Refactoring di un Software per la Prenotazione di Servizi Sanitari

Refactoring of a Software for Booking Healthcare Services

Relatore
Prof. Michele Amoretti

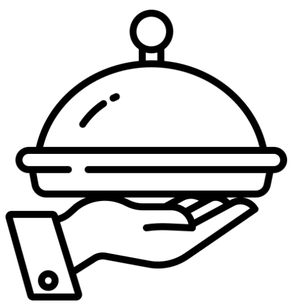
Correlatore
Prof. Andrea Prati
Dott. Fabio Strozzi

Tesi di Laurea di
Daniele Pellegrini

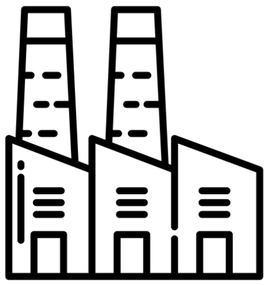
Anno Accademico 2019-2020

MAPS

SHARING KNOWLEDGE

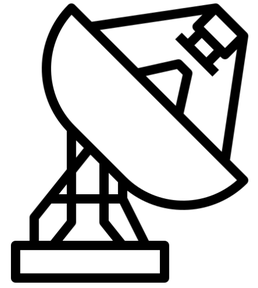


Services



Manufacturing

Public Administration



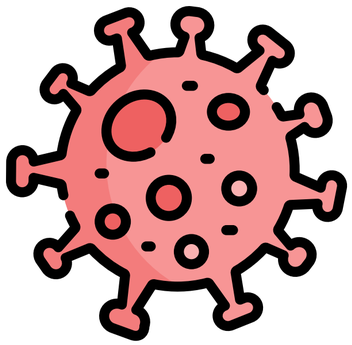
Telco & Utilities

Healthcare

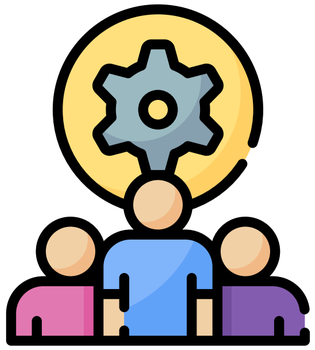
APPLICAZIONE PER LA GESTIONE DELLA CODA NELLE STRUTTURE OSPEDALIERE

- Profilazione degli utenti
- Miglior gestione logistica della struttura
- Prevenzione di assembramenti
- Velocità di accesso ai servizi

