

Automatic Speech Recognition at Scale Using Tensorflow, Kubernetes and Airflow



A faint, light gray wireframe globe is centered in the background, showing latitude and longitude lines.

May 23—27, 2022

AIRFLOW SUMMIT

Large, bright green abstract shapes are positioned on the left and right sides of the image. On the left, there is a thick, curved line and a solid rectangular block below it. On the right, there is a thick, curved line. Small green triangles point towards the center from the top right and bottom left.

What we **will** talk about today

- Automatic Speech Recognition
- Data Centric AI
- Task Orchestration
 - Airflow
 - Kubernetes
 - Kubernetes Pod Operator
- MLOps
- ~~ML Model Training~~

Who am I?

- [Rafael V. Pierre](#), MSc. @ University of Amsterdam
- [Solutions Architect](#) @ Databricks
- [15 years experience](#) in technology and data intensive industries
- Passionate about [Machine Learning](#), [Data Engineering](#), [MLOps](#) and [Call of Duty](#)



Business Context



Top european bank, 38
million customers globally



Millions of customers calls
per year



Sizable customer service
team



Some quotes to get inspired

*“Success comes from
listening to your customer”*



Sir Richard Branson, Virgin Group

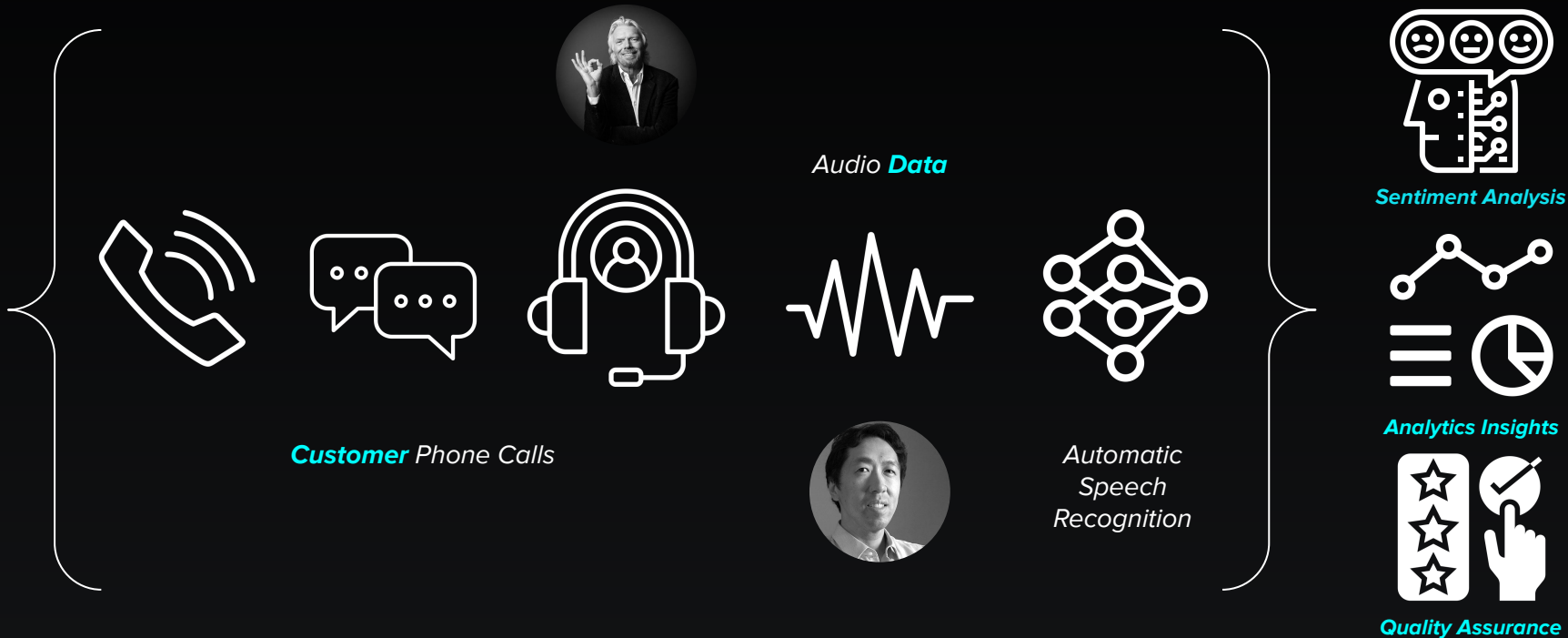
“Data is the new electricity”

