



NoTIP
NO TESTS IN PRODUCTION

Verbale Interno del 2026-03-09

Versione	0.0.1
Data Modifica	2026-03-09
Utilizzo	Interno
Presenze	Valerio Solito Leonardo Preo Alessandro Mazzariol Matteo Mantoan Alessandro Contarini Mario De Pasquale Francesco Marcon

Abstract dei contenuti

Meeting interno tenutosi in seguito all'attività di sportello con il prof. Cardin per chiarimenti di natura tecnica riguardanti progettazione e sviluppo dell'*MVP_G*.

Indice

1. Info e ordine del giorno	3
2. Discussione	3
2.1. Suddivisione <i>microservizi_G</i> e comunicazione interna/esterna.	3
2.2. Strategia di separazione delle <i>repository_G</i>	3
2.3. Approccio alla documentazione architettuale per la PB.	4
2.4. Struttura del Manuale Utente.	4
3. Esiti e decisioni finali	5

1. Info e ordine del giorno

In data **09 marzo 2026** si è svolta una riunione interna del gruppo in modalità telematica dalle ore 09:30 alle ore 10:00.

L'ordine del giorno prevede quattro questioni di natura tecnica:

- Suddivisione *microservizi_G* e comunicazione interna/esterna (API *REST_G* vs *NATS_G* Request-Reply e/o a eventi);
- Strategia di separazione delle *repository_G*;
- Approccio alla documentazione architettuale per la PB;
- Struttura del Manuale Utente.

2. Discussione

2.1. Suddivisione *microservizi_G* e comunicazione interna/esterna.

Dibattito:

- **Accoppiamento e Granularità:** La comunicazione sincrona genera forti dipendenze (disponibilità simultanea, necessità di retry/circuit breaker).
 - **Accorpamento:** Valutare l'unificazione se il servizio A dipende strettamente da B.
 - **Separazione:** Mantenere distinti i servizi con volumi di carico differenti per garantire lo scaling indipendente.
 - **Approccio prof. Cardin:** Prediligere una suddivisione iniziale granulare per poi unificare solo se necessario.
- **Persistenza:** Confermato l'**anti-pattern** del database condiviso; ogni microservizio deve possedere il proprio strato di persistenza dedicato.
- **Comunicazione e Pattern:**
 - **Modello Ibrido:** Adottare un approccio sincrono verso l'esterno (richiesta-risposta) e asincrono internamente (event-driven).
 - **Resilienza:** Implementazione obbligatoria di timeout, throttling, circuit breaker e logiche di retry.
 - **Transazionalità:** Deve essere circoscritta a un singolo microservizio per evitare la complessità delle transazioni distribuite.
- **API Gateway_G:** Le API esterne possono fungere da aggregatori per più *microservizi_G* interni (es. gestione *tenant_G* e logica applicativa).

Decisioni:

Si è concordato di seguire le indicazioni del Prof. Cardin: suddivisione granulare dei *microservizi_G*, comunicazione esterna sincrona e interna asincrona tramite *NATS_G*.

Azioni da intraprendere:

- [doc-specifica_tecnica-v0.1.0](#) - Creazione e inizio stesura documento di specifiche tecniche

2.2. Strategia di separazione delle *repository_G*.

Dibattito:

Il Prof. Cardin sconsiglia nettamente l'uso di monorepo, ritenendole raramente giustificate. Con un'architettura a *microservizi_G* la scelta consigliata è **una *repository_G* per microservizio**,

così da permettere a sotto-gruppi del team di lavorare in autonomia. L'alternativa, qualora non si adottino *microservizi_G*, sarebbe il modular monolith.

Decisioni:

Si è concordato di adottare *repository_G* separate per ciascun microservizio e una *repository_G* di infrastruttura.

Azioni da intraprendere:

- [Setup repo servizi MVP_G](#)

2.3. Approccio alla documentazione architetturale per la PB.

Dibattito:

Per la Product *Baseline_G* è fondamentale descrivere le **scelte architetturali**, non i dettagli implementativi. Vengono valutate le relazioni fra le componenti, le motivazioni, la correttezza e i trade-off (sincrono vs asincrono, *microservizi_G* vs monolite, ecc.). Vanno descritti i pattern architetturali ad alto e basso livello, ma senza scendere nel codice. Il tutto andrà esposto in una **presentazione di 20 minuti**.

Decisioni:

Si è concordato di focalizzare la documentazione PB sulle scelte architetturali, le relazioni tra componenti e le relative motivazioni, evitando dettagli implementativi.

Azioni da intraprendere:

- [doc-specifica_tecnica-v0.1.0](#) - Creazione e inizio stesura documento di specifiche tecniche

2.4. Struttura del Manuale Utente.

Dibattito:

Servono **documenti distinti per tipologia di utente**: ad esempio, un utente *frontend_G* non necessita della documentazione API. La specifica OpenAPI è un artefatto utile, ben gestibile anche da strumenti automatici e LLM, ma **non è sufficiente da sola**: serve una descrizione aggiuntiva dei flussi operativi (es. autenticarsi prima di accedere ad altre API).

Decisioni:

Si è concordato di predisporre manuali utente separati per ciascuna tipologia di utente, affiancando alla specifica OpenAPI una documentazione descrittiva dei flussi ove necessario.

Azioni da intraprendere:

- [doc-manuale_utente-v0.0.1](#) - Creazione documento
- [doc-manuale_sysadm-v0.0.1](#) - Creazione documento
- [doc-manuale_api-v0.0.1](#) - Creazione documento
- [doc-manuale_infra_monitoring-v0.0.1](#) - Creazione documento

3. Esiti e decisioni finali

La riunione è da considerarsi conclusa con successo. Il confronto con il Prof. Cardin ha fornito indicazioni chiare su quattro aspetti chiave: suddivisione granulare dei *microservizi_G* con comunicazione esterna sincrona e interna asincrona; adozione di *repository_G* separate per ciascun microservizio; documentazione PB incentrata sulle scelte architetturali e i relativi trade-off, da presentare in 20 minuti; manuali utente differenziati per tipologia di utilizzatore, con specifica OpenAPI affiancata da descrizione dei flussi.