# Sensor Readings MVP

MVP: Using SignalR to send and receive 1000 datapoints/second and plot it to a graph.

**MVP Architecture:**Angular front end with SignalR library

Dot net core service with broadcast of data

Single page application as MVP for demonstrating sending and receiving up to 1 lakh entries

**Product Architecture.**

**Front end**

Standalone angular app to display data (based on requirement)

**Backend**

Dot net core service listening to sensor data continuously

**Scenario 1** – Arranged and collected sensor data

In case there is an existing mechanism where all the sensor data (1000 datapoints) can be communicated to the dot net service then no extra effort is required

**Scenario 2** – Data from different devices.

In this case a mechanism has to be in place to accumulate all the sensor data and to communicate it to the dot net service. (Separate MVP if required)

**Front end app architecture.**

Components: Dashboards, History etc based on requirement

Services: SignalR Service(for real time data), Api Service(rest of the functionalities)

UI Layer: Chats etc

**Backend app architecture:**

Middleware and Filters for data validation and security based on requirements

Controllers – Sensor/SignalR Hub(for receiving and broadcasting info), Other Api Endpoints

Services – Sensor Service for saving data

Cache Service – For temporary retention for broadcast if required

Infrastructure: Entity framework core and sql server

Thread based segregation of receive – broadcast and saving data.

**Other features and considerations:**

Implementing SignalR with background service:  
[background service with SignalR](https://learn.microsoft.com/en-us/aspnet/core/signalr/background-services?view=aspnetcore-9.0)

Performance and scalability:

[SignalR Performance](https://learn.microsoft.com/en-us/azure/azure-signalr/signalr-concept-performance)