Global-e
Smart Cross-Border™

Global-e Shipping APIs

Version: 2.3

Table of Contents

1. Global-e Shipping API - Introduction

Global-e's Shipping API provides merchants with fully integrated shipping capabilities in getting the relevant shipping labels and documentation to ship straight from their own warehouses or through a 3PL. For this purpose the integration is based on implementing the Global-e shipping solution akin to a simplified carrier integration.

This API works seamlessly across Global-e direct integrations and Shopify Markets Pro merchants.

This document describes the implementation of the API calls required to perform the required shipping activities.

1.1. Shipping model Scenarios

Global-e produces:

- 6x4 PDF/ZPL/EPL (for specific UK carriers) shipping labels

- 8.5x11 inch letter size / A4 PDF Commercial Invoices (required only when a carrier and country do not support Paperless Trading (PLT)

In order for Global-e to fulfill a cross border order, our APIs must be used to produce the shipping documents - labels & commercial invoices to ship the package, to ensure that the shipment is documented correctly, compliantly and allow for duty guarantee.

Global-e offers different shipping models to leverage our extended carrier network with specialised services via the Global-e facility.

Scenario	Documents returned by the API
Direct dispatch from warehouse with final mile carrier	- Carrier label - PDF invoice(s) - Possible additional documents (eg. Dangerous Goods Note)
Shipping via Global-e Hub	- 6x4 ZPL/PDF crossdocking label for the Global-e hub

1.2. Using Global-e API

To implement API calls with Global-e, the only required information is the *merchantGUID* provided during onboarding for each merchant, which acts as both authorization and authentication to the Global-e API.

Production endpoint : https://api.global-e.com/

All Shopify Markets Pro calls should be made through the production instance of the Global-e API.

For Global-e direct integration test environments should be used:

- INT : https://connect.bglobale.com/
- STAGING : https://connect2.bglobale.com/

All information required to use the Shipping API endpoints to generate shipping documents is contained in this document.

2. Integration Overview

It is essential that the label and CI accurately describe the contents of the shipment to avoid being overcharged for duties and taxes. The parcel(s) contents declared in the API should match the actual shipment.

