ArTeller

The future of audio-guide

Marco Cerino 1755069
Giuliano Martinelli 1915652
Mattia Pocci 1688770

https://github.com/g185/ArTeller

What is ArTeller?

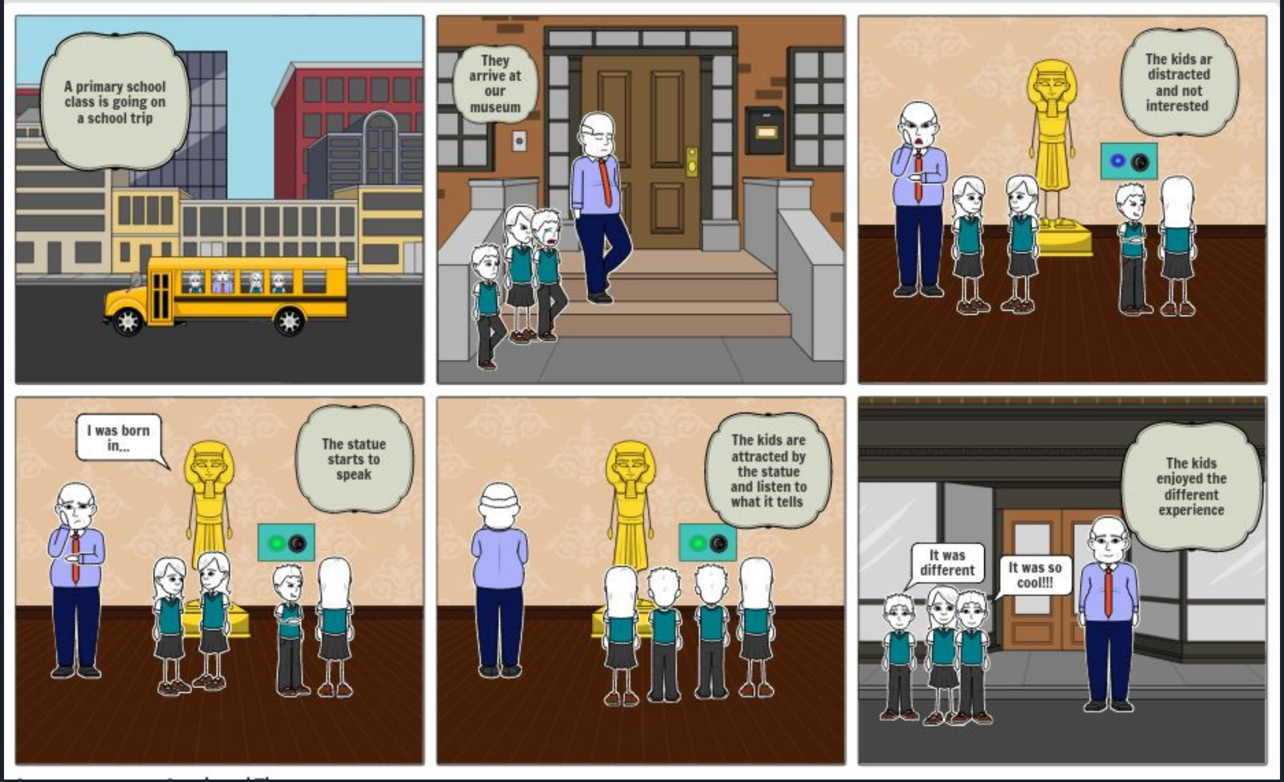
The aim of ArTeller is to provide a free audio-guide service to the Sapienza Museum visitors, while providing useful insights to the museum manager.

This service will make the museum visit more interesting and interactive, particularly for students and kids.

