

UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

DIPARTIMENTO DI INGEGNERIA "ENZO FERRARI"

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

Cloud computing: Amazon Web Services e distribuzione di una Django web application

CANDIDATO:

Davide Gesualdi

MATRICOLA: 119939

RELATORE:

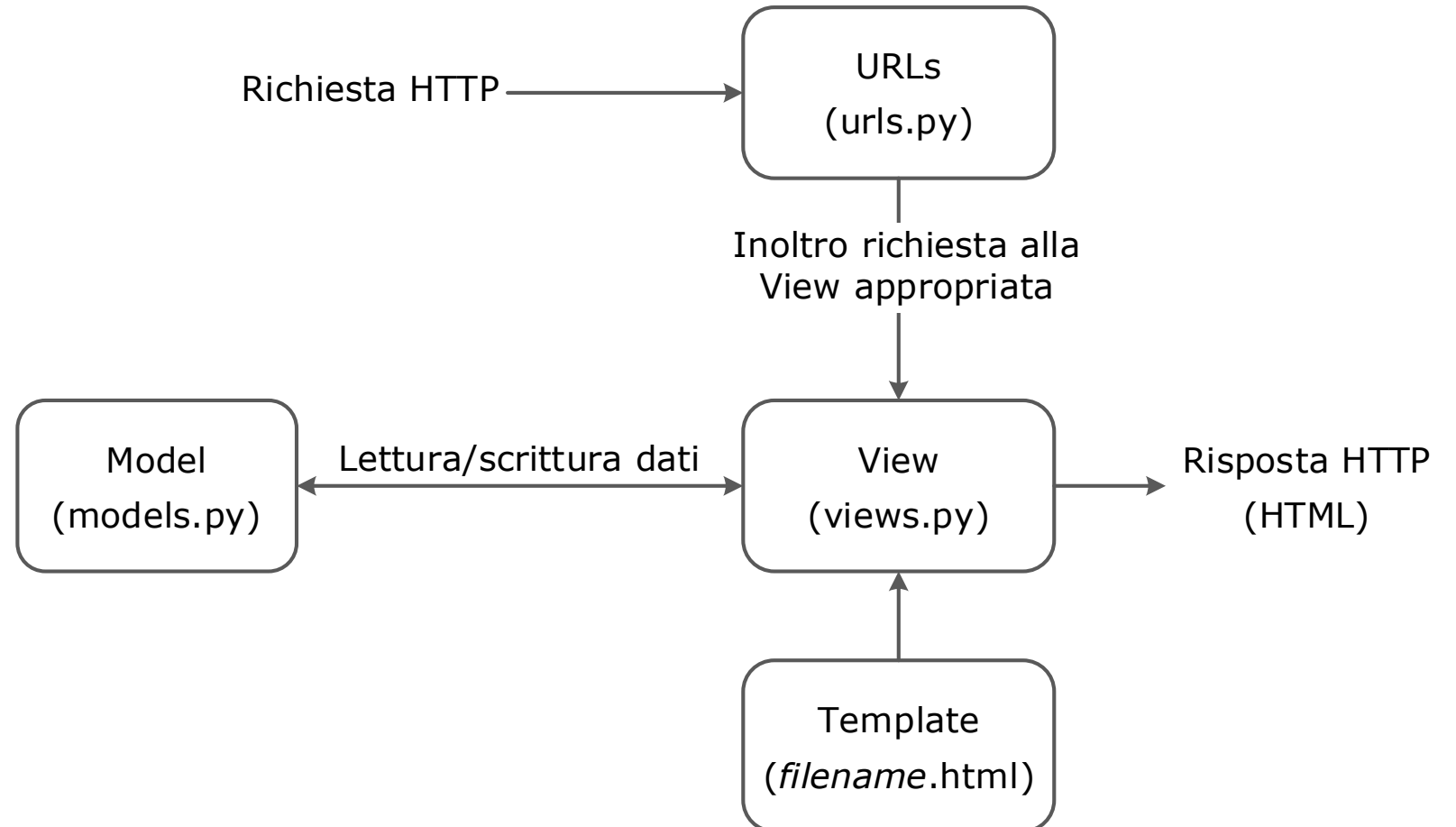
Prof. Nicola Bicocchi

ANNO ACCADEMICO 2019 - 2020

Obiettivi

- Realizzazione Django web application con un responsive web design.
- Distribuzione su Amazon Web Services (AWS) ed utilizzo dei servizi di cloud computing offerti.
- Elaborazione serverless dei video delle sessioni con AWS Lambda.