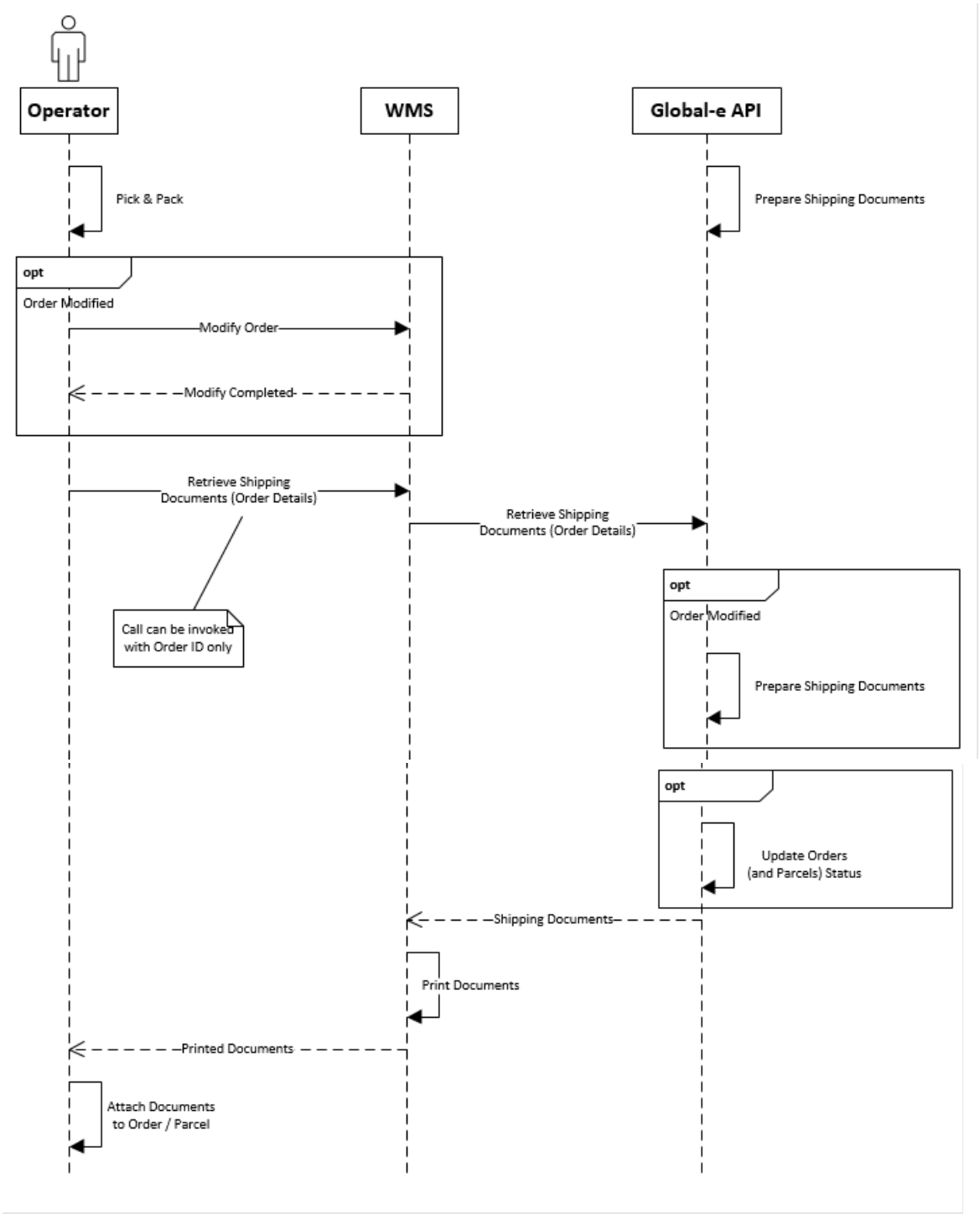
The shipping process and associated integration is built with the following approach:

1. **Receive and Identify Global-e managed orders**
2. **Declare orders fulfilled and ready for dispatch in API call #1**
3. **Retrieve Shipping documents as a response to API call #1**
    a. Carrier label
    b. Commercial invoice (For non-paperless countries)
    c. Other documents, e.g. Dangerous Goods note
4. **Declare effective dispatch of orders as API call #2**
5. **Retrieve the carrier manifest for carrier sign-off as a response to API call #2**

