



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

GLITCHHub
TEAM

Verbale esterno 03/03/2026 (M31)

Ordine del giorno

1. Presentazione e dubbi su design
2. Discussione su dev container

•
Versione **1.0.0**

Stato	Verificato
Partecipanti	Alessandro Dinato Riccardo Graziani Elia Ernesto Stellin Jaume Bernardi Michele Dioli
Distribuzione	GlitchHub Team M31 SRL Prof. Cardin Riccardo Prof. Vardanega Tullio

Indice

1. Introduzione	2
2. Resoconto	2
2.1. Presentazione e dubbi su design	2
2.1.1. Design del Gateway	2
2.1.2. Component Diagram	2
2.1.3. Code Diagram	2
2.1.4. Osservazioni della proponente sul design del Gateway	2
2.1.5. Design del Data Consumer	2
2.2. Discussione su dev container	3

1. Introduzione

Il presente verbale attesta che in data 3 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 presentato alla proponente lo stato di avanzamento del design logico del progetto, sviluppato secondo il modello C4, e ha esposto i diversi dubbi emersi durante la progettazione.

2. Resoconto

2.1. Presentazione e dubbi su design

Il gruppo ha illustrato alla proponente il design logico in fase di sviluppo, basato sul modello C4:

- i diagrammi di System e Container sono stati ritenuti soddisfacenti dalla proponente;
- rimangono alcuni dubbi aperti sui diagrammi di Component e Code, discussi nei punti successivi.

2.1.1. Design del Gateway

Sono emersi dubbi bloccanti relativi al componente *gateway*, la cui gestione prevede un elevato grado di multithreading tramite l'uso di goroutine.

2.1.2. Component Diagram

Il gruppo ha esposto il dubbio riguardante la tecnologia da adottare per il *buffer* del *gateway*, che necessita di supporto alla persistenza dei dati. La proponente ha confermato la scelta di utilizzare SQLite con una colonna di tipo JSON, in modo da poter salvare dati eterogenei senza la necessità di modellare esplicitamente lo schema del database.

2.1.3. Code Diagram

Il gruppo ha illustrato la scelta di applicare l'architettura esagonale al *design* del *gateway* ed ha esposto il seguente dubbio: il componente `NatsGatewayCommandRepository` riceve comandi da NATS e crea delle classi di comando, le quali vengono successivamente passate alla goroutine `GatewayCommandDispatcher`. Quest'ultimo riceve una classe che implementa un'interfaccia comune; il gruppo desidera adottare una soluzione polimorfica senza fare ricorso a RTTI e ha chiesto indicazioni su dove debba avvenire la creazione del comando poi eseguito da `GatewayCommandDispatcher`. La proponente ha dichiarato che la soluzione va determinata con uno studio più approfondito ed ha consigliato di esporre questo dubbio attraverso una mail in maniera da poter fornire una risposta più completa.

2.1.4. Osservazioni della proponente sul design del Gateway

La proponente ha fatto notare che sono state impiegate numerose risorse sul *design* del simulatore del *gateway* tramite architettura esagonale. Questo impegno andrebbe dedicato anche al *backend* della dashboard, ritenuto dalla proponente la parte più rilevante del progetto. Il lavoro svolto è stato comunque riconosciuto come approfondito e dettagliato. Tuttavia la proponente ha consigliato di volgere maggiore attenzione al design del sistema *cloud*.

2.1.5. Design del Data Consumer

Il gruppo ha presentato alla proponente il design sviluppato fino a questo momento per il Data Consumer, illustrando i *services* e i *components* di Angular elaborati. Non sono emersi dubbi particolari in merito al *frontend*.

2.2. Discussione su dev container

La proponente ha suggerito caldamente di adottare i Dev Container al fine di garantire una linea di sviluppo comune, basata su ambienti di sviluppo virtualizzati e indipendenti dal sistema operativo. Come alternativa ai Dev Container, è stato consigliato lo strumento ASDF per la gestione degli strumenti di lavoro.

Jaume Bernardi



Firma del revisore interno

Cristian Pirlog



Firma del revisore esterno