

Sample: Shopify + Amazon, COD-heavy D2C

Business requirements document

Version: 0.1

Status: Draft (27 open decisions)

Date: 19 Oct 2026

Prepared: AI-generated, reviewed in OpsOS

Evidence: 3 confirmed · 2 derived · 2 recommended · 0 missing

Contents

1. Summary	3
2. Scope	3
3. Guardrails	3
4. Requirements	4
5. Integration flows	5
6. Decisions and open questions	6
7. Exceptions and edge cases	7
8. Market practices	9
9. UAT scenarios	10
10. Validation baseline	10

1. Summary

Evidence: 3 confirmed · 2 derived · 2 recommended · 0 missing · 27 open decisions

- Enforce payment clearance gates before order processing and warehouse packing across Shopify and Amazon channels.
- Implement multi-channel real-time inventory synchronization with safety stock buffers to eliminate overselling during high-volume sales events.
- Automate bi-directional tracking number synchronization back to sales channels to achieve 100% update reliability.
- Streamline cash-on-delivery (COD) risk management and return logistics loops to prepare operations for the Diwali peak.

2. Scope

Scenario: Hi, we sell on Shopify and Amazon, around 400 orders a day. Orders keep getting packed before payment clears (COD mostly) and we oversold the blue kurta twice last sale. Tracking numbers don't go back to Shopify half the time. Need this sorted before Diwali.

Industry: Ecommerce

Channels: Shopify, Amazon

3. Guardrails

Financial trigger: Orders enter the OMS only after payment is confirmed.

Logistics loop: Every inbound order flow must have a matching fulfillment return flow.

Inventory buffer: Use configurable safety stock to prevent overselling during peak demand.

Data mapping: Every integration must explicitly define its required fields.

Controlled uncertainty: Mark assumptions clearly and ask one confirmation question per unknown.

4. Requirements

REQ-001 — Financial Trigger & Payment Gate for Order Ingestion (MUST, confirmed)

Orders from Shopify and Amazon enter the OMS only after payment confirmation. For Cash on Delivery (COD) orders, orders enter a pending-verification state until payment status is validated or RTO risk scoring is completed.

Recommended standard: Financial trigger: orders enter the OMS strictly after payment is confirmed.

Assumptions:

- Payment gateway webhooks or channel API payloads accurately reflect payment status.
- COD orders require an explicit gate prior to warehouse release.

Confirmation questions:

- What specific payment gateway is used for prepaid Shopify orders to verify settlement instantaneity?

Acceptance criteria:

- Prepaid orders automatically flow to OMS only when payment_status equals 'paid'.
- COD orders are held in 'payment_pending' and never advance to packing without explicit verification.

REQ-002 — Inventory Buffer and Multi-Channel SKU Mapping (MUST, confirmed)

Maintain a configurable safety stock buffer per channel for high-demand items (e.g., blue kurta) to prevent cross-channel overselling. Implement multi-channel SKU mapping to unify product catalogues across Shopify and Amazon.

Recommended standard: Inventory buffer: use configurable safety stock to prevent cross-channel overselling.

Assumptions:

- Stock levels must account for safety buffers before publishing available-to-sell quantities.
- Channel SKUs differ from internal master SKUs.

Confirmation questions:

- What is the exact safety stock buffer quantity configured for the blue kurta during peak sales?

Acceptance criteria:

- Available-to-sell inventory equals on-hand stock minus soft allocations and channel safety buffers.
- Unmapped channel SKUs trigger an integration exception without zeroing the catalogue.

REQ-003 — Bi-Directional Tracking Number Synchronization (MUST, confirmed)

Ensure fulfillment status and tracking numbers generated by the shipping system are successfully pushed back to Shopify and Amazon without failure.

Recommended standard: Outbound parcel orders move through explicit states driven by documents, ensuring fulfillment data is pushed back immediately upon dispatch.

Assumptions:

- Carriers provide valid tracking numbers upon manifest closure or label generation.

Confirmation questions:

- What retry mechanism and alert threshold are currently configured for failed tracking updates?

Acceptance criteria:

- 100% of fulfilled orders transmit tracking_number and carrier_id back to Shopify and Amazon within 15 minutes of dispatch.

REQ-004 — Logistics Return Loop and RTO Processing (MUST, derived)

Every outbound fulfillment flow must have a matching fulfillment return flow, segregating customer returns from courier Return-to-Origin (RTO) shipments.

Recommended standard: Logistics loop: every inbound order flow must have a matching fulfillment return flow.