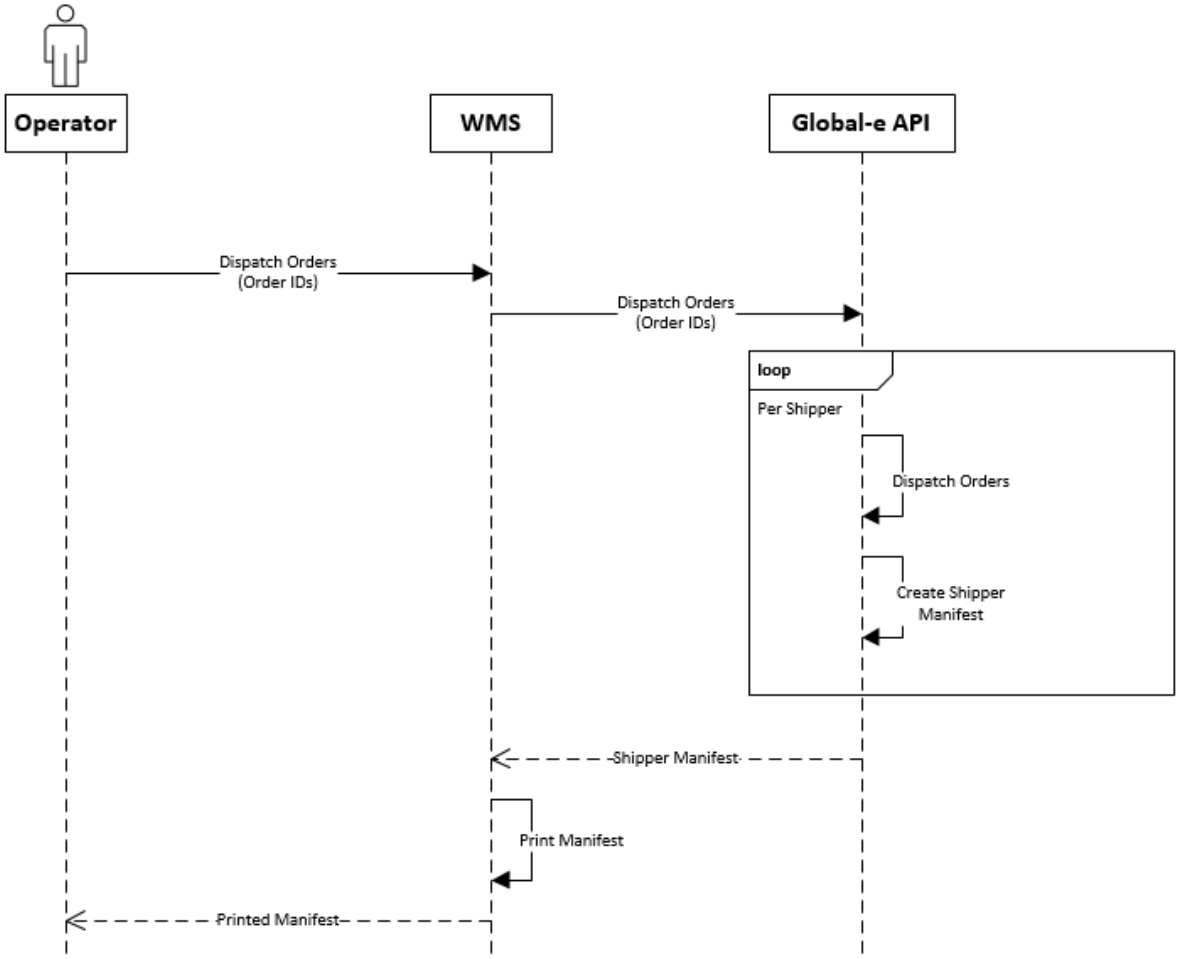
6. Declare the fulfilment back to the eCommerce platform

The diagrams below describe the typical fulfilment and dispatch flow using the end-to-end API process.

2.1. Declaring order fulfilment and retrieving shipping documents



2.2. Orders dispatch notification



3. Prerequisites

The merchants OMS or WMS should have knowledge of the following information and allow, either directly or through an escalation process, to:

- Have the full list of items in the order
- Enable selection of items to be dispatched

After performing the effective pick & pack and required actions in the OMS/WMS, the following integration can be used to process the orders for dispatch.

4. Integration Steps

4.1. Identifying orders to be processed via the Global-e API

Orders may be received directly from the eCommerce platform or via 3rd party integrations.
Accordingly, the below scenarios should be supported to associate orders with the Global-e integration to generate shipping documentation:

- Shipping/Carrier code as provided in the order payload received
  - In Shopify's context such indication will be available via order shipping_lines code or title field
    Mapping of values presented in these fields should allow to map orders to the Global-e integration
  - Outside Shopify, standard carrier mapping as currently performed should be used

- Additionally, when using a direct integration with Shopify to retrieve orders, for example Shopify Markets Pro, the *merchant_of_record_app_id* field will be provided.
  When Global-e is responsible for an order, this value will be set to 2745565185.
  - The *merchant_of_record_app_id* field is available on Shopify webhooks, RESTful GET Orders calls or GraphQL queries. Version 2022-10 or later of the Shopify API is required for this field to be present.

Both methods should be implemented to comply with different merchant setups in pushing orders to the WMS/3PL. However, if you are an independent merchant doing this integration, you may choose either of the methods above per your integration.
Orders will arrive via the same channel as all other orders for a merchant. It's important to map such orders and prevent them from being shipped via means other than Global-e.

