**23/02/2025:**

**FATTO:**

* Fatto funzionare l’associazione ID tra Central (enroll) e acquisition. Durante l’accensione il bridge aspetta una prima enroll. Questo genera un ID casuale che viene passato al bridge per poi trasmetterlo all acquisition libera (una delle disponibili, già gestita). Non è gestito però la richiesta di apertura da parte dell’utente (funziona solo il primo giro, vedi primo punto DA FARE)
* Merge su un unico branch (dev-alpha2.0) e cancellato quelli non necessari. Quando abbiamo da fare una feature, apriamo branch e poi merge su dev-alpha2.0. Al termine merge su master
* Gestita la compatibilità del bridge per il sistema operativo macOS
* Controllo dell’invio ad ogni iterazione del pacchetto placholder da parte del central verso il bridge(“0000000000”) e la relativa gestione se è una richiesta di enroll oppure una richiesta di apertura della cassetta.
* Gestione del lock locale (open-close)
* parte gestione infringement
* risolto il motivo per cui invertendo le seriali su mac si bloccava

**DA FARE:**

* **Telegram BOT (Fre)**
* ~~Controllo dell’invio ad ogni iterazione del pacchetto placholder da parte del central verso il bridge(“0000000000”) e la relativa gestione se è una richiesta di enroll oppure una richiesta di apertura della cassetta.~~
* ~~Capire come gestire la chiusura, con il punto precedente abbiamo deciso che la cassetta non è lockata quindi si può aprire, il byte di lock sarà quindi a 1, quando lo rimettiamo a 0? (da discutere assieme, idea personale: lo lasciamo a 1 finche la cassetta non viene chiusa fisicamente e in quel momento lo rimettiamo a 0)~~
* ~~Gestione dello sblocco dall’infragment lock (lato central~~)
* ~~Capire il motivo per cui invertendo le seriali su mac ovvero mettendo il central su cu.usbmodem1101 e l’acquisition su~~
* Test con più acquision (2)