

Cove Commerce accelerator for Meridian OMS

About this sample

Document type: Integration reference

Audience: Enterprise architects and OMS developers

Purpose: Describes the architecture, modules, and deployment of a pre-built integration between an enterprise order management system (OMS) and a commerce and fulfillment platform.

Note: Company, product, and platform names are fictional. The technical content is representative of the published version.

You can use the Cove Commerce accelerator for Meridian OMS to integrate Cove Commerce with your Meridian order management system (OMS). The following sections describe the technical requirements, solution architecture, implementation modules, and deployment considerations for the accelerator.

Accelerator benefits

The Cove Commerce accelerator for Meridian OMS reduces integration time and technical complexity. You can use the accelerator to implement Cove commerce capabilities without extensive modifications to your existing inventory management, order processing, fulfillment tracking, and returns handling systems.

The accelerator works alongside your current OMS configurations with minimal system changes. This approach lets development teams focus on merchant-specific customizations instead of building core integration components.

Solution overview

The Cove Commerce accelerator for Meridian OMS processes data asynchronously without disrupting your existing Meridian OMS operations. The accelerator manages data transformations and maintains state consistency between Cove services and your OMS, adding Cove fulfillment capabilities while preserving existing workflows.

Use cases

The following table describes several use cases for the Cove Commerce accelerator for Meridian OMS:

Use case	From	To	Description
Fulfill with Cove	OMS	Cove	Create and manage Cove Commerce orders from Meridian OMS.
Cove fulfillment updates	Cove	OMS	Update your Meridian OMS with delivery cancellations, delivery failures, shipping, and tracking info from Cove.
Returns through Cove	Cove	OMS	Handle returns initiated from Cove, create return orders in Meridian OMS, and sync return delivery status.
Returns outside Cove	OMS	Cove	Sync merchant-initiated returns with Cove and maintain return visibility.
Cove refund requests	Cove	OMS	Process refund requests initiated from Cove due to delivery cancellations, delivery failures, and returns.
Synchronize issued refunds	OMS	Cove	Send refund updates from OMS to Cove to ensure synchronized status.
Merchant order cancellation	OMS	Cove	Request order cancellation through various channels.
Synchronize inventory	Cove	OMS	Sync inventory in real time and batch from Cove to Meridian OMS to maintain accurate stock levels.

Design principles and technical architecture

Integration principles

The Cove Commerce accelerator for Meridian OMS uses principles such as keeping client systems authoritative, maintaining system boundaries, and using asynchronous, event-driven communication.

Architecture layers

The Cove Commerce accelerator for Meridian OMS uses three distinct architectural layers:

- Cove Connector layer: Handles technical connectivity and data mapping between Cove and OMS formats.
- OMS Integration Services layer: Handles business logic and orchestration.
- OMS Event Handler layer: Provides validation and routing logic. You deploy the OMS Event Handler layer on your OMS platform to identify and direct Cove orders to the accelerator's OMS Integration Services. This layer serves as a reference implementation that you can configure or extend based on your requirements.

Event processing architecture

The architecture of the accelerator uses asynchronous event processing to reduce system coupling and increase scalability.

- Cove to Meridian OMS: Webhooks send Cove events to the Cove Connector, which validates and transforms data before routing to Java Message Service (JMS) queues. The OMS Integration Services layer processes these events asynchronously, maintains OMS state, and coordinates Cove responses. Meridian OMS uses PKCS#12 certificates and basic authentication for security. The accelerator processes JSON-based Cove events and communicates with Meridian OMS using the native XML format that its APIs require.
- Meridian OMS to Cove: The accelerator validates Meridian OMS transaction events for Cove orders and routes them to JMS queues for processing. The OMS Integration Services layer processes these events asynchronously, maintains Cove state, and coordinates OMS responses. Access to Cove services requires OAuth-based authentication. The accelerator communicates with Cove services using Cove Commerce APIs and Cove Seller APIs.

