MASMOVIL

Call transcription using AI

Alejandro Sánchez Muñoz

#bbs

MASMOVIL

Overview

Call transcription using AI

#bbs

Project Scope

- **Transcription** of the **calls** to the **call center** from all the brands of **MASMOVIL Group**
- **Data Sources**:
  - **Genesys Cloud** (Platform that manages call center calling data)
  - **External Platforms** (audio files stored in GCS Buckets)

Analysis of S2T tools

- There are several **S2T tools** in the market. We have chosen three of them to evaluate:
  - Genesys Cloud transcription tool
  - Google Speech-to-Text
  - **OpenAI Whisper**

MASMOVIL

OpenAI Whisper

- **Whisper** is a popular Speech-to-Text tool from **OpenAI**
- Whisper has **accurate results** and splits the whole transcription into paragraphs/words, adding **start & end timestamps**
- We have decided to **download the model** instead using the Whisper API

```
0.0;5.0; Hola, bienvenido a Yoigo. Estamos grabando esta llamada.
12.0;15.0; Un momento, no te retires, te paso con un agente.
```

MASMOVIL

PyAnnote

- **PyAnnote** is a tool for **speaker diarization** that helps in **speaker recognition** in a conversation and establishes **start & end timestamps**.

```
start=2.8s stop=4.2s speaker_SPEAKER_01
start=5.9s stop=14.6s speaker_SPEAKER_01
start=14.6s stop=19.2s speaker_SPEAKER_00
start=20.0s stop=20.3s speaker_SPEAKER_00
start=20.3s stop=30.2s speaker_SPEAKER_01
start=30.9s stop=36.9s speaker_SPEAKER_01
```

MASMOVIL

Workload

- We need to be able to process **all the calls of one day** in **less than 24 hours**

- **Estimated workload per day**:
  - Number of calls → **>100k calls**
  - Call time → **36k hours**

- **Estimated processing times (using GPUs)**:
  - **Whisper** (model Medium) -> **4x** (audio 10 min processed in 2m 30s)
  - **PyAnnote** -> **14x** (audio 10 min processed in 40s)

#bbs

MASMOVIL

Architecture

Call transcription using AI

#bbs

MASMOVIL

Architecture principles

- **Processing parallelism**: It is **mandatory** to **parallelize processes** to be able to **get the work done in time**
- **Event driven architecture**: With this principle we can get **independent components decoupled**
- **Easy scaling up & down:** We don't want to have machines running with **no workload** (it's an unnecessary **waste of money**)

#bbs

MASMOVIL

Technologies

- **GKE Cluster**
- **Deployments written in Python**
- **GCS Buckets (with Pub/Sub notifications)**
- **PubSub**
- **Airflow**

#bbs

MASMOVIL

Hardware specifications

- **Preemptible** & **Guaranteed** instances **n1-standard-4** (4vCPU 15GB)
- **GPU Nvidia Tesla T4**
- Region: **europe-west1** (there is not GPUs available in europe-southwest1)

