

BOOK SHARING

Progetto IoT
Corso "IoT & 3D Intelligent Systems"

Paolo Bruno
Alessandro Castellucci



Introduzione

Rapporto libri e tecnologia

PRO

- gestione ottimizzata di biblioteche e banche dati
- diffusione di formati digitali (pdf, epub, mobi, cbr, cbz) grazie agli ebook reader
- contenuti disponibili con un click
- acquisti da casa
- portabilità

CONTRO

- abbandono del caro e vecchio libro cartaceo
- scarso supporto alla condivisione degli ebook
- assenza di community e gruppi lettura

Bike sharing
Car sharing

...

BOOK SHARING

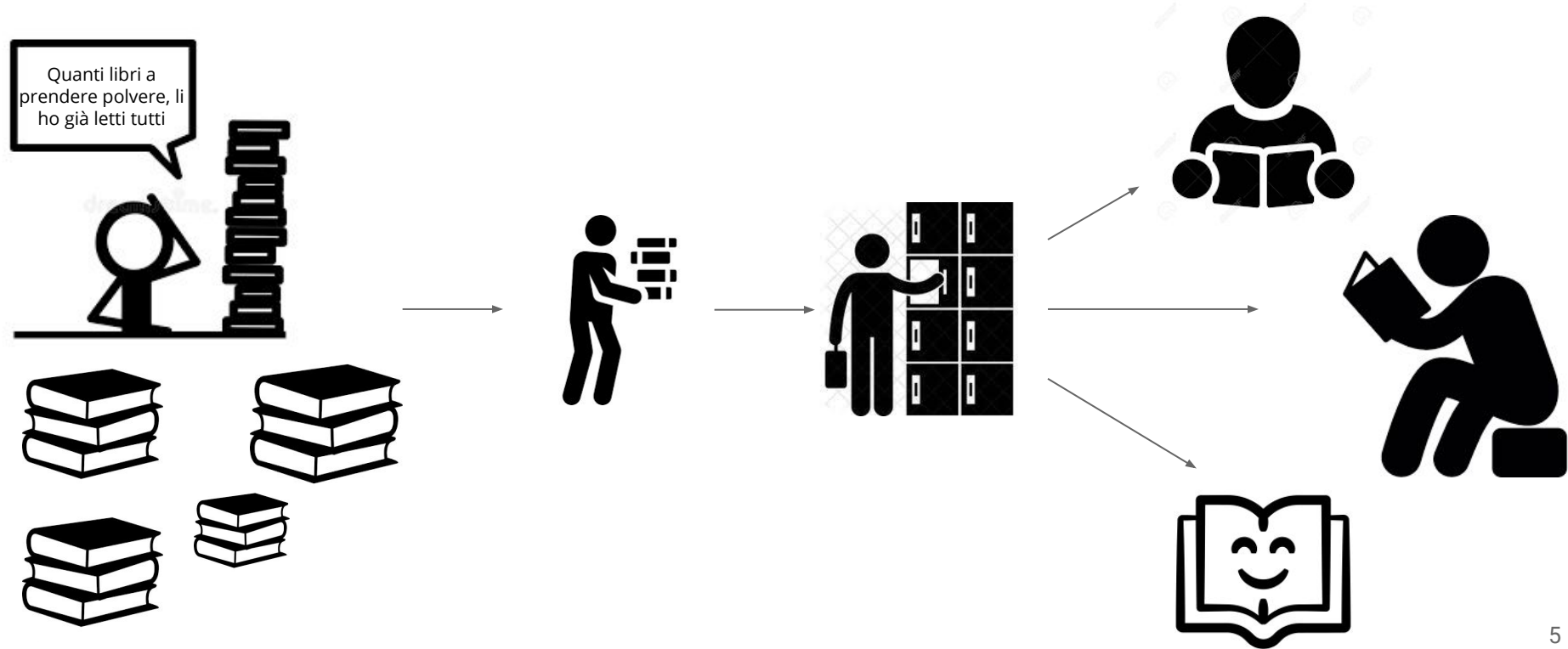
Condividiamo auto, bici, monopattini, ...

Perché non condividere anche la cultura?

Perché non condividere libri?



