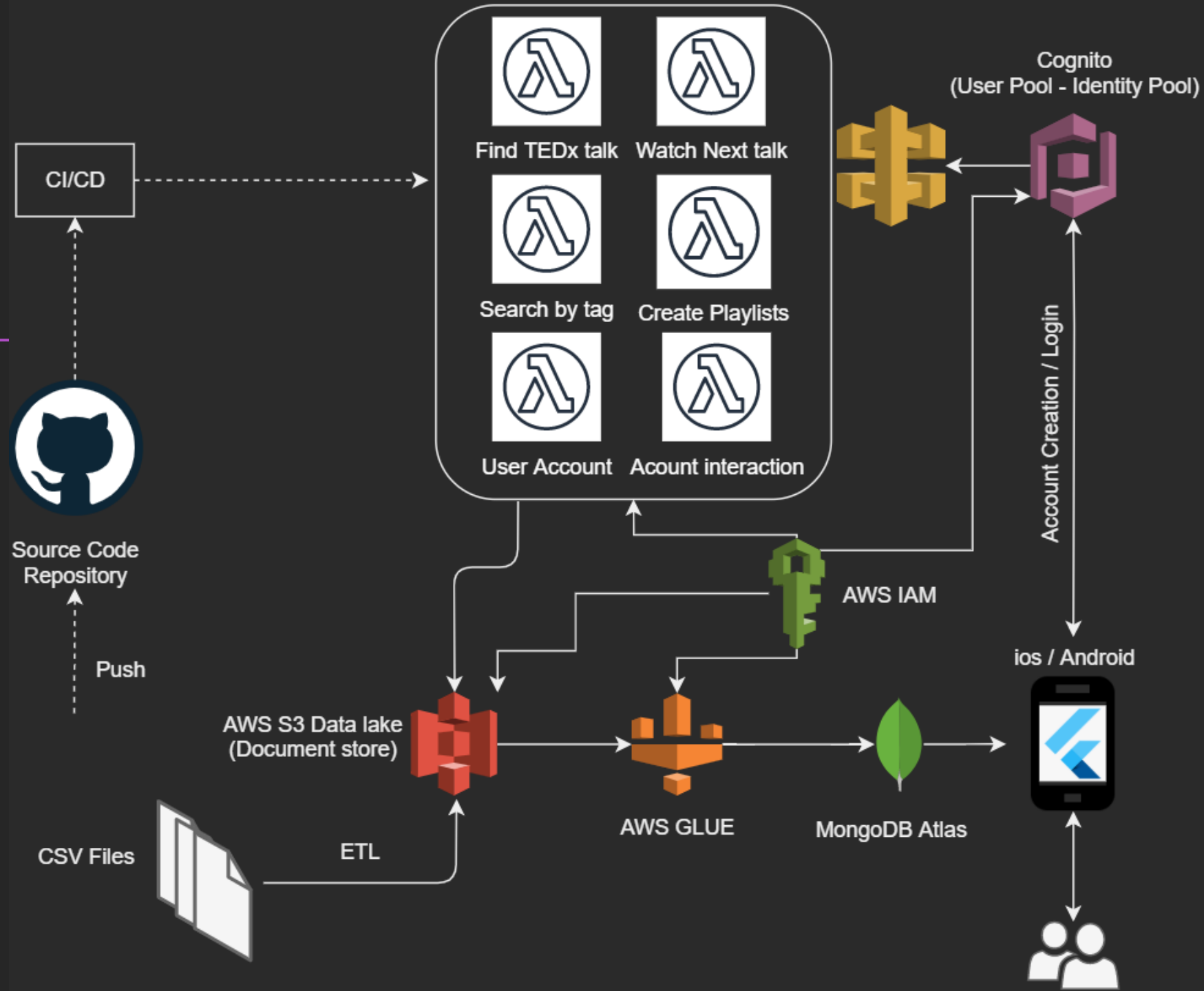


MyTEDx



# SCENARIO



# RELEASE PLAN

0.1

Define requirements

Design system

+ Aggiungi un'altra scheda

0.2

Build the development environment

test the environment

+ Aggiungi un'altra scheda

0.3

Build infrastructure

Build APIs

0/4

+ Aggiungi un'altra scheda

0.4

Build user interface

+ Aggiungi un'altra scheda

0.9

Testing

+ Aggiungi un'altra scheda

1.0

RELEASE

+ Aggiungi un'altra scheda

# OBIETTIVI

- Implementare ricerca video per titolo
- Implementare suggerimento prossimo video attinente
- Implementare ricerca video per tag
- Implementare possibilità di creare playlist
- Implementare interazione tra account
- Tema scuro

