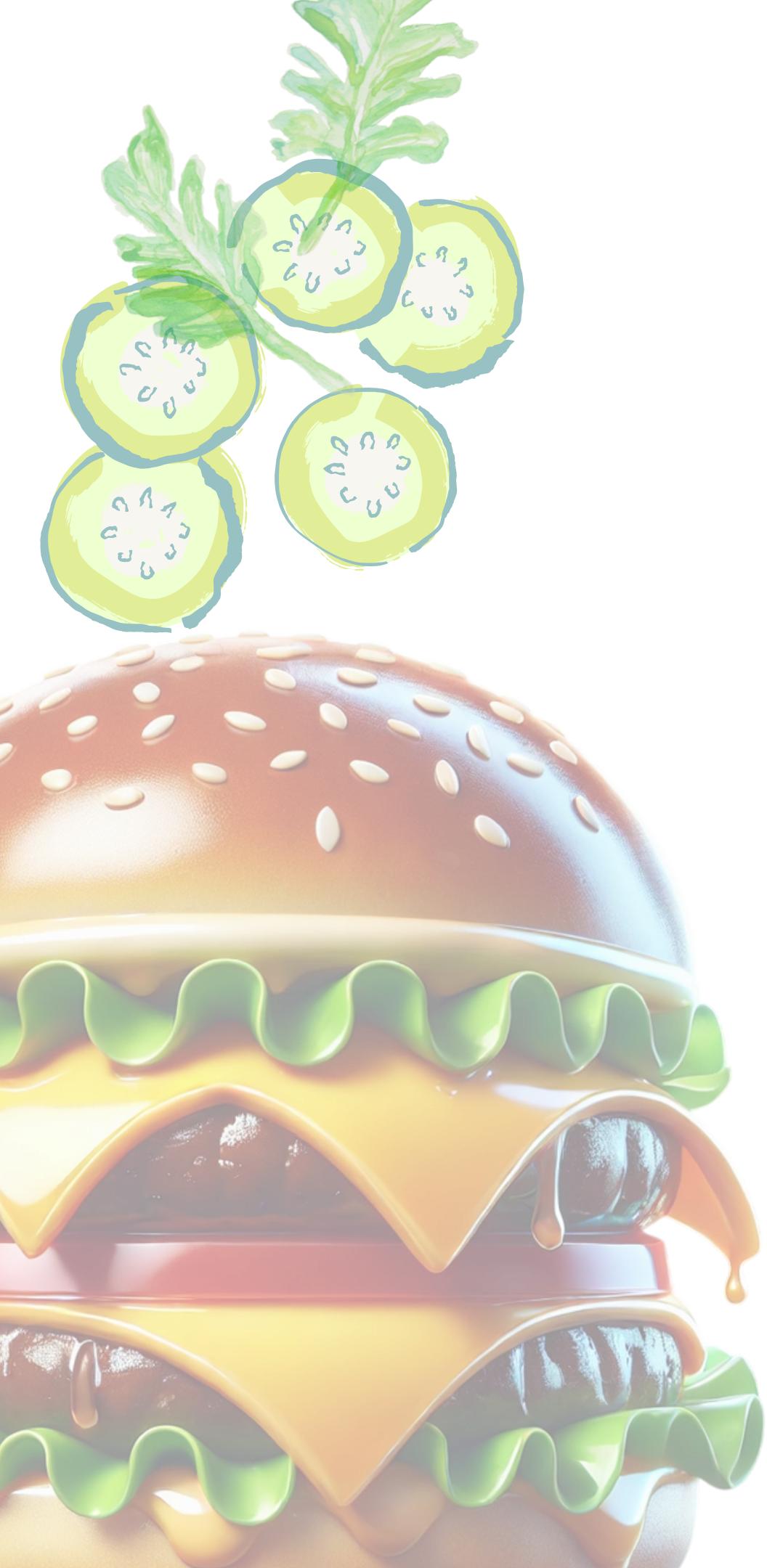




Food Delivery Application

Rider Side	Restaurant Side	Customer Side
Expected Functionalities		
<ul style="list-style-type: none"> • Create and edit rider accounts. • Rider login with credentials. • Update location and availability for deliveries. • Accept delivery requests from restaurants. • Collect food from restaurants. • Deliver to customers and update delivery status. 	<ul style="list-style-type: none"> • Create and edit restaurant profiles. • Restaurant login with credentials. • Manage menus by adding, editing, or deleting food items. • Accept customer orders. • Request riders for deliveries. 	<ul style="list-style-type: none"> • Create and edit customer profiles. • Customer login with credentials. • Search for restaurants or food items. • Place food orders. • View payment amount and pay the rider upon delivery.
Additional Functionalities		
<ul style="list-style-type: none"> • Real-Time Live Location Updates Interactive Map View • Path finding Using A Algorithm* • Concurrency Handling in Request Acceptance 	<ul style="list-style-type: none"> • Password recovery functionality. • Real-time order notifications using WebSocket. • Caching for restaurants, menus, and menu items with Redis. 	<ul style="list-style-type: none"> • Track orders with rider live location • Order management with real time order status

