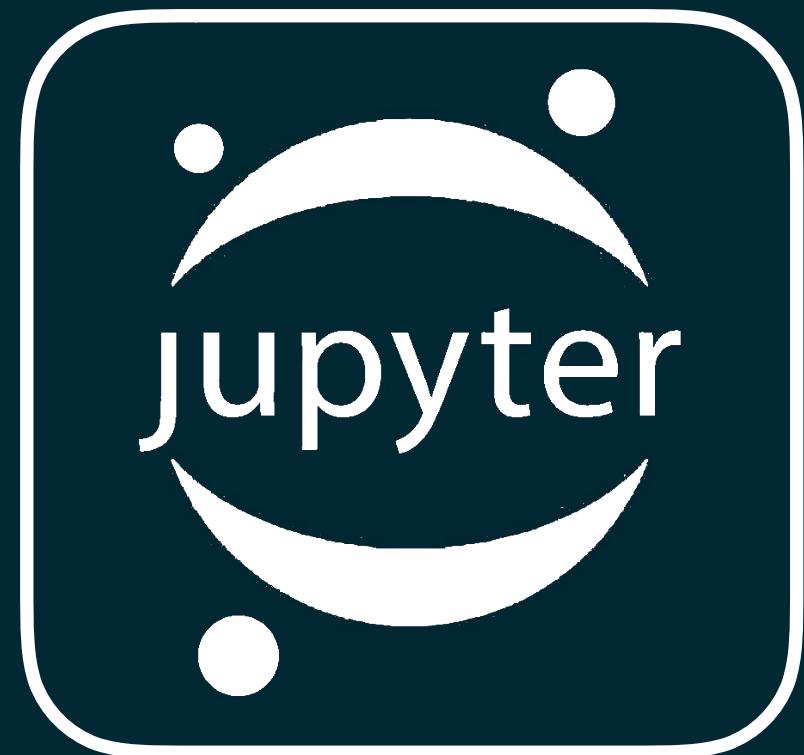
Docker Command

```
run in background  
docker run -d -p 8088:8088 tylerfowler/superset  
run container      map ports  
image name
```

The diagram illustrates a Docker command line with the following annotations:

- A vertical line with a dot at its top points to the `-d` option, labeled "run in background".
- A vertical line with a dot at its top points to the `-p` option, labeled "map ports".
- A vertical line with a dot at its top points to the image name `tylerfowler/superset`, labeled "image name".
- A vertical line with a dot at its bottom points to the `run` command, labeled "run container".