Pensato per le persone



L'IoT per rendere “tecnologici” anche i libri cartacei

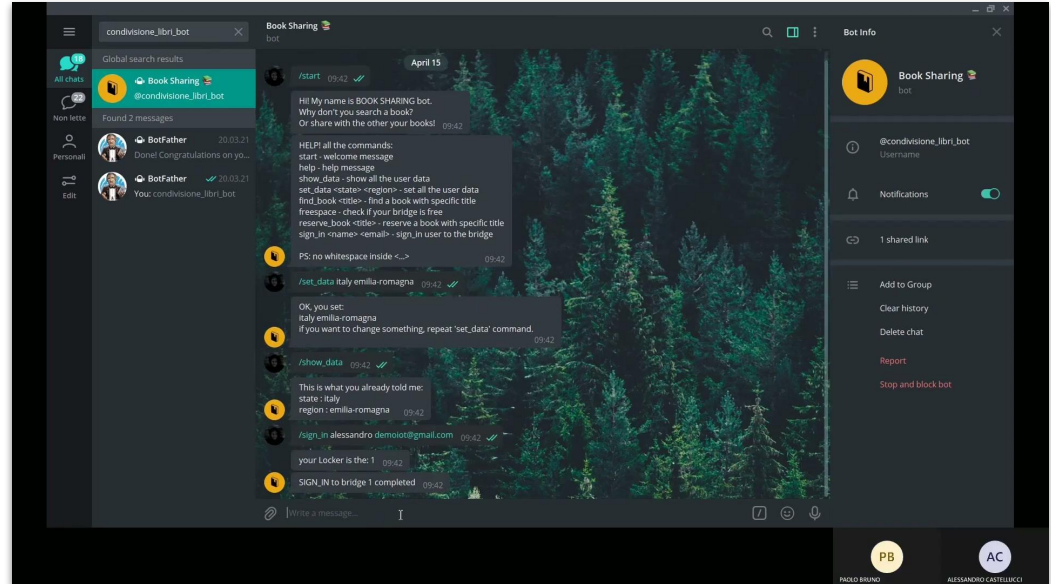
- Locker di libri distribuiti sul territorio
 - gestiscono i libri in modo smart
 - interagiscono per gestire utenti e richieste
- Utenti
 - condividono libri portandoli nei locker
 - ritirano i libri di loro interesse resi disponibili da altri utenti
 - lasciano una valutazione
 - ottengono raccomandazioni periodiche via email



Cosa serve all'utente?