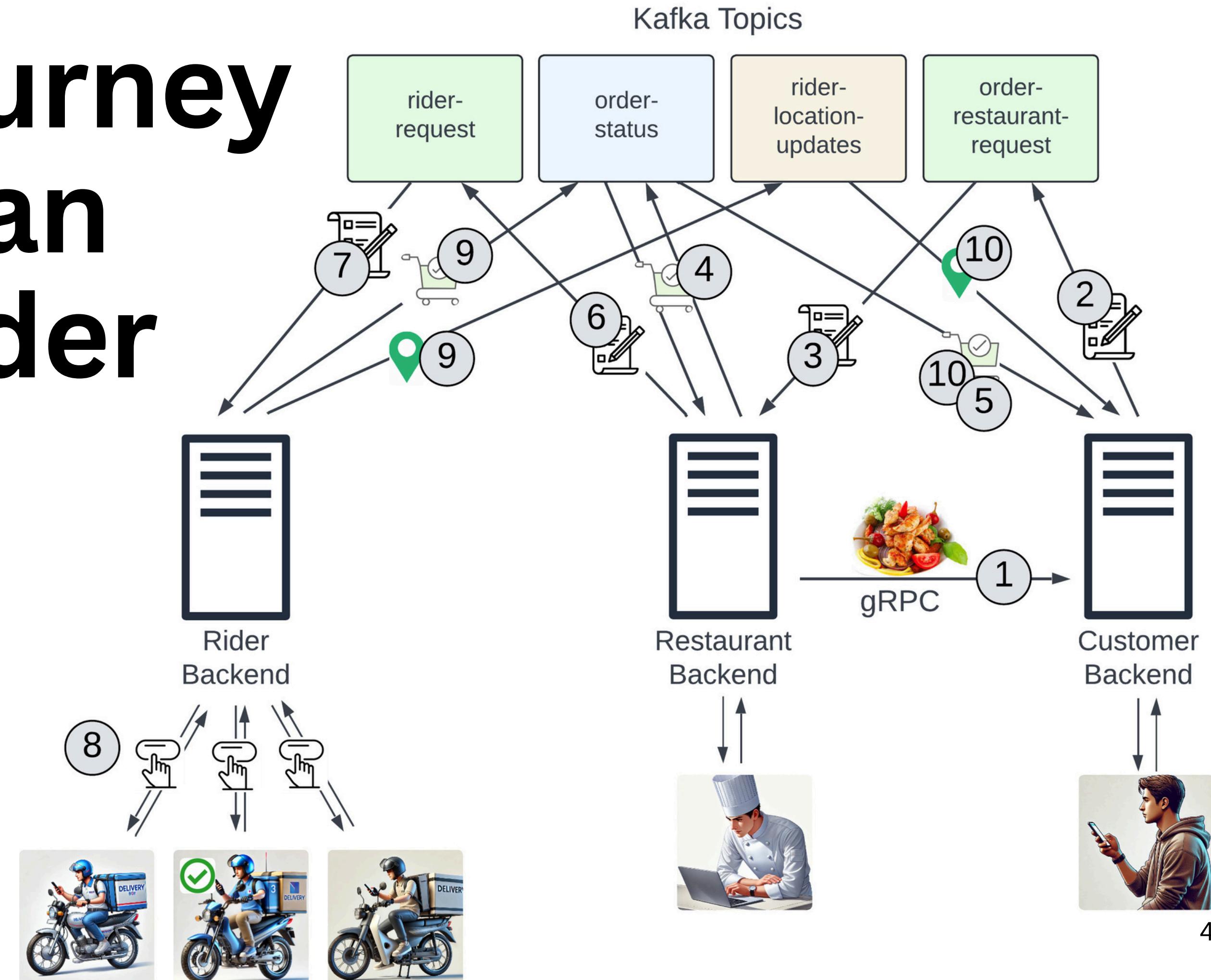


Assumptions

- **Single Restaurant Orders**
 - Each order can only be placed from one restaurant.
- **Rider Availability**
 - The rider is visible on the map page and is ready to accept orders.
- **Simulated Movements**
 - Rider movements are demonstrated using a predefined array of locations on the map.
- **Order Acceptance Rule**
 - After accepting an order, the rider cannot cancel it under any circumstances.