Architecture



Computation



Visualization

The ugly truth

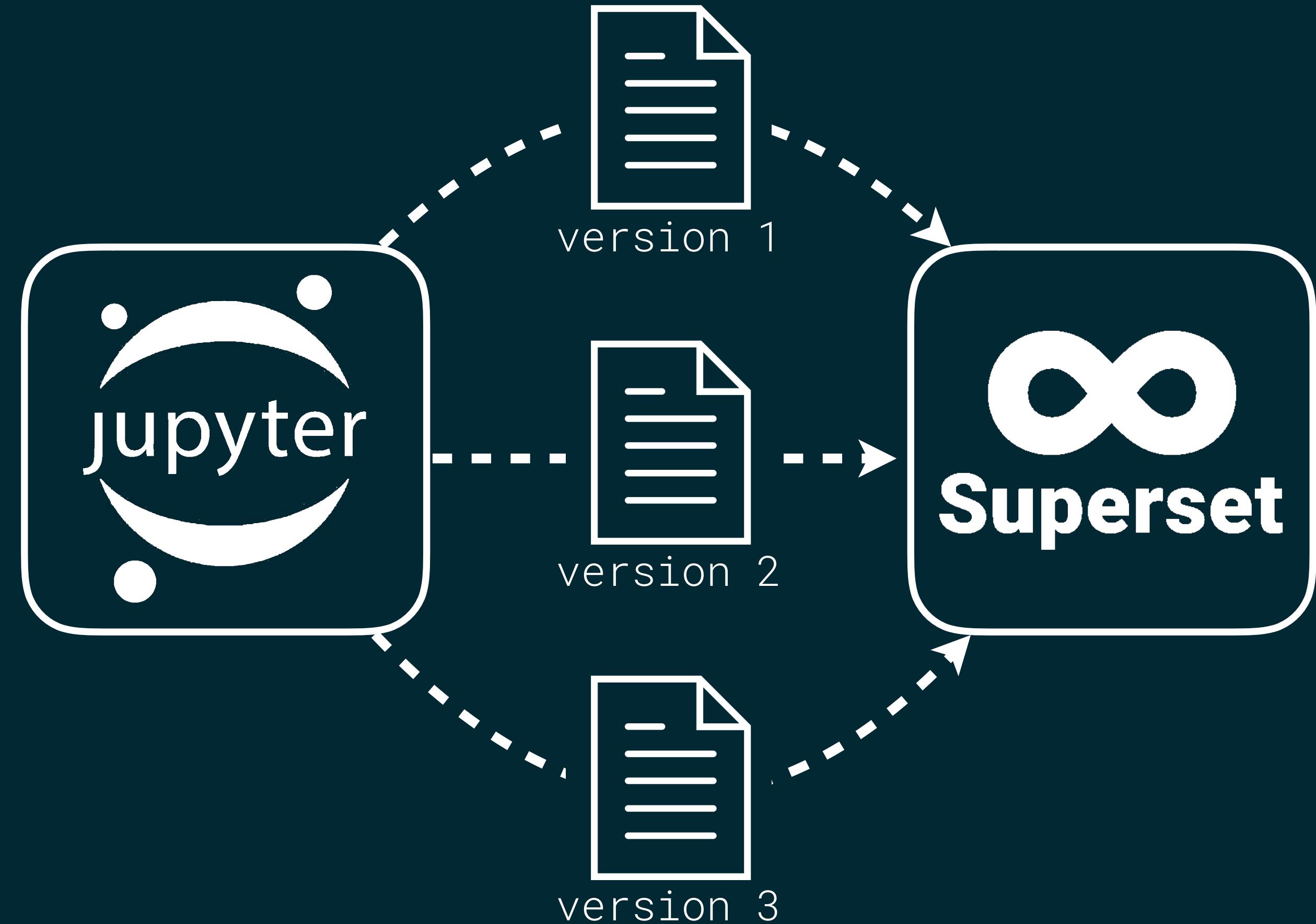
Static
Visualisations

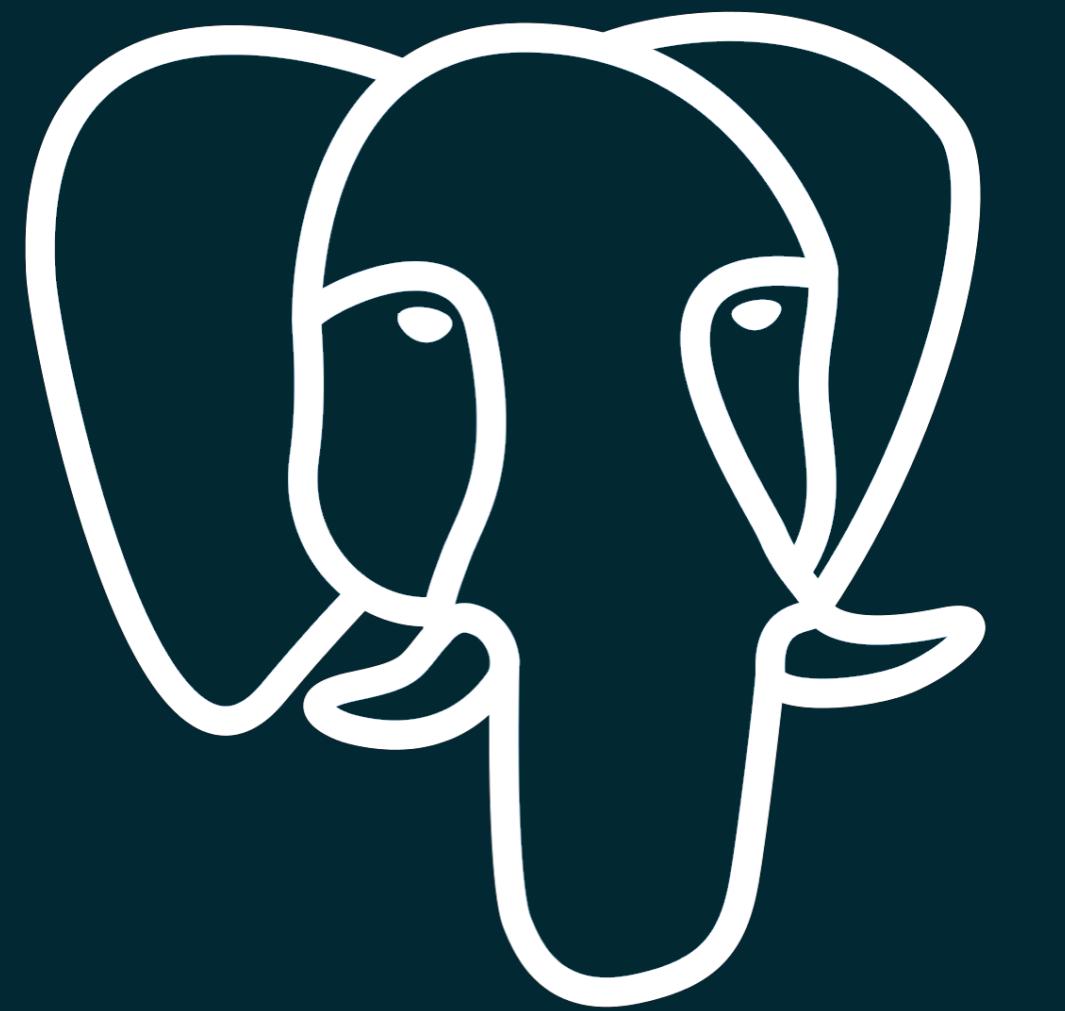
Isolated
Flat Data Files

Difficult
Model Retraining

Challenging
Model Exposure

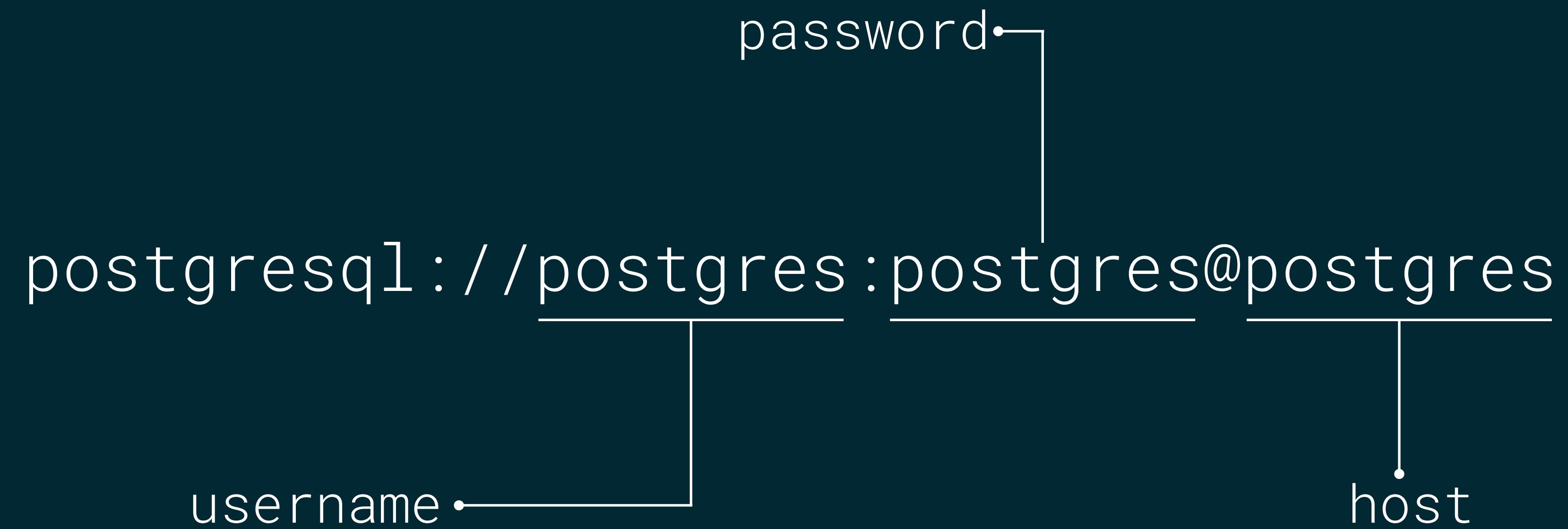
Manual Data Push



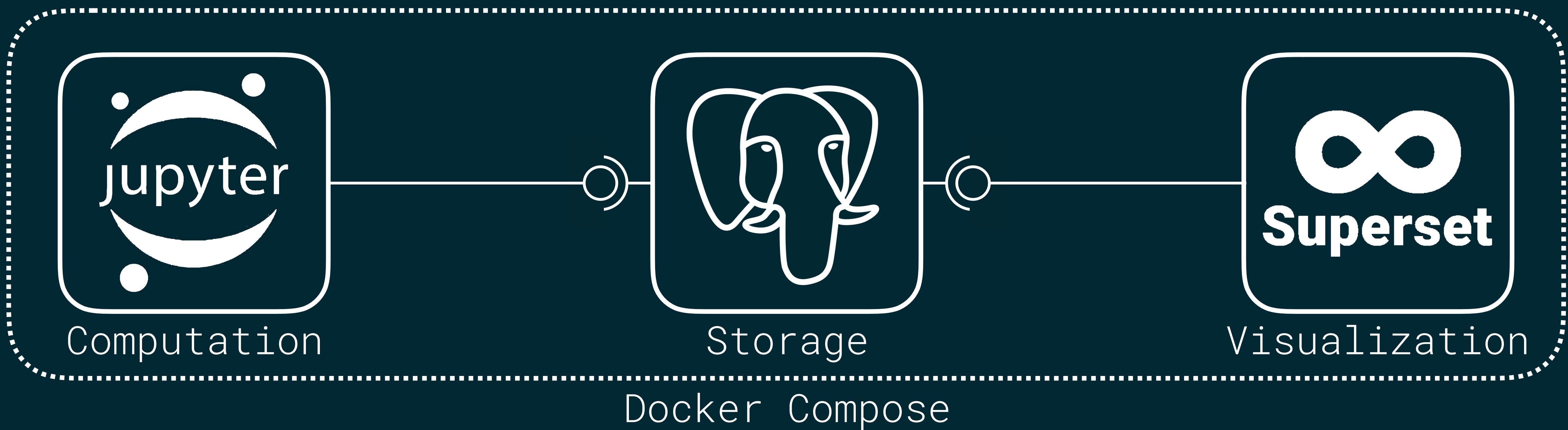


101

Postgres Connections



Architecture



Docker Compose File I/II

```
01 version: "3"
02 services:
03
04   jupyter:
05     container_name: "pycon_jupyter"
06     image: jupyter/scipy-notebook
