SmartGlasses

Abaco Group challenge

Mission

Il progetto si basa sulla volontà di realizzare una soluzione in grado di far coesistere tecnologia e praticità d'uso con un occhio di riguardo verso l'alta scalabilità e il portafoglio.

Per ottimizzare la praticità dell'esperienza la soluzione proposta nasce sul principio di voler ridurre l'input manuale dei dati che, a causa di numerosi Physical Constraint, può risultare un processo invadente e macchinoso.

Quanto realizzato / predisposto nasce per rispondere alle esigenze espresse dall'azienda:

- Navigabilità
- Gestione delle operazioni
- Digital Twin
- Tracciabilità

L'idea - 1

Per poter realizzare a livello pratico quanto indicato nella slide prcedente abbiamo optato per l'impiego di Smart Glass in dotati di video camera e microfono in grado di comunicare con apposite *torrette* mobili posizionate a ridosso delle parcelle.

Tramite il microfono sarà possibile registrare delle note vocali riguardo le operazioni svolte che, una volta processate e trasformate in stringhe, verranno inserite all'interno dei DB.

La video camera presente sugli occhiali permetterà di scattare delle immagini che verranno caricate in cloud affinchè siano processabili da soggetti / algoritmi terzi.

L'idea - 2

Ad ogni parcella è assegnata una torretta che si accende / spegne tramite la verifica di prossimità di un operatore al fine di minimizzare i consumi.
Dalla vicinanza degli occhiali alla torretta scaturisce l'accensione e il relativo funzionamento di una torretta spenta.

Il test di prossimità e lo scambio dati avverrà tramite l'uso del protocollo BLE sia per quanto riguarda lo scanning delle torrette da parte dell'occhiale sia per quanto riguarda lo scambio dati.

All'estermità della torretta è installato un monitor che permette una visualizzazione pratica dei seguenti dati:

- Codice, Descrizione e Tesi sulla parcella
- Operazioni da svolgere
- Operazioni svolte