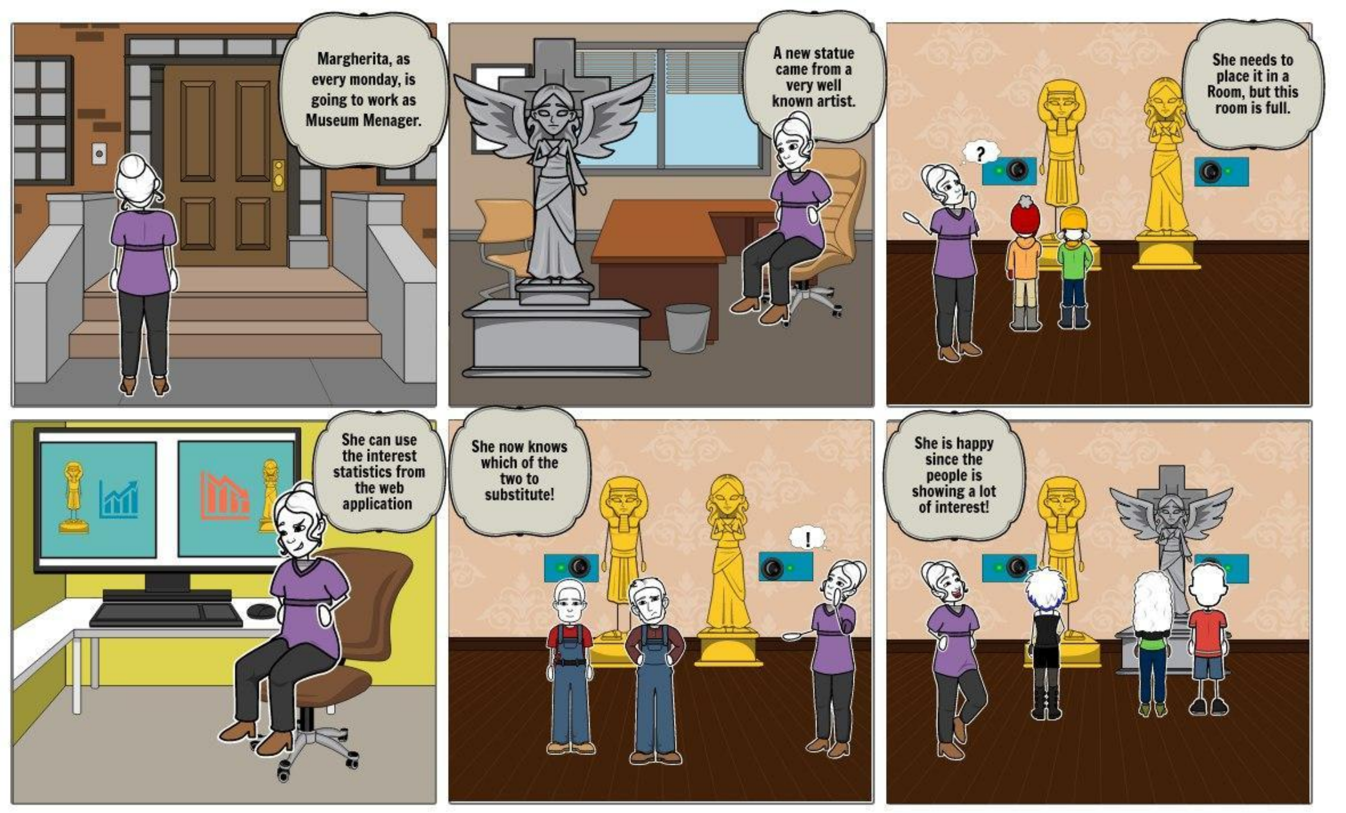
The idea consists in locating a device in proximity of each statue which will look for interested people through face detection. Once it has detected one or more users, it will start to play the audio description for the given opera.

A web page will be accessible by the museum manager in order to retrieve meaningful statistics about the different engagement rates of the operas.

