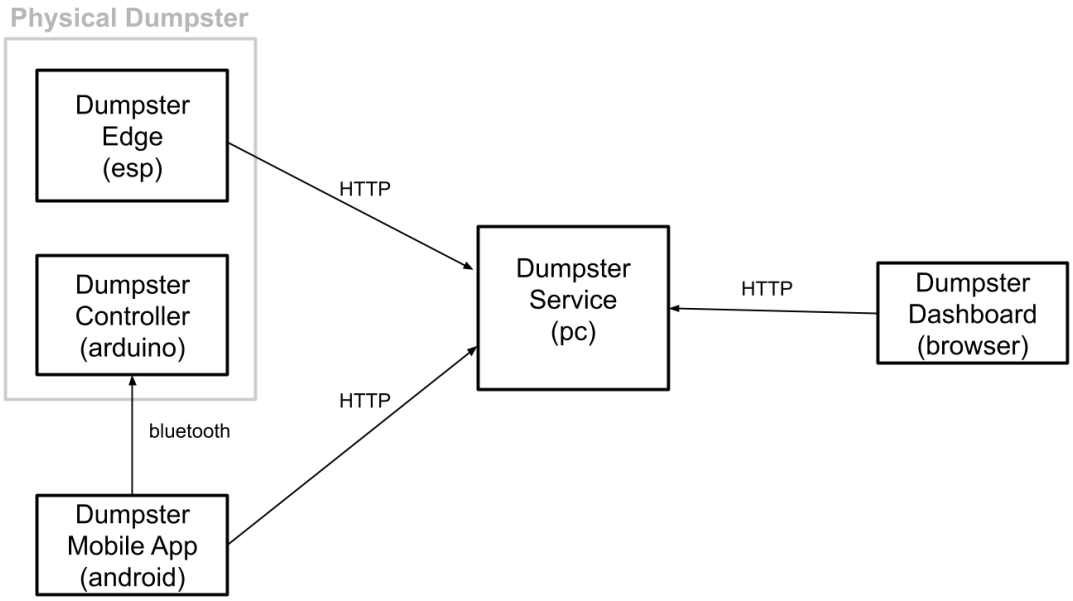
Sistemi Embedded e IoT - a.a. 2019-2020

Progetto #3 - *Smart Dumpster*

Si vuole realizzare un sistema IoT che implementi una versione semplificata di uno *smart dumpster*, ovvero un cassonetto dei rifiuti “intelligente”. Il sistema complessivamente è costituito da 5 sotto-sistemi:



- **Dumpster Controller (Arduino)**
  - sistema embedded che controlla il cassonetto e interagisce con utenti dotati di dispositivo mobile su cui è in esecuzione la parte Mobile App (via Bluetooth)
- **Dumpster Mobile App (Android)**
  - permette all’utente di interagire con il cassonetto e il processo di deposito di un rifiuto. Interagisce sia con la parte Controller (via Bluetooth), sia con la parte Service (via HTTP).
- **Dumpster Edge (ESP)**
  - sistema embedded sempre parte del cassonetto che ha come compito monitorare la quantità di rifiuti depositati e interagire con la parte Service (via HTTP)
- **Dumpster Server (ESP)**
  - servizio REST che tiene traccia dello stato del cassonetto.
- **Dumpster Dashboard (Browser o Client su PC)**
  - Front end per visualizzazione/osservazione/analisi dati

Dettaglio componenti HW di Controller e Edge

- Dumpster Controller
    - Microcontrollore Arduino UNO con board che include:
        - tre led verdi L_A, L_B, L_C
        - 1 servo motore M con cui si attua l'apertura e chiusura dello sportello
        - 1 modulo Bluetooth HC-06 o HC-05

- Dumpster Edge
    - SoC ESP 8266 con board che include:
        - un led verde L_{avail} e un led rosso L_{not_avail}
        - 1 sensore che misura il peso (simulato da un potenziometro)

Comportamento dettagliato del sistema

Lo Smart Dumpster si può trovare in due stati possibili: *available* o *not-available*, il primo segnalato dalla configurazione led L_{avail} acceso e led L_{not_avail} spento, mentre il secondo dalla configurazione opposta. Nello stato *available*, gli utenti possono depositare i rifiuti, nello stato *not-available* il Dumpster non è disponibile (in quanto, ad esempio, pieno e in attesa di essere svuotato). L'utente può controllare lo stato del Dumpster via Dumpster Service.

Per depositare un rifiuto in uno Smart Dumpster:

- L'utente ottiene un *token* dal servizio Dumpster Service e quindi si reca in prossimità del cassonetto. E' possibile ottenere un token solo se il Dumpster è nello stato *available*.
- Ottenuto un token, l'utente può interagire con il cassonetto via mobile app selezionando il tipo di rifiuto da depositare (A, B, C). Selezionato il tipo di rifiuto, il Dumpster accende il led corrispondente (L_A, L_B, L_C) e apre il portello. L'utente ha $T_{deliver}$ secondi per depositare il rifiuto, dopodiché il portellone si chiude e contestualmente si spegne il led.Nel caso all'utente serva più tempo, lato app si può chiedere (più volte) un'estensione.
- Se il processo va a buon fine, l'app comunica al servizio Dumpster Service il successo dell'operazione, in modo che il servizio tenga aggiornato il numero di depositi fatti.
- Il Dumpster Edge monitora la quantità di rifiuti presenti. Non appena questo valore supera una certa soglia W_{max}, il Dumpster passa in stato *not-available*, comunicandolo al servizio. ~~Nel caso ci sia una operazione di deposito in corso, si chiude il portellone e l'operazione viene conclusa.~~
- Il Dumpster Dashboard, interagendo con il Dumpster Service, permette di visualizzare lo stato del Dumpster - disponibilità, numero depositi fatti e quantità corrente - e fornisce l'UI sia per ripristinare lo stato *available,* sia per forzare lo stato *not-available* in caso di necessità. Inoltre la Dashboard deve permettere di visualizzare l'andamento dell'utilizzo del Dumpster (numero depositi, quantità depositate) negli ultimi N_{days} giorni.

Realizzare il sistema con le seguenti specifiche:

- **Dumpster Controller** basato su piattaforma Arduino
    - Implementare la logica in termini di macchina a stati finiti sincrona
- **Dumpster Edge** basato su piattaforma ESP (o equivalenti)
    - nessun vincolo sulla tecnologia da usare
- **Dumpster Service** in esecuzione su un PC
    - nessun vincolo sulla tecnologia da usare
- **Dumpster Mobile App** basato su piattaforma Android
    - Fisica o emulata: nel caso fisico, la comunicazione con Controller deve avvenire mediante bluetooth; nel caso emulato, la comunicazione con Controller può avvenire mediante seriale, usando lato PC ove è in esecuzione l'emulatore Android il bridge emulatore-seriale presentato in laboratorio
- **Dumpster Dashboard** - basato su browser o client su PC
    - Può essere implementata come web app (quindi basato su protocollo HTTP) con la tecnologia che si ritiene più opportuna o anche come client usando socket TCP o UDP

Per tutti gli aspetti non specificati, fare le scelte che si credono più opportune.