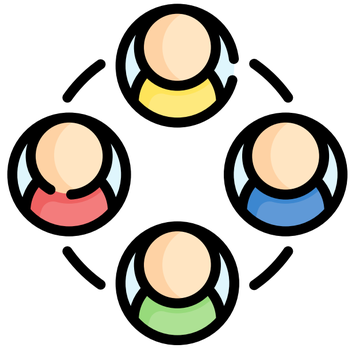
IL PROBLEMA



EMERGENZA SANITARIA



CAMBIAMENTO DEL
TEAM DI SVILUPPO

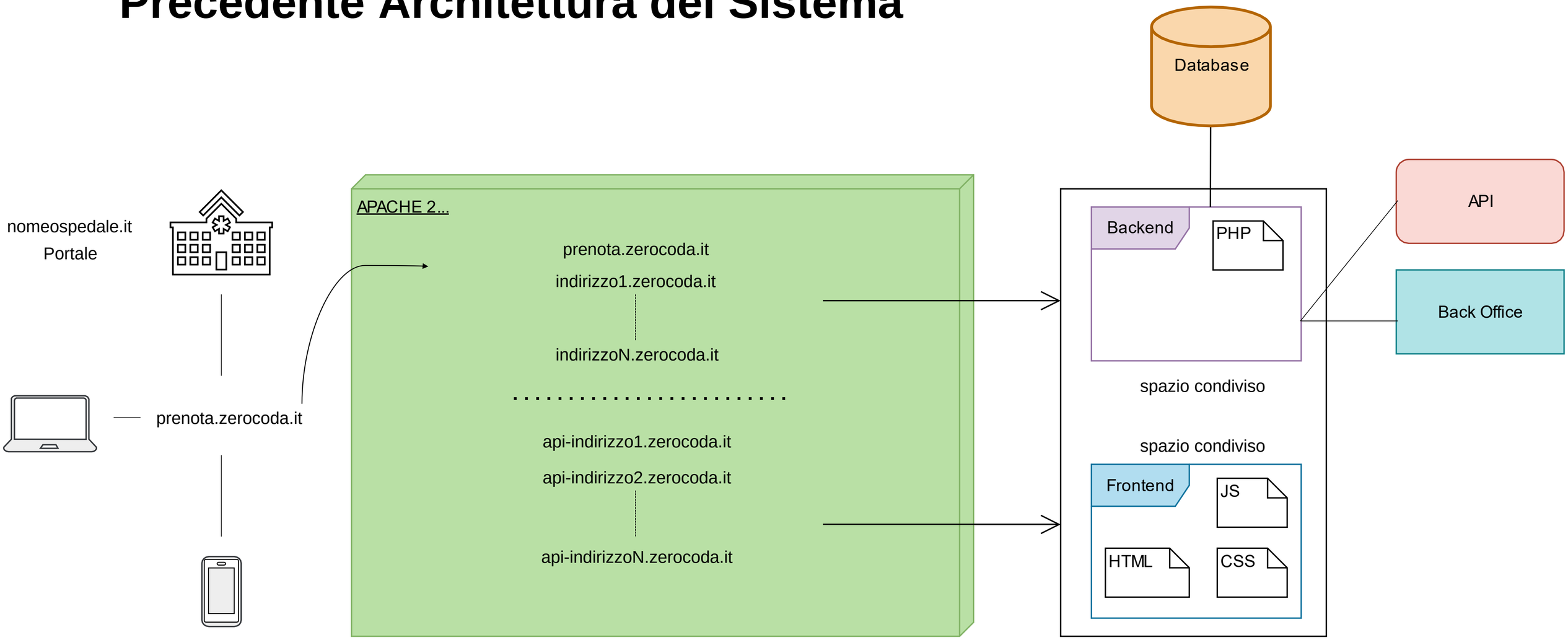


AUMENTO DEL
NUMERO DI UTENZE

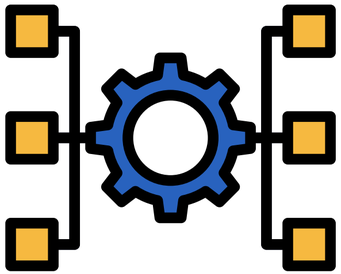
DOCUMENTAZIONE
OBSOLETA

CAMBIAMENTI
COMPLESSI E COSTOSI

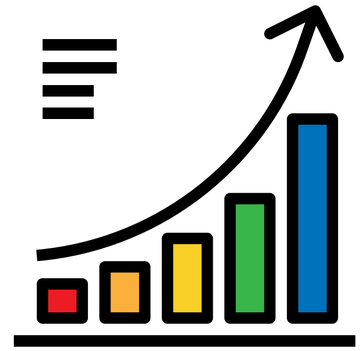
Precedente Architettura del Sistema



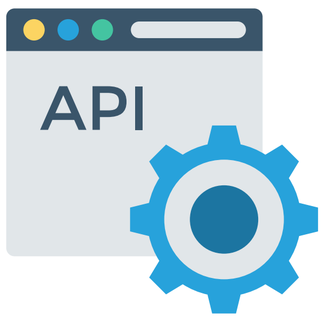
OBIETTIVO



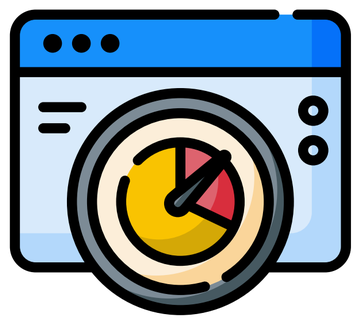
ARCHITETTURA A MICROSERVIZI



AUMENTO DELLA SCALABILITÀ

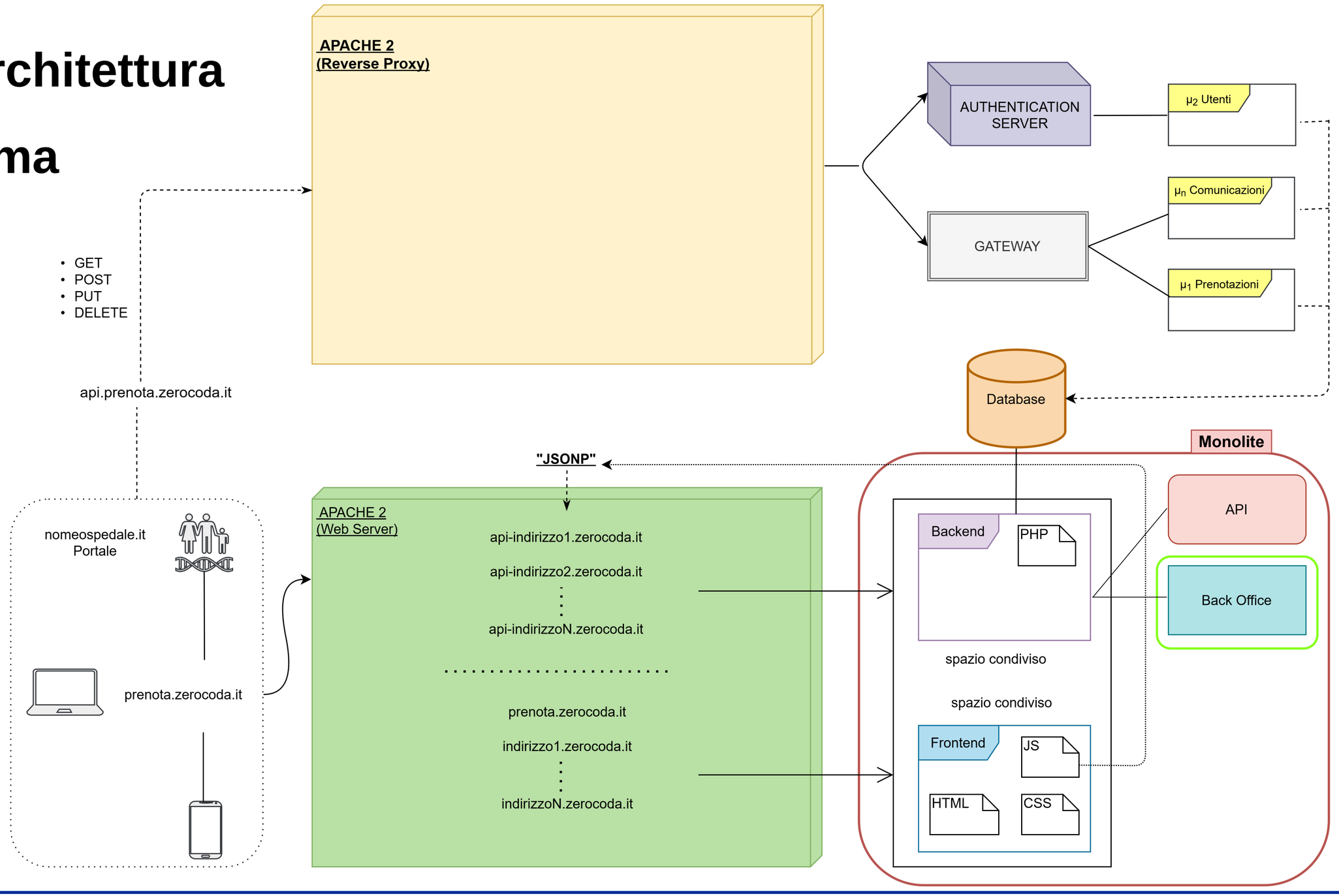


NUOVE API REST



MIGLIORAMENTO DELLE PERFORMANCE

Nuova Architettura del Sistema



Nuove API REST – Booking Server

3 SERVIZI

booking

GET	**/v1/bookings**	Get all user's reservations
POST	**/v1/bookings**	Add a new Reservation for the user
DELETE	**/v1/bookings/{reservationId}**	Delete a user's reservation

calendar

GET	**/v1/calendars/{functionId}**	Get the Calendar of a Facility Service
GET	**/v1/calendars/{functionId}/{day}**	Get the available slots for the day

facility

GET	**/v1/facilities**	Get the available facilities

Documentazione delle API

Swagger

Tool composto da un set di software open source per progettare, creare e documentare *RESTful APIs* attraverso l'*OpenAPI Specification*