Meridian OMS platform integration

The accelerator design aligns with Meridian OMS order lifecycle transactions, events, and processing pipeline. This design principle supports integration with your existing OMS implementation and reduces customization requirements. The architecture uses established Meridian OMS mechanisms for order processing, fulfillment, and inventory management.

Accelerator configuration

The Cove Commerce accelerator for Meridian OMS lets you configure properties to change the behavior of the accelerator based on your business needs and account setup. You can modify these properties using API calls or manually modify them with the Meridian OMS application manager.

Merchant-specific customizations and configurations

Minimize impact to your Meridian OMS environment

To minimize impact to your Meridian OMS environment, the Cove Commerce accelerator for Meridian OMS does the following:

- Excludes configurations like hub rules, user exits, and event actions from the installation package to preserve your existing Meridian OMS configurations.
- Externalizes common, potentially overlapping configurations like transaction and status codes.
- Provides sample pipelines for order, shipment, and return processing to avoid conflicts with your existing pipelines.
- Processes most accelerator operations in a separate transaction boundary from core OMS transactions to minimize performance impact.
- Prefixes all Cove configurations with `cov` to avoid conflicts with your existing setup.

Required configurations and customizations

To use the Cove Commerce accelerator for Meridian OMS, you need to apply the following common configurations and customizations. Your specific Meridian OMS implementation might require additional customizations.

- **Configurations:** Review and assess configurations like conflicting status codes and ship node values for applicability.
- **Pipelines:** The accelerator provides sample pipelines for order, shipment, and return processing. You might need to modify these to include common transactions such as sales and returns posting.
- **Servers:** The accelerator includes several server configurations, but you might prefer to consolidate infrastructure usage with your existing servers for common processes like creating orders and scheduling and releasing orders. This requires manual changes to the server configurations after deployment.
- **OMS Event Handler:** Implement the OMS Event Handler. We provide this as a reference implementation but do not include it in the accelerator deployment package.
- **Code:** You can make changes to the accelerator source code as necessary. For example, you can update the source code to add support for segmentation when updating inventory.

Implementation modules

The Cove Commerce accelerator for Meridian OMS uses several modules to integrate Cove services with Meridian OMS. These modules handle order management, fulfillment, returns processing, refund management, cancellations, and inventory synchronization. Each module uses specific APIs and event triggers to maintain data consistency between systems.

Fulfill with Cove

The Fulfill with Cove module supports both storefront-initiated and OMS-initiated orders. For OMS-initiated orders, the module handles creation and routing to Cove, managing the fulfillment initialization process. For storefront-initiated orders, the accelerator initiates fulfillment.

The following table lists fulfillment APIs and events:

Component type	Name	Purpose
Cove APIs	<code>mutation: createOrder</code>	Creates new order in Cove.
	<code>mutation: updateOrder</code>	Updates order status.
Cove events	N/A	-
OMS APIs	<code>getOrderList</code>	Get complete order details.
	<code>changeOrder</code>	Update order with Cove <code>orderId</code> ; cancel OMS order line if failed to create Cove order.
OMS events	<code>ORDER_CREATE.0001.ON_SUCCESS</code>	Triggers on creation of OMS order, used to create Cove order or initiate Cove fulfillment.
	<code>RELEASE.0001.ON_SUCCESS</code>	Triggers on release of OMS order, used to initialize Cove fulfillment.

Cove fulfillment updates

The fulfillment status update component provides real-time visibility through event-based status updates. It processes various fulfillment events including in-transit notifications, delivery confirmations, package milestones, cancellations, and delivery failures. After Cove starts fulfilling an order, this module automatically updates OMS with package delivery milestones such as current package location, delivery attempts, and package status.

After the accelerator creates a shipment in OMS, your OMS is expected to create a shipment invoice and complete the payment processing.

The following table lists shipment and delivery APIs and events:

Component type	Name	Purpose
Cove APIs	<code>query: order</code>	Retrieves delivery/package details.
Cove events	<code>PACKAGE_DELIVERY_IN_TRANSIT</code>	Indicates shipment started.
	<code>PACKAGE_DELIVERED</code>	Confirms delivery completion.
OMS APIs	<code>getShipmentContainerList</code>	Get shipment details for <code>CovDeliveryId</code> .
	<code>getOrderLineList</code>	Get order line details for Cove item alias.
	<code>createShipment</code>	Creates shipment record.
	<code>changeShipmentStatus</code>	Updates shipment status.