Scenario: School trip at the museum


A primary school class is going on a school trip
They arrive at our museum
The kids ar distracted and not interested
I was born in...
The statue starts to speak
The kids are attracted by the statue and listen to what it tells
It was different
It was so cool!!!
The kids enjoyed the different experience

Scenario: The importance of insights


Margherita, as every monday, is going to work as Museum Menager.
A new statue came from a very well known artist.
She needs to place it in a Room, but this room is full.
?
She can use the interest statistics from the web application
She now knows which of the two to substitute!
!
She is happy since the people is showing a lot of interest!

User Experience Evaluation

To evaluate the user experience we will use a review based method on a questionnaire that will be handed to visitors at the end of the museum visit. Some heuristics we will consider are:

1. The system is prevedible
2. The system is consistent
3. The system gives appropriate feedbacks

Technical Evaluation

In order to test our solution, since we cannot have access to real devices due to the Covid-19 emergency, we will proceed in this way:

1. **Face Recognition**: we will test our face recognition technology using the webcams on our laptops.
2. **MQTT architecture**: we will test the MQTT architecture using Iot-Lab
3. **Cloud infrastructure**: deployed on AWS

Architecture

Overall Project Architecture

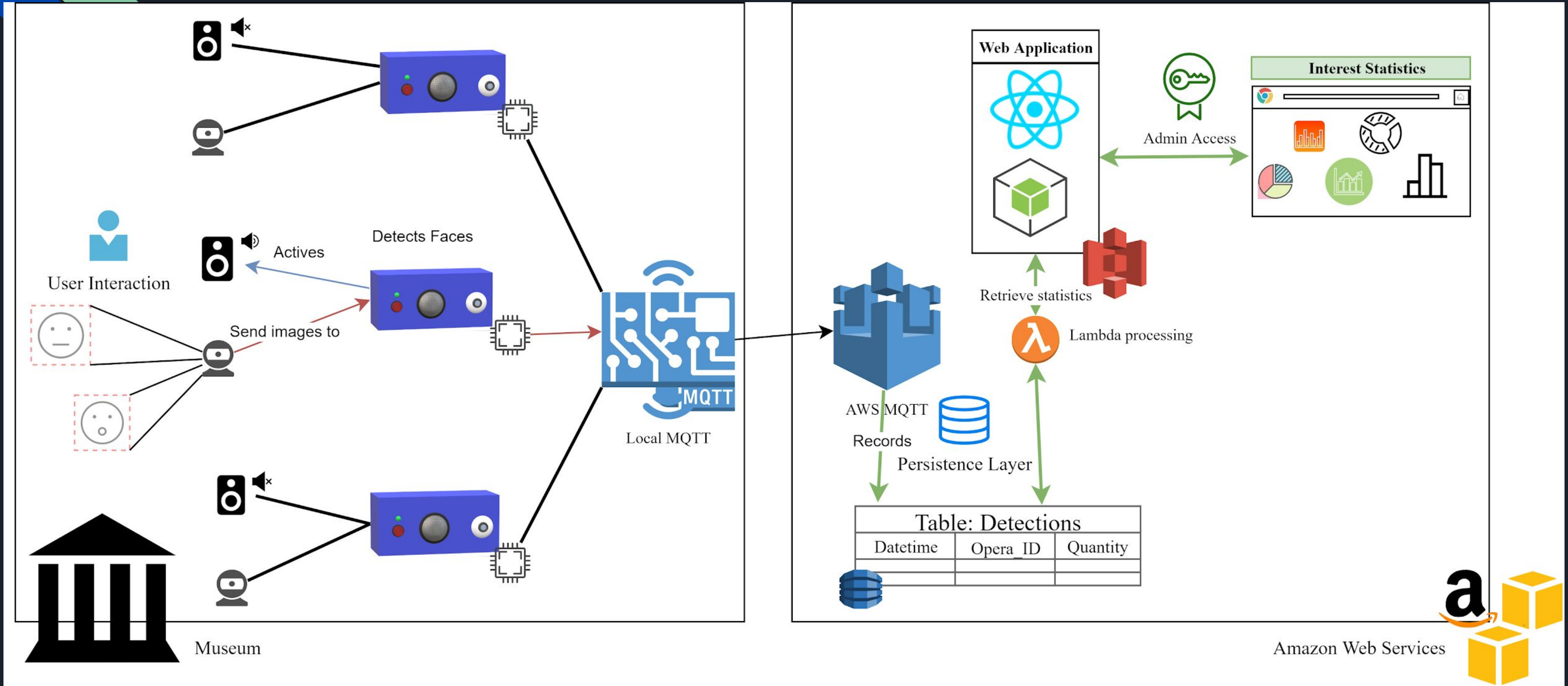
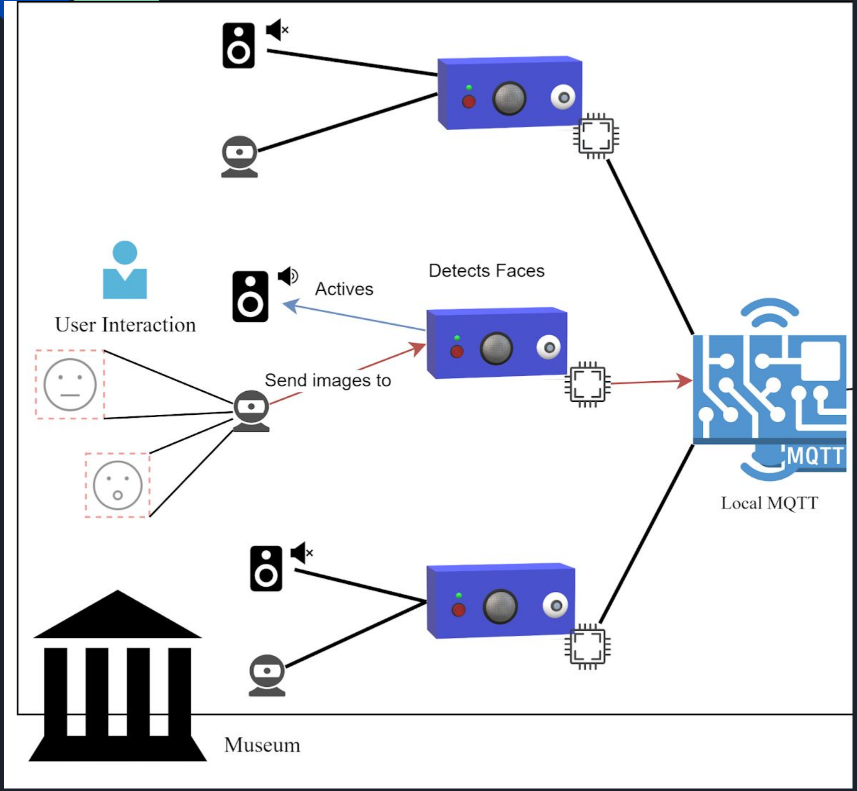


