**Solution 1(Local Server)**:

This solution fits both ERP-A and ERP-B. The only way to access data in ERP-A is direct access to DB. And if there is no access to Mssql DB from Clould we cann not use colud solution for ERP-A. in that case each solution must be locally not cloud:

Three-tier Architecture

* Data Access Layer (Database/ REST API)
* Business-Server Layer (Java)
* Presentation Layer (web-vue.js)

**Data Access Layer**

Interface with two implementations:

* ERP-A: Hibernate/JAP with direct access to MSSQL database on-premise
* ERP-B: Make customers and orders queries though REST API

**Business-Server Layer**

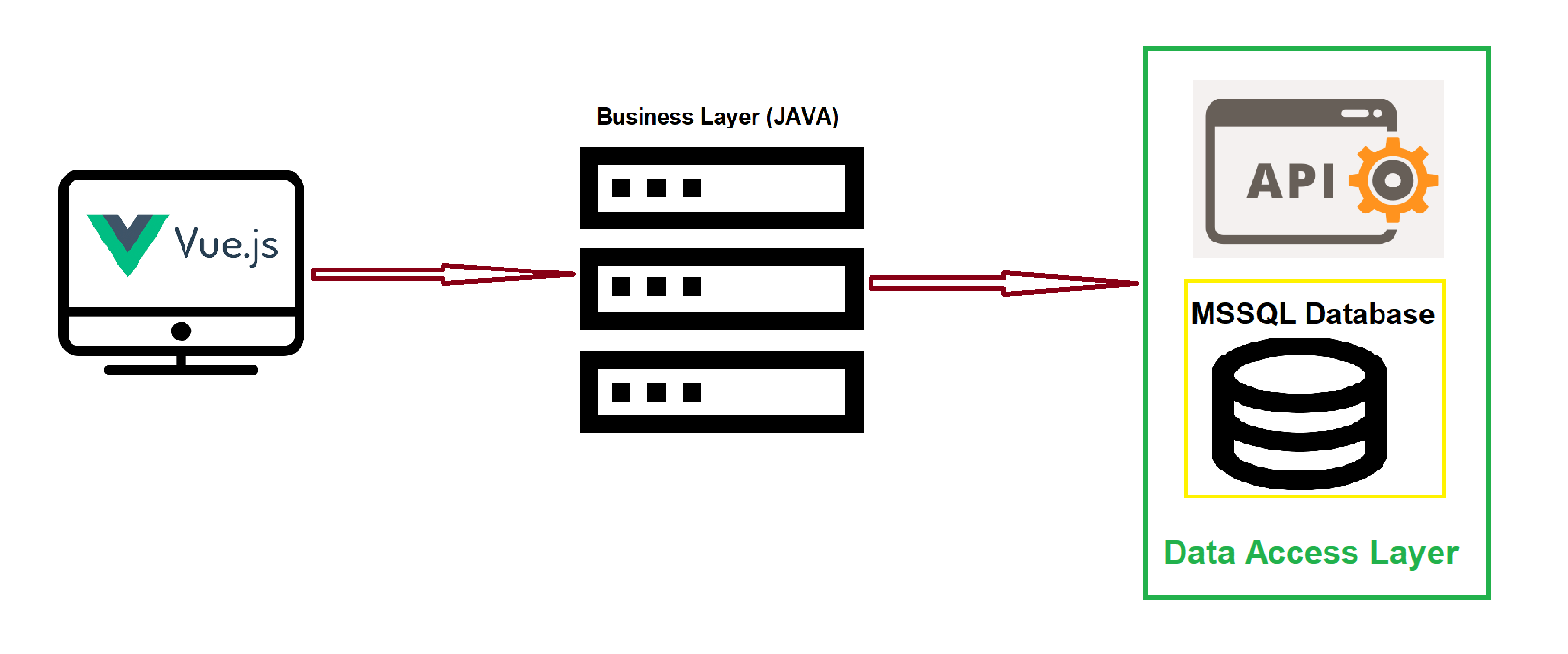
Java Web-Application make

* Interpretation the requests of UI
* Decide to use with implementations of Data Access Layer based on application runtime settings.
* Catch the customers and orders data through Data-Access-Layer
* Prepare the response fort he UI

**Presentation Layer**

A light web interface with vue.js to:

* intract with the users.
* Gathering the request and send to Server.
* Represent the server-response as a web rich format fort he users

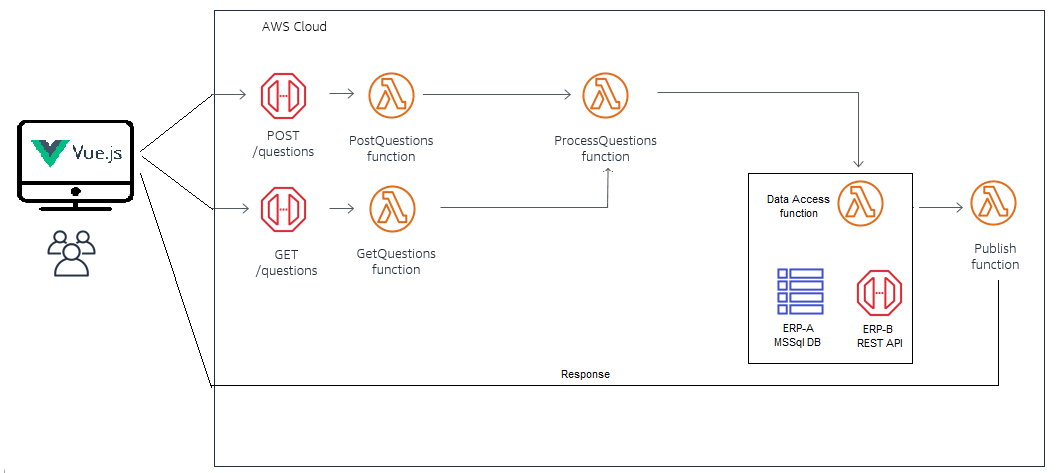


**Solution 2(Cloud)**:

If there is access from could to MSSql from ERP-A we can use could solution:

Serverless solution based on AWS Serverless Application Model (AppSync/Lamda)





**AppSync**:

Entrypoint for the request’s (GET/POST) and Access-Point fort he response in cloud

**Lambda**:

1. Process request’s (GET/POST).
2. Decide about how to make the query.
3. Decice to make query through Database (ERP-A) or REST API (ERP-B)
4. Prepare and publish the response tot he AppSync

**Presentation**:

A light web interface with vue.js to:

* intract with the users.
* Gathering the request and send to Server.
* Represent the server-response as a web rich format fort he users