# UTENTI

---

# VALORE

MyTEDx si propone come alternativa all'app ufficiale di TED Conferences LLC 'TED' (20M di download)

MyTEDx punta sulla migrazione di utenti in cerca di un'alternativa:

- 'easy on the eyes'
- con possibilità di creare più playlists
- con possibilità di interazione tra account

# CRITICITA'

---



L'ALTA REATTIVITA' è una caratteristica garantita da una gestione efficiente degli algoritmi di ricerca e da un'interazione efficace con il database.



INTERFACCIA SEMPLICE ED INTUITIVA: Il design user-friendly dell'interfaccia di MyTEDx la rende intuitiva e di semplice utilizzo. L'aspetto lineare e pulito permette un comodo e piacevole utilizzo e previene potenziali errori da parte dell'utente.



INTEGRAZIONE DATI AGGIORNATI : Il DataBase permette una comoda integrazione dei dati e di conseguenza la facile possibilità di aggiornamento degli stessi.



GESTIONE ACCOUNT: La gestione degli account impedisce il sovraccarico dei profili, inserendo alcune limitazioni nella condivisione e nel salvataggio di video. Gli account inattivi per un periodo di tempo prolungato vengono eliminati.

# Parte 2

A thin, vertical white line is positioned to the right of the text 'Parte 2', extending from the top of the text area down to the bottom of the slide.

# Aggiunta watch\_next\_dataset

Lo script del job pyspark è stato aggiornato con le seguenti righe di codice:

## **## READ WATCH\_NEXT DATASET**

```
ted_watch_next_dataset_path = "s3://pirotta-bucket-mytedx/watch_next_dataset.csv"
watch_next_dataset = spark.read.option("header","true").csv(ted_watch_next_dataset_path)
```

## **# ADD WATCH\_NEXT TO TEDX\_DATASET**

```
watch_next_dataset_agg = watch_next_dataset.groupBy(col("idx").alias("idx_ref2")).agg(collect_list("url").alias("url_next"),collect_list("watch_next_idx").alias("id_next"))
watch_next_dataset_agg.printSchema()
tedx_dataset_agg = tedx_dataset_agg.join(watch_next_dataset_agg, tedx_dataset_agg._id == watch_next_dataset_agg.idx_ref2, "left") \
    .select(col("*")) \
    .drop("idx_ref2") \
tedx_dataset_agg.printSchema()
```

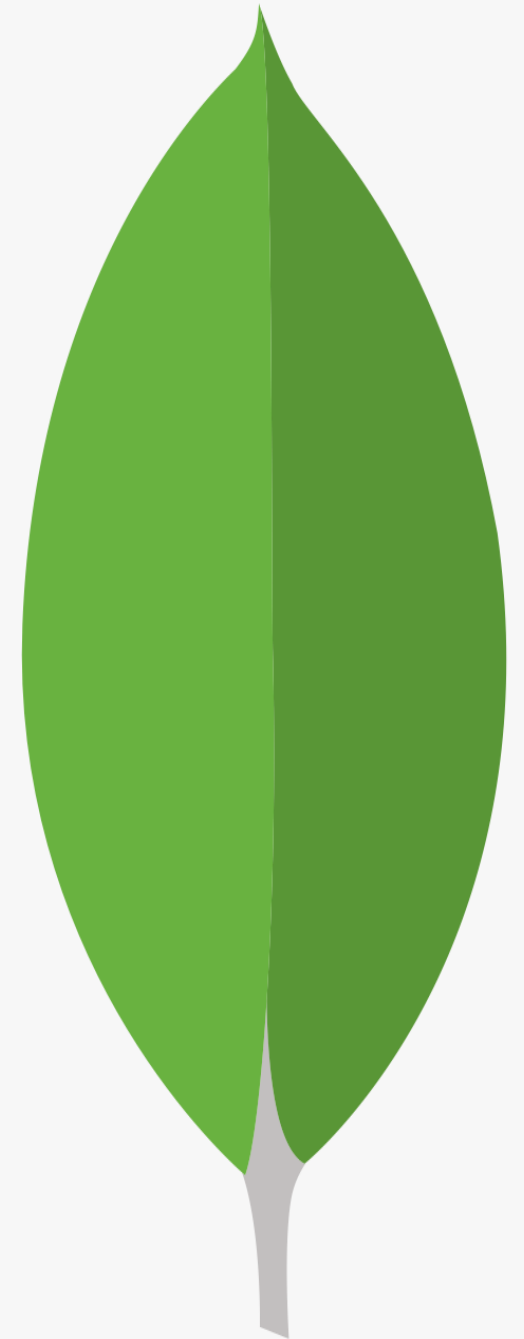


# Collection risultante

---

Lo script è stato modificato al fine di eseguire un ulteriore join con il dataset contenente i video consigliati. Lo schema risultante in MongoDB è il seguente:

<i>_id:</i>	id del video in riproduzione
<i>main_speaker:</i>	relatore del talk in riproduzione
<i>details:</i>	dettagli sul video in riproduzione
<i>posted:</i>	data di pubblicazione del video in riproduzione
<i>url:</i>	url del video in riproduzione
> <i>tags: Array</i>	tags del video in riproduzione
> <i>url_next: Array</i>	url dei video consigliati
> <i>id_next: Array</i>	id dei video consigliati



# Criticità tecniche

---

Sintassi del  
linguaggio query  
Pyspark

Difficoltà nel testare  
lo script causa  
tempi di avvio  
dell'architettura  
cloud prolungati

# Possibili evoluzioni

---



RICERCA VIDEO IN BASE  
A DATA PUBBLICAZIONE



RICERCA VIDEO IN BASE  
A RELATORE

# Parte 3

A thin, vertical white line is positioned to the right of the text 'Parte 3', extending from the top of the text area down to the bottom of the slide.

# Get\_Next\_By\_Url

---

Abbiamo creato una nuova lambda function basandoci su Get\_Talk\_By\_Tag. E' stato necessario modificare il file **handler.js** per cambiare la natura della query

```
module.exports.get_by_url = (event, context, callback) => {
  context.callbackWaitsForEmptyEventLoop = false;
  console.log('Received event:', JSON.stringify(event, null, 2));
  let body = {}
  if (event.body) {
    body = JSON.parse(event.body)
  }
  // set default
  if(!body.url) {
    callback(null, {
      statusCode: 500,
      headers: { 'Content-Type': 'text/plain' },
      body: 'Could not fetch the talks. url is null.'
    })
  }
}
```

```
connect_to_db().then(() => {
  console.log('=> get_all talks');
  talk.find({url:body.url})
    .skip((body.doc_per_page * body.page) - body.doc_per_page)
    .limit(body.doc_per_page)
    .then(talks => {
      callback(null, {
        statusCode: 200,
        body: JSON.stringify(talks)
      })
    })
})
```

# Dati ottenuti

```
{
  "url": "https://www.ted.com/talks/ethan_lisi_what_it_s_really_like_to_have_autism",
  "doc_per_page": 10,
  "page": 1
}
```

Tramite Postman abbiamo testato il corretto funzionamento della funzione.