#bbs

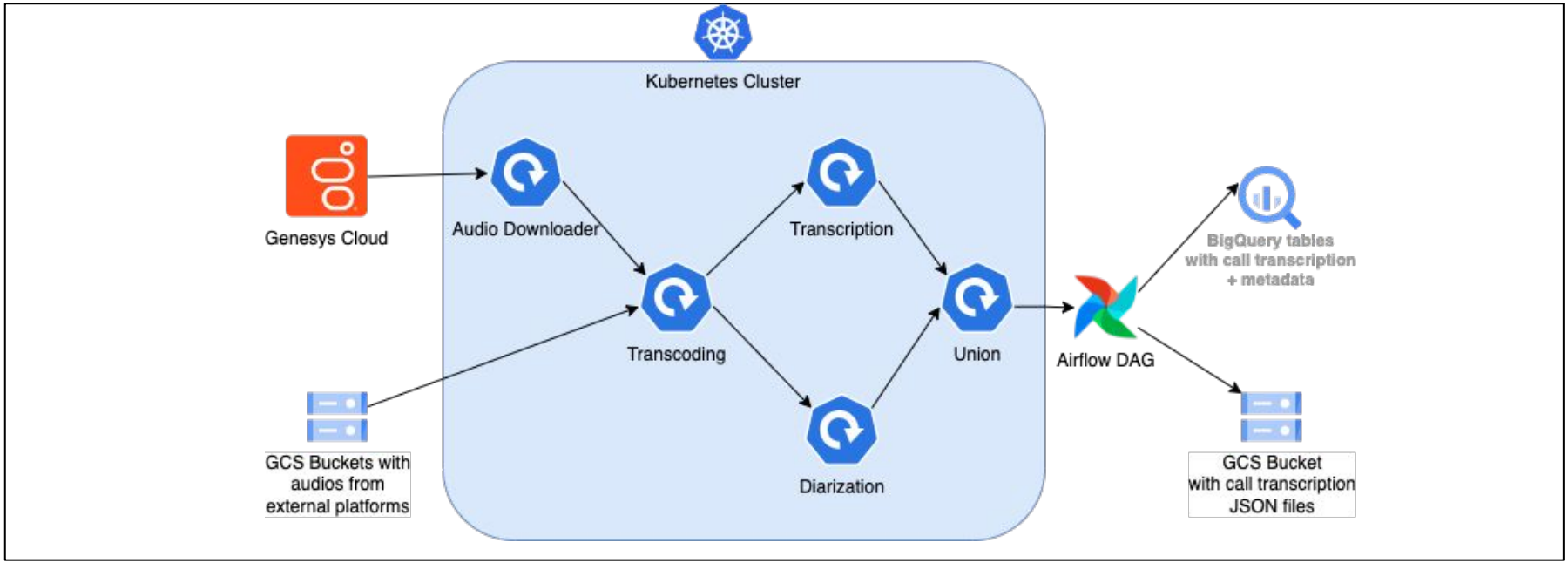
MASMOVIL

Component Design

- Every **component** of this architecture was designed as a **PubSub consumer** that consumes messages from a **PubSub topic**
- When a **file** is stored in a **GCS Bucket**, a new **message** is **published** into a **PubSub topic** and a consumer receives it
- This architecture allows us to have **multiple consumers decoupled** working **independently**

#bbs

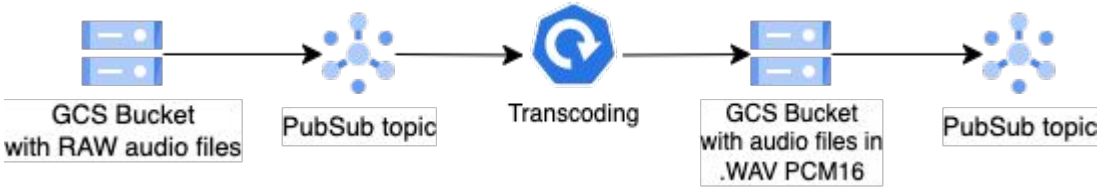
Call Transcription Architecture



MASMOVIL

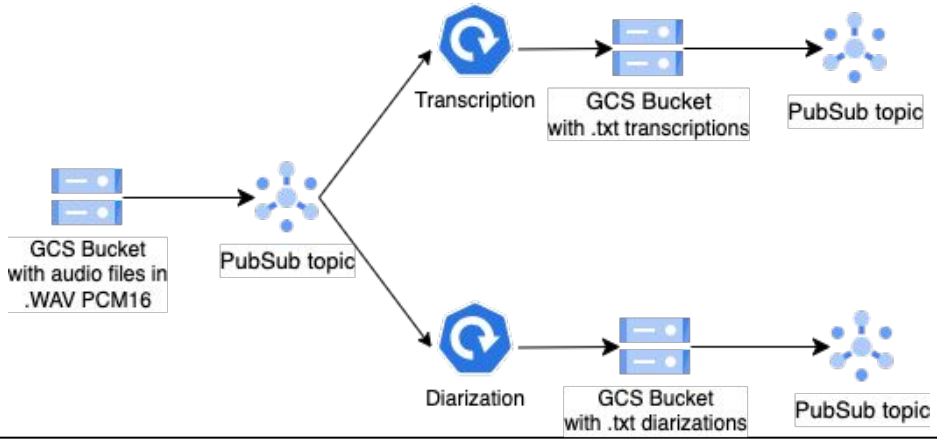
Transcoding

- This component receives a **PubSub message** when an audio files is saved into a **GCS bucket**
- The process uses **ffmpeg library** to convert the input audio into a .**wav audio** encoded in **PCM16 encoding**