07     ports:
08       - 8888:8888
09
10   superset:
11     container_name: "pycon_superset"
12     image: tylerfowler/superset
13     ports:
14       - 8088:8088
```

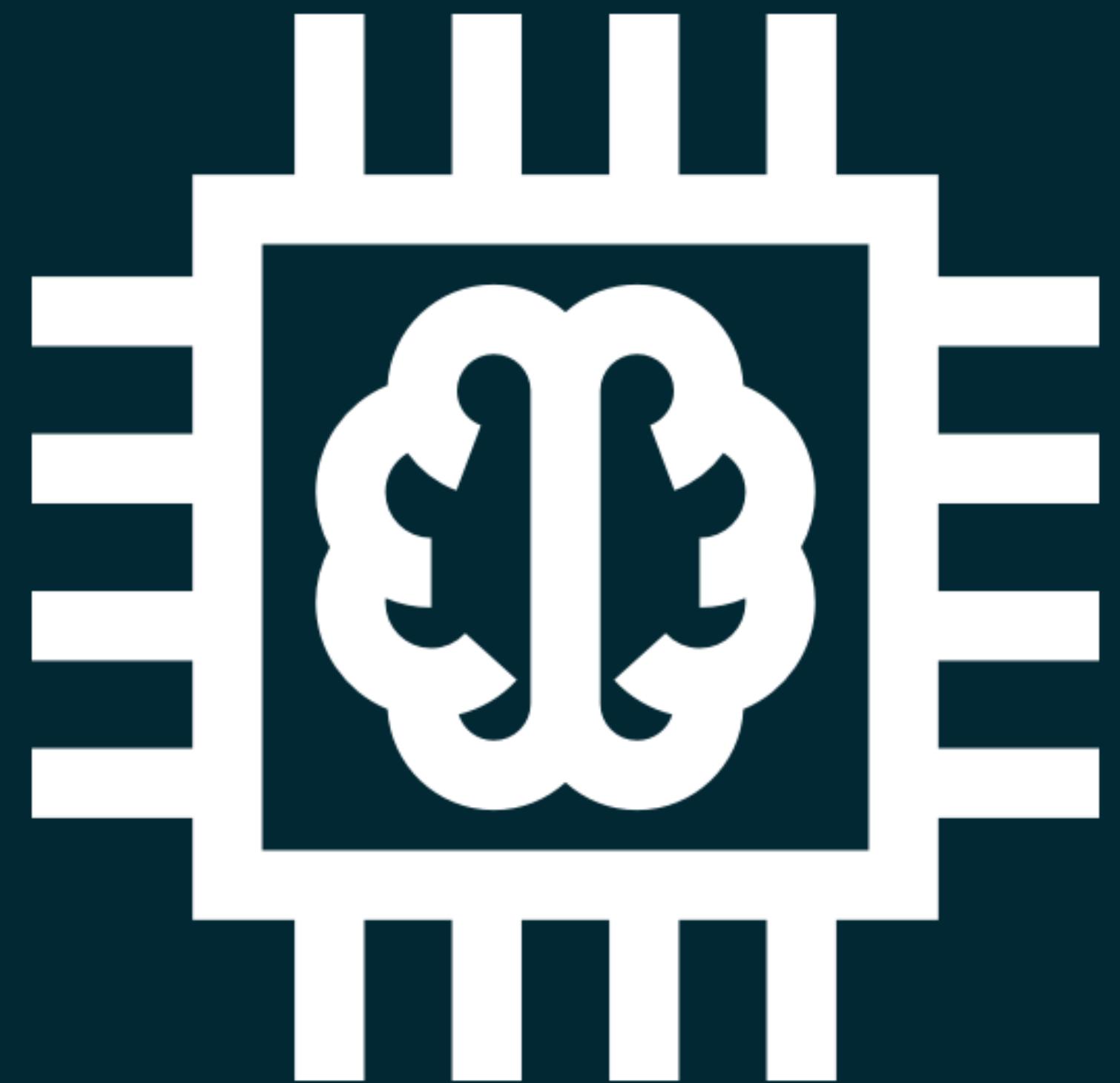
Docker Compose File II/II

```
01    version: "3"
02    services:
03
04    // jupyter & superset services //
05
06    postgres:
07        container_name: "pycon_postgres"
08        image: postgres
09        volumes:
10            - pg_volume:/var/lib/postgresql/
11        ports:
12            - 5432:5432
13
14    volumes:
15        pg_volume
```

