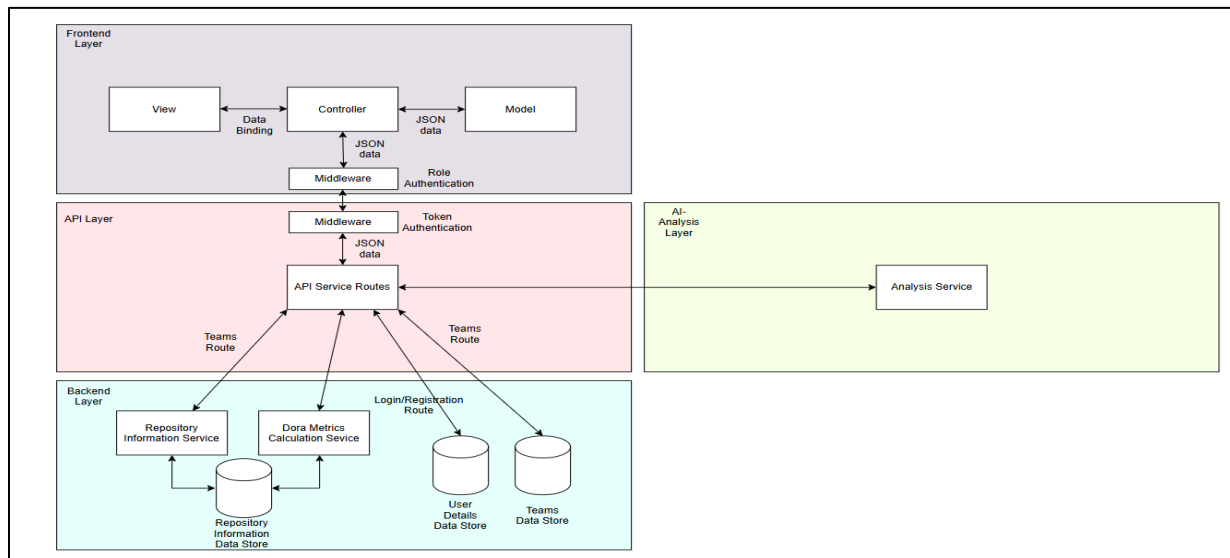


# DEVX360

Sipho Sehlapelo  
Kesley Hamann  
Sibusiso Mngomezulu  
Lwando Msindo  
David Musa-Aisien

## ARCHITECTURE DIAGRAM

This document presents the system's architecture diagram, illustrating how the frontend, API layer and backend interact. User requests flow from the frontend through the API, which routes them to the appropriate backend service or AI analysis, ensuring clear separation of concerns and controlled communication between components.



The system makes use of three architectural patterns namely MVC (Model-View-Controller), N-Layered Architecture and Component-Based architecture.

### MVC

The Frontend architecture makes use of MVC. It stores data received from the API – including dashboard DORA metrics, user profile data, and the optional AI analysis. Data can be modified using the login component and profile management component residing in the controller layer. The profile management component allows users to update their profile information, manage team memberships, and for team leaders, create new teams. The view layer consists of the dashboard and profile components, which simply present the data from the model to the user in an organized and accessible format.

### N-Layered Architecture

Our system follows an N-layered architecture with four distinct layers: frontend, API, backend, and AI analysis. The frontend implements an MVC structure to manage user interaction and presentation. The API acts as a central unit, exposing well-defined service routes for each request, ensuring seamless communication between layers. The backend is responsible for calculating and storing DORA metrics as well as user information using its

database functionality. The AI analysis layer consumes this backend data, performing advanced analysis to generate feedback and actionable insights.

### **Component-based Architecture**

The backend contains modularized capsules that carry out specific functions. The API may call upon specific component services to carry out functions based on requests from the frontend.