## Exercise 3

Describe in words how you can extend the solution of Exercise 1 to help Oscar manage his wine warehouse and inventory.

Here reported some improvement to permit to Oscar to manage his wine warehouse and inventory

As first step to decrease the human intervention on standardized, well-defined and recursive activities we can:
- digitalize the calculation of the stocks for the warehouse using the information stored into the DB
- make automatic via the Application the comunication of all reservations received by the e-commerce to the warehouse

The data model have to include the followings tables to maintain Warehouse's data:

- **WHAREHOUSE**: WINE_ID, AMOUNT, LAST_UPDATE

  WINE_ID identify the wine
  AMOUNT represent the amount of that specific wine into the Warehouse
  LAST_UPDATE is the moment of the insert of the row and represent the moment of the last check into the Warehouse for that specific wine

- **PROVIDER**: ID, PHONE, DATE_OF_JOIN, LAST_UPDATE

  ID row's identification value
  PHONE of the provider
  DATE_OF_JOIN date of the first add of the row
  LAST_UPDATE date of the last update of the provider's data

- **WINE_SUPPLY_ORDER**: WINE_ID, AMOUNT, PRICE_CAD, TOTAL, CURRENCY, DATE_OF_ORDER, EAT, PROVIDER_ID

  AMOUNT represent the amount of the supply order
  PRICE_CAD the price for bottle
  TOTAL the result of PRICE_CAD * AMOUNT (column added to simply data analysis)
  CURRENCY the currency used in the order (ex: EUR, USD, GBP)
  EAT is the estimated arrival time announced by the provider of the order
  PROVIDER_ID identify the provider for the order

- **PERIODICAL_ORDER**: WINE_ID, AMOUNT, PRICE_CAD, TOTAL, CURRENCY, THRESHOLD

  THRESHOLD the threshold set to determine when the batch have to prepare a new order based of the periodical order information

The Application should provide the followings functionalities:

- GET /showWarehouse(Filter filter): Rest resource that will return the state of the warehouse in terms of: wine's amount, customer's and supply orders disposal. The functionality have to give the as-is state of the Warehouse and also to compute the hypothetical state in a X time in the future using the information provided by the provider of the supply order

