

# FTGO — Layered Architecture Map

Clear, layered view of gateways, orchestrator, participants, supporting services, and CQRS.

Generated: Oct 06, 2025 12:55

Gateway

Orchestrator

Participant

Supporting

CQRS

Arrow colors: REST • Saga • Events

## Layered Dependency Map

Gateways

API Gateway  
(REST)

API Gateway  
(GraphQL)

Core Orchestrator

order-service  
(Saga Orchestrator)

ftgo\_order\_service MySQL

Participants

consumer-service

ftgo\_consumer\_service MySQL

kitchen-service

ftgo\_kitchen\_service MySQL

accounting-service  
(Event Sourced)

ftgo\_accounting\_service MySQL / Event Store

restaurant-service  
(Master Data)

ftgo\_restaurant\_service MySQL

Supporting

delivery-service

ftgo\_delivery\_service MySQL

CQRS View

order-history-service  
(CQRS Read Model)

ftgo\_orderhistoryservice DynamoDB

This diagram emphasizes **directional dependencies** and **ownership of data**. Export using the Print button above.

## Notes

- **Gateways** are stateless and compose responses across services.
- **order-service** orchestrates multi-service transactions with *Saga* commands and compensations.
- **restaurant-service** publishes master-data events consumed by order, kitchen, and delivery.
- **order-history-service** maintains a *CQRS* read model in DynamoDB fed by order events (via CDC/Kafka).

Need swimlanes per communication type (REST/Saga/Events) or per domain (Ordering/Kitchen/Delivery)? I can add that too.