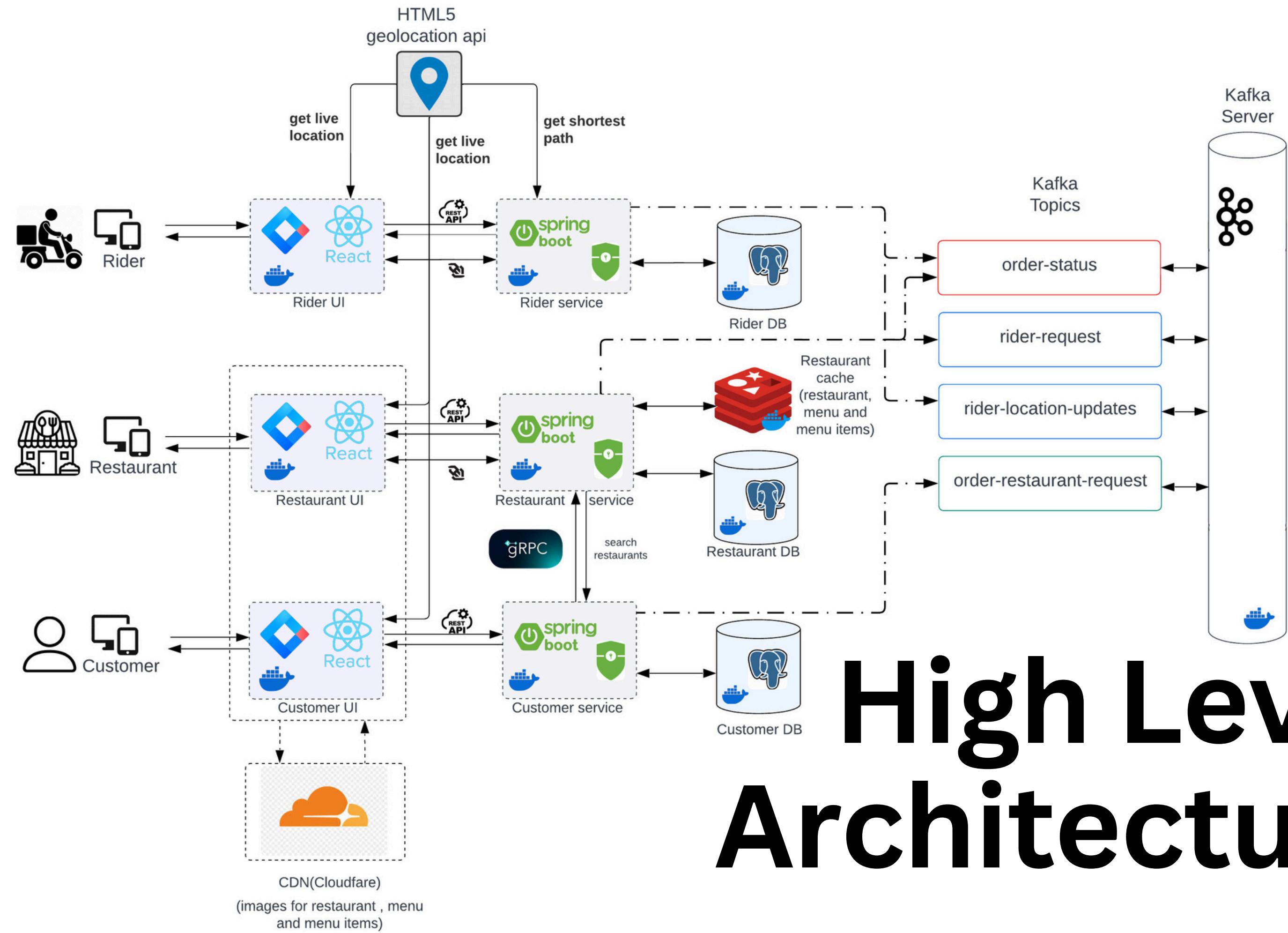
Kafka Topics

Journey of