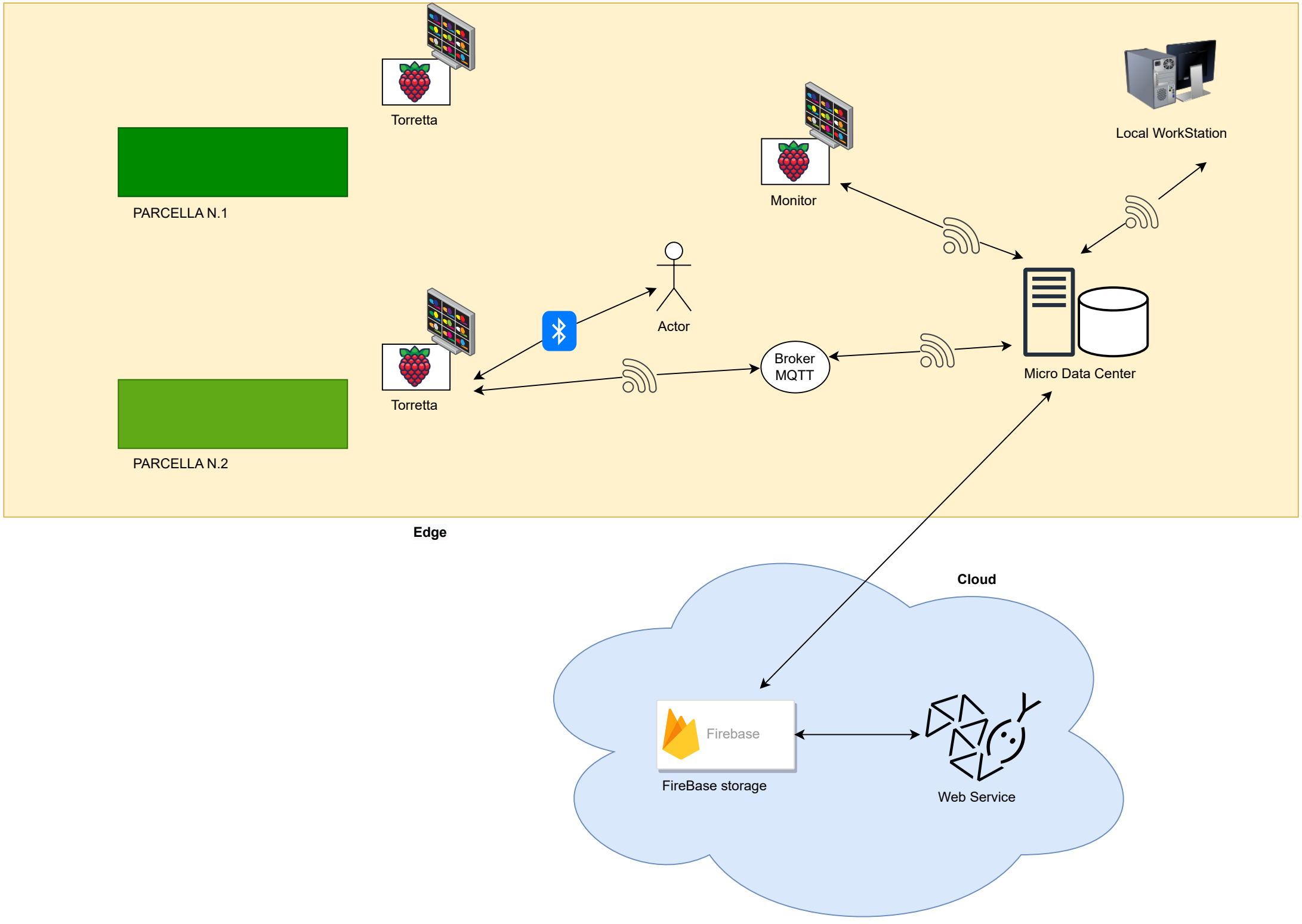
L'idea - 3

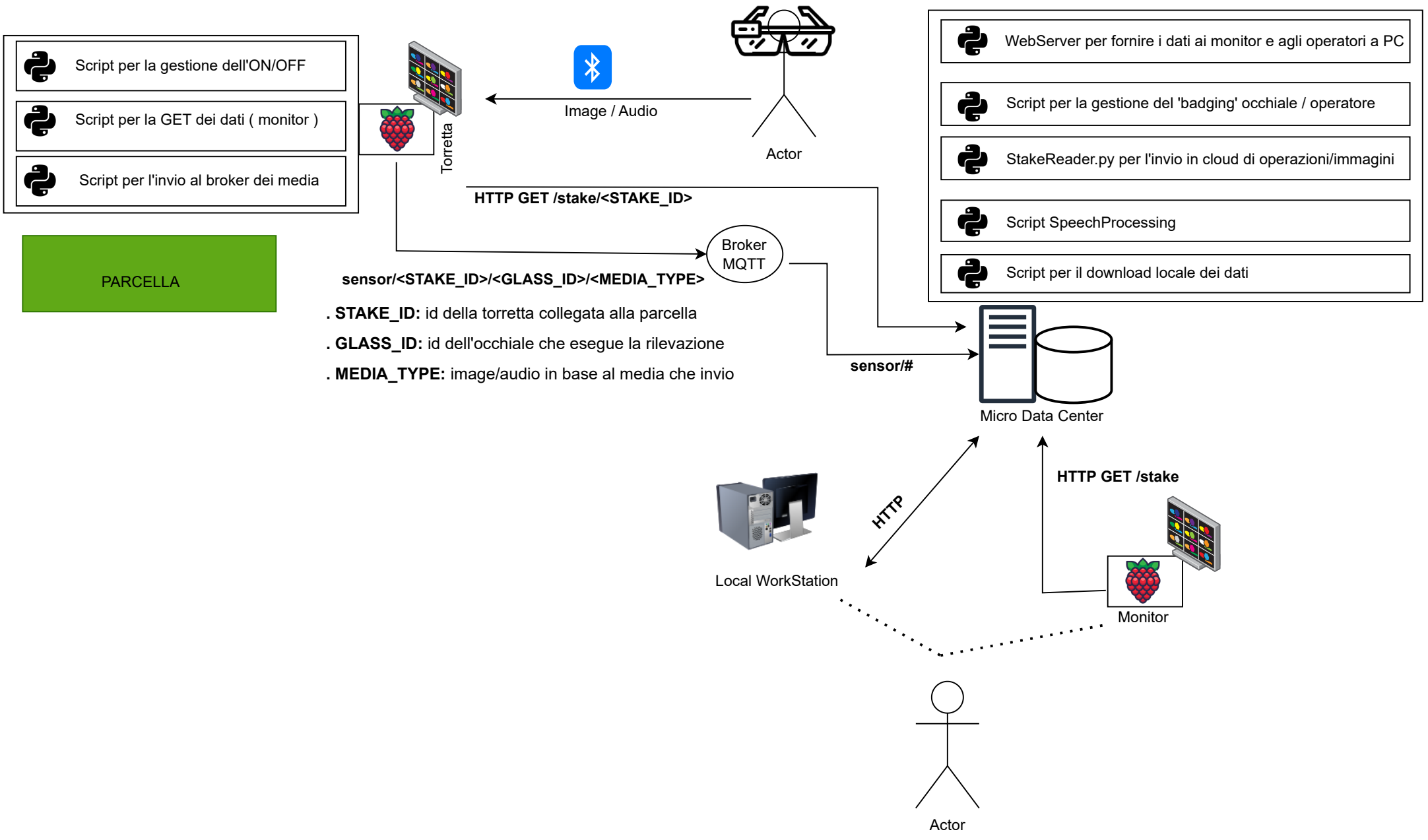
On Edge saranno presenti dispositivi per l'associazione operatore / smart glass al fine di mantenere la tracciabilità delle operazioni svolte nel tempo.
Oltre a questo saranno presenti delle workstation al fine di poter garantire la gestione manuale dei dati in caso di guasti o errori.

