



Gruppo 9 - A.A. 2025-2026



# Automated EN18031 Compliance Verification

**Capitolato C1** : Automated EN18031 Compliance Verification

**Obiettivo:** Sviluppare un'applicazione web che automatizzi e guidi la verifica di conformità dei dispositivi radio allo standard EN 18031, che definisce dei requisiti di cybersicurezza.

Ciò viene fatto eseguendo i decision tree dei requisiti in modo strutturato e tracciabile.



Atlas

# Tecnologie utilizzate

## Python v3.11

- Richiesto dal Capitolato
- Sintassi semplice



## FastAPI v0.124.4

- Altamente utilizzato
- Uno dei framework Python più veloci
- Documentazione automatica
- Supporto asincrono



Atlas

## Librerie:

- Pydantic, per la validazione dei dati
- Python-multipart, per ricevere i file caricati
- Pytest, per il testing automatizzato
- HTTPX, per il testing di chiamate API



# Tecnologie utilizzate

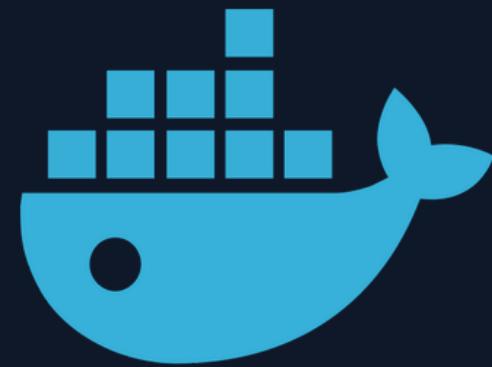
## React v19.2.0

- Semplicità concettuale (componenti)
- Diffuso e consolidato
- Librerie considerate:
  -  React Router, per il routing
  -  Axios, per la gestione delle richieste HTTP
  -  React Hook Form + Zod, per la gestione dei form
  -  Zustand, per la condivisione dello stato
  -  Vitest + React Testing Library, per il testing di unità, dei componenti e di integrazione
- Utilizzo di Context API + Service Layer, per le Dependency Injection



Atlas

# Tecnologie utilizzate



## Docker

- Deploy veloce e affidabile
- Facile riproducibilità degli ambienti di sviluppo



Atlas

# Alternative considerate

- Flask: più lavoro per validazione, documentazione e gestione asincrona
- Angular: struttura complessa e molti concetti avanzati
- Vue.js: React offre maggiore diffusione industriale e un ecosistema più maturo
- Svelte: comunità ed ecosistema più piccolo rispetto a React
- Podman: Docker è più diffuso e conosciuto dal team