Assumptions:

- RTO parcels are received against a carrier return manifest and handled distinctly from customer returns.

Confirmation questions:

- What is the designated warehouse staging area for processing RTO vs. customer return parcels?

Acceptance criteria:

- RTO shipments are scanned back into inventory without triggering credit note issuance.
- Customer returns undergo mandatory inspection before refund authorization.

REQ-005 — Data Mapping and Mandatory Field Enforcement (MUST, derived)

Every integration must explicitly define and validate mandatory fields including order_id, sku, quantity, shipping_address, customer_email, payment_status, tracking_number, carrier_id, and fulfillment_status.

Recommended standard: Data mapping: every integration must explicitly define its mandatory fields.

Assumptions:

- Incoming payloads missing core mandatory fields are rejected to staging with explicit error logs.

Confirmation questions:

- Which secondary metadata fields are required for statutory invoicing compliance?

Acceptance criteria:

- Integrations reject payloads missing any mandatory field and generate an actionable exception log.

REQ-006 — Duplicate Order and Webhook Replay Handling (SHOULD, recommended)

Handle duplicate order submissions caused by webhook replays, retries, or double clicks by enforcing idempotency checks using channel order IDs.

Recommended standard: Idempotency controls must deduplicate incoming webhooks before order creation.

Assumptions:

- Sales channels retry webhooks upon non-200 HTTP responses.

Confirmation questions:

- What is the deduplication time window enforced by the OMS middleware?

Acceptance criteria:

- Duplicate webhooks with identical channel order IDs do not create duplicate records in the OMS.

REQ-007 — Unmapped SKU and Catalog Exception Management (MUST, recommended)

Orders arriving with a channel SKU that has no internal mapping must be held in an exception queue and prevented from releasing to warehouse packing.

Recommended standard: Integration pipelines must halt unmapped items to prevent fulfillment errors.

Assumptions:

- Mapping tables are maintained centrally within the OMS.

Confirmation questions:

- Who receives automated alerts when an unmapped SKU enters the order queue?

Acceptance criteria:

- Unmapped SKUs route directly to an 'Unmapped Catalog Exception' queue and block fulfillment until resolved.

5. Integration flows

INT-001: Shopify -> OMS (inbound)

Event: Order Created | Trigger: Webhook fired upon checkout completion with confirmed payment

Fields: order_id, sku, quantity, shipping_address, customer_email, payment_status

Return path: OMS to Shopify Fulfillment API (tracking_number, carrier_id, fulfillment_status)

INT-002: Amazon -> OMS (inbound)

Event: Order Ready to Ship | Trigger: API poll for unfulfilled marketplace orders with verified payment

Fields: order_id, sku, quantity, shipping_address, customer_email, payment_status

Return path: OMS to Amazon Feeds API (tracking_number, carrier_id, fulfillment_status)

INT-003: OMS -> Shopify / Amazon (outbound)

Event: Inventory Level Update | Trigger: Inventory change or periodic batch sync after safety stock calculation

Fields: sku, quantity, fulfillment_status

Return path: Channel inventory acknowledgment webhook

INT-004: OMS -> Shipping Partner (bidirectional)

Event: Manifest Dispatch & Tracking Push | Trigger: Carton packed and shipping label generated

Fields: order_id, tracking_number, carrier_id, shipping_address

Return path: Shipping Partner to OMS (Delivery Status, RTO Scan, NDR updates)

6. Decisions and open questions

Decision log (1)

DEC-025 · Platform

What are normal and peak volumes (orders per day and per hour at sale peaks), and what latency is acceptable for stock and order sync?

Agreed: around 400 orders a day

Requirements: REQ-003

Open questions (27)

DEC-001 · Channels

For each channel account, which fulfilment model applies (self-ship, marketplace-ship, marketplace-fulfilled, JIT, vendor PO), and who issues the invoice and the shipping label?

Requirements: REQ-001

DEC-002 · Channels

Per channel account: order intake on/off, sync-from date, polling interval, and which capabilities are enabled (order pull, acknowledgement, stock push, dispatch update, cancel, returns)?

Requirements: REQ-001, REQ-002, REQ-003

DEC-003 · Routing

Which facilities fulfil orders, what is each one's role and GSTIN, and which channels does each facility serve?

Requirements: REQ-001

DEC-004 · Routing

How is the fulfilling facility chosen (stock, distance, cost, priority), and what breaks a tie?

Requirements: REQ-001

DEC-005 · Routing

May an order split across facilities or ship partially? Are backorders allowed, or is an unfulfillable line cancelled?