Table: Detections

Datetime	Opera_ID	Quantity

Edge IoT Network



- **Camera sensor:** sends images to the device
- **Speaker:** produces audio tracks
- **Board:** processes face detections and MQTT exchanges
- **Local MQTT broker:** manages mqtt messages from broker and to AWS cloud

Cloud Back-end

- **AWS IoT Core**: receives MQTT messages and forward them to the persistence layer

- **DynamoDB**: stores detection events

- **Lambda Functions**: processes statistics using DB data

- **Dashboard**: outputs statistics and insights

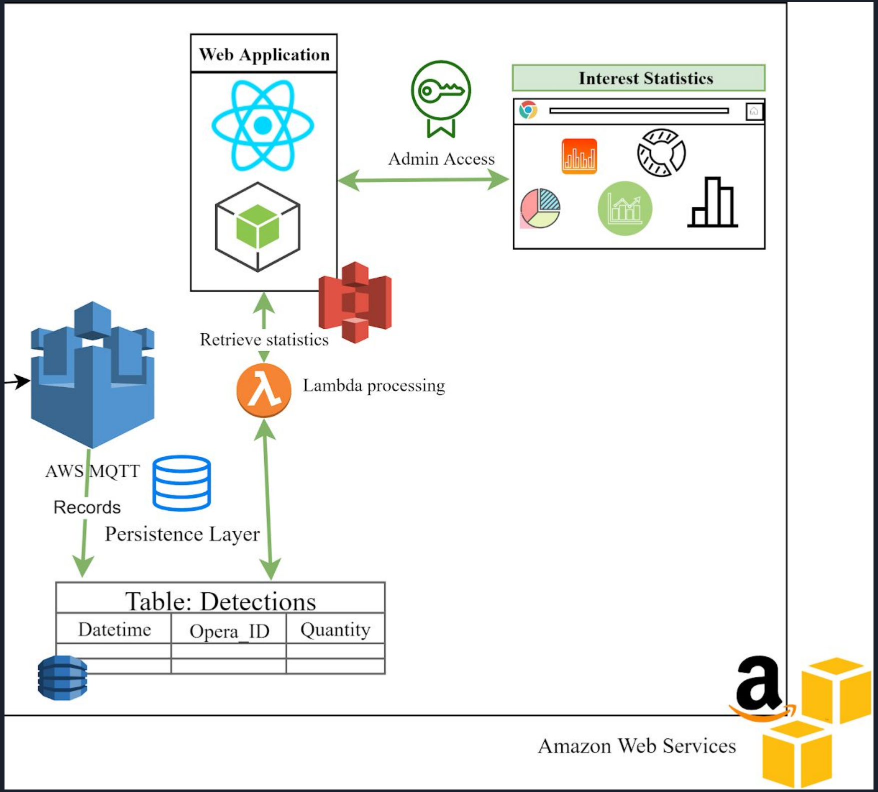


Table: Detections

Datetime	Opera_ID	Quantity

Focus: Face Detection

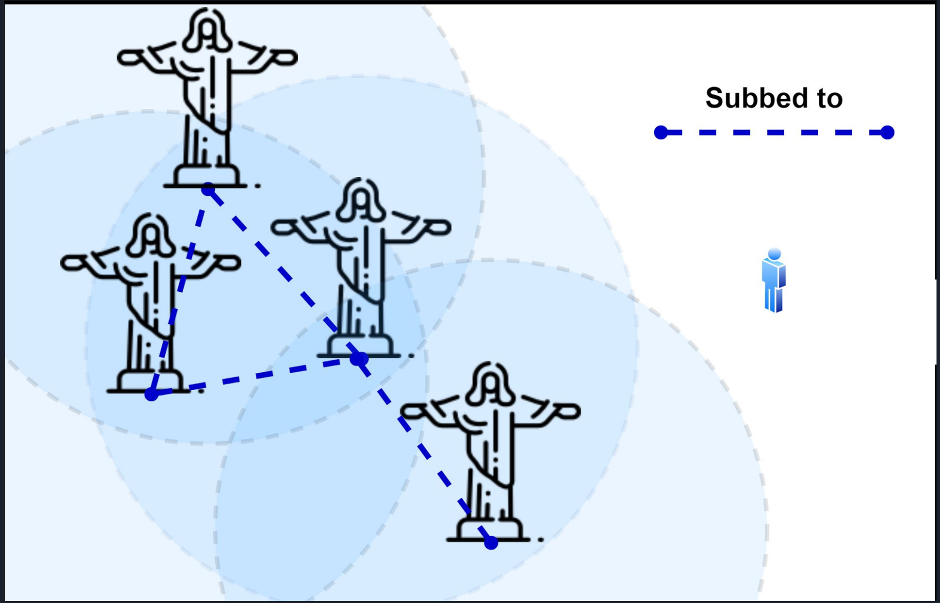
In the implementation of Face Detection we will have to put a particular attention in order to avoid false positives.