Docker Compose Command

```
docker compose up -d
```

What about the Model?





101

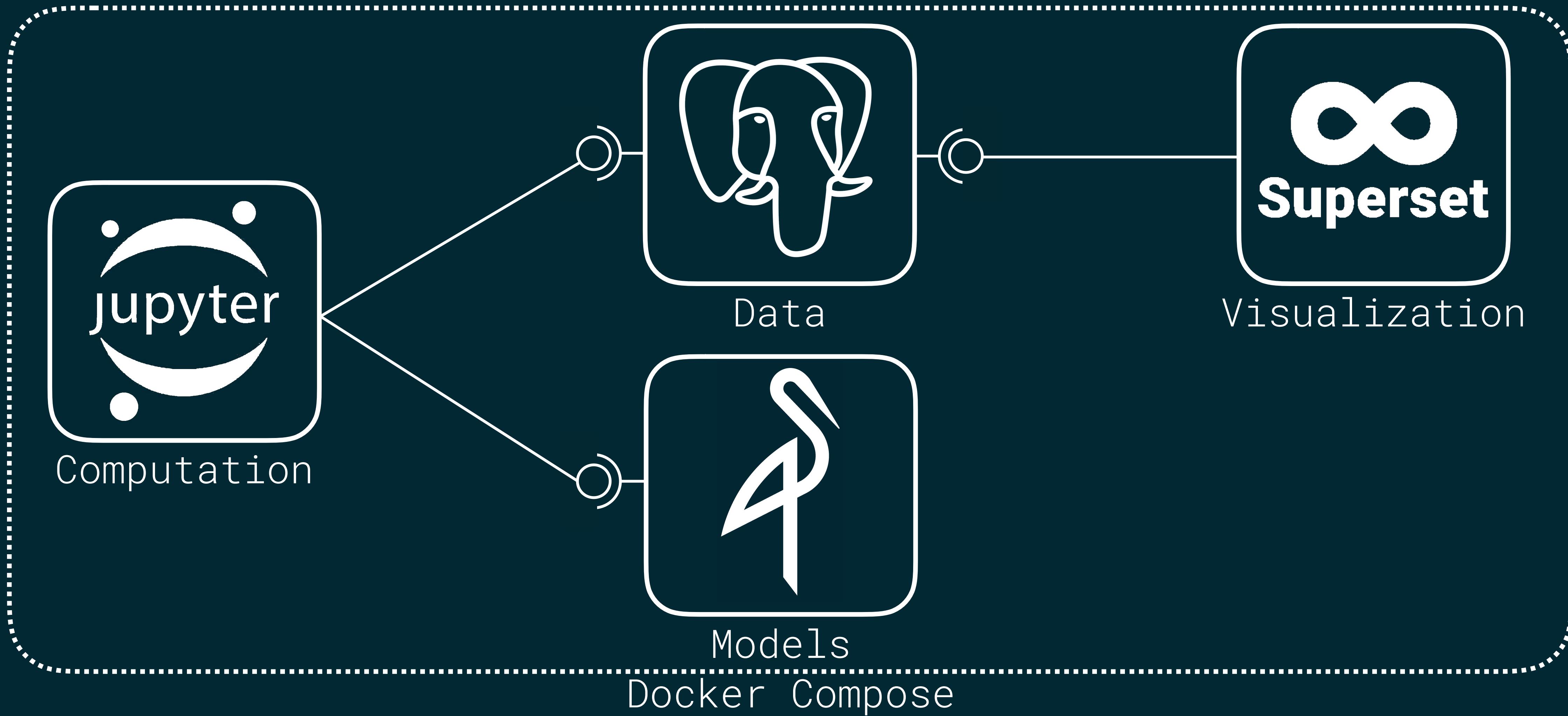
Minio: Live demo

<http://localhost:9000>

Docker Compose Enhancement

```
01  version: "3"
02  services:
03
//    jupyter & superset services, postgres //
23
24  minio:
25      container_name: "pycon_minio"
26      image: minio
27      volumes:
28          - minio_volume:/data
29      ports:
30          - 9000:9000
31      command:
32          - server /data
33
//    volumes for postgres & minio //
```