Model – Template – View (MVC)



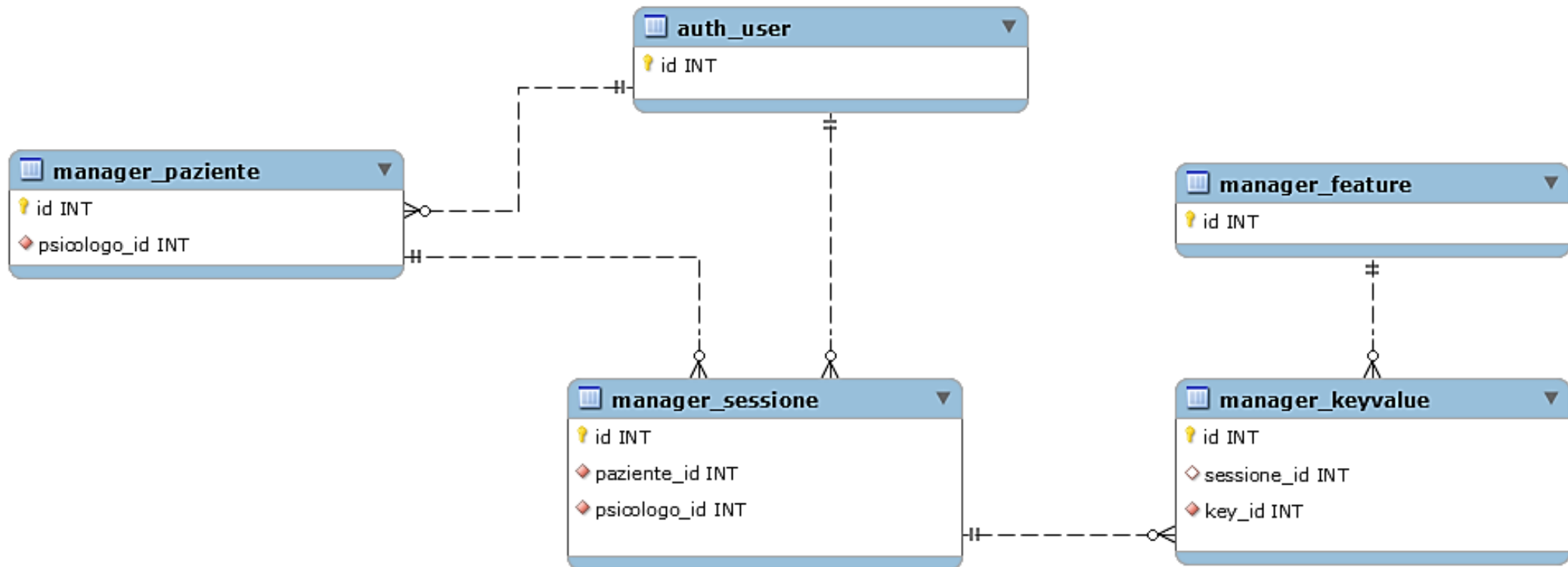
Struttura progetto Django

Il progetto si compone delle seguenti applicazioni:

- **accounts:** gestione account utente (psicologo).
- **core:** home page e cerca.
- **data_list:** elenco e dettagli dei pazienti e delle relative sessioni.

Struttura progetto Django

- **manager:** gestione pazienti e relative sessioni.



Amazon Web Services (AWS)

- Modello di responsabilità condivisa.
- 77 zone di disponibilità distribuite su 24 regioni in tutto il mondo:
 - Costi
 - Latenza
 - Sicurezza e conformità
 - Disponibilità dei servizi
- Ottimizzazione dei costi.