The following table lists package delivery milestone APIs and events:

Component type	Name	Purpose
Cove APIs	query: Order	Retrieves tracking details.
Cove events	PACKAGE_TRACKER_MILESTONE_CHANGED	Updates tracking milestones.
OMS APIs	getShipmentContainerList	Get shipment details for TrackingNo.
	getContainerMilestoneList (custom)	Get milestones for CovDeliveryId and TrackingNo.
	updateTrackingMilestones (custom)	Updates tracking information.

The following table lists delivery cancellation and delivery failure APIs and events:

Component type	Name	Purpose
Cove APIs	query: order	Retrieves delivery/package details.
Cove events	PACKAGE_DELIVERY_CANCELLED	Indicates delivery cancellation.
	PACKAGE_DELIVERY_FAILURE	Indicates delivery failures.
OMS APIs	getShipmentContainerList	Get shipment details for CovDeliveryId.
	getOrderLineList	Get order line details for Cove item alias.
	changeRelease	Cancel release.
	changeOrder	Cancel for Cove checkout or backorder line for Cove fulfillment.
	changeShipmentStatus	Updates shipment status.

Returns through Cove

This module processes returns initiated through the Cove online returns center or Cove checkout shopper order details page. It automatically creates return records in OMS and maintains synchronization of return delivery status updates, ensuring consistent return tracking across systems.

The following table lists Cove return APIs and events:

Component type	Name	Purpose
Cove APIs	<code>query: Order</code>	Retrieves return details.
Cove events	<code>RETURN_STARTED</code>	Indicates return initiation.
	<code>RETURN_PACKAGE_IN_TRANSIT</code>	Updates return shipment status.
	<code>RETURN_PACKAGE_DELIVERED</code>	Confirms return delivery.
	<code>RETURN_ITEM_GRADED</code>	Updates return item inspection status.
	<code>RETURN_COMPLETED</code>	Indicates return completion.
OMS APIs	<code>getOrderList</code>	Get order details for <code>CovReturnId</code> .
	<code>getOrderLineList</code>	Get order line details for <code>CovOrderId</code> .
	<code>createOrder</code>	Creates return order in OMS.
	<code>changeOrderStatus</code>	Change order status.
	<code>changeOrder</code>	Save events.

Returns outside Cove

This module manages returns initiated outside the Cove platform, such as through your OMS customer service portal. It ensures Cove has visibility into all returns, preventing duplicate return authorizations.

The following table lists external return APIs and events:

Component type	Name	Purpose
Cove APIs	<code>mutation: updateOrder</code>	Creates or updates return in Cove.
OMS APIs	<code>getOrderList</code>	Retrieves order details.
	<code>changeOrder</code>	Updates return status.
OMS events	<code>ORDER_CREATE.0003.ON_SUCCESS</code>	Triggers external return processing.
	<code>ORDER_CHANGE.0003.ON_SUCCESS</code>	Triggers return status updates.

Cove refund requests

This module processes refund request notifications from Cove. It handles refund requests for various scenarios including successful returns, order cancellations, and delivery failures.

After the accelerator creates a return invoice or a credit memo, your OMS is expected to complete the payment refund processing.

The following table lists refund request APIs and events:

Component type	Name	Purpose
Cove APIs	<code>mutation: updateOrder</code>	Updates refund status in Cove.
	<code>query: Order</code>	Query refund details.
Cove events	<code>REFUND_REQUESTED</code>	Triggers refund processing.
OMS APIs	<code>getReceiptLineList</code>	Get receipts for <code>CovReturnId</code> and <code>CovRefundId</code> .
	<code>receiveOrder</code>	Mark return line as received.
	<code>closeReceipt</code>	Close receipt.
	<code>createOrderInvoice</code>	Create return invoice.
	<code>changeOrderInvoice</code>	Update invoice with <code>CovRefundId</code> .
	<code>recordInvoiceCreation</code>	Creates credit memo invoice.
	<code>getInvoiceDetails</code>	Retrieves invoice based on <code>CovRefundId</code> .
	<code>getOrderLineList</code>	Retrieve order lines based on item ID alias.