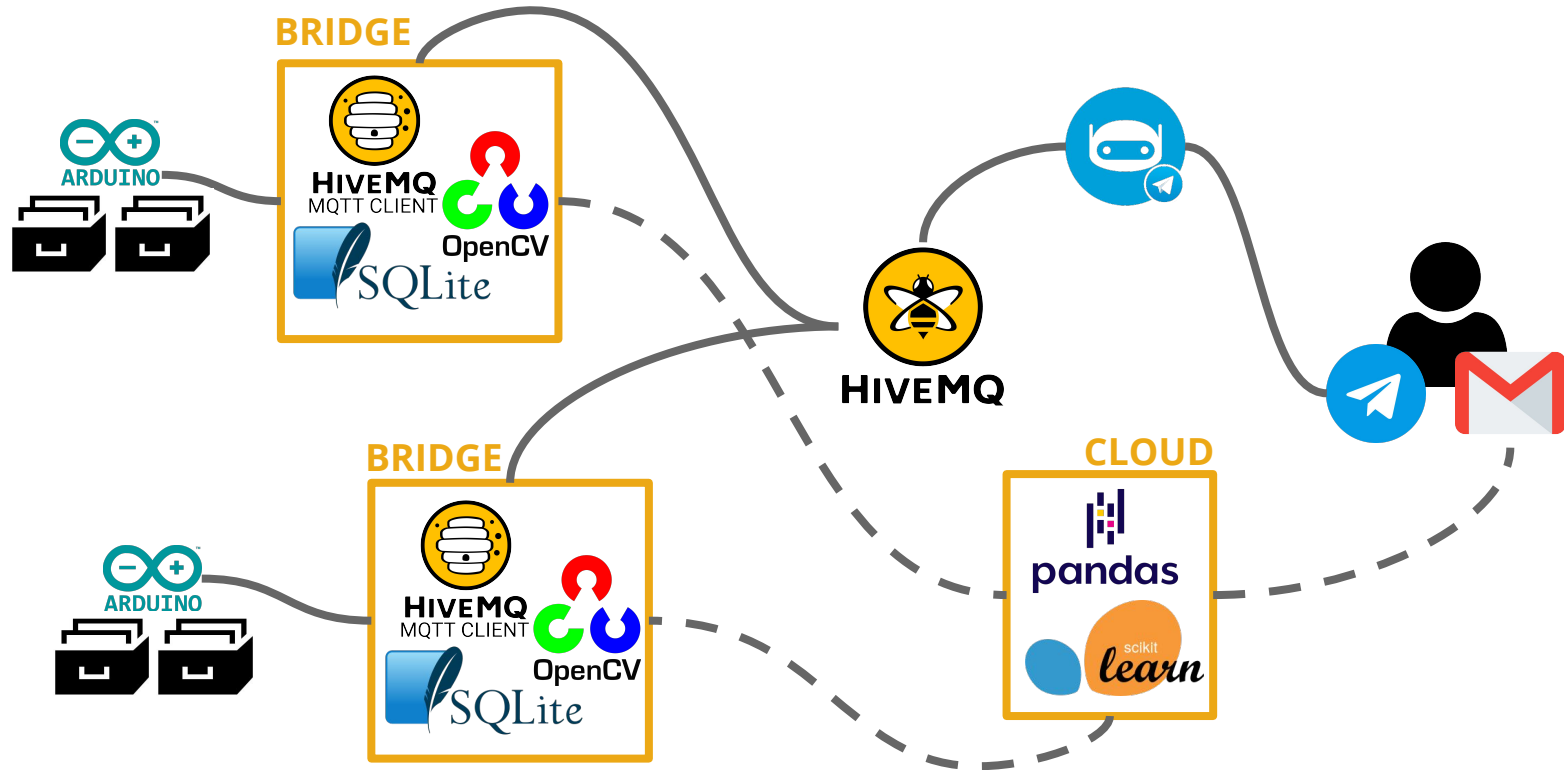


Solo Telegram!
... e voglia di leggere



Descrizione tecnica

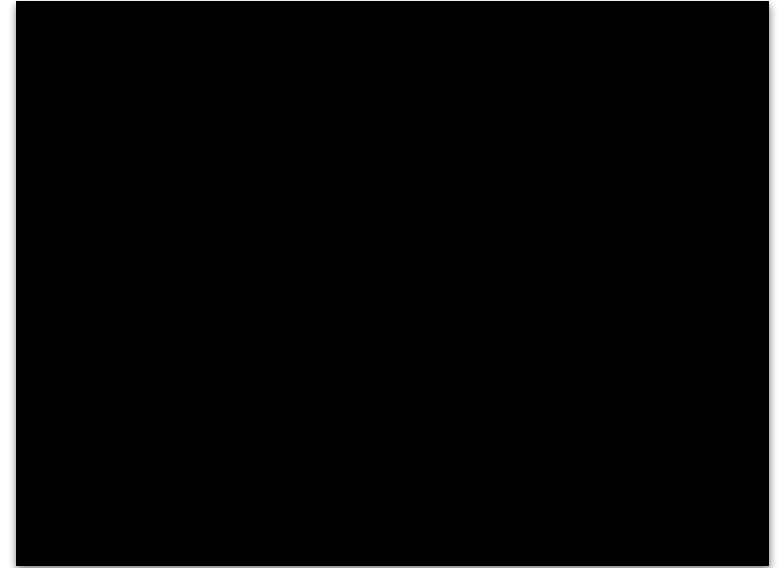
Panoramica del sistema



Bridge



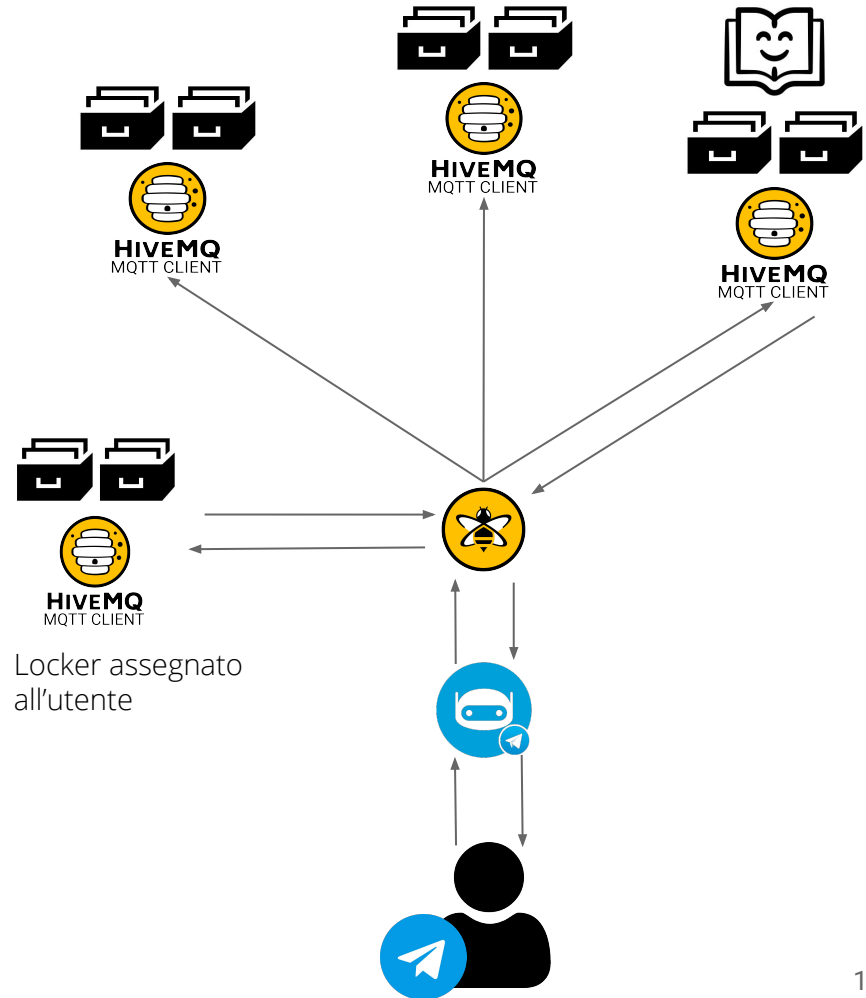
- connessione seriale bridge-microcontrollore
- database locale con libri, utenti e ratings
- API Google per il riconoscimento dei libri
- gestione delle interazioni utente-bridge
- analisi del flusso video da webcam
- comunicazione MQTT verso client Telegram e altri bridge
- comunicazione bridge-cloud



Riconoscimento dell'utente e condivisione di un libro

Broker MQTT HiveMQ

- comunicazione attraverso protocollo MQTT
- scambio di messaggi client-bridge e bridge-bridge
- utilizzo di topic per sottoscrizione ad-hoc



Client → Bridges			
DESCRIPTION	ACTION	TOPIC	PAYLOAD