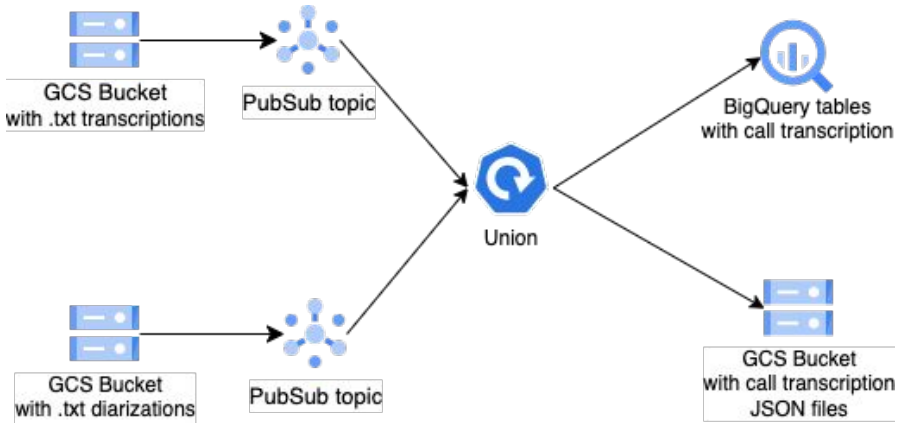
Transcription & Diarization

- **Transcription** process uses **Whisper** to convert .wav audio files into **.txt files** with the **call transcription**
- **Diarization** process uses **PyAnnote** to convert .wav audio files into **.txt files** with the **call diarization**



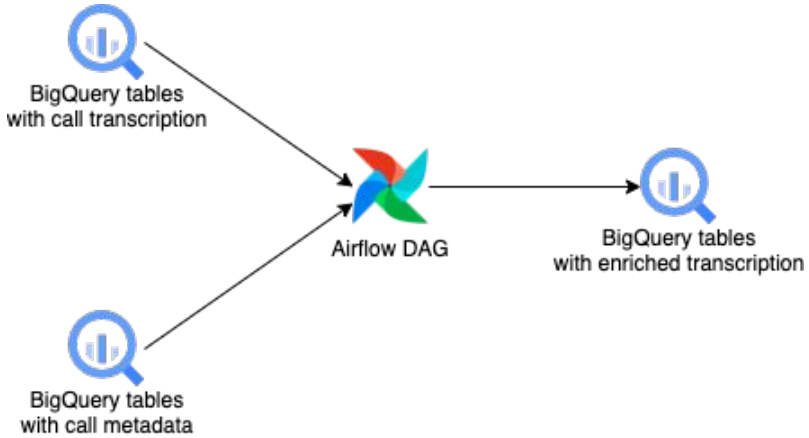
Union

- Union component **joins** the **transcription** and **diarization** data into a **final product**
- Its **output** is a **JSON file** per conversation, as well as a new element in a **BigQuery table**



Airflow DAG

- Finally, the information generated in the union process is **enriched** with **extra information**
- This process consists on many **SQL scripts** orchestrated in an **Airflow DAG**
- The result is the **call transcription** ready to be consulted by the multiples **areas** of the **company**



MASMOVIL

Problems & Solutions

Call transcription using AI

#bbs

Scale pods up & down automatically

- To solve this problem, we have decided to use **k8s** feature **HPA** with metric ‘targetCPUUtilizationPercentage’
- HPA (HorizontalPodAutoscaler) automatically **updates the pods** of the **deployments** with the aim of **automatically scaling the workload** to match **demand**
- It allows us to **parallelize processes** and get the **work done in time**