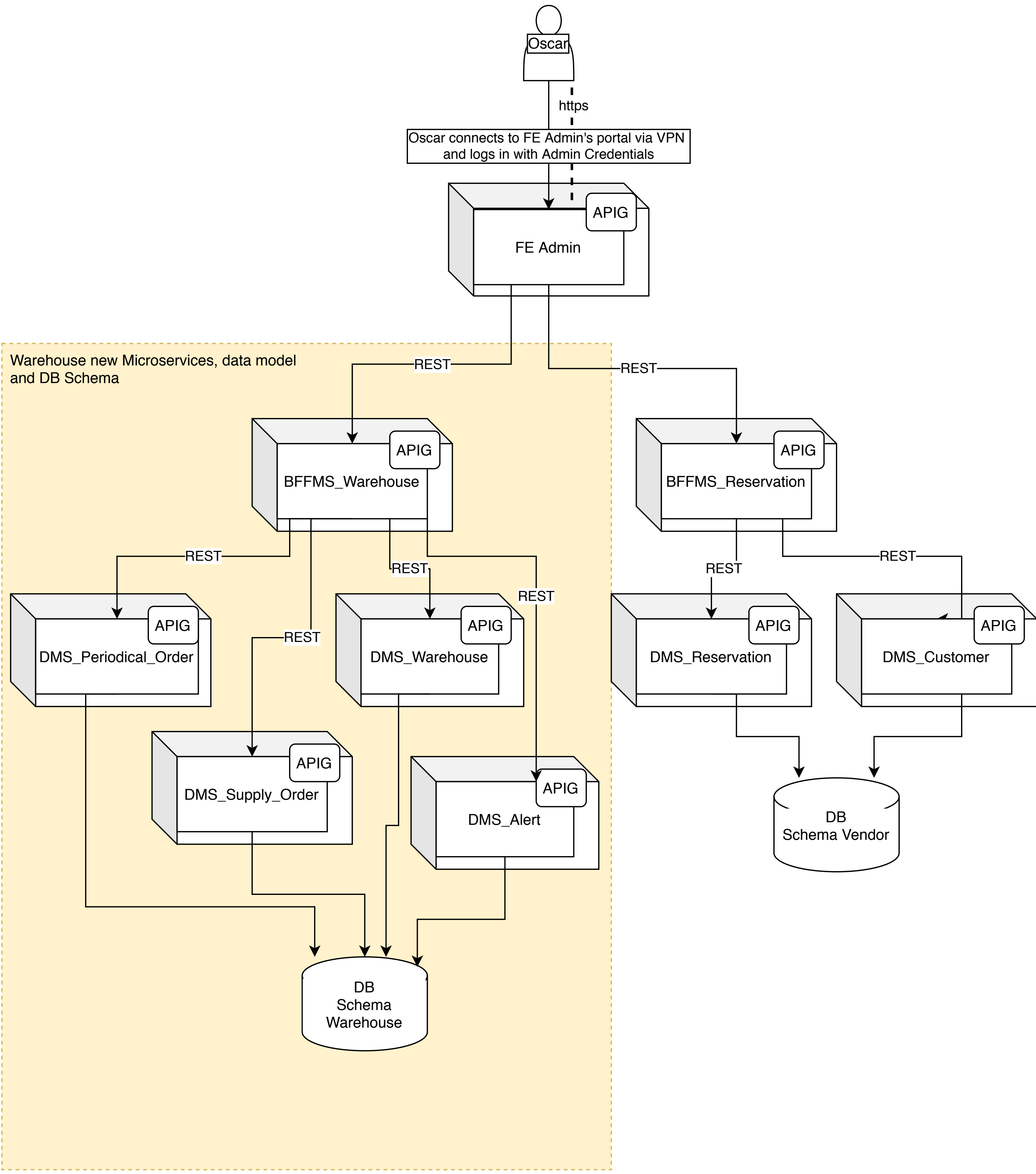
- POST /estimateNexySupplyOrders(Filter filter): Rest resource that return a list of possible supply orders to maintain constant the supplying for determined wines. By the filter and the information retrieved from the DB the Application will help Oscar to be ready for the peak of customers' orders

- PUT /addSupplyOrders(SupplyOrder supplyOrder): Rest resource to add information about a supply order
- POST /updateSupplyOrders(SupplyOrder supplyOrder): Rest resource to update information about a supply order
- GET /listSupplyOrders(SupplyOrderFilter soFilter): Rest resource to get list of supply orders' information
- DELETE /deleteSupplyOrders(SupplyOrder supplyOrder): Rest resource to delete information about a supply order

- PUT /addPeriodicalOrders(PeriodicalOrder periodicalOrders): Rest resource to add periodical order
- POST /updatePeriodicalOrders(PeriodicalOrder periodicalOrders): Rest resource to update periodical order
- GET /listPeriodicalOrders(PeriodicalOrderFilter poFilter): Rest resource to get list of periodical orders
- DELETE /deletePeriodicalOrders(PeriodicalOrder periodicalOrders): Rest resource to delete periodical order

- PUT /addAlert(Alert alert): Rest resource to add alert
- POST /updateAlert(Alert alert): Rest resource to update alert
- GET /listAlert(AlertFilter alertFilter): Rest resource to get list of alert
- DELETE /deleteAlert(Alert alert): Rest resource to delete alert

- A batch functionality "sendAlert" that analyze all the data stored on the DB to check if any of the rules set for the alerts is violated, in that case the batch sends an alert to Oscar with the information gathered


Oscar
https
Oscar connects to FE Admin's portal via VPN and logs in with Admin Credentials
APIG
FE Admin
Warehouse new Microservices, data model and DB Schema
REST
REST
APIG
BFFMS_Warehouse
APIG
BFFMS_Reservation
REST
REST
REST
REST
REST
REST
APIG
DMS_Periodical_Order
APIG
DMS_Warehouse
APIG
DMS_Reservation
APIG
DMS_Customer
APIG
DMS_Supply_Order
APIG
DMS_Alert
DB Schema Vendor
DB Schema Warehouse