an Order

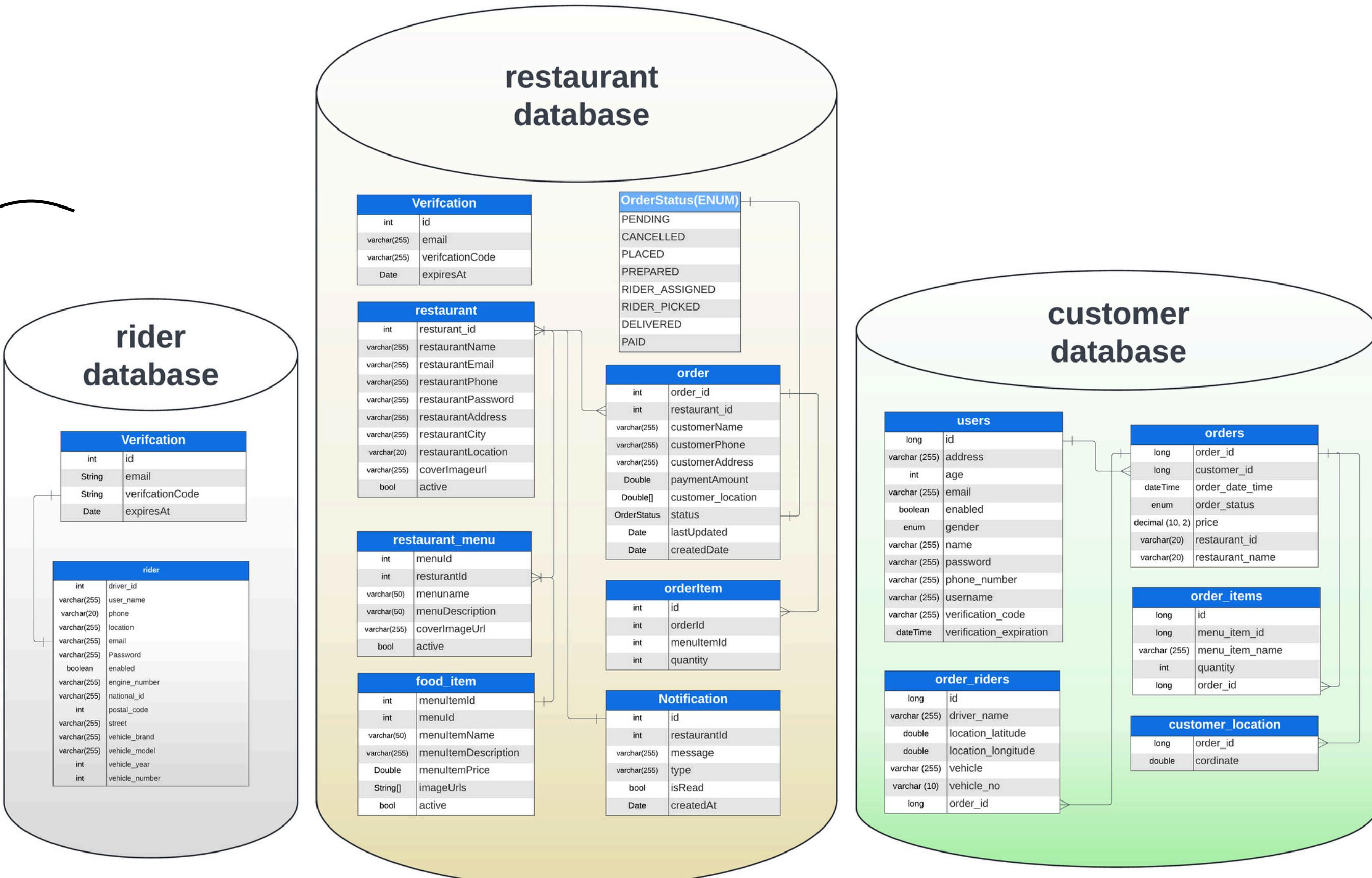