MASMOVIL

Stuck PubSub consumers

- Sometimes we have found some **pods** running **PubSub consumers** that are **not consuming** any message from its **PubSub Subscription**
- This leads to a **waste of hardware** and a **decrease of speed processing**
- To solve this problem, we have decided to use the **k8s** feature **livenessProbe**, checking periodically a dummy file in the pod. If this file is not updated since a period of time, the pod is **automatically restarted**

MASMOVIL

High costs on GPUs

- To reduce costs, we have applied two solutions:
  - **Faster Whisper** (3x times faster than normal Whisper)
  - **Time-Sharing** (two pods sharing one GPU)

MASMOVIL

Next Steps

Call transcription using AI

#bbs

MASMOVIL

Audio Preprocessing

- We have **some problems** with **audio quality**:
  - **IVR music** makes **Whisper hallucinations**
  - **Bad quality** of audio calls

- We have to work on **audio preprocessing** to solve or **minimize** this **problems** and achieve the **best quality** of the **transcriptions**

MASMOVIL

Participant Identification

- In one call, can be **2 or more participants**, and **identify** the number of participants in a call is **not trivial**
- We are using **PyAnnote** with **min and max thresholds** of number of **participants**, but we are searching for a better solutions to **exactly identify** the **number or participants**

#bbs

MASMOVIL

Change HPA Metric

- Actually we are using the **HPA** metric **'targetCPUUtilizationPercentage'**, that is based on **CPU utilization**
- We want to use a **more specific metric**, related to **unacked messages** in a **PubSub subscription**

#bbs

MASMOVIL

DLP

- Call transcriptions can have **personal data** in its content (IDs, street address…) so this could result on a **data leak** when these transcriptions are shared with **another areas** of the company

- We are working on a solution of **anonymization** and **tokenizing** these transcriptions using **Google DLP**