1

2

3

4

5

POST **/v1/bookings** Add a new Reservation for the user

Store a new reservation in the database

Parameters **Try it out**

Name	Description
X-Api Id *required integer ($int 32) *(header)*	*X-Api-Id here*
X-Forwarded-For *required string *(header)*	*X-Forwarded-For here*

Request Body *required

```
{
    "slotId": 0
}
```

Responses

```
{
    "number": "A 12",
    "activationKey": "string"
}
```

Code	Description
200	OK

Design Pattern Applicati

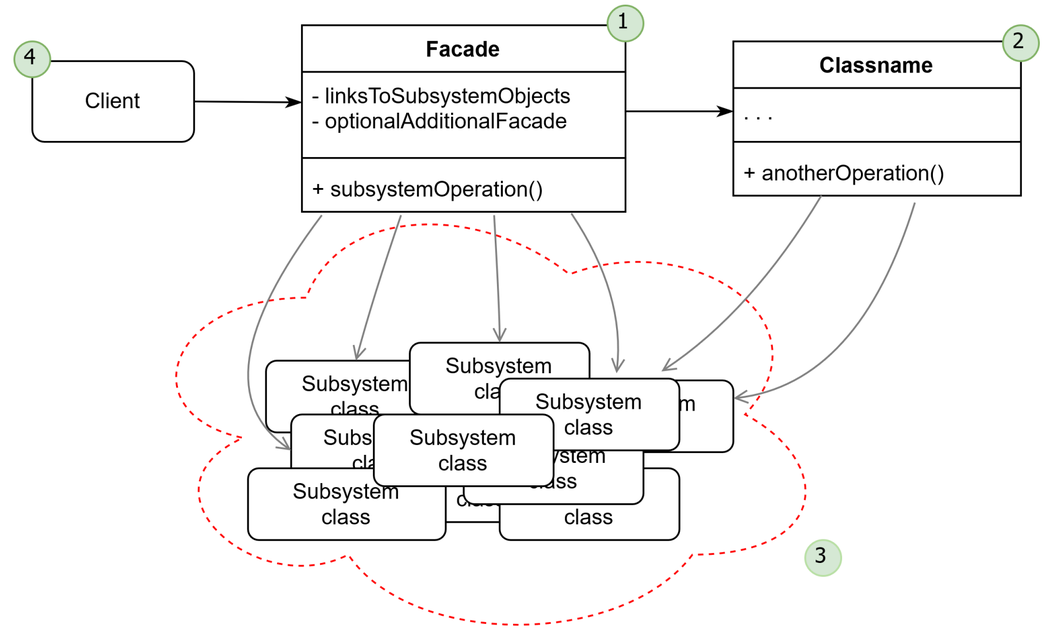
Inversion of Control (IoC)

Il codice viene richiamato dai componenti del framework

↓

Dependency Injection

Façade



Design Pattern Architetturali

DATA TRANSFER OBJECT
(DTO)

Utilizzato per trasferire dati

tra sottosistemi di

un'applicazione software

DATA ACCESS OBJECT
(DAO)