In order to avoid uninterested people to trigger the audio-description we will analyze:

- **Distance from the sensor:** face detection of people that is far from the statue is not a good trigger

- **Time in the frame**: people that show interest in the opera will watch it with curiosity

Focus: priority management



Audio interference from nearby statues could be a big problem

We would like to implement a priority queue locally for each device, that will listen to the nearby devices.

When a device detects a face, it will first make sure nobody is already talking

Overall Project Architecture

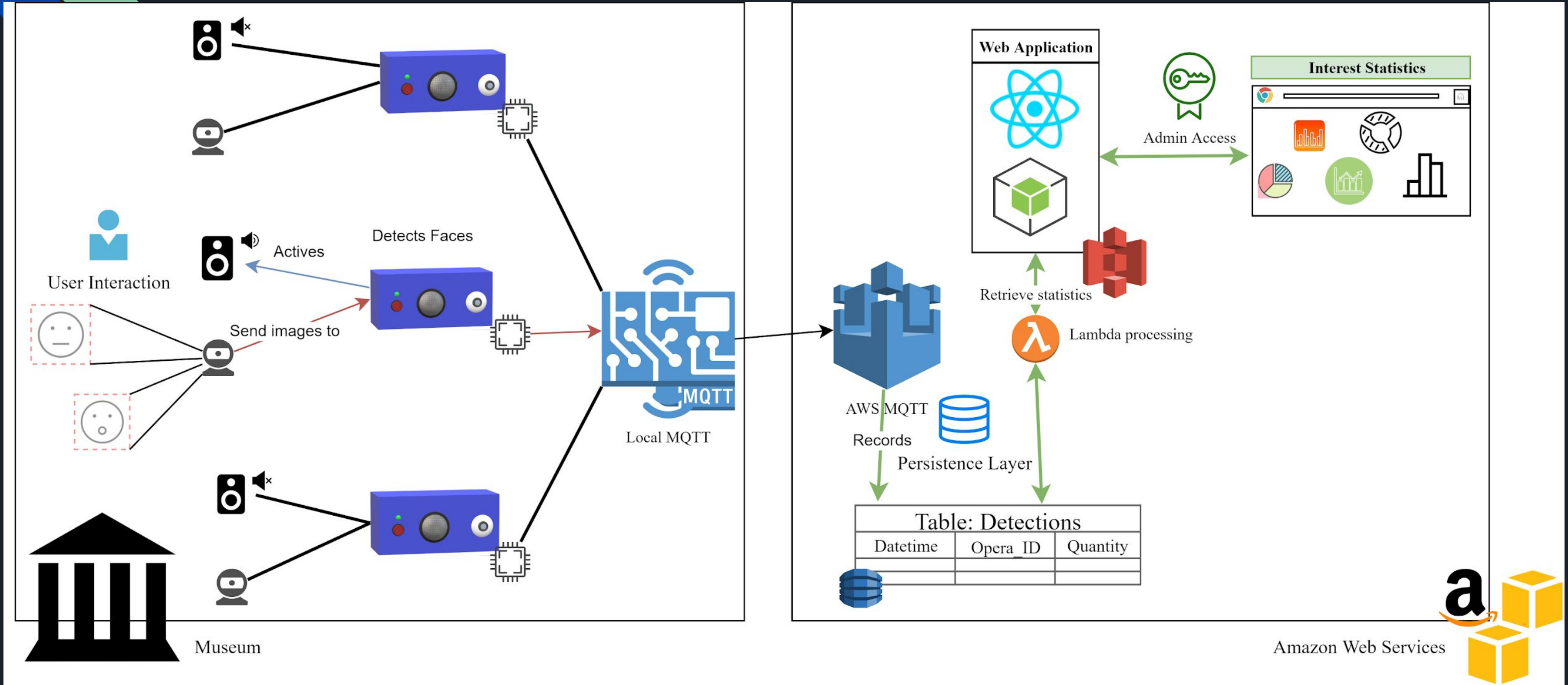


Table: Detections

Datetime	Opera_ID	Quantity