# ARCHITECTURE PROPOSAL.

**Database**

SQL Server database called “Healthcare”.

A secured connection to a failover system is recommended for high availability and fault tolerance.

**Back-end**

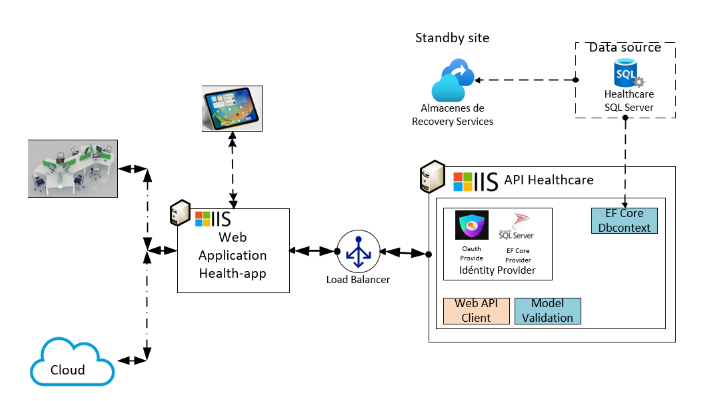
The back-end called “HealthAPI” is deployed in IIS on its own server and is protected with domain security policies. The Api service is implemented using .NET Framework 9 and is divided into the following layers:

**HealthcareAPI**: Controls client requests and operates as the presentation layer of the REST API. It includes a method to store traces and errors in a physical path called Log.

**Healthcare.BR**: It has business logic and validations.

**Healthcare.Data**: It has direct connection to the database.

**Healthcare.Common**: Cross-cutting layer for common utilities.



**Font-end**

Web application developed in Angular. It consumes the back-end APIs through the load balancer. The load balancer is recommended in case of high demand, this includes having another server for the API service.

The web application is deployed on another IIS server keeping separate back-end and front-end resources.

To conclude, you keep in mind that the separation of functions not only increases costs but also increases the complexity of configuration and maintenance. It is necessary to say that handling separate servers by functionality in addition to the load balancer ensures that the system can handle high traffic demands and gives the possibility to add more servers in case of growth.