Amazon Simple Storage Service (S3)

- Quantità infinita di oggetti in un bucket (fino a 5 TB di dati).
- Struttura flat:

Delimitatori

Sessioni/Elaborazioni/transcribe.json

Prefissi dei nomi di chiave

Nomi di chiave



- Bucket creati:
 - **brainiostorage**: memorizzazione output elaborazioni sui video.
 - **brainio-mediatemp**: memorizzazione temporanea video.
 - **brainio-resulttemp**: memorizzazione file temporanei.

AWS Elastic Beanstalk e Amazon RDS

- Creazione applicazione Elastic Beanstalk.
- Creazione di un ambiente Elastic Beanstalk e distribuzione di una versione dell'applicazione.
- Configurazione istanza database MySQL di Amazon RDS.

Elastic Beanstalk > Ambienti

Tutti gli ambienti 🔄 Azioni ▼ Crea un nuovo ambiente

< 1 > ⚙️

	Nome dell'ambiente ▲	Stato di integrità ▼	Nome dell'applicazione ▼	URL ▼	Versioni in esecuzione: ▼	Piattaforma ▼
<input type="radio"/>	django-env	Ok	BrainIO	django-env.eba-22m9ksej.eu-central-1.elasticbeanstalk.com	app-210226_210910	Python 3.6 running on 64bit Amazon Linux

AWS Identity and Access Management (IAM)

Definizione di:

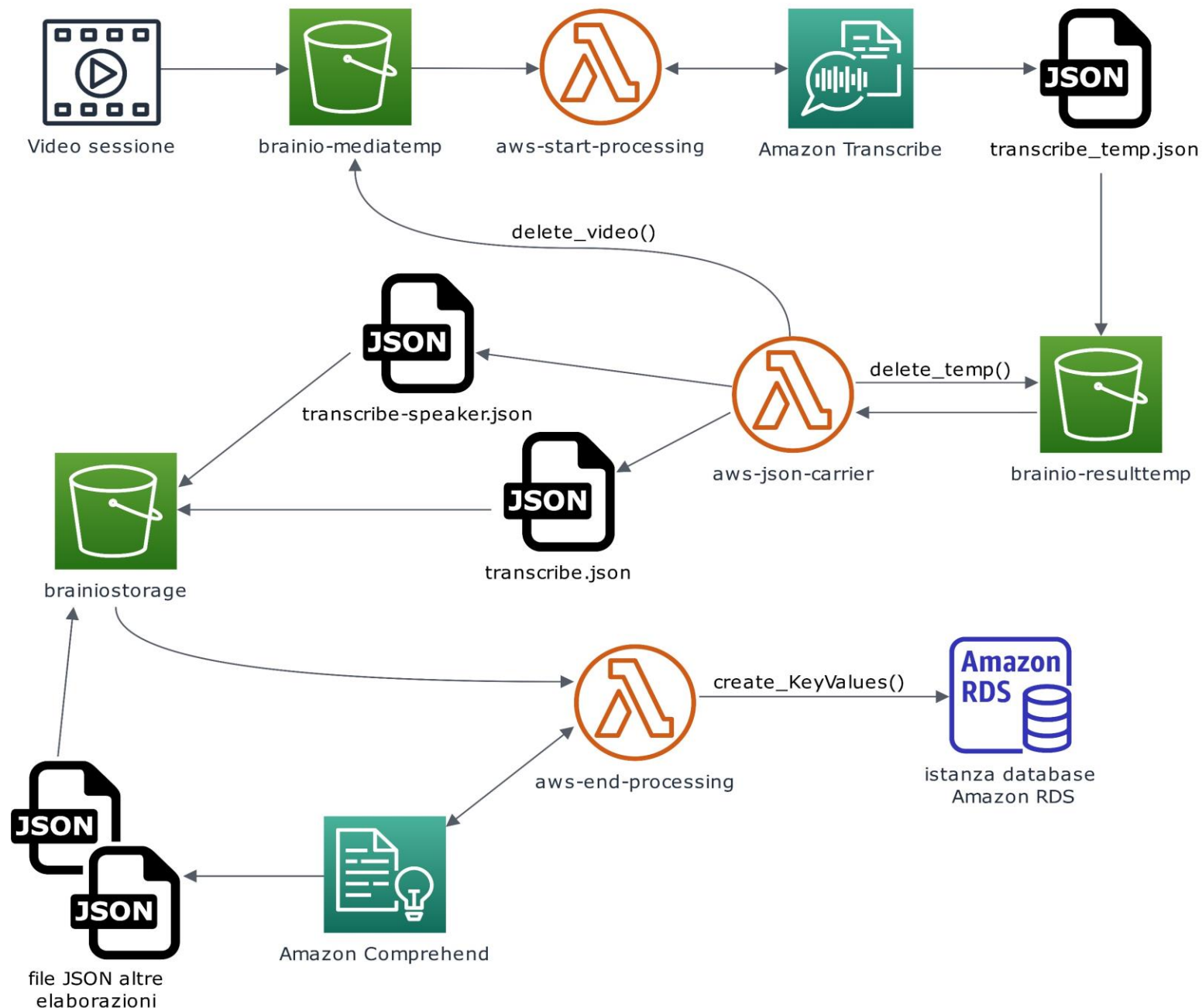
- policy basate su identità (utenti, gruppi o ruoli IAM).
- policy basate su risorse.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::696482742108:role/aws-elasticbeanstalk-ec2-role"
      },
      "Action": [
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::brainiostorage/*"
      ]
    }
  ]
}
```

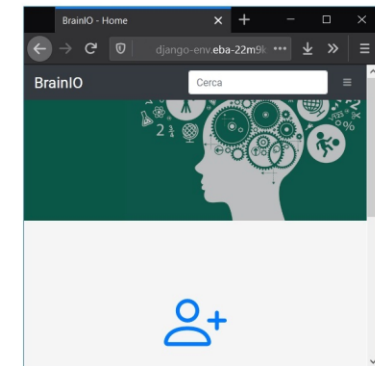
