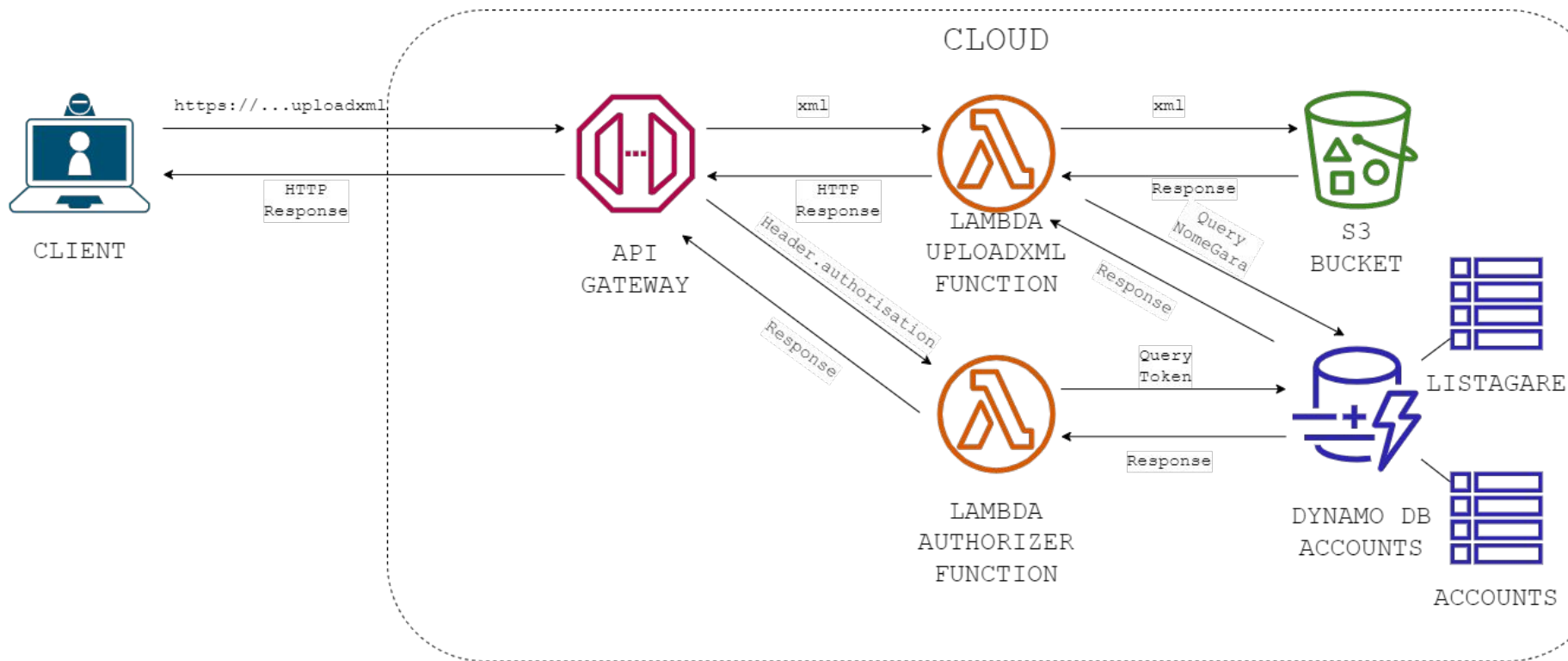


JQuelli: Capelli Luca 1064893
Ragosta Stefano 1064527
Vedovati Matteo 1064586

Architettura Cloud



Client: Esegue un upload dell'xml e delle credenziali (contemporaneamente)

API Gateway (uploadAPI): Interfaccia per il metodo POST in /uploadXML

Lambda Function (Authorizer): Legge il campo '*Authorization*' nell'header della richiesta, contenente il token di autorizzazione, e lo confronta con quelli presenti nel database, se non ha corrispondenza l'upload dell'xml verrà rifiutato. I token vengono generati con Basic Auth.

Lambda Function (uploadXML): Prende l'xml in upload, esegue una convalida secondo le direttive del file .xsd dello standard IOF presente su github, aggiunge la gara all'interno del database e salva il file xml con un nome univoco. In caso fosse già presente la gara permette il suo aggiornamento (se non già finita).

Bucket S3 (xmlresults): Contiene i file xml contenenti i dati

DynamoDB: Contiene 2 tabelle:

 '**ListaGare**', con al suo interno le informazioni delle gare, data e ora di inizio e fine, nome della gara, l'URI del file salvato nel bucket S3 e id univoco.

 '**Accounts**' con al suo interno i token per l'autenticazione

Funzionamento di uploadXML (Lambda Function)

Nella fase di caricamento del file XML della gara, la funzione lambda, uploadXML, verifica che all'interno del DB non sia già stata registrata la gara. In presenza della gara si controlla che non sia ancora registrata la sua conclusione (presenza di data e ora di fine gara), in caso non vi sia, si aggiorna il record e viene sovrascritto il file XML della gara presente nel bucket con quello aggiornato (N.B. non c'è controllo se il nuovo file ha più o meno informazioni). Altrimenti nulla verrà aggiornato e il nuovo XML verrà scartato.

Se il file non è presente viene inserito nel bucket e registrato sul DB.

Parte dell'utente (Client)

L'utente che vuole eseguire l'upload dell'xml deve caricare il file, inserire username e password, e selezionare il numero di minuti trascorsi dall'inizio dell'evento (per simulare l'invio dei dati raccolti fino a quel momento). Verrà inviata dunque una richiesta POST HTTP contenente i dati raccolti al server.

Strumenti utilizzati:
JavaScript (jQuery)
Ajax



The screenshot shows a web form with the following elements:

- A file selection button labeled "Scegli file" followed by the filename "gara.xml".
- An "Upload Xml file" button.
- Input fields for "User:" and "Pass:".
- A text label: "l'evento è durato un totale di 240 minuti".
- A label "selezionare il numero di minuti passati dall'inizio della gara:" followed by a text input field containing the value "0".
- An "invia i dati" button at the bottom.