Architecture



The ugly truth

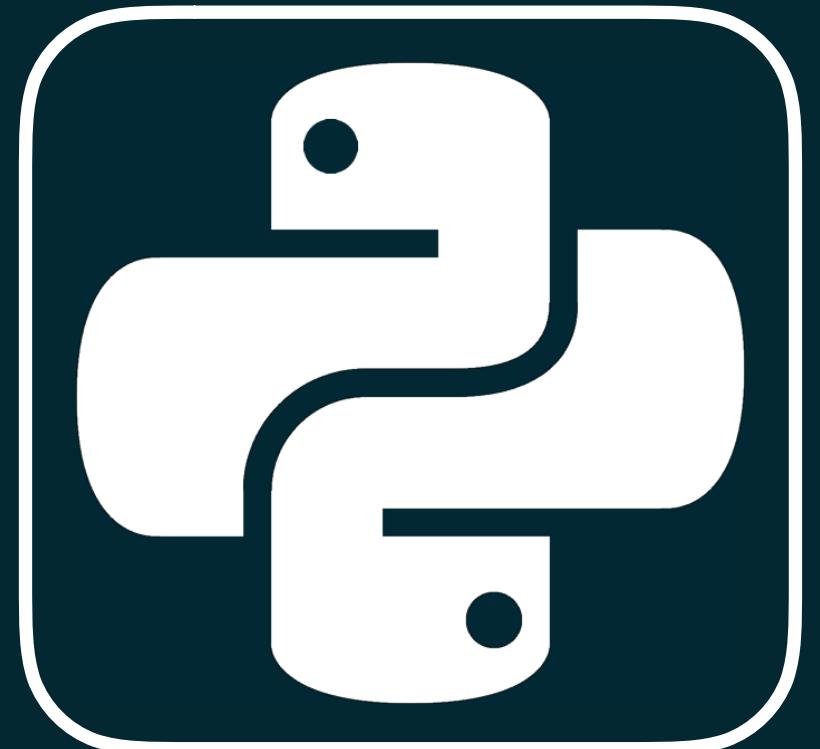
~~Static
Visualisations~~

~~Isolated
Flat Data Files~~

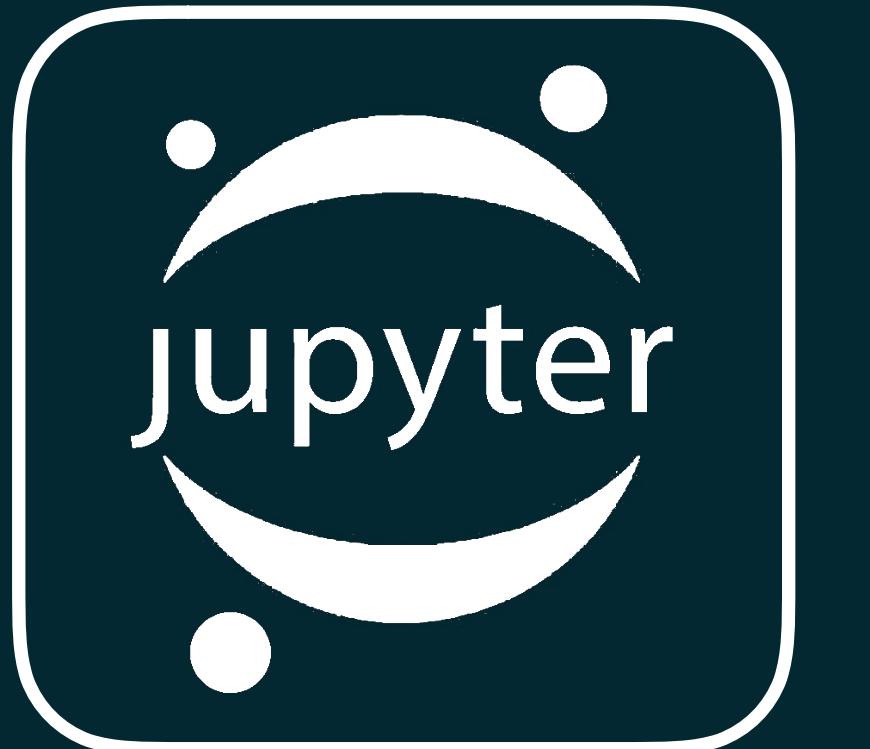
Difficult
Model Retraining

Challenging
Model Exposure

How to Predict-as-a-Service?



Binary Model



Jupyter Notebook



Minio



101

API Star: Live demo

<http://localhost:8000>

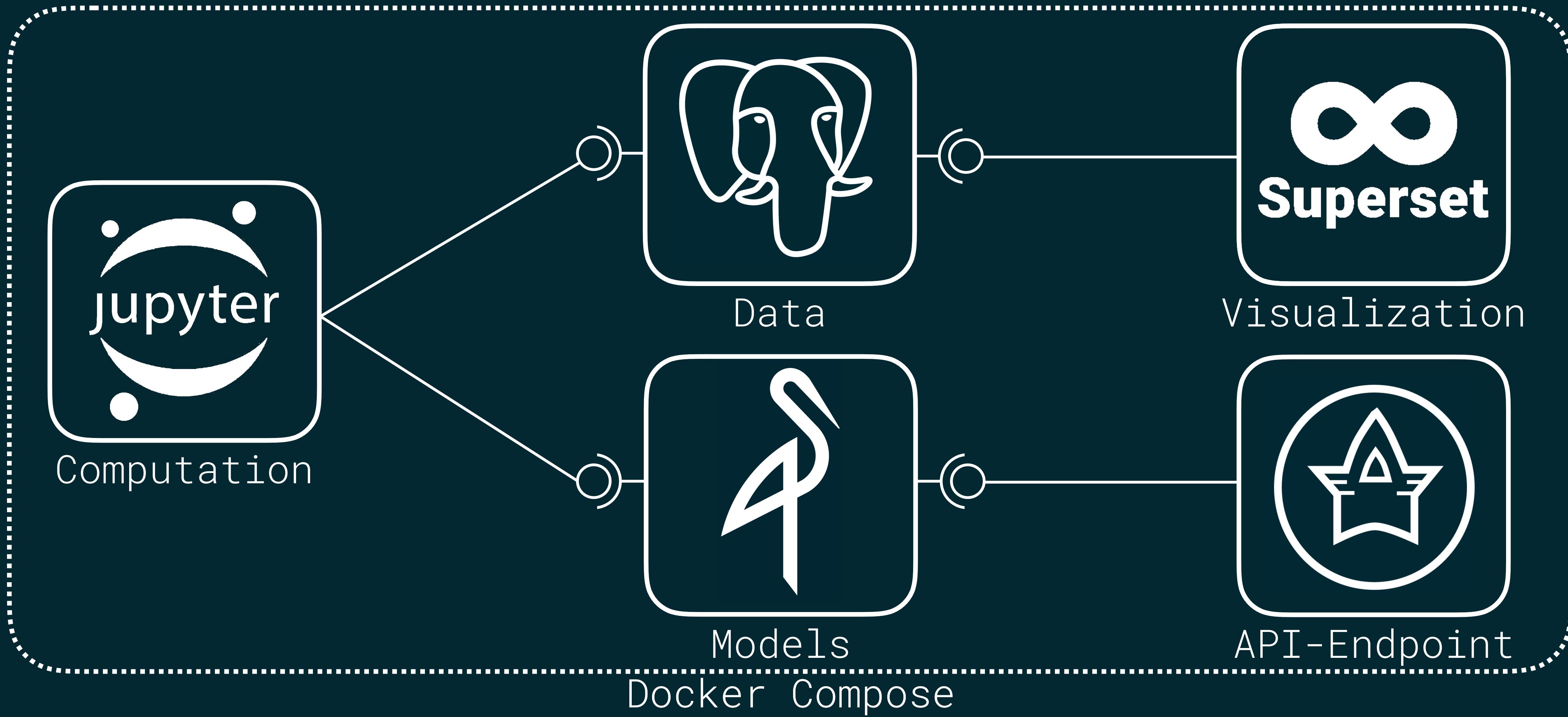
Docker Compose Enhancement

```
01  version: "3"
02  services:
03
04    // jupyter, superset, postgres, minio //
05
06    apistar:
07      container_name: "pycon_apistar"
08      build: services/apistar
09      volumes:
10        - ./services/apistar/api:/usr/src/app
11      ports:
12        - 8000:8000
13      command: gunicorn app:app -b 0.0.0.0:8000
14
15    // volumes for postgres & minio //
```