Requirements: REQ-001

DEC-006 · Orders

Are orders confirmed automatically? Which hold types exist (payment, fraud/RTO risk, address, manual), who releases them, and does a hold block picking?

Requirements: REQ-001

DEC-007 · Orders

What COD limits apply (order value, pincode, carrier), is COD verified before dispatch (call, WhatsApp, IVR), and are COD-to-prepaid nudges used?

Requirements: REQ-001

DEC-008 · Orders

What happens to a cancellation at each stage (before allocation, after pick, after pack/invoice, after handover), and who may cancel after invoicing?

Requirements: REQ-001

DEC-009 · Orders

Is an order complete on delivery or on carrier handover, and how many days after settlement does it auto-close?

Requirements: REQ-003

DEC-010 · Inventory

What stock is published to each channel: buffers, soft blocks or channel shares, which stock statuses count as sellable, and how often is stock pushed?

Requirements: REQ-002

DEC-011 · Inventory

How often is stock counted (ABC, by location), is counting blind, and what variance needs approval before an adjustment posts?

Requirements: REQ-002

DEC-012 · Outbound

Which picking method is used per facility or channel (single order, batch, cluster, zone, waves), with totes or not, and how is a short pick handled?

Requirements: REQ-001

DEC-013 · Outbound

Is every unit scanned at packing, is weight captured at carton close (with what tolerance), and how is the box size chosen?

Requirements: REQ-001

DEC-014 · Outbound

In what order are invoice, e-invoice (IRN), shipping label and e-way bill produced, and which failure blocks dispatch?

Requirements: REQ-003

DEC-015 · Shipping

How is the carrier chosen (pincode, weight, COD/prepaid, channel, cost), what is the default and fallback carrier, and is an aggregator used?

Requirements: REQ-003

DEC-016 · Shipping

How are manifests grouped (per carrier, per channel), what are the pickup cut-off times, and what proof is captured at handover?

Requirements: REQ-003

DEC-017 · Shipping

Who acts on NDRs and within what SLA, how many reattempts are allowed, and when is RTO triggered automatically?

Requirements: REQ-004

DEC-018 · Shipping

How is COD apportioned across multi-carton shipments, and who reconciles carrier COD remittances against collections, how often?

Requirements: REQ-001

DEC-019 · Returns

How are RTO (undelivered) and customer returns kept apart in stock and accounting, how is a partial RTO handled, and what evidence supports marketplace claims?

Requirements: REQ-004

DEC-020 · Compliance

Which invoice number series applies per GSTIN (and per channel or client), and are prices tax-inclusive?

Requirements: REQ-004

DEC-021 · Compliance

Does e-invoicing apply to this business, who raises e-way bills and when, and is dispatch blocked without a valid e-way bill?

Requirements: REQ-003

DEC-022 · Integrations

Which system owns the item master, invoices, credit notes and stock valuation (OMS, ERP, accounting), and how often do they sync?

Requirements: REQ-001, REQ-002

DEC-023 · Integrations

Who reconciles marketplace settlements (commission, fees, TCS) and COD remittances, and from which reports?

Requirements: REQ-001

DEC-024 · Platform

Who may override routing, cancel after invoicing, adjust stock, force dispatch and release holds?

Requirements: REQ-001

DEC-026 · Platform

How will opening stock, open orders and in-flight returns be migrated, and what are the go/no-go criteria?

Requirements: REQ-001

DEC-027 · Orders

Is editing an order before dispatch in scope (address, phone, quantities, payment mode)? If so, up to which stage and by whom?

Requirements: REQ-001

DEC-028 · Platform

Is a handheld scanning app in scope for receiving, picking, packing and counting, and on which devices?

Requirements: REQ-001

7. Exceptions and edge cases

EC-001 — Duplicate order from a webhook replay, retry or double click

Deduplicate on channel account + channel order ID; replays with an older or equal revision are ignored; every create call carries an idempotency key.

Requirements: REQ-006

EC-002 — Order arrives with a channel SKU that has no mapping

Hold the order and fail the event visibly, naming the unmapped SKU; never fall back to a default SKU; allow re-mapping and replay.

Requirements: REQ-002, REQ-007

EC-003 — Channel API returns a partial or throttled page

Fail the sync run without moving the cursor; never create orders with empty or invented lines; pace calls centrally per channel.

Requirements: REQ-005

EC-004 — Channel answers HTTP 200 with an error in the body

Judge success from the response body, not the status code; log the request body on terminal errors.

Requirements: REQ-005

EC-005 — Order date defaults to ingest time instead of the channel purchase date

Store the channel order number and purchase date separately from record creation time; SLAs measure from the purchase date.