4.2. Declaring order fulfilment and getting documents

4.2.1. API Call

API call (POST request) - `API Endpoint: GetShippingDocuments`

From the orders indicated as fulfilled with possible exceptions the Global-e API will return the required shipping documentation as a base64 encoded byte array and URL for printing.

Processing attributes description

UpdateOrderDispatchRequest			
OrderId	-	-	"eCommerce order number" as passed down in order payload For Shopify, should be either: - Shopify Order Name (default), eg. #12345 - Shopify Order Number, eg. 12345
DeliveryReferenceNumber *(optional)*	-	-	Additional informative dispatch reference
HubCode* *(optional)*			When dispatching from multiple hubs, indicates for a given call which hub the related shipment will be dispatched from
List <Parcel> Parcels	ParcelCode *(Required & MAX 20 characters)*	-	Unique identifier for each parcel(/package) to be shipped Can be freely generated in any range as long as unique and no longer than 20 characters.
	List <Product> Products	ProductCode *(Required)*	Product reference from merchant as available in eCommerce platform For Shopify, should be either: • Shopify SKU or Barcode for product to be shipped
		DeliveryQuantity *(Required)*	Quantity of product to be shipped

API call (POST request)

https://connect.bglobale.com/Order/GetShippingDocuments?merchantGUID=abcdabcd

Body: *UpdateOrderDispatchRequest*


```
{
    "OrderId": "GE123874638GB",
    "HubCode": "hub001",
    "DeliveryReferenceNumber": "123756483",
    "Parcels": [
        {
            "ParcelCode": "123454321",
            "Products": [
                {
                    "DeliveryQuantity": 1,
                    "ProductCode": "sku121212",
                },
                {
                    "DeliveryQuantity": 2,
                    "ProductCode": "sku131313"
                }
            ]

        }
    ]
}
```


4.2.2. Success Response

OrderDocumentsResponse with *List<OrderDocument> Documents*


```
{
    "IsSuccess": true,
    "ErrorText": null,
    "Documents":
    [
        {
            "DocumentTypeCode": "4",
            "DocumentTypeName": "AWB",
            "DocumentExtension": "zpl",
            "URL": "https://assets.global-e.com/documents/478D-AA72-1EB8BBDD7C27.zpl",
            "DocumentData":"JVBERi0xLjUNCiW1tbW1DQoxIDAgb2[...]mDQoxNDgzNDkNCiUlRU9G"
        },
        {
            "DocumentTypeCode": "1",
            "DocumentTypeName": "CommercialInvoiceAndPackingList",
            "DocumentExtension": "pdf",
            "URL": "https://assets.global-e.com/documents/AE09-FAB014DAA421.pdf",
            "DocumentData": "JVBERi0xLjUNCiW1tbW1DQoxIDAgb2[...]mDQoxNDgzNDkNCiUlRU9G",
            "ErrorMessage": null
```

```
        }
    ],
    "ParcelsTracking": [
        {
            "ParcelTrackingNumber": "9895722141",
            "ParcelTrackingUrl": "
https://mydhl.express.dhl/us/en/tracking.html#/results?id=9895722141",
            "ParcelCode": "123454321"
        }
    ]
}
```


4.2.3. Error Responses

HTTP Error code as 500, 400 (or 200)

1. **General API error** as *ErrorInfo :*


```
{
  "Code": "error code",
  "Error": "error message",
  "Description": "error description"
}
```


2. **Object processing error** as *OrderDocumentsResponse:*


```
{
    "IsSuccess": false,
    "ErrorText": "Could not retrieve documents and/or process order. Either order is in wrong
status (Cancelled) or partial information provided.",
    "Documents": null,
    "ParcelTracking": null,
    "TrackingDetails": null,
    "Errors": [
        {
            "OrderID": "GE3008553US",
            "ErrorCode": "A200",
            "ErrorText": "Could not retrieve documents and/or process order. Either order is i
n wrong status (Cancelled) or partial information provided.",
            "MerchantOrderID": null
        }
    ]
}
```

The response of the Get Shipping Documents API call (GSD) will include the tracking information of the shipment(s). Tracking information includes:

- Shipper Name (order level)
- Tracking Number (for each parcel and order level)
- Tracking URL (for each parcel and order level)

Tracking should always be stored from ParcelsTracking array by matching the relevant parcel code tracking.