Apistar Dockerfile

```
01  FROM python:3
02
03  WORKDIR /usr/src/app
04
05  RUN pip install --no-cache-dir \
06      apistar==0.5.41 \
07      gunicorn
08
09  EXPOSE 8000
```

Architecture



The ugly truth

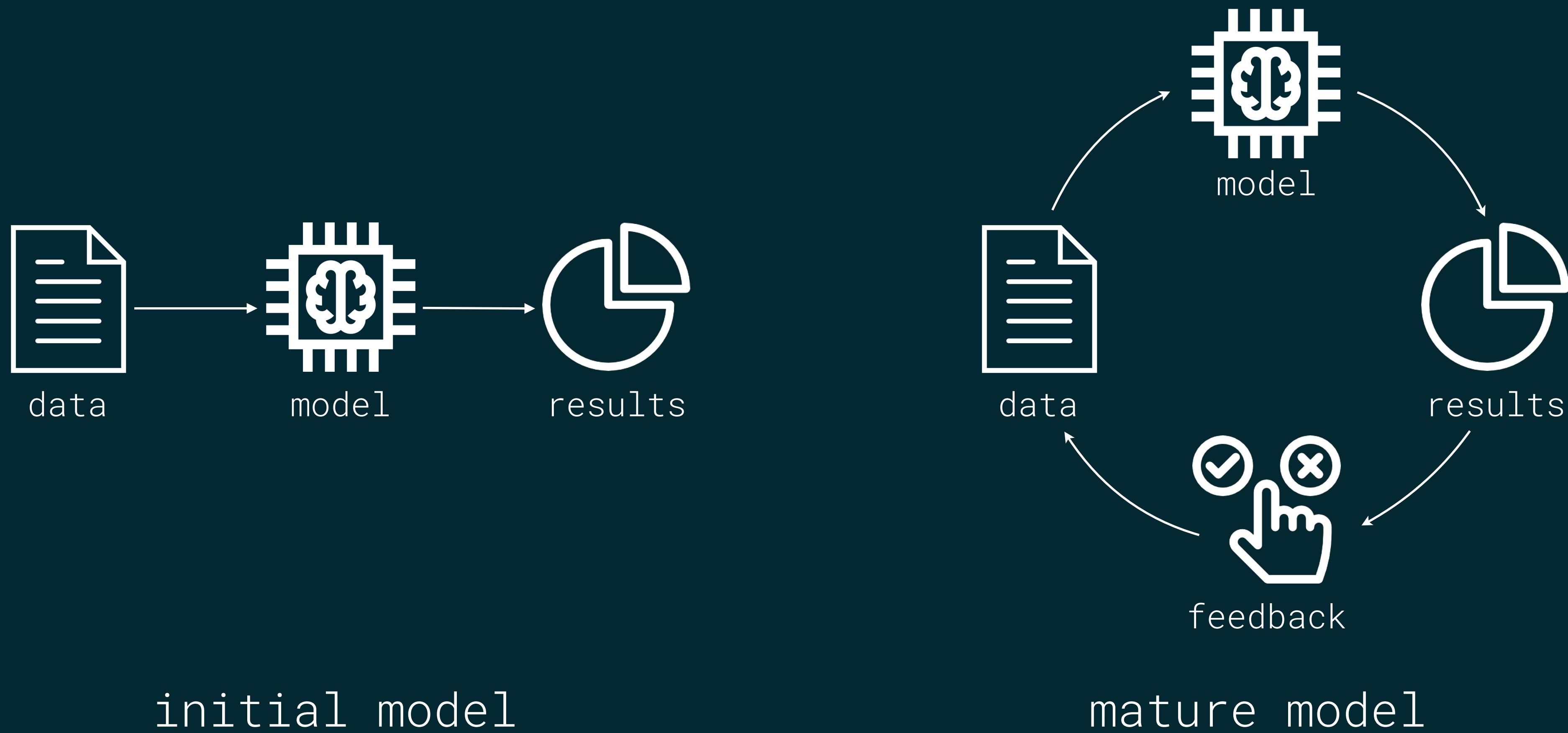
~~Static
Visualisations~~

~~Isolated
Flat Data Files~~

Difficult
Model Retraining

~~Challenging
Model Exposure~~

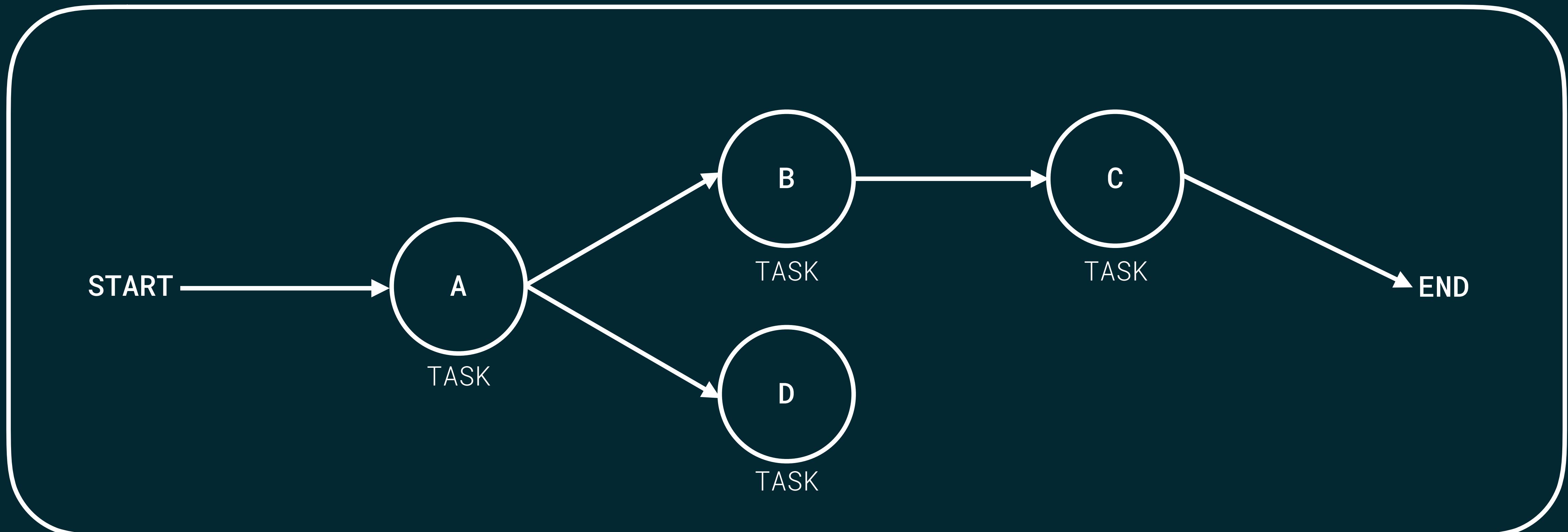
Model Maturity





101

Airflow Core Concepts



DIRECTED ACYCLIC GRAPH (DAG)

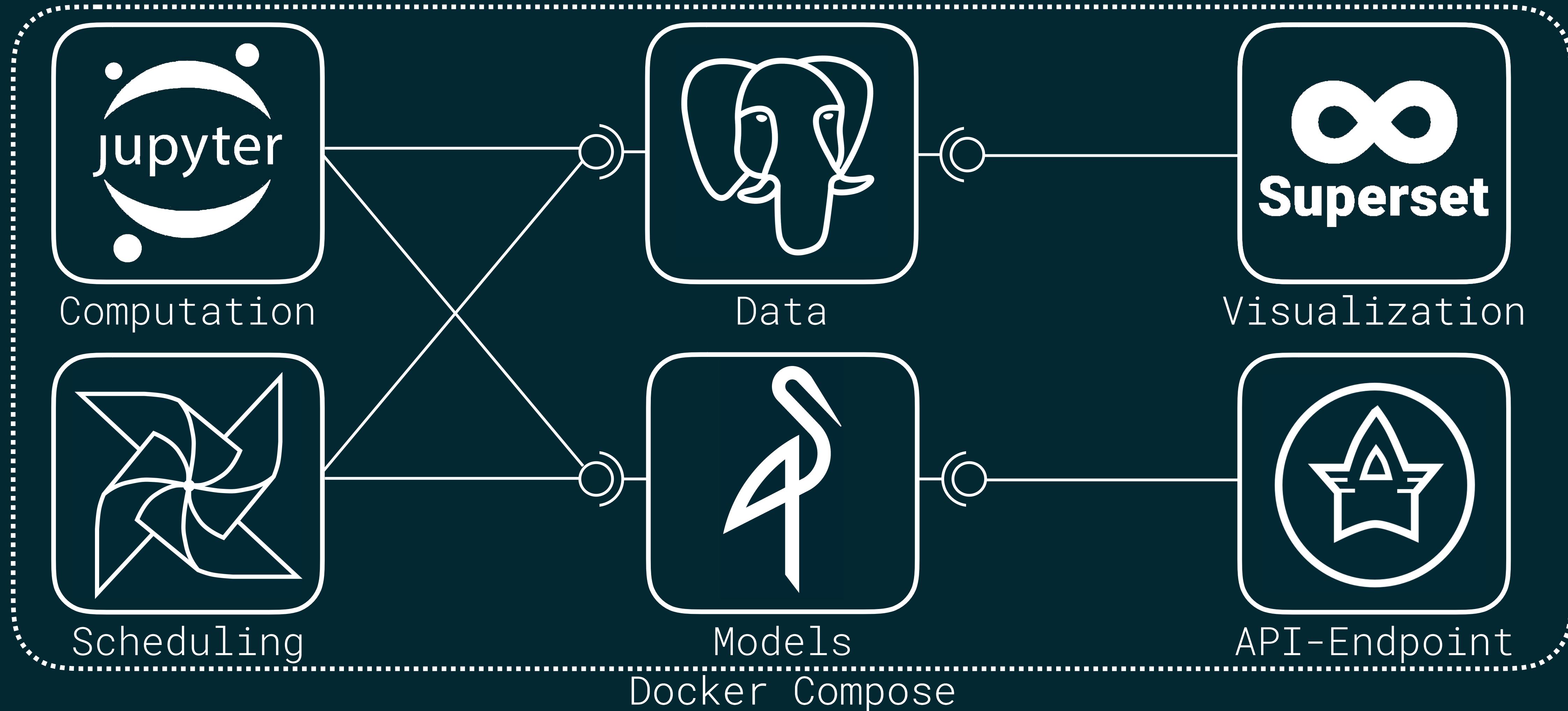
Airflow: Live demo

<http://localhost:8080>

Docker Compose Enhancement

```
01    version: "3"
02    services:
03
04        // jupyter, superset, postgres, minio, apistar //
05
06    airflow:
07        container_name: "pycon_airflow"
08        image: puckel/docker-airflow
09        depends_on:
10            - postgres
11        environment:
12            - EXECUTOR=Local
13        ports:
14            - 8080:8080
15        command: webserver
16
17    // volumes for postgres & minio //
```

Architecture



The ugly truth

Static
Visualisations

Isolated
Flat Data Files

Difficult
Model Retraining

Challenging
Model Exposure

The beautiful truth

Portability

Maintainability

Flexibility

Customize to your needs!