Disaccoppia il server

dall'accesso al database

Framework Utilizzati

GESTIONE DELLA COMPLESSITÀ DEL SOFTWARE

Implementazione della IoC

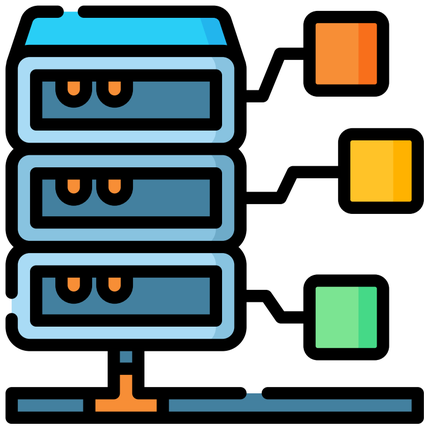
mediante Dependency Injection

MAPPING METODI JAVA E QUERY SQL

Query SQL a carico

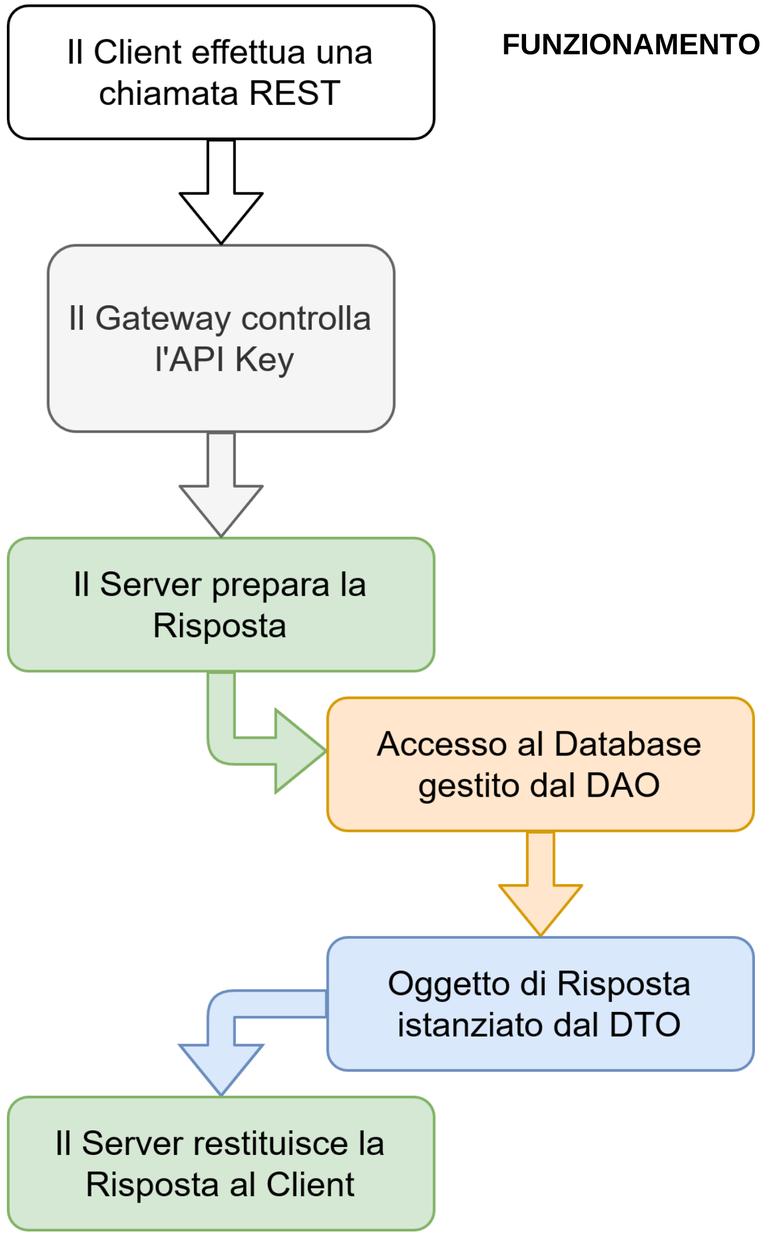
del programmatore

Booking Server



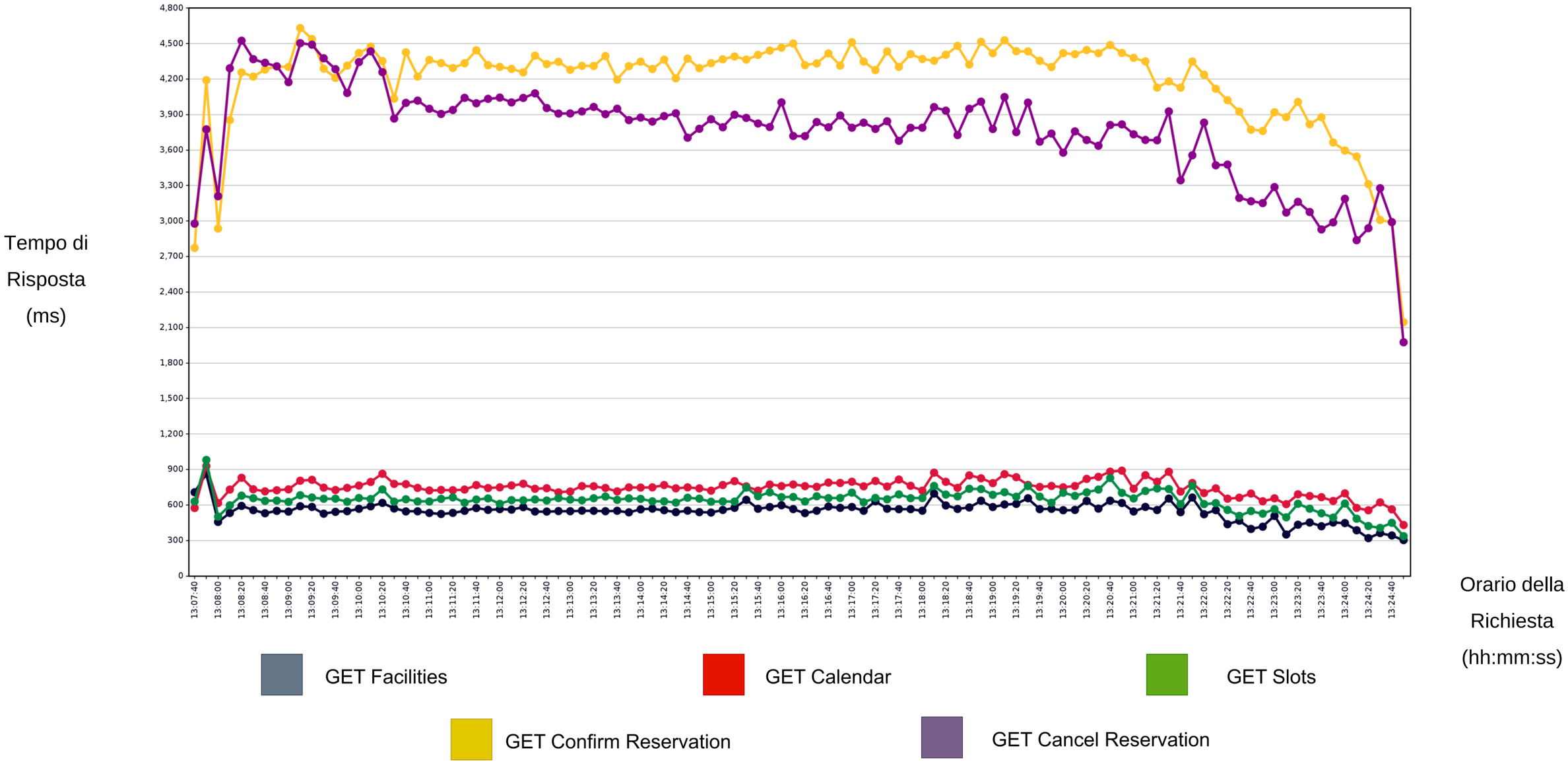
BOOKING SERVER

