

The Real Composable Commerce OMS Checklist

Use these questions when evaluating an OMS for composable commerce.

GATE 1 • ORDER STATE

- ☐ **Order state** Can the system handle concurrent order updates without destructive cancel-and-recreate workflows?
- ☐ **Order mutations** Can customers remove items, change addresses, or swap variants while preserving the original order ID and audit trail?
- ☐ **State transitions** Does the system enforce valid transitions based on the physical fulfillment state of the order?

GATE 2 • INVENTORY

- ☐ **Inventory lifecycle** Does it model inventory beyond a simple available/unavailable count, including reservations, allocation, picking, shipping, and safety buffers?
- ☐ **Reservation expiry** Can temporary inventory reservations expire automatically when checkout fails, or a cart is abandoned?
- ☐ **Real-time ATP** Can it calculate Available-to-Promise inventory in real time across multiple fulfillment nodes?

GATE 3 • ROUTING

- ☐ **Fulfillment splitting** Can one commercial order be split into multiple fulfillment units without custom database logic or middleware?
- ☐ **Dynamic routing** Can the engine evaluate factors such as proximity, fulfillment cost, split minimization, warehouse capacity, and cutoff times?
- ☐ **Routing control** Can operations teams adjust routing rules without waiting on developer sprints?

GATE 4 • INTEGRATION

- ☐ **System boundaries** Is the OMS the authoritative system for in-flight orders, while the ERP and other downstream systems rely on it as the source of truth?
- ☐ **Event-driven integration** Can the OMS emit asynchronous business events without making checkout dependent on downstream system availability?
- ☐ **Failure handling** Does it provide native idempotency, retries, and failure recovery for integrations with ERP, WMS, 3PL, payment, and other systems?

GATE 5 • OPERATIONS

- ☐ **Operational interface** Does the platform include a ready-to-use back-office UI for operations teams without requiring custom internal dashboards?
- ☐ **Operational parity** Can the same operational actions available in the UI be performed through APIs and embedded into existing systems?