



**EBU**

OPERATING EUROVISION AND EURORADIO

# **MCMA WORKSHOP AT BLOOMBERG**

ALEXANDRE ROUXEL

13 FEBRUARY 2020– NYC

# EBU

OPERATING EUROVISION AND EURORADIO

## MERGING SERVICES

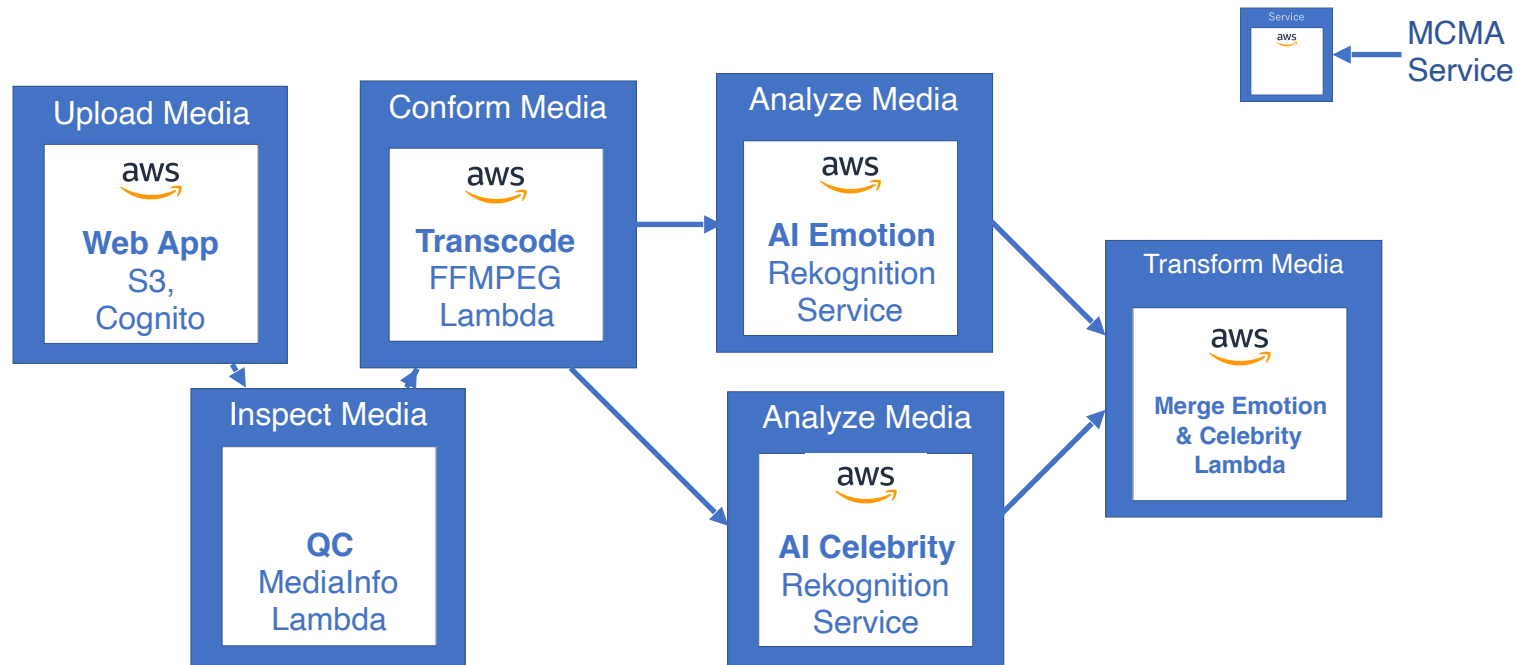
EMOTIONS AND CELEBRITIES





# EMOTIONS AND CELEBRITIES DETECTION

## > Workflow overview



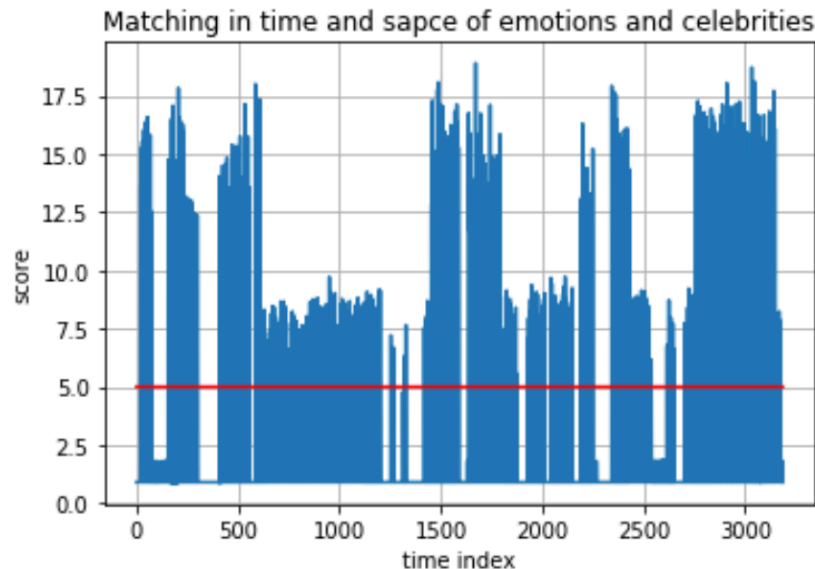


# MERGING CELEBRITIES AND EMOTIONS

- › The objective is to associate emotions to celebrities:
  - › On AWS it is two different services : Emotions of Face detection and Celebrity recognition
- › But these two services use two different algorithms to detect faces and personalities:
  - › The bounding boxes and the time stamps of Celebrity don't match the ones of Emotions, bounding boxes can be faces or persons.
- › We merge the two outputs with a spatio-temporal match filter approach
- › We compute a PyChar representing the average emotion per celebrity in videos

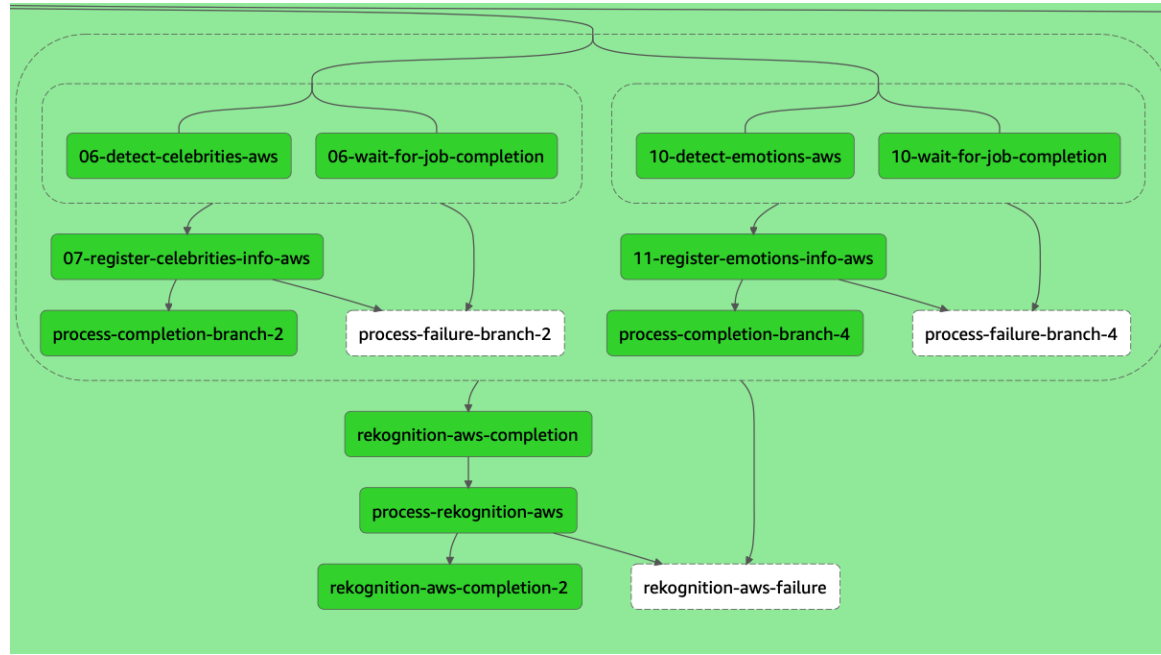
# MERGING CELEBRITIES AND EMOTIONS

- › The algorithm computes a global score made of :
  - › Intersection area of bounding boxes
  - › Difference between time stamp of the bounding boxes
- › A threshold is applied to associate (or not) an emotion to a celebrity at a time  $t$ .