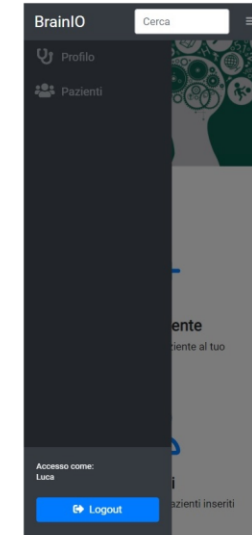
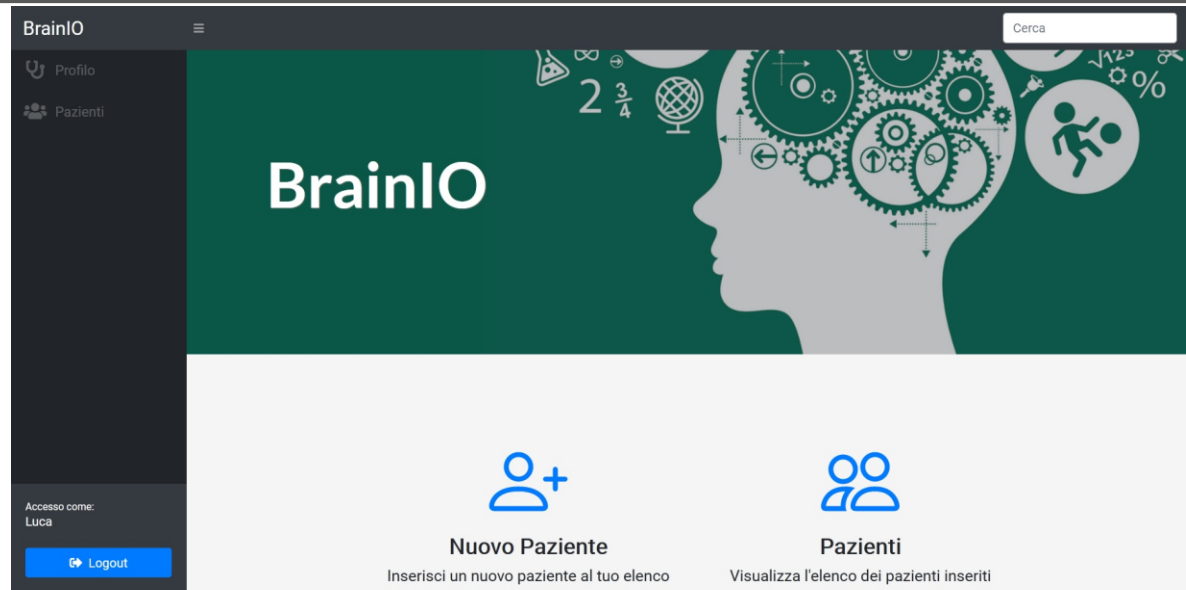
AWS Lambda

Definizione di 3 funzioni Lambda:

- aws-start-processing
- aws-json-carrier
- aws-end-processing



Responsive Web Design




Effettua il login per accedere a BrainIO

I campi contrassegnati con * sono obbligatori.

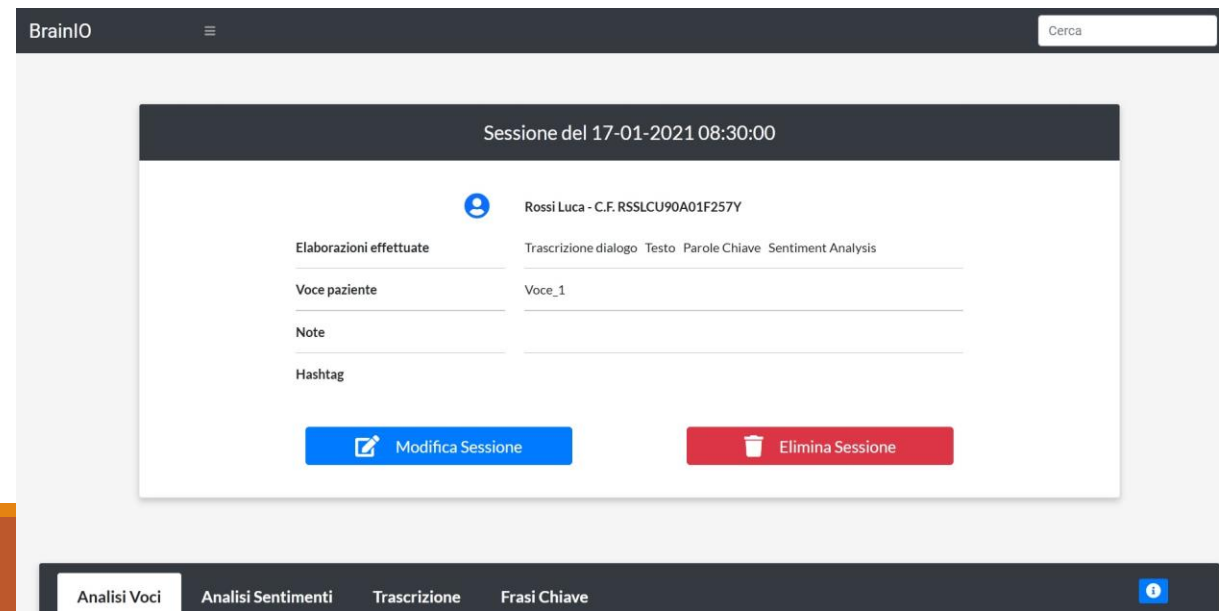
Nome utente*

Password*

Login

 [Password dimenticata?](#)