Requirements: REQ-001

EC-006 — Channel sends no payment mode, or an unpaid order looks prepaid

Unpaid or unknown -> COD pending (never assume prepaid); partially paid -> partial COD with the outstanding balance as the COD amount.

Requirements: REQ-001

EC-007 — Every line is cancelled but the order header stays open

Cancelling the last line cancels the header and releases stock; a periodic sweep closes any that slipped through.

Requirements: REQ-001

EC-008 — Order cancelled after picking or packing, before handover

Keep the order open as "aborting"; void AWBs, cancel cartons, reverse invoice, e-invoice and e-way bill, withdraw any ASN, raise reversal putaways; only then mark it cancelled.

Requirements: REQ-001

EC-009 — Marketplace-fulfilled (FBA-type) orders sit in "created" forever

Raise no pick work for them; close them on ingest when the channel reports shipped, with shipped quantity taken from the channel so invoicing and returns still work.

Requirements: REQ-001

EC-010 — Overselling during the stock-sync delay at peak

Apply the safety buffer on the stock-push path, not just in settings; push absolute quantities; alert on oversell incidents.

Requirements: REQ-002

EC-011 — A bulk stock push zeroes the whole catalogue

Guard bulk pushes (threshold on the share of SKUs going to zero); never push from a sandbox account as if it were live.

Requirements: REQ-002

EC-012 — A stock hold that still lets the units be allocated or published

A hold must block allocation and remove the units from channel stock; prove it with a test.

Requirements: REQ-001, REQ-002

EC-013 — An RTO parcel is counted as a delivery

RTO runs in its own lane with its own date; RTO-delivered never counts as delivered in SLAs or carrier scorecards; no credit note on RTO.

Requirements: REQ-004

EC-014 — Carrier booking times out and a retry books a second AWB

Hold the consignment as "booking uncertain"; resolve by idempotent re-book, lookup or an operator decision; never re-book blindly.

Requirements: REQ-003

EC-015 — An unrecognised carrier scan is read as "delivered"

Store every raw scan; only mapped scans move status; keep carrier status maps configurable.

Requirements: REQ-003, REQ-004

EC-016 — A carton is cancelled while its AWB stays live and billable

Cancelling a carton on a live consignment cancels the consignment and voids the AWB with the carrier first.

Requirements: REQ-003

EC-017 — The full COD amount is printed on every carton of a multi-carton order

Apportion COD across consignments; check the carrier supports multi-piece COD.

Requirements: REQ-001

EC-018 — A marketplace-provided label is voided or rebooked by the seller

Channel labels are never voided or rebooked; withdrawing one is a channel-side cancel.

Requirements: REQ-003

EC-019 — A lost shipment means the order never settles or closes

Mark lost as a terminal status that writes off the units, so delivered + RTO + written-off settles the line.

Requirements: REQ-003, REQ-004

EC-020 — A manifest without a carrier, or one that strands cartons when cancelled

Every manifest has one carrier and only accepts its labelled cartons; cancelling a manifest releases them; chunk when a carrier caps AWBs per manifest.

Requirements: REQ-003

EC-021 — A split or partial shipment is billed the full order amount, or reuses the first invoice

One invoice per shipment for the packed quantity × the order's stamped unit price; the second shipment gets its own invoice (and ASN).

Requirements: REQ-004

EC-022 — A SKU has no HSN or tax rate

Block or report the invoice; never default to 0%; keep the channel-stated HSN so the invoice and ASN agree.

Requirements: REQ-005

EC-023 — A setting is captured and shown but never enforced

Every configuration requirement has an acceptance test that proves the behaviour changes when the setting changes.

Requirements: REQ-002

8. Market practices

MP-001 — Outbound

Parcel orders move through explicit states driven by documents: failed (with the reason) -> ready to confirm -> confirmed -> packed when the invoice is raised -> ready to ship when the label prints -> dispatched when the manifest closes, with one invoice per package.

Requirements: REQ-003

MP-002 — Orders

COD orders are scored for RTO risk; risky ones are held automatically for verification or a prepaid-conversion offer before packing, and NDRs are worked the same day.

Requirements: REQ-001, REQ-006

MP-003 — Integrations

Reconciliation runs as a routine, order by order: marketplace settlements, COD remittances and their dates, carrier bills against packed weight (weight disputes), returns and stock, with exceptions listed for action.

Requirements: REQ-003, REQ-005, REQ-006, REQ-007

MP-004 — Outbound

Pack and return stations record video linked to the order and AWB, and claim-filing reports are kept, so marketplace and courier claims carry evidence.