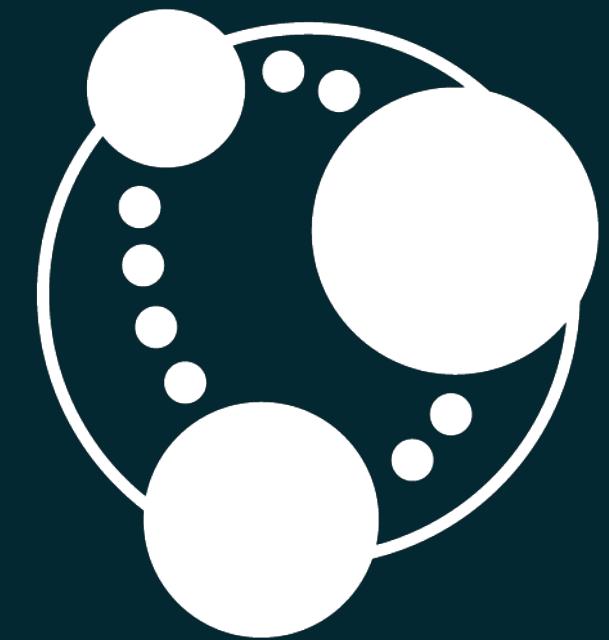
in-memory broker



webserver



search platform



graph database



monitoring & alerting

503 - Information Overload?



Use the Cookiecutter!

```
joshuagorner@Le-Big-Mac ➜ cookiecutter https://github.com/jgoerner/data-science-stack-cookiecutter.git  
You've downloaded /Users/joshuagorner/.cookiecutters/data-science-stack-cookiecutter before. Is it okay to delete and re-download it? [yes]: yes  
project_name [project_name]:
```

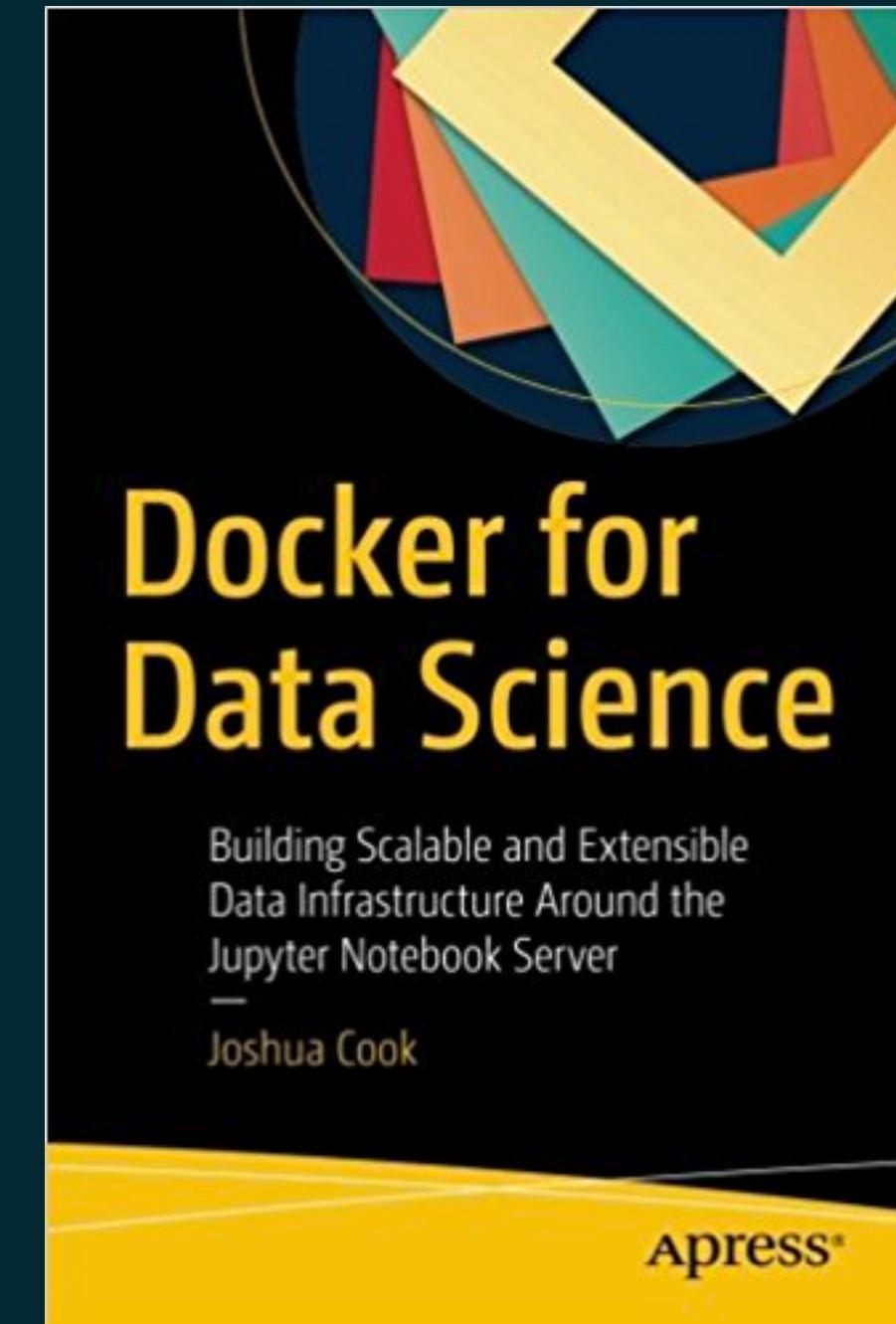


Thanks for listening!

Further Resources



“Data Science is Software”
(Talk)



“Docker for Data Science”
(Book)

Beyond Jupyter



>> Take your data science skills to the next level



(beyond-jupyter.io)