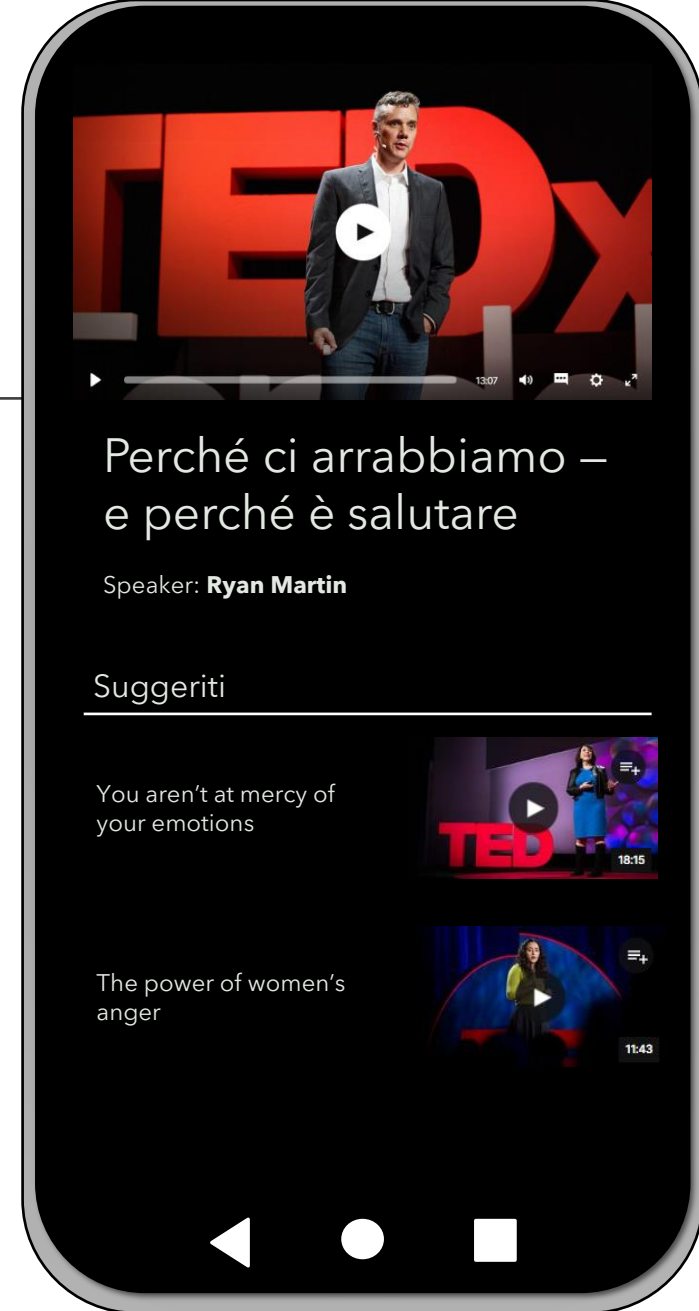
In base all'url passato, la funzione restituisce il JSON corrispondente, compreso di **url\_next**

```
{
  "main_speaker": "Ethan Lisi",
  "title": "What it's really like to have autism",
  "details": "\"Autism is not a disease; it's just another way of thinking,\" says Ethan Lisi. Offering a glimpse into the way he experiences the world, Lisi breaks down misleading stereotypes about autism, shares insights into common behaviors like stimming and masking and promotes a more inclusive understanding of the spectrum.",
  "posted": "Posted Apr 2020",
  "url": "https://www.ted.com/talks/ethan_lisi_what_it_s_really_like_to_have_autism",
  "tags": [
    "TED",
    "talks",
    "autism",
    "social change",
    "Autism spectrum disorder",
    "community",
    "diversity",
    "humanity",
    "TED-Ed",
    "inclusion"
  ],
  "url_next": [
    "https://www.ted.com/talks/steve_silberman_the_forgotten_history_of_autism",
    "https://www.ted.com/talks/steve_silberman_the_forgotten_history_of_autism",
    "https://www.ted.com/session/new?context=ted.www%2Fwatch-later",
    "https://www.ted.com/talks/rosie_king_how_autism_freed_me_to_be_myself",
    "https://www.ted.com/talks/rosie_king_how_autism_freed_me_to_be_myself",
    "https://www.ted.com/session/new?context=ted.www%2Fwatch-later",
    "https://www.ted.com/talks/alix_generous_how_i_learned_to_communicate_my_inner_life_with_asperger_s",
    "https://www.ted.com/talks/alix_generous_how_i_learned_to_communicate_my_inner_life_with_asperger_s",
    "https://www.ted.com/session/new?context=ted.www%2Fwatch-later",
    "https://www.ted.com/talks/ami_klin_a_new_way_to_diagnose_autism",
    "https://www.ted.com/talks/ami_klin_a_new_way_to_diagnose_autism",
    "https://www.ted.com/session/new?context=ted.www%2Fwatch-later",
    "https://www.ted.com/talks/wendy_chung_autism_what_we_know_and_what_we_don_t_know_yet",
    "https://www.ted.com/talks/wendy_chung_autism_what_we_know_and_what_we_don_t_know_yet",
  ]
}
```

# Esperienza utente

Tramite questa funzione è possibile reperire gli url dei video consigliati come successivi a quello al momento in esecuzione.

Tramite questi url vengono ottenuti titoli e anteprime dei video consigliati, che vengono esposti sotto al video in esecuzione. L'utente può selezionarli e passare alla visione del video suggerito.



# CRITICITA' TECNICHE

---



**UTILIZZO DI UN TOOL ESTERNO:** per il test del corretto funzionamento della funzione creata abbiamo dovuto utilizzare il tool Postman che offre la possibilità di eseguire la chiamata GET e passare la query JSON dopo l'inserimento del link di collegamento alla nostra API creata attraverso i servizi AWS.



**CODICE:** comprensione del codice, della sintassi e sviluppo della funzione. Comprensione del funzionamento delle Lambda function e della loro corretta implementazione.