Specifiche OpenAPI e classi service che implementano le chiamate REST

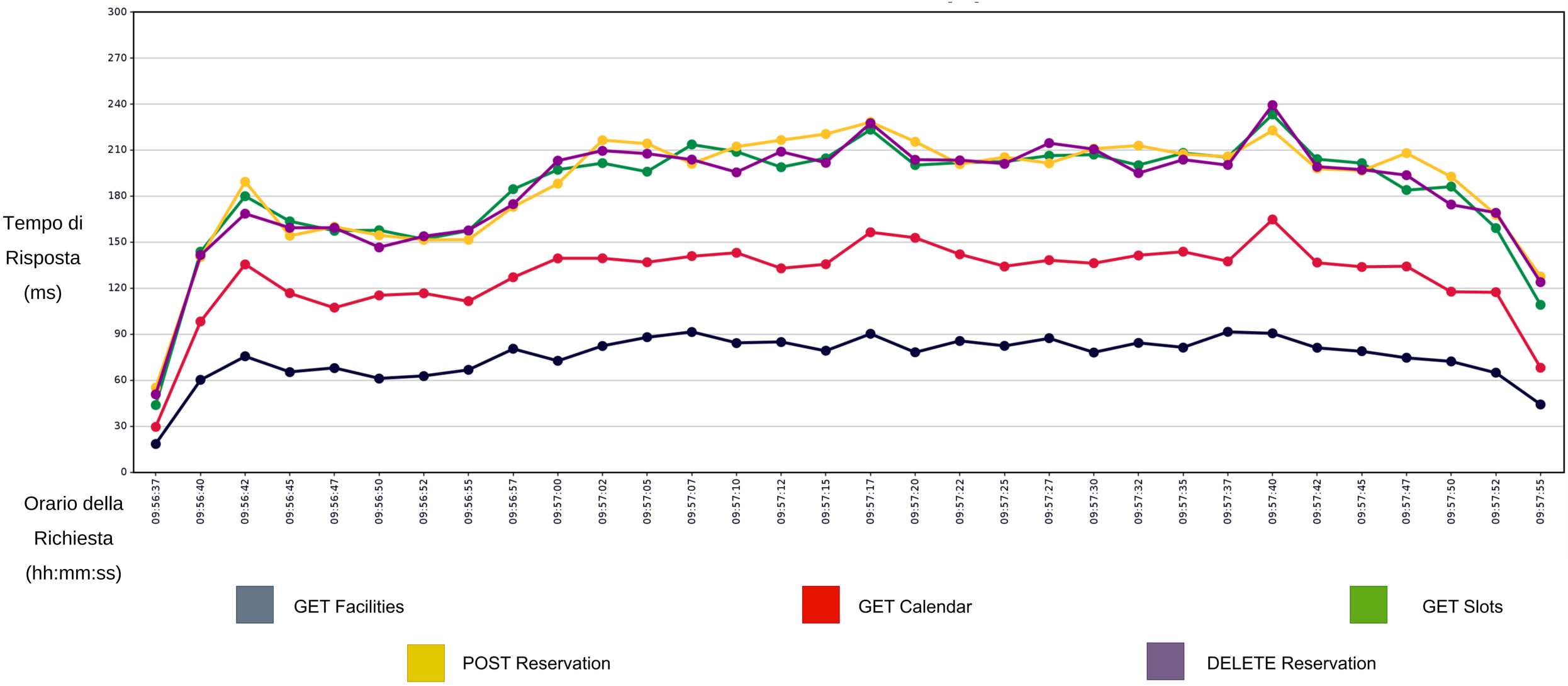


FUNZIONAMENTO

Test – Vecchie API



Test – Nuove API



Test – Risultati Ottenuti

- Tempo di esecuzione dei thread delle nuove api quasi 20 volte minore
- Incremento della velocità di risposta in tutte le richieste
- Valori di picco meno frequenti e più controllati
- Netto incremento del valore di throughput
- Aumento globale delle performance

Perché Questo Miglioramento?

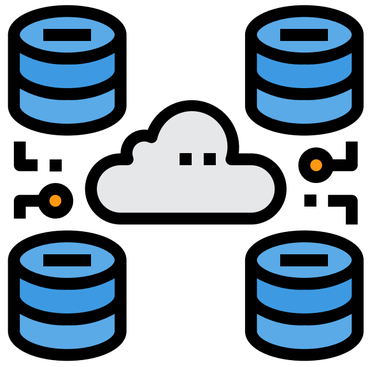
A ciascuna nuova richiesta di accesso al database corrisponde una **nuova connessione**

Richieste di accesso al database mantenute attraverso una **connection pool**

Sviluppi Futuri

ADATTAMENTO DEL FRONTEND

INSTALLAZIONE



REFACTORING DEL DATABASE

Grazie per l'attenzione.