channel to receive response	SUB	world/state/region/clients/client_id	
find bridge	PUB	world/state/region/bridges	{client_id}

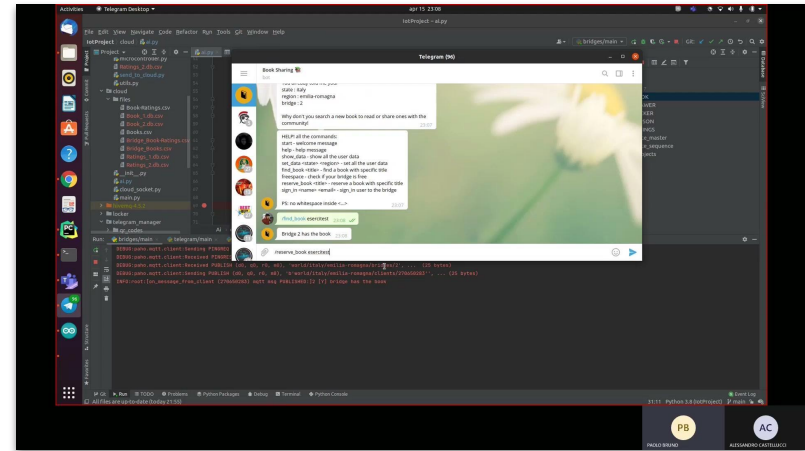
Bridges → Client			
DESCRIPTION	ACTION	TOPIC	PAYLOAD
channel to receive response	SUB	world/state/region/bridges/bridge_id	
positive response to find bridge	PUB	world/state/region/clients/client_id	{bridge_id} is a bridge that can be achieved
positive response to find book	PUB	world/state/region/clients/client_id	{bridge_id} [Y] bridge has the book
negative response to find book	PUB	world/state/region/clients/client_id	{bridge_id} [N] bridge has NOT the book
positive response to check freespace	PUB	world/state/region/clients/client_id	{bridge_id} [Y] bridge has freespace
negative response to check freespace	PUB	world/state/region/clients/client_id	{bridge_id} [N] bridge has NOT freespace