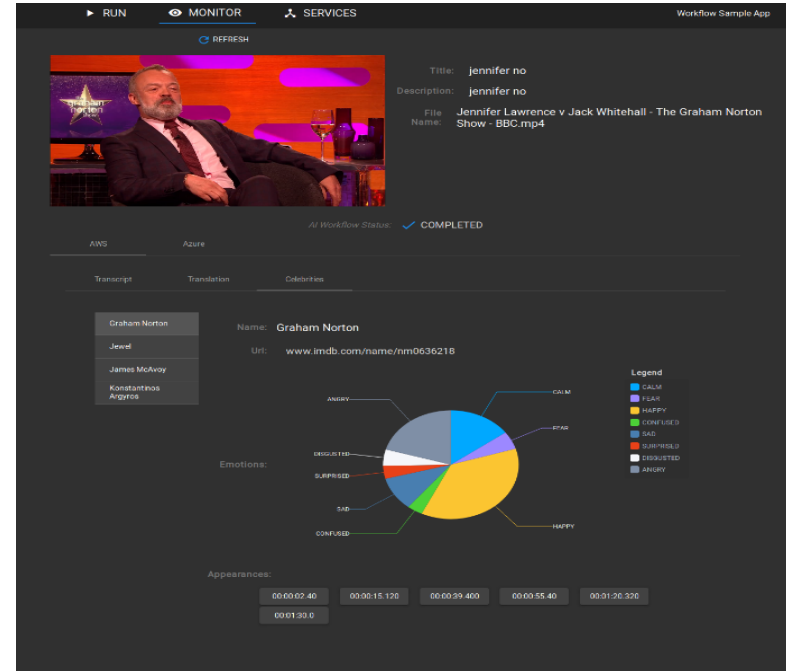
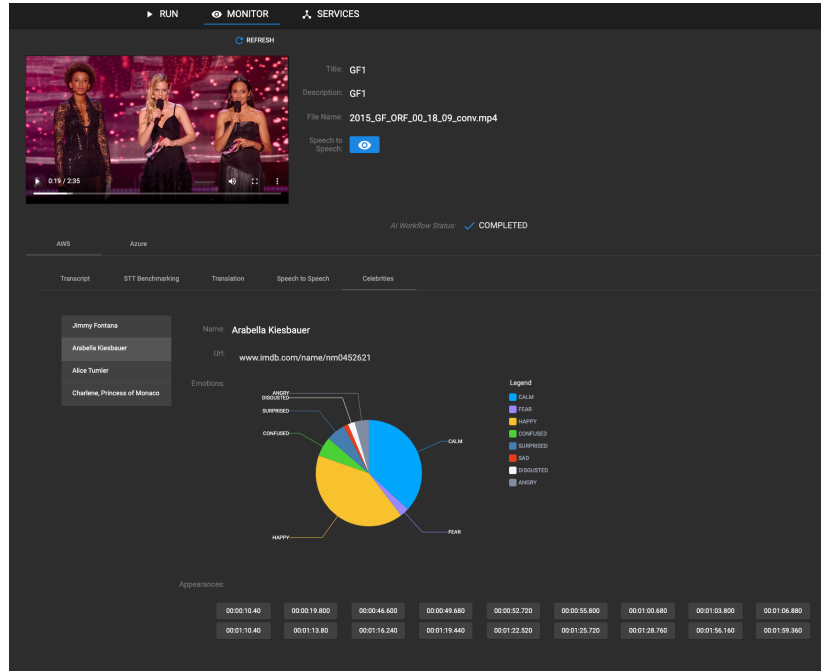


# STEP FUNCTIONS





# IBC 2019 DEMO





# WHAT I HAVE LEARNT

- › General principles on
  - › Serverless cloud computing, cost effectiveness, FaaS.
  - › Micro service architecture for Media contents, how to manage the timing constraints, merging services outputs
- › MCMA architectures
  - › Common Service Pattern : REST interface, API Handler, Worker, DB
  - › Service instantiation : Job repo; Job Processor, Service Repo, Service
- › How to create a workflow with MCMA



# EBU

OPERATING EUROVISION AND EURORADIO

## ANNEX





# HOW TO CREATE A WORKFLOW

- › Step 1 : Register a new service
  - › Add a new job profile in the post deployment script
    - › mcma-projects/multi-cloud-ai-workflow/deployment/post-deploy/src/index.js
  - › Deploy and check that the new job profile and service have been added in the logs
  - › Service registry : two end points one to get the job profile, one to get the service
    - › Invoke the api and check the interface with Postman
- › Step 2 : Do the implementation : a simple exemple
  - › In mcma-projects/multi-cloud-ai-workflow/services/aws-ai-service/worker/src/index.js
    - › Add profile in WorkerBuilder and Handler
  - › In mcma-projects/multi-cloud-ai-workflow/services/aws-ai-service/worker/src/profiles/detect-celebrities.js
    - › Define the job profile, do the same for emotion
- › Step 3 : Modify the workflow : add a merging step
  - › In multi-cloud-ai-workflow/workflows/ai/20-rekognition-aws/src/index.js
  - › Declare the new .tf



## TWO MAIN FILES

- › In multi-cloud-ai-workflow/services/aws-ai-service/worker/build/staging/profiles/ and mcma-projects/multi-cloud-ai-workflow/services/aws-ai-service/worker/src/profiles/process-reko-results.js to :
  - › Call the services RekognitionGetCelebrityRecognition and RekognitionGetFaceDetection, iterate over 1000 time-stamps and write the results in S3
- › multi-cloud-ai-workflow/workflows/ai/20-rekognition-aws/src/index.js to :
  - › Get the files on S3 and process the score