Università degli Studi di Padova

NoTIP
NO TESTS IN PRODUCTION

Colloquio tecnico - PB

Gruppo 12 - A.A. 2025/2026

Francesco Marcon
Alessandro Contarini

Leonardo Preo
Alessandro Mazzariol
Valerio Solito

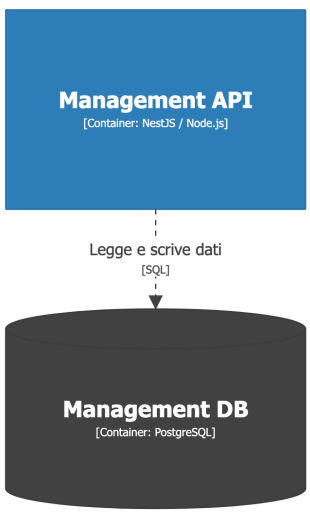
Matteo Mantoan
Mario De Pasquale

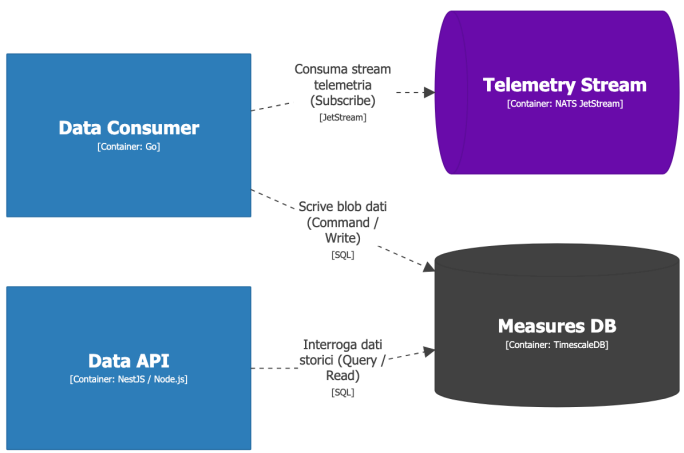
2026-04-17

High Level Design

Persistenza

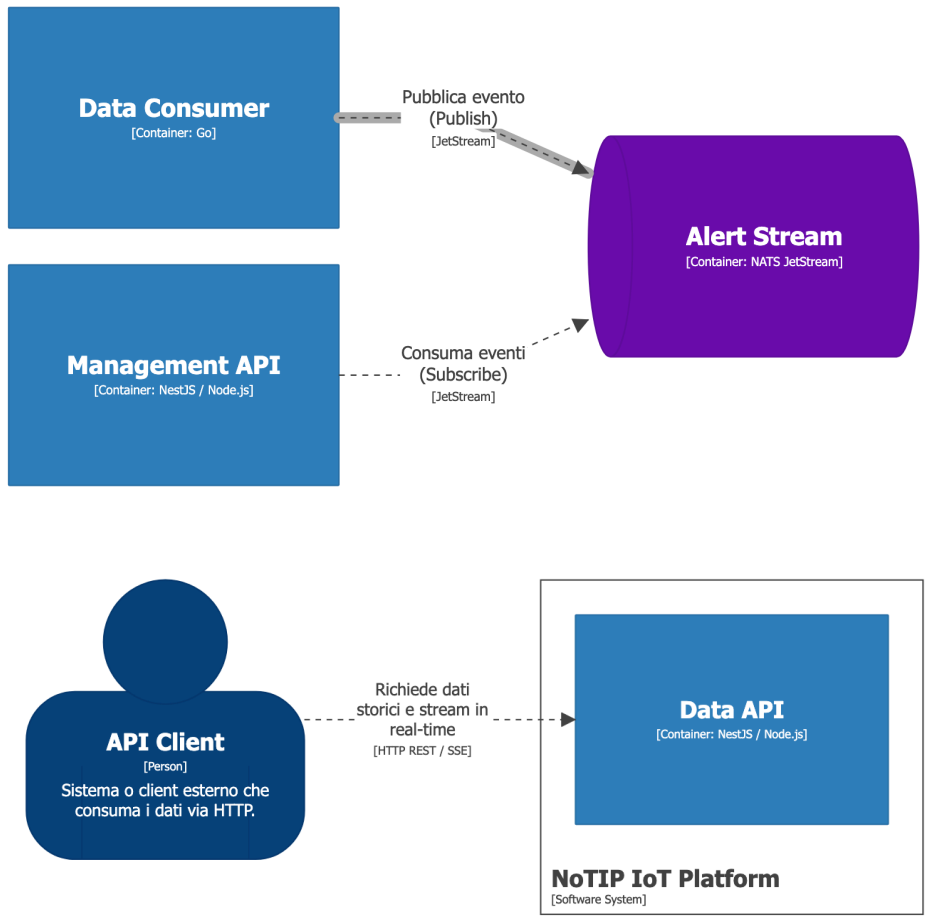
- Database-per-Service
- **CQRS**: Command-Query-Responsibility-Segregation