Bridge → Bridges			
DESCRIPTION	ACTION	TOPIC	PAYLOAD
channel to receive freespace-request	SUB	world/state/region/bridges/freespace	
channel to receive book-request	SUB	world/state/region/bridges/book	
bridge ask to other bridge space	PUB	world/state/region/bridges/freespace	{client_id}
bridge ask to other bridge book	PUB	world/state/region/bridges/book	{client_id} {title}

Request-Response Pattern con MQTT

Bot Telegram

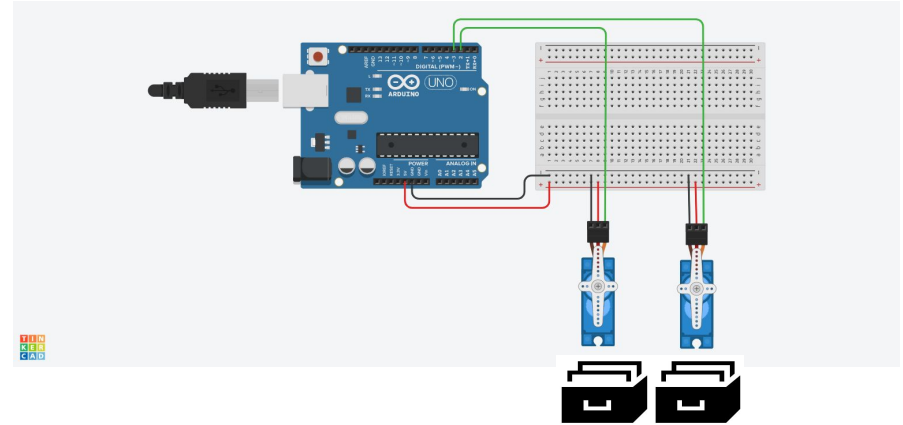
- comunicazione utente-bridge
 - registrazione dell'utente
 - ricerca di un libro
 - prenotazione di un libro
 - verifica di spazio libero in un locker
- gestione persistente dei dati utente
- generazione QR code per identificare l'utente



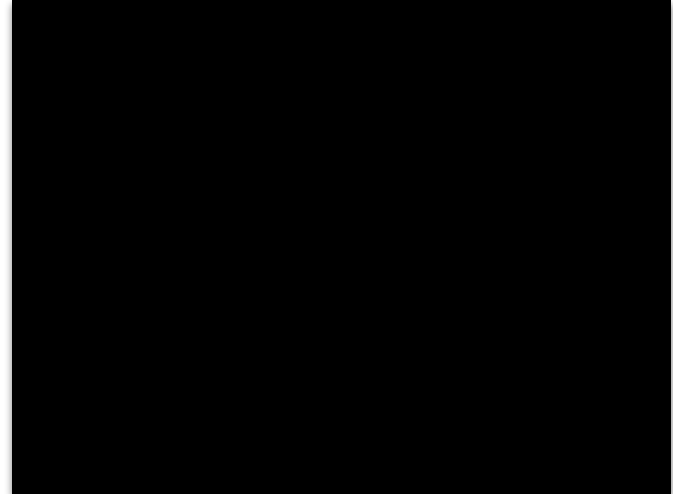
Dimostrazione di ricerca libro, verifica spazio libero e prenotazione libro

Microcontrollore Arduino Uno

- comunicazione seriale con bridge
- FSM per apertura e chiusura dei cassetti nel locker
- servomotori per l'attuazione



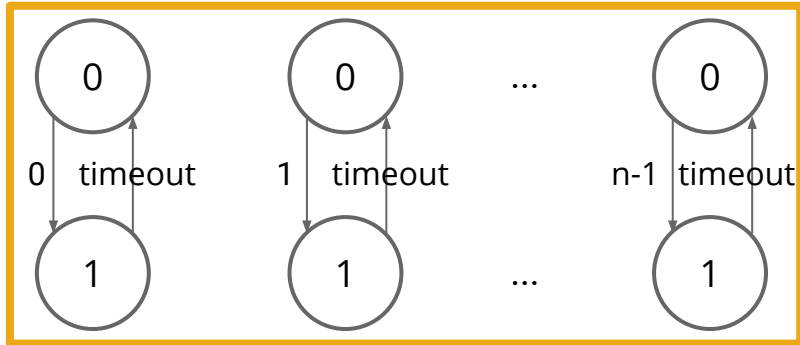
Attuazione servo motore



Uno sguardo al codice

```
// allocation and initialization of the drawers
drawer* d = (drawer*) malloc(TOT_DRAWERS*sizeof(drawer));

char inNum;          // input number from serial
int* cState = (int*) malloc(TOT_DRAWERS*sizeof(int));        // current states
int* futureState = (int*) malloc(TOT_DRAWERS*sizeof(int));    // future states
unsigned long* lasttime = (unsigned long*) malloc(TOT_DRAWERS*sizeof(unsigned long));
```