The Rise of Data-Centric AI



*Andrew Ng, PhD, Stanford University,
Coursera, DeepLearning.ai*

Steps to success





Minimum Viable Product



Crontab Job

```
[
  {
    "word": "hi",
    "start": "0.2",
    "end": "1.1",
    "probability": "0.88"
  },
  {
    "word": "good",
    "start": "1.5",
    "end": "2.7",
    "probability": "0.79"
  },
  {
    "word": "morning",
    "start": "3.0",
    "end": "4.6",
    "probability": "0.79"
  }
]
```

Transcription

Challenges

Resilience

Fault tolerance
No retry mechanism

Hard to scale

Long time to transcribe
Sequential processing
No task dependencies

Complexity

Not reusable
Hard to monitor
Credentials management

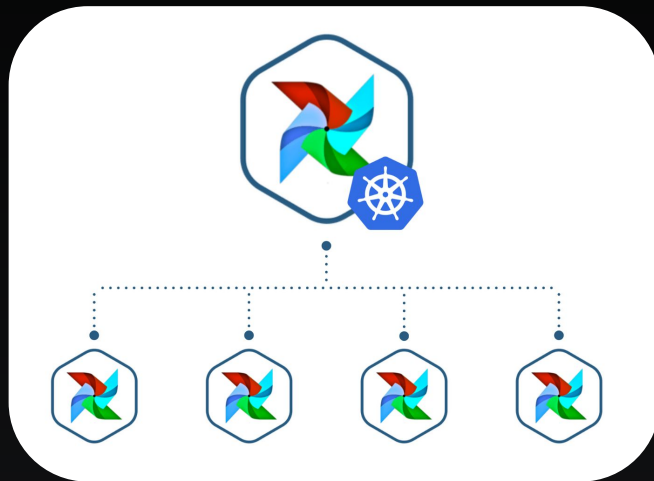


Ingredients

Kubernetes Executor (On-Premise)



KubernetesPodOperator



Recap: Kubernetes Pod

```
apiVersion: v1
kind: Pod
metadata:
  name: frontend
spec:
  containers:
    - name: app
      image: images.my-company.example/app:v4
      resources:
        requests:
          memory: "64Mi"
          cpu: "250m"
        limits:
          memory: "128Mi"
          cpu: "500m"
    - name: log-aggregator
      image: images.my-company.example/log-aggregator:v6
      resources:
        requests:
          memory: "64Mi"
          cpu: "250m"
        limits:
          memory: "128Mi"
          cpu: "500m"
```

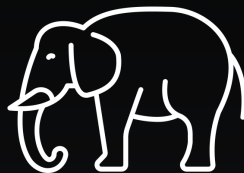

Audio files



Dynamic DAGs



ELT + Postgresql



Achievements & Takeaways

- **100X Transcription Throughput Increase**
- **Data Centric AI**
 - **MLOps** is important
 - **Productionizing** ML Models is harder than **training**
- **Airflow Kubernetes Pod Operator**
 - **Powerful** combination!
 - Great **flexibility**
 - **Horizontal** and **Vertical** Scaling
 - From 1 to 100s of workers without any code change
- **DAGs** are **reusable** and **parameterizable**
- **Credentials** are safely stored as **Airflow Secrets**

Future Work

- **DAG Proliferation**
 - More reusable DAGs
 - Same logic for different languages
- **Hugging Face**
 - Pre-trained models
 - State of the art for ASR: Wav2Vec2 and Data2Vec2
- **From Batch to Streaming**
 - Example: [Databricks Delta Auto Loader](#)
- **Distributed Training & Inference**
 - **Airflow + Databricks**
 - [DatabricksSubmitRunOperator](#)
 - [DatabricksRunNowOperator](#)
 - [Spark + Horovod + Petastorm](#)

Thanks!

<https://www.linkedin.com/in/rafaelpierre>

<https://mlopshowto.com>

***Disclaimer:** Opinions stated in this presentation are my own and do not reflect in any way my current or previous employers. The use case and technical decisions described here are not related to my current employer and are presented strictly for educational purposes; in no way they represent any kind of recommendation or endorsement.*