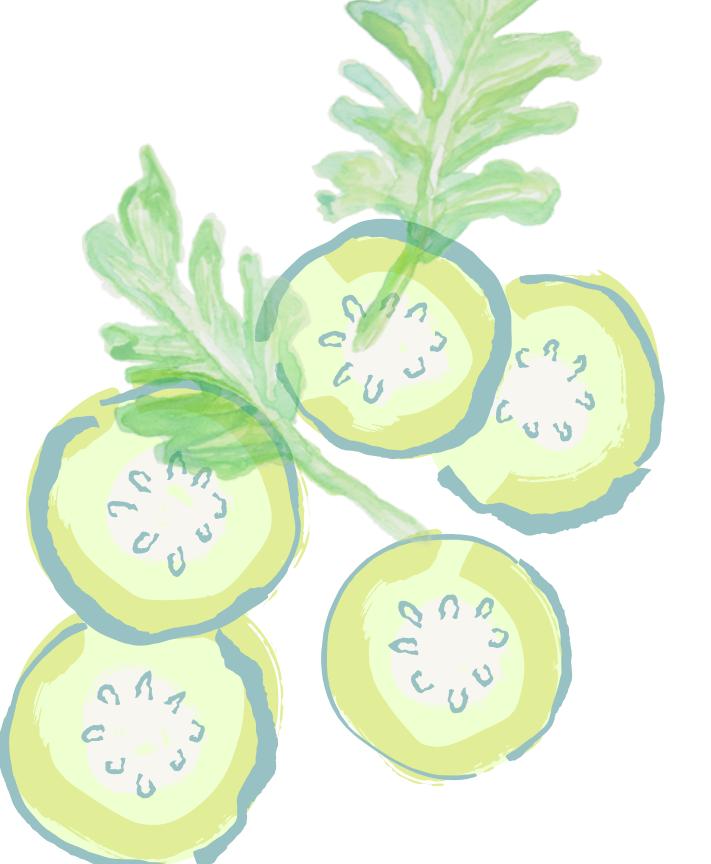
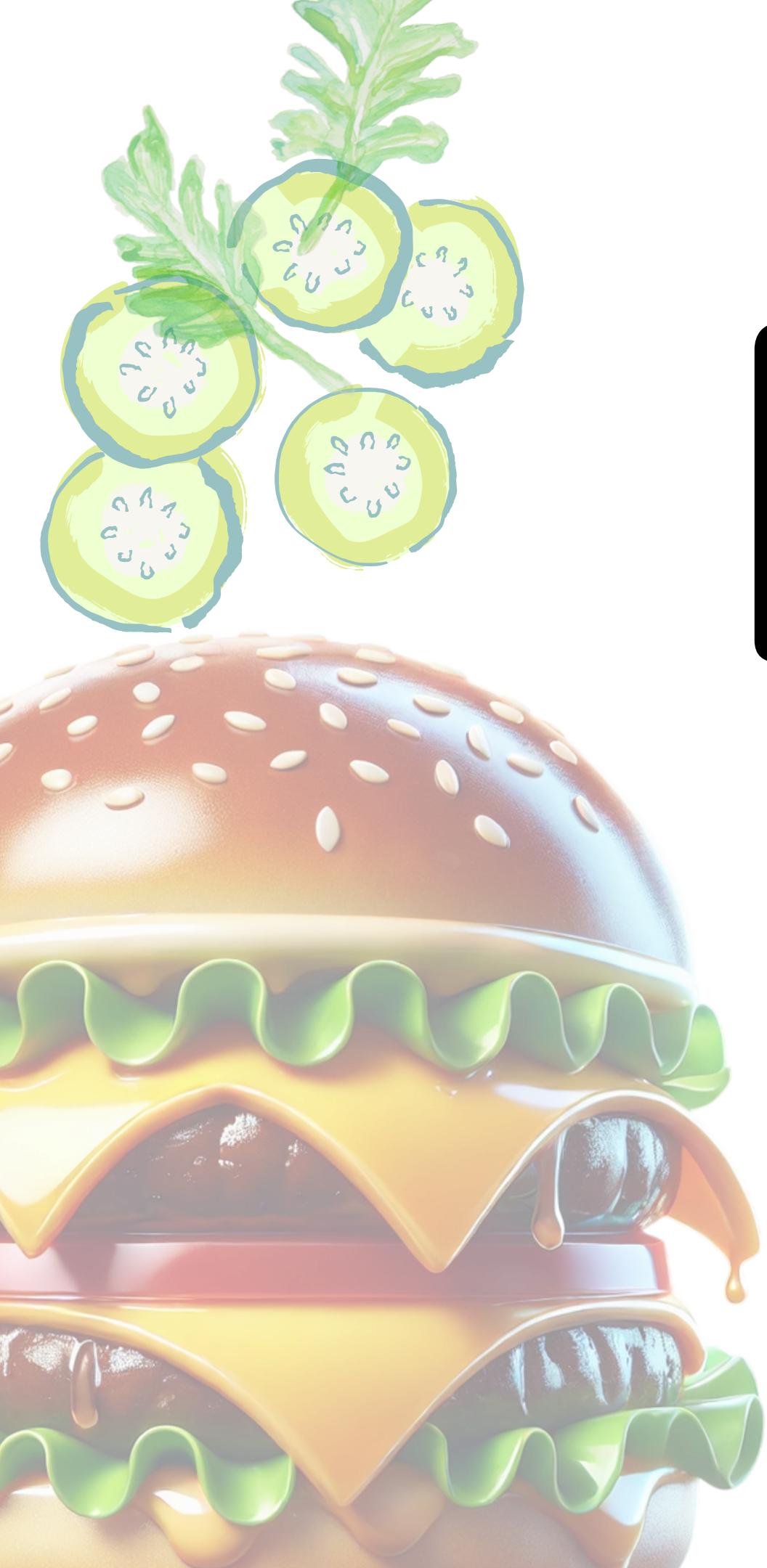
High Level Architecture

5



Database Architecture





demo