Synchronize issued refunds

This module updates Cove with refund processing status and details after OMS completes refund payment processing. This ensures Cove has accurate records of completed refunds and can update customer communications accordingly.

The following table lists credit memo refund APIs and events:

Component type	Name	Purpose
Cove APIs	<code>mutation: updateOrder</code>	Updates refund status in Cove.
OMS events	<code>PAYMENT_COLLECTION.ON_INVOICE_COLLECTION</code>	Triggers refund status updates.

The following table lists Cove-initiated refund APIs and events:

Component type	Name	Purpose
Cove APIs	<code>mutation: updateOrder</code>	Updates refund status in Cove.
OMS events	<code>PAYMENT_COLLECTION.ON_INVOICE.0003_COLLECTION</code>	Triggers refund status updates.

The following table lists merchant-initiated refund APIs and events:

Component type	Name	Purpose
Cove APIs	<code>mutation: updateOrder</code>	Updates refund status in Cove.
OMS APIs	<code>getOrderList</code>	Retrieve order details.
	<code>getInvoiceDetailList</code>	Retrieve invoice details.
	<code>changeOrderInvoice</code>	Update invoice with <code>CovRefundId</code> .
OMS events	<code>PAYMENT_COLLECTION.ON_INVOICE.0003_COLLECTION</code>	Triggers refund status updates.

Merchant order cancellation

You can initiate cancellation of Cove checkout or Cove fulfillment orders through various channels like your website or call center. You can cancel orders for reasons such as customer remorse, out-of-stock items, payment issues, or other business scenarios. The accelerator evaluates and forwards valid cancellation requests to Cove for processing.

The following table lists order cancellation APIs and events:

Component type	Name	Purpose
Cove APIs	<code>mutation: cancelOrder</code>	Request to cancel Cove order.
OMS events	<code>CHANGE_ORDER_ON_CANCEL</code>	Triggers request to cancel Cove order.

Synchronize inventory

The inventory management module synchronizes inventory between Cove and your Meridian OMS. This module supports both real-time inventory updates through events and periodic full synchronization to ensure accurate inventory levels across systems. The periodic full synchronization operates through a scheduled agent process that systematically retrieves inventory data from Cove and updates the OMS.

The following table lists real-time inventory APIs and events:

Component type	Name	Purpose
Cove APIs	<code>getInventorySummaries</code>	Get fulfillable quantity for seller <code>SKU</code> .
Cove events	<code>INVENTORY_CHANGED</code>	Triggered on inventory availability change.
OMS APIs	<code>getAggregatedDemand</code>	Get sum of all open demand.
	<code>syncSupply</code>	Update supply quantity.

The following table lists periodic inventory synchronization APIs:

Component type	Name	Purpose
Cove APIs	<code>getInventorySummaries</code>	Get fulfillable quantity for batch of seller <code>SKUs</code> .
	<code>query: Products</code>	Get external ID for seller <code>SKU</code> .
OMS APIs	<code>getAggregatedDemand</code>	Get sum of all open demand.
	<code>syncSupply</code>	Update supply quantity.

Installation and deployment overview

Prerequisites

The installation process requires a functioning Meridian OMS environment with a Java Message Service (JMS) MQ platform, core components, code packages, and configurations. The accelerator supports both on-premises and cloud-hosted editions of Meridian OMS.

Deployment

We recommend the following approach to deploy the accelerator:

1. Deploy to a sandbox environment to identify gaps and conflicts.
2. Merge configurations into the management configuration environment.

The implementation spans multiple layers, including application code deployment, JMS queue configuration, and server infrastructure setup. Module-specific configurations require enabling specific primitives, setting module properties, configuring database extensions, and establishing integration points aligned with your business requirements.