#bbs

MASMOVIL

Use Cases

Call transcription using AI

#bbs

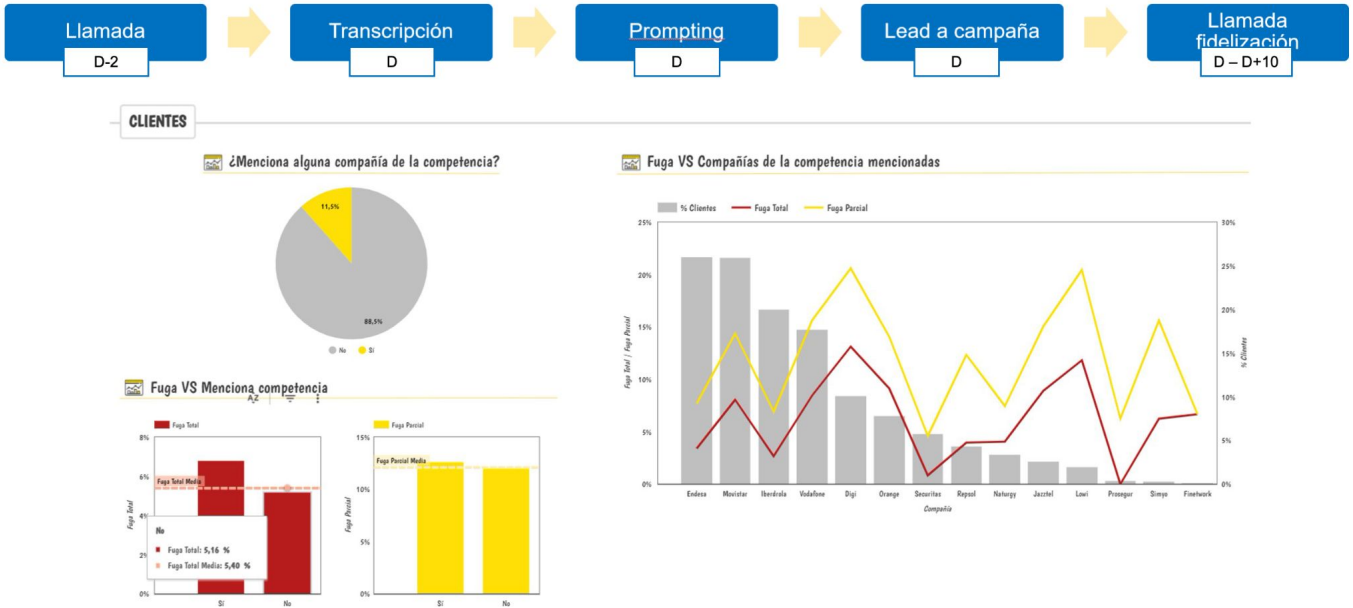
MASMOVIL

Generation of leads in fidelization Campaigns

- Generation of **leads** that are used on **fidelization campaigns** using **GenAI**
- This use case works with **transcriptions** to get:
  - **Sentiment Analysis**
  - **Problem identification**
  - **Reference to competitors companies**

#bbs

Generation of leads in fidelization Campaigns

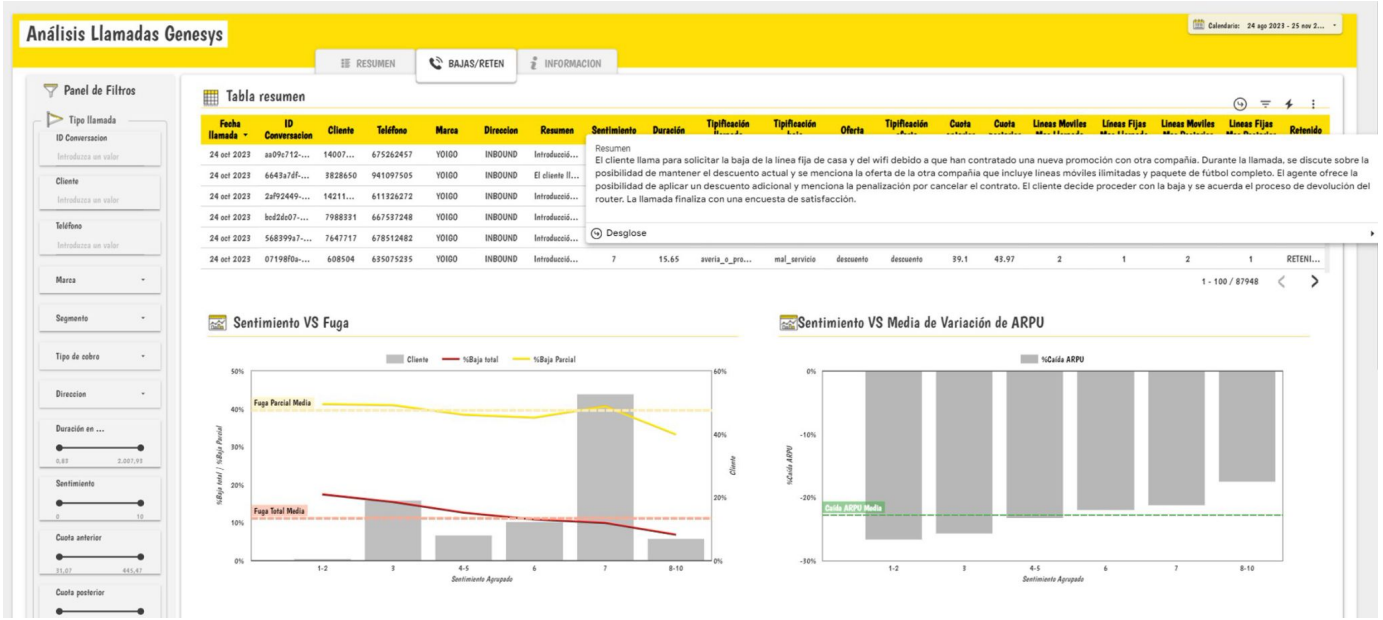


#bbs

Monitor Commercial Activity on retention calls

- Generate **KPIs** automatically to **monitor commercial activity** in **retention calls** (with no human interaction) using **GenAI**
  - **Dashboard** with **summary, sentiment analysis** and **KPIs** associated with the call
  - Automatically **typification of calls**

Monitor Commercial Activity on retention calls



#bbs

Monitor Commercial Activity on retention calls

#bbs

MASMOVIL

Q & A

Muchas gracias!!

#bbs