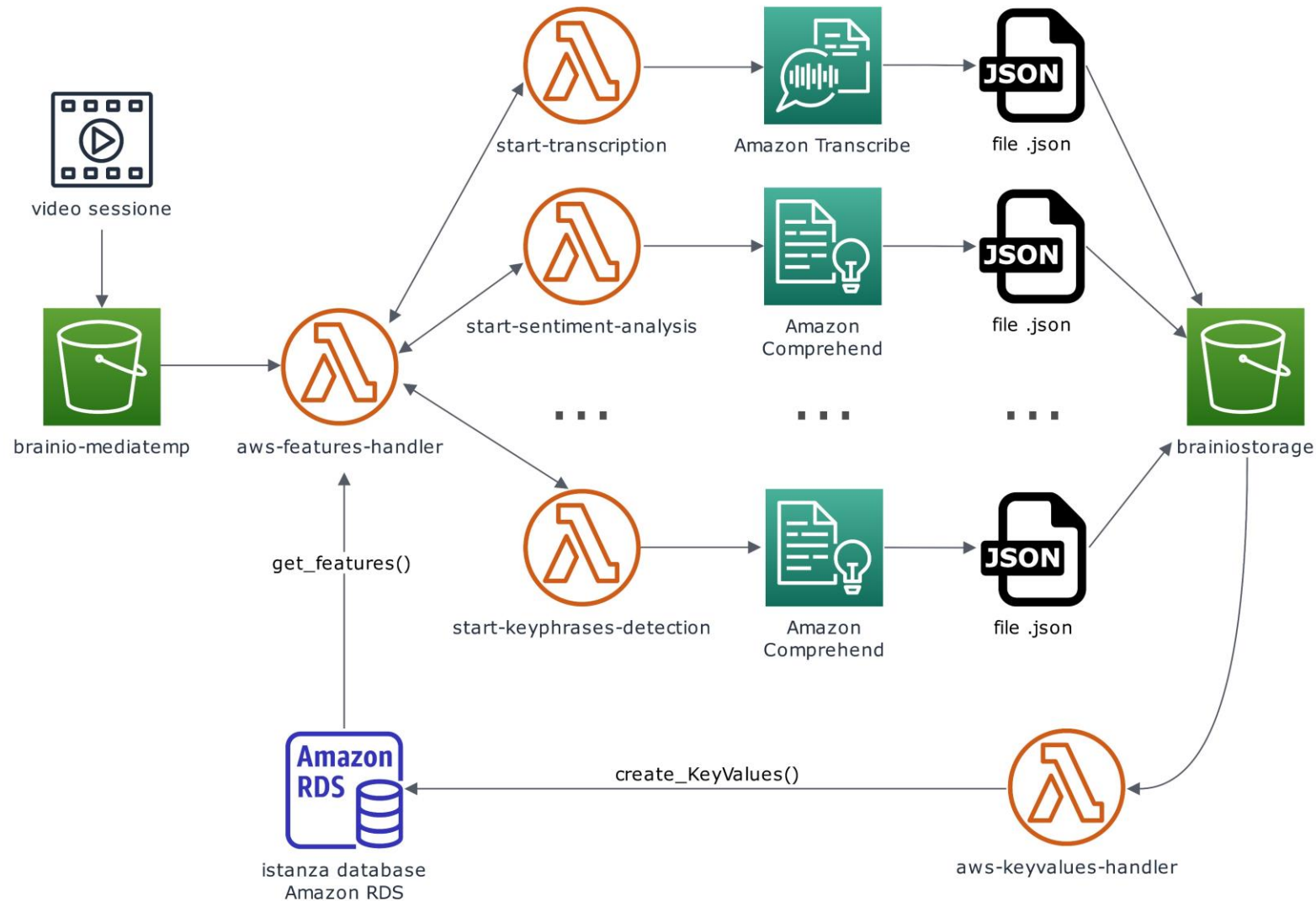
Non sei ancora registrato? [Registrati](#)



Conclusioni

- Necessario approfondimento sulle norme a tutela dei dati personali dei pazienti.
- Configurazione ottimale dei servizi scelti con una maggiore esperienza con quanto offerto da AWS.
- Possibile miglioramento della modularità dell'applicazione.

Possibile miglioramento



Grazie per l'attenzione