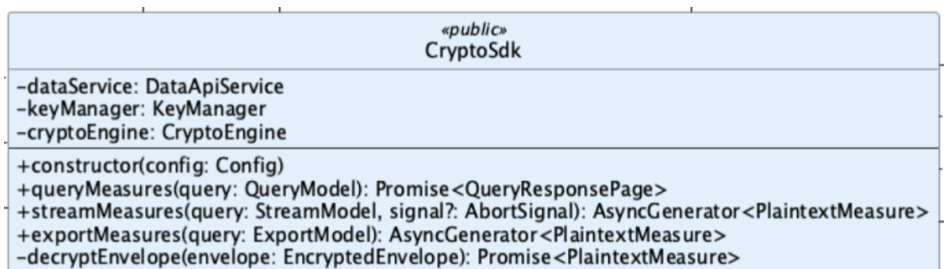
Comunicazione

- Async API inter-service via NATS (+ RR pattern)
- API REST HTTP pubbliche (+ SSE)



SDK

- Problema: semplificare integrazione e sviluppo per clienti
- **CryptoSDK library**, pacchetto NPM

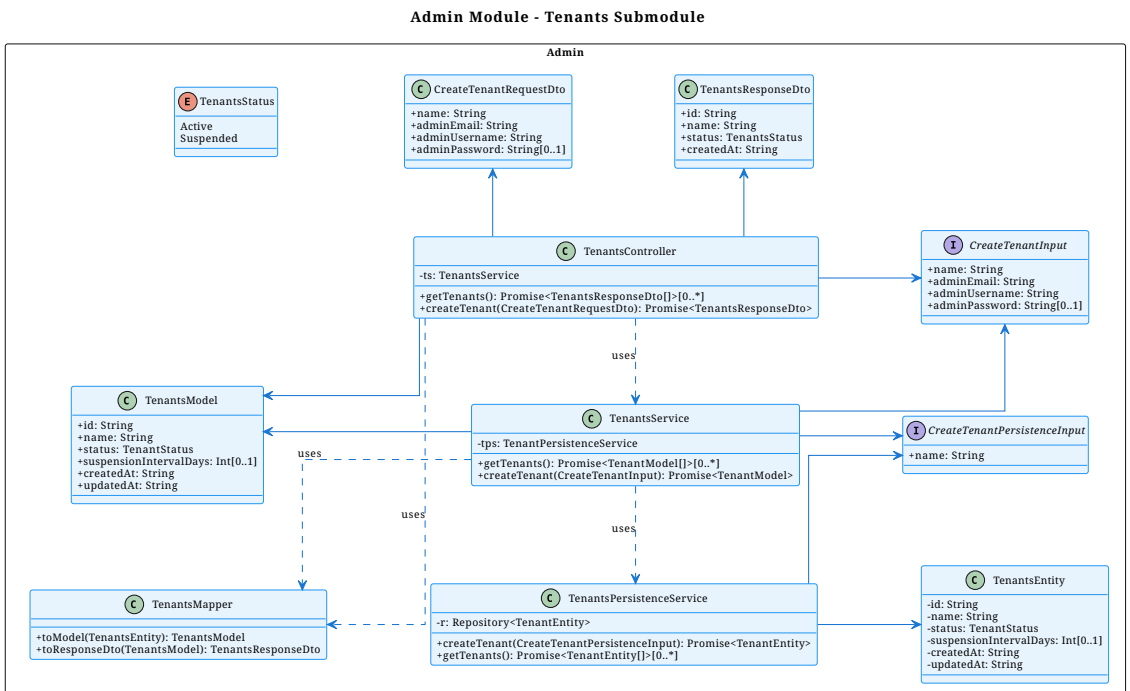


Management API

- **Servizio rappresentativo**: backend **NestJS** layerizzato.
- **Scelta architetturale**: **Layered Architecture** + **logic modules**.

