



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

GLITCHHub
TEAM

Verbale esterno 10/03/2026 (M31)

Ordine del giorno

1. Presentazione design del **gateway** e dubbi correlati
2. Presentazione design del backend **cloud**
3. Ritrattazione di alcuni requisiti
4. Consigli per il futuro e utilizzo di modelli ad agenti

•
Versione **1.0.0**

Stato Verificato

Partecipanti Alessandro Dinato
Riccardo Graziani
Elia Ernesto Stellin
Jaume Bernardi
Michele Dioli
Siria Salvalaio

Distribuzione GlitchHub Team
M31 SRL
Prof. Cardin Riccardo
Prof. Vardanega Tullio

Indice

1. Introduzione	2
2. Resoconto	2
2.1. Presentazione diagramma gateway	2
2.1.1. Frequenza di invio dati	2
2.1.2. Creazione nuovi tenant	2
2.2. Presentazione design del backend cloud	2
2.3. Ritrattazione dei requisiti	3
2.4. Consigli per il futuro e utilizzo di modelli ad agenti	3
3. Attività conseguenti	3

1. Introduzione

Il presente verbale attesta che in data 10 marzo 2026 dalle 14:30 alle 15:20, si è svolto l'incontro con la **proponente** **M31 SRL**, in modalità remota.

Durante l'incontro il gruppo ha mostrato l'avanzamento dei diagrammi C4 del sistema e ha espresso alcuni dubbi relativi alla progettazione.

2. Resoconto

2.1. Presentazione diagramma gateway

Il gruppo ha presentato il code diagram aggiornato del **gateway**, illustrando le nuove implementazioni corrette a seguito dell'incontro con il Prof. Cardin.

Il gruppo ha posto il seguente dubbio: ha senso che il GatewayManagerService mantenga uno stato per quanto riguarda la configurazione, al fine di far partire il data sender e i sensori simulati? La **proponente** ha risposto dicendo che è opportuno che il **gateway** mantenga uno stato a runtime in modalità "push", in modo che ogni modifica alla configurazione si rifletta direttamente sullo stato del **gateway**.

Secondo la **proponente**, è prioritario concentrarsi sulla corretta implementazione delle funzionalità destinate al prodotto finale, considerando che componenti come il simulatore e i gateway non troverebbero applicazione in un futuro contesto reale.

2.1.1. Frequenza di invio dati

Il gruppo ha discusso il problema della frequenza d'invio variabile. In precedenza era stato stabilito che ogni tipo di sensore (**profilo GATT**) avrebbe potuto avere una frequenza d'invio dedicata.

Tuttavia questa soluzione risulta complessa da implementare, dato l'utilizzo di **SQLite** come buffer interno e la necessità di un data sender per ogni tipo di sensore, il che rende difficile propagare e gestire lo stato a livello di **gateway**.

La **proponente** ha approvato il compromesso proposto dal gruppo, ovvero l'implementazione di una frequenza di trasmissione variabile per singolo **gateway**. Il parametro di frequenza sarà applicato all'intera istanza, garantendo l'invio complessivo indipendentemente dalla natura del dato.

L'obiettivo finale è garantire la precisione direttamente sul **gateway** per poi distribuire la configurazione ai sensori. Al momento, questa ottimizzazione è stata esclusa dal rilascio attuale e rimandata a sviluppi futuri.

2.1.2. Creazione nuovi tenant

Il gruppo, data la difficoltà implementativa, ha chiesto alla **proponente** come gestire la creazione di nuovi **tenant**, dato che la gestione tramite **NATS** risulterebbe troppo onerosa in termini di tempo. La proposta di una figura dedicata alla creazione delle credenziali è stata accettata dalla **proponente**, con la presenza di una nota indicante la futura miglioria.

2.2. Presentazione design del backend cloud

Il gruppo ha presentato il diagramma del sistema **cloud**, illustrando l'individuazione dei servizi principali:

- User Service, per l'autenticazione;
- Gateway Service;
- Sensor Service;
- Authentication Service;

- Historical Data Service;
- Real-Time Service.

Dalla discussione con la **proponente**_G è emersa l'alta desiderabilità dell'Alert Service, un servizio che non era stato modellato. È stato consigliato al gruppo di proseguire comunque con il suo inserimento nel diagramma del sistema e, solo in caso di mancanza di tempo, rinunciare alla sua realizzazione.

2.3. Ritrattazione dei requisiti

A seguito della discussione, la **proponente**_G ha ritrattato il livello di importanza di alcuni requisiti:

- frequenza d'invio variabile: da considerarsi requisito desiderabile;
- sospensione dell'invio dati: da considerarsi requisito opzionale;
- alert: da considerarsi requisito desiderabile con alto grado di desiderabilità.

2.4. Consigli per il futuro e utilizzo di modelli ad agenti

La **proponente**_G ha espresso soddisfazione per il lavoro di design svolto, ritenendo che esso indirizzi bene il prosieguo dello sviluppo, rendendo il lavoro meno dispersivo.

Riguardo all'utilizzo di **LLM**_G nello sviluppo, la **proponente**_G ha consigliato di supervisionare sempre il lavoro prodotto dagli strumenti di intelligenza artificiale e di effettuare doppi controlli. In particolare, è stato suggerito di inserire nella cartella .github del workspace dei file di istruzioni e skill per personalizzare gli agenti di sviluppo (ad esempio, creare agenti con esperienza in **Go**_G). Consigliando l'utilizzo e lo studio di awesome-copilot.

3. Attività conseguenti

Task	Assegnatari	Issue
Modificare AdR con requisiti ritrattati	Siria Salvalaio	<u>Docs/#429</u>
Modificare code diagram con frequenza d'invio variabile per gateway	Alessandro Dinato	<u>Docs/#431</u>
Aggiungere AlertService in code diagram	Elia Ernesto Stellin	<u>Docs/#432</u>

Jaume Bernardi



Firma del revisore interno

Cristian Pirlog



Firma del revisore esterno