Lo script che si occupa di raccogliere i dati provenienti dalle torrette installate invierà online i dati al fine di rendere l'accesso ai dati sempre aggiornato per poter visualizzare fedelmente il *gemello virtuale* della propria serra.

Se all'interno del Data Center installato localmente verranno eseguite operazioni di pre-processing e filtering in cloud avverrà il training di algoritmi di ML come possono essere:

- SpeechRecognition
- Object Detection


Torretta
PARCELLA N.1
Monitor
Local WorkStation
Actor
Torretta
Broker
MQTT
Micro Data Center
PARCELLA N.2
Edge
Cloud
Firebase
FireBase storage
Web Service


Script per la gestione dell'ON/OFF
Script per la GET dei dati (monitor)
Script per l'invio al broker dei media
Torretta
Image / Audio
Actor
WebServer per fornire i dati ai monitor e agli operatori a PC
Script per la gestione del 'badging' occhiale / operatore
StakeReader.py per l'invio in cloud di operazioni/immagini
Script SpeechProcessing
Script per il download locale dei dati
HTTP GET /stake/<STAKE_ID>
PARCELLA
Broker MQTT
sensor/<STAKE_ID>/<GLASS_ID>/<MEDIA_TYPE>
. STAKE_ID: id della torretta collegata alla parcella
. GLASS_ID: id dell'occhiale che esegue la rilevazione
. MEDIA_TYPE: image/audio in base al media che invio
sensor/#
Micro Data Center
HTTP
HTTP GET /stake
Local WorkStation
Monitor
Actor

Devices & Servizi

- **Smart Glass:** dotati di un chip per la comunicazione BLE, videocamera e microfono.
- **Torrette:** rapsberry per la ricezione dei dati dagli smart glass e l'invio al broker MQTT dei dati con un annesso monitor per la visualizzazione dei dati relativi alla parcella.
- **Lettore Badge:** collega gli smart glass all'utilizzatore.
- **Monitor:** display posizionato all'interno della serra per visualizzare genericamente il posizionamento delle parcelle.
- **WorkStation**: postazione per l'inserimento manuale dei dati
- **Micro Data Center**
- **Broker MQTT:** broker Mosquitto locale per l'interscambio dati disaccoppiato script / torretta.
- **Firebase BaaS**: impiegati i servizi di Storage e Real-Time DB.
- **PythonAnywhere:** piattaforma in cloud con hosta la web app per utenti terzi.

Tecnologie & Protocolli

- **Google Speech Recognition:** viene usato per trasformare le note vocali provenienti dagli smart glasses tramite le torrette in stringhe contenti le operazioni svolte.
- **BLE:** protocollo con caratteristiche simili al tradizionale Bluetooth, adatto per low consumption appliances.
- **MQTT**
- **Flask Framework:** gestisce le API REST sul server locale e su PythonAnywhere. La separazione tra l'applicativo locale e quello in cloud è legato al fatto che sono offerte funzionalità diverse e inoltre i dati già presenti in locale non devono essere richiesti in cloud riducendo su larga scala l'interrogazione da parte di tutte le WorkStation dell'applicazione in cloud.
- **FireBase-SDK:** rispetto ai tradizionali DB Firebase risulta essere molto pratico in fase di installazione non necessitando di alcun driver.
  Offre inoltre soluzioni *Pay-as-you-use* per la scalabilità e l'ottimizzazione dei costi.

Pricing

- *Lettore Bage* **~ 20,00 €**
- *Smart Glass* **~ 60,00 €**
  - Arduino Nano 33 BLE **~ 25,00 €**
  - Video camera **~ 15,00 €**
  - Microfono **~ 10,00 €**
  - Batteria al litio **~ 0,15 €/Kw**
- *Torretta* **~ 70,00 €**
  - Display **~ 30,00 €**
  - Raspberry Pi Zero W **~ 20,00 €**
  - Batteria al litio **~ 0,15 €/Kw**
- *Firebase*
  - *Real-Time DB*
    - GB Stored: **$5/GB**
    - GB Downloaded: **$1/GB**
  - Storage **~ 15,00 €/Mese**

Credits

- *Edoardo Torrini*
- *Mirco Sandonini*
- *Alessandro Pansera*