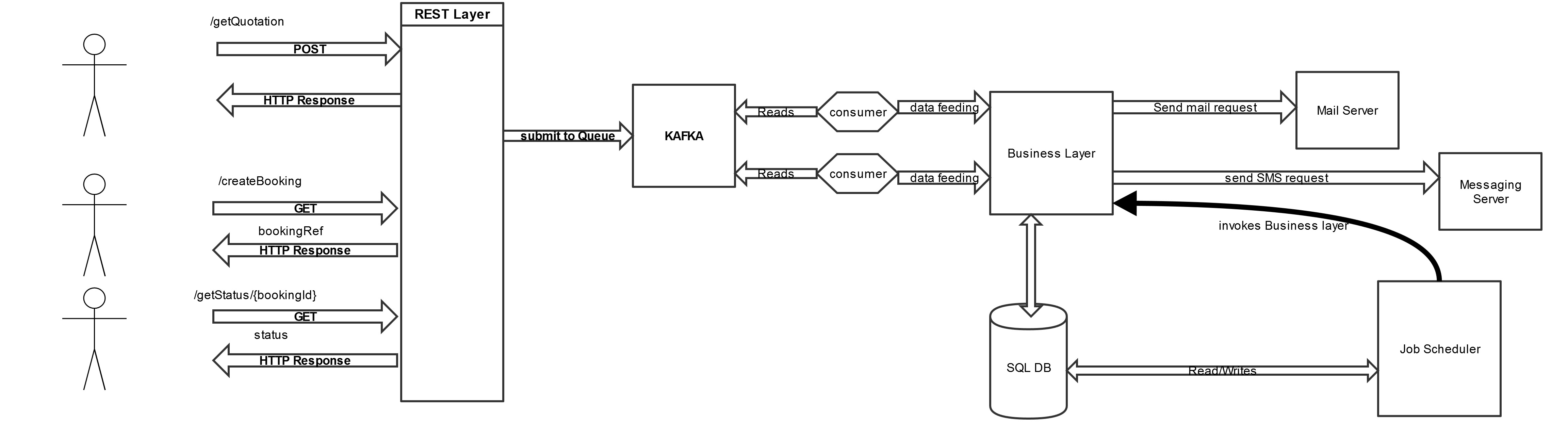
**Design Architecture**

*System Architecture:*

**

*Module Description*

1. **Rest Layer:** This is the first layer which will be interacting with Business. It will expose 3 APIs :

**a) POST /api/v1/quotation** to get the quotation

**Sample Payload:**

{

"businessId": "1",

"typeofGood" :"glass",

"noofUnit": "2",

"weight": "3",

"typeOfBooking":"fragile"

}

**OutPut:**

**200OK response code**

**b) GET /api/vi/booking?quotationId=** to confirm the booking

**c) GET /api/vi/booking/status?bookingId=** to get the status of booking.

**Output:**

**200OK response code**

Rest Layer will put the incoming request into kafka messaging queue for asynchronus processing.

**2. Kafka Consumer :** Kafka Consumer will listen to incoming requests.

If it is get Quotation request, it will interact with mysql db to get the price based on payload and mail the quotation to business. Redis cache can be used to improve performance so that query can be avoided.

If it is to confirm booking, it will update the warehouse entity.

**3. Job Scheduler:** A job will run at 12 PM everyday and check warehouse entity to get list of all good to be delivered. It will have logic to find the optimal route based on type of delivery chosen by business and category of goods. Based on that it will assign the goods to transport system .

**4. Mail Daemon:** Once the delivery is done, business will be sent mail regarding confirmation.

**5. FSM for Booking:** Once the booking is received, state of the booking will be maintained. And same will be used to server

*DAO Entities*

1. Goods
2. Category
3. Warehouse
4. Price
5. Booking