**Design Solution:**

There are 3 main components in the solution.

1. Push raw data for machines
2. Refine Raw data
3. Consume Refine Data

* Pushing Raw Data for Machines:

1. There would be two separate tables for Repair and Session entities.
2. Session Entity:
   1. Machine recordings will be pushed on Event Hub bus such as Kafka, Service bus every 1 hour by IOT devices.
   2. Scheduler will keep on polling the data from event hub bus and save into NoSQL.
3. Repair Entity:
   1. Source data will be sent to the Event Hub bus.
   2. Scheduler will keep on polling the data from the event hub bus and save into NoSQL.

* Refine Raw Data
  + Prerequisites:

1. Show machine feeds (Repair and session) on UI which contains less data
2. Data should be sorted based on date
   * Logic:
3. Refine the data from Raw Data tables Repair and Session
4. Keep few data columns to show on UI for index purpose
5. Scheduler will run in every say 1 hour to fetch data.
6. Save the data sorted based on date into ElasticSearch or Cosmos DB

* Consume Refine Data

1. Send paginated data since there could be millions of data.

**Architecture Pattern:**

1. Architecture would be event driven and loosely coupled between raw data and processed data
2. Microservices pattern would be for scalable and loose coupling. Ex: Web API

**Technologies**

1. Front End Technology: Angular and React
2. BackEnd Technology:
   1. WebAPI Core
   2. ADO.Net: Connect between DB and API Logic
3. Database: NoSQL for Raw and Processed Data
4. Cache: Redis for Azure

**Cloud Deployment:**

1. App service for Web App and Web API
2. APIM for API Management
3. Cosmos DB for low latency and high volumn