0: cassetto chiuso
1: cassetto aperto



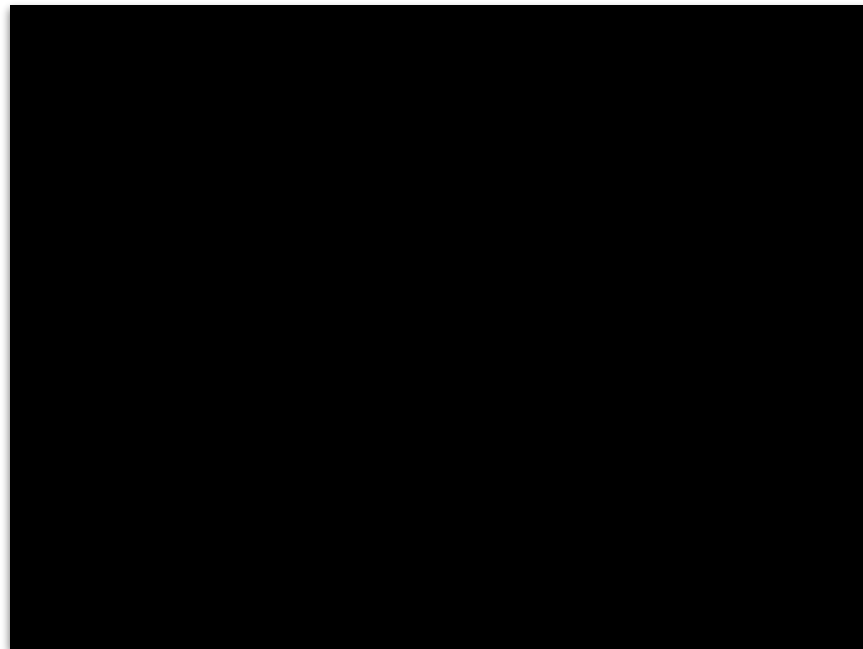
cassetto 0

cassetto 1

cassetto n-1

Cloud + AI

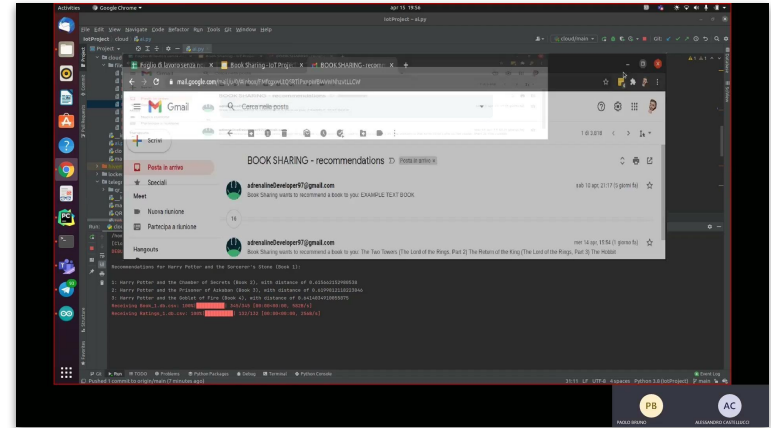
- comunicazione bridge-cloud con socket:
 - invio di file csv
 - backup delle informazioni dei bridge
- analisi dei dati attraverso un sistema di raccomandazione
 - invio di email all'utente



Valutazione di un libro alla sua riconsegna

Sistema di raccomandazione

- Sistema di raccomandazione a filtro collaborativo basato sui ratings dei libri effettuati dagli utenti
- Preprocessing con Pandas
- Implementa l'algoritmo K-Nearest Neighbors di Scikit Learn per la ricerca di cluster di utenti simili (basato sui libri letti in comune)
- Invio di email periodiche contenenti libri consigliati agli utenti



Conclusione

Implementazioni future e migliorie

- Protocollo XMPP:
 - service discovery
 - multi-user chat
 - request-response nativo
- Interfaccia grafica
- Miglioramento nel sistema di attuazione
- Miglioramento dell'integrazione utente - sistema per quanto riguarda:
 - bot Telegram
 - funzioni per il riconoscimento di QR code e ISBN barcode
- Sensoristica avanzata



**Grazie per
l'attenzione!**