Requirements: REQ-003

MP-005 — Inventory

Available-to-sell is set per channel on purpose: on hand minus soft blocks and safety buffers per channel (a virtual catalogue per channel at scale), or full exposure where stock sync is fast.

Requirements: REQ-002, REQ-007

MP-006 — Returns

Courier returns (RTO) are received against a return manifest per carrier, and a customer return is accepted only for items already dispatched.

Requirements: REQ-004

MP-007 — Channels

The webstore's fulfilment orders are mirrored per location, fulfilment and tracking go back per fulfilment order, and its webhooks are treated as at-least-once: deduplicated, then reconciled by a periodic pull.

Requirements: REQ-002, REQ-007

9. UAT scenarios

UAT-001 — Prepaid and COD Order Ingestion Gate

Validates: REQ-001

1. Simulate a Shopify prepaid order arrival with payment_status 'unpaid'.
 2. Verify the order does not enter OMS active orders.
 3. Simulate a Shopify prepaid order arrival with payment_status 'paid'.
 4. Verify the order successfully enters OMS.
 5. Simulate a COD order arrival.
 6. Verify the order is held in 'payment_pending' state and blocked from advancing to packing.
- Expected: Prepaid unpaid orders are blocked, prepaid paid orders enter OMS, and COD orders hold in payment_pending state without advancing to packing.**

UAT-002 — Multi-Channel Safety Buffer Inventory Calculation

Validates: REQ-002

1. Configure on-hand stock for the blue kurta SKU to 10 units with a safety buffer of 2 units.
 2. Create a soft allocation of 3 units.
 3. Check the available-to-sell inventory calculation for the SKU across Shopify and Amazon.
 4. Verify available stock correctly reflects on-hand minus allocations and buffers.
- Expected: Available-to-sell inventory equals on-hand stock minus soft allocations and channel safety buffers.**

UAT-003 — Bi-Directional Tracking Number Synchronization

Validates: REQ-003

1. Dispatch a fulfilled order in the OMS containing tracking_number and carrier_id.
 2. Monitor the synchronization logs to Shopify and Amazon.
 3. Verify that the fulfillment status, tracking number, and carrier ID are successfully pushed back to the respective channel within 15 minutes of dispatch.
- Expected: 100% of fulfilled orders transmit tracking_number and carrier_id back to Shopify and Amazon within the specified time window.**

UAT-004 — Logistics Return Loop and RTO Processing

Validates: REQ-004

1. Process an incoming courier Return-to-Origin (RTO) shipment scan.
 2. Verify the RTO items are scanned back into inventory.
 3. Confirm that no credit note is generated for the RTO shipment.
 4. Process a customer return and verify it enters the mandatory inspection state before refund authorization.
- Expected: RTO shipments return to inventory without credit notes, and customer returns undergo mandatory inspection.**

UAT-005 — Mandatory Field Validation and Exception Logging

Validates: REQ-005

1. Simulate an incoming order payload missing the customer_email and shipping_address fields.
 2. Attempt ingestion into the OMS.
 3. Verify that the payload is rejected and an actionable exception log is generated containing all mandatory fields.
- Expected: Payloads missing mandatory fields are rejected and generate an actionable exception log.**

UAT-006 — Webhook Replay Idempotency Check

Validates: REQ-006

1. Send a valid order webhook payload with a unique channel order ID.
 2. Verify the order is successfully created in the OMS.
 3. Send an identical webhook payload with the exact same channel order ID a second time.
 4. Verify that no duplicate record is created in the OMS.
- Expected: Duplicate webhooks with identical channel order IDs do not create duplicate records in the OMS.**

UAT-007 — Unmapped SKU Exception Routing and Blocking

Validates: REQ-002, REQ-007

1. Simulate an order arrival containing an unmapped channel SKU.
 2. Verify the order routes directly to the Unmapped Catalog Exception queue.
 3. Verify that fulfillment release is blocked until the SKU mapping issue is resolved.
- Expected: Unmapped SKUs route to the exception queue, do not zero the catalogue, and block fulfillment until resolved.**

10. Validation baseline

Completeness (25%): No obvious gap that a senior analyst would immediately catch.

Integration logic (25%): Every inbound flow has a defined return path.

Edge-case depth (20%): Duplicate, partial, retry, and latency cases are covered.

Testability (20%): Every requirement has at least one pass/fail acceptance condition.

Hallucination control (10%): No unverified API, field, SLA, or platform behavior is stated as fact.