Design patterns:

- **Controller-Service-Persistence**.
- **Dependency Injection** tramite provider NestJS e constructor injection.
- **Mapper Pattern** per `Entity -> Model -> DTO`.
- **Guard** per policy e meccanismo **RBAC**.
- **Integration pattern**: **NATS Request-Reply** per API interne tra microservizi.



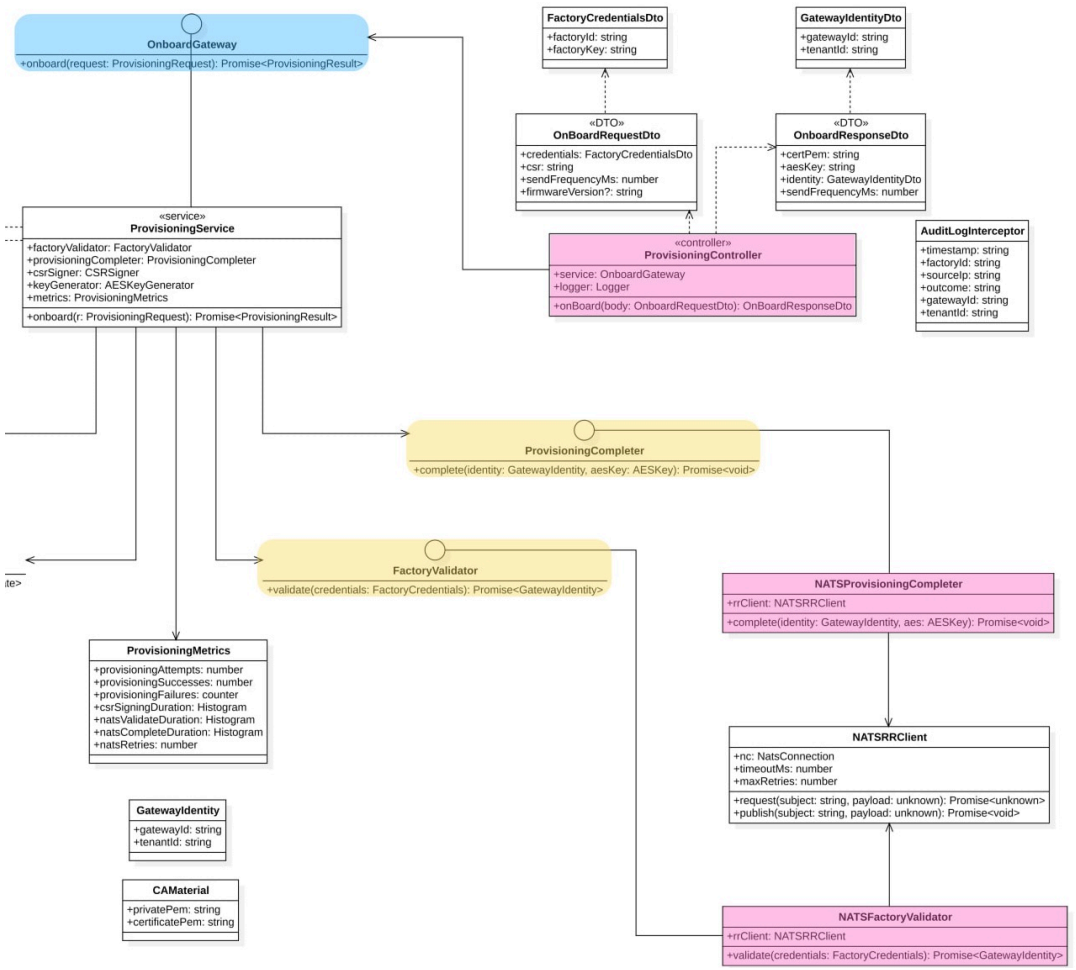
Provisioning Service

- Microservizio di backend in **NestJS** con architettura **Layered** orientata a **ports e adapters**.

