

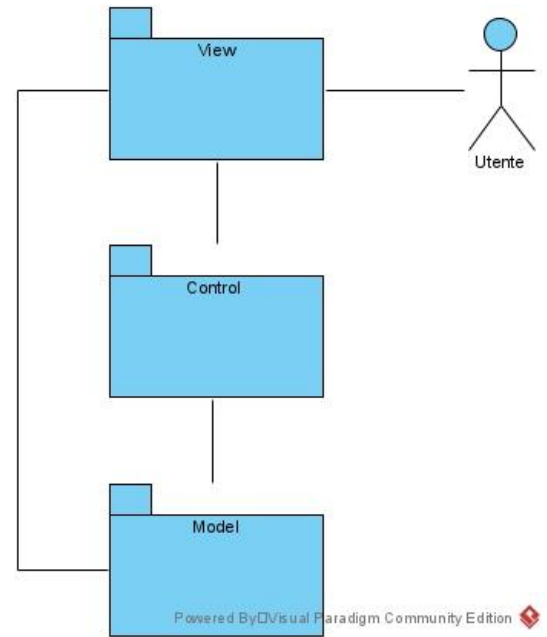
3.2 Decomposizione in sottosistemi

La decomposizione adottata per il sistema è il pattern MVC composto da tre layer, ognuno dei quali gestisce determinate funzionalità:

- **Controller:** trasforma le interazioni dell'utente con la View in azioni che possono essere eseguite dal Model;
- **Model:** definisce la gestione dei dati e la gestione della logica del sistema, cioè le operazioni che possono essere eseguite;
- **View:** gestisce l'implementazione dell'interfaccia grafica, la quale rappresenta il mezzo tramite il quale gli utenti possono interagire col sistema.

Dopo attente valutazioni, abbiamo ritenuto opportuno adottare questo tipo di architettura per il nostro sistema poiché lo migliorerebbe sia in termini di efficienza che di manutenibilità.

Per via dei criteri mostrati nella sezione dei Design Trade-off, e quindi dalle esigenze del nostro sistema, tale decomposizione mira alla suddivisione delle componenti, facendo sì che risultino con un basso accoppiamento e un'alta coesione.



Component Diagram

Nella gestione delle componenti, ogni layer ha in gestione determinate componenti, di seguito esposte.

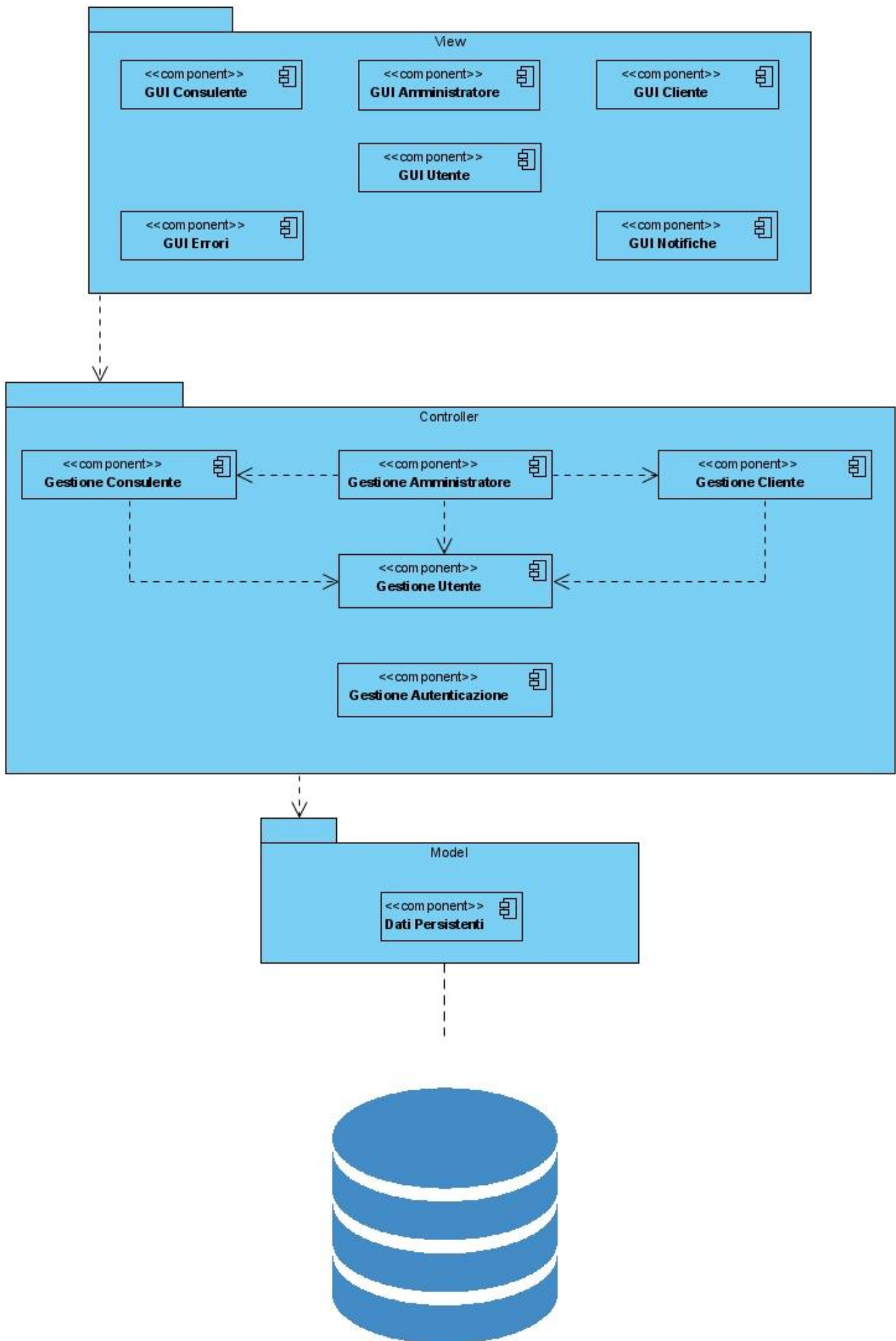
Il layer **Controller** gestisce cinque componenti:

- Gestione Amministratore
- Gestione Consulente
- Gestione Cliente
- Gestione Auto
- Gestione Autenticazione

Il layer **Model** gestisce una sola componente, cioè quella che si occupa di gestire i dati persistenti nel sistema, interfacciandosi con il Database.

Il layer **View** gestisce sei componenti:

- GUI Amministratore
- GUI Consulente
- GUI Cliente
- GUI Utente
- GUI Errori
- GUI Notifiche



Deployment Diagram

L'utente, tramite un browser compatibile con l'interpretazione delle funzionalità di JavaScript, può richiedere le funzionalità del sistema tramite l'interfaccia presente sul Web. Le richieste dell'utente vengono ricevute ed elaborate poi dal sistema tramite il modello MVC, che si occupa di comunicare con il Database ed i dati persistenti.