Design patterns:

- Scelta architetturale: **Layered**:
  - Presentation
  - Application
  - Infrastructure
  - Persistence

  con **driving ports** e **driven ports**.
- **Adapter Pattern**
- **Dependency Injection**
- **Interceptor Pattern** e **Exception Filter**



Simulator Backend & Data Consumer

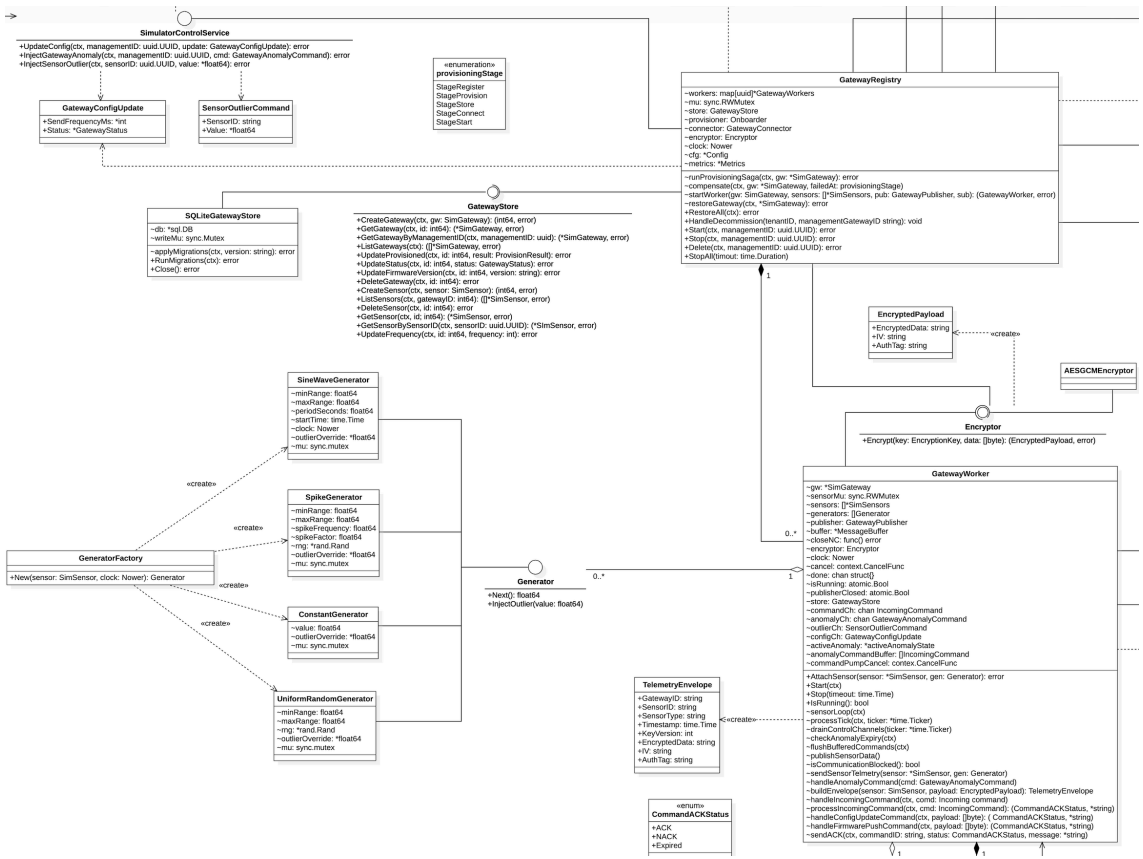
Hexagonal Architecture

Design patterns:

- **Adapter Pattern**
- **Repository Pattern**
- **Strategy** + **Factory** per i gateway simulati
- **Observer Pattern** per il decommissioning dei gateway

Scelte implementative:

- concorrenza `1 goroutine = 1 gateway`
- Heartbeat tracking dei gateway e cache lock-free di configurazione
- Batch processing e buffering



Frontend

Layered Feature-Based Architecture

Design patterns:

- **Interceptor Pattern** HTTP

Scelte implementative:

RouteGuards per protezione rotte

State Management ibrido: Signal Angular + RxJS