If the tracking is done on the order level, the tracking information of the order (Tracking Number and URL) will be populated on the parcel level as well (with the same values).

4.2.4. OrderDocument DocumentTypeCode list

DocumentTypeCode should be used to identify the type of document returned by the API and how to print it.

Labels for Zebra/Thermal Printer

DocumentTypeCode	Description	Paper format	Notes
4	Carrier Label	6x4	For direct dispatch
7	GELabel	6x4	
9	ArchiveLabel	6x4	*Only for specific*

A4 / Letter Size for Laser printer

DocumentTypeCode	Description	Paper format	Notes
1	CommercialInvoice	A4 / *Letter Size*	Provided only for destinations not supporting electronic customs declarations
5	*VATInvoice*	*A4 / Letter Size*	*legacy*
6	DangerousGoods	A4 / *Letter Size*	Only for merchants shipping Dangerous Goods
10	Delivery Advice	A4 / *Letter Size*	For specific UAE Free Trade Zones requirements

4.3. Notifying Order Dispatch and retrieving Carrier Manifest

API Endpoint: DispatchOrders

For all orders that had labels printed since the last call to DispatchOrders, the Global-e API will return the required carrier manifest as a base64 encoded byte array and URL for printing.
For non Shopify orders they will also be marked as "Dispatched to Customer" and if configured so notification emails with tracking will be sent out.

Processing attributes description

UpdateOrderDispatchRequest			
HubCode* *(optional)*			When dispatching from multiple hubs, indicates for a given call which hub the related shipment will be dispatched from
List <String> OrderIds	-	-	"eCommerce order number" as passed down in order payload For Shopify, should be either : - Shopify Order Name (default), eg. #12345 - Shopify Order Number, eg. 12345

API call (POST request)
https://connect.bglobale.com/Order/DispatchOrders?merchantGUID=abcdabcd

Body: *GetOrdersManifestRequest*

```
{
    "OrderIds": [
       "GE11111111GB", "GE2222211GB"
    ],
    "HubCode":"hub001"
}
```

Response: *OrderDocumentsResponse*

```
{
    "IsSuccess": true,
    "ShipperManifests":
    [
        {
            "DocumentTypeCode": "3",
            "DocumentTypeName": "ShipperManifest",
            "DocumentExtension": "pdf",
```

```
            "URL": "https://assets.global-e.com/documents/04552836-ED8F-4362-AE09-
FAB014DAA421.pdf",
            "DocumentData": "JVBERi0xLjUNCiW1tbW1DQoxIDAgb2[...]mDQoxNDgzNDkNCiUlRU9G"
            "ErrorMessage": null
        },
        {
            "DocumentTypeCode": "3",
            "DocumentTypeName": "ShipperManifest",
            "DocumentExtension": "pdf",
            "URL": "https://assets.global-e.com/documents/8F0B8A31-B424-478D-AA72-
1EB8BBDD7C27.pdf",
            "DocumentData": "JVBERi0xLjUNCiW1tbW1DQoxIDAgb2[...]mDQoxNDgzNDkNCiUlRU9G"
        }
    ]
}
```


4.4. Mark the Order Fulfilled

Once a label has been used to ship a set of goods, update the fulfilment state back to the merchant platform to complete the workflow.

- Status: success
- Tracking Company: Use the ShipperName returned by the GetShippingDocuments call
- Tacking Number: Use the TrackingNumber returned by the GetShippingDocuments call
- Items: Only those items that have shipped with this label should be identified

5. Use Cases

Examples of request body as *UpdateOrderDispatchRequest*

Order description for this section :

- product-A in quantity 1
- product-B in quantity 2

5.1. Single parcel shipping

Single call

```
{
    "OrderId": "100018322",
    "Parcels": [
        {
            "ParcelCode": "100018322-1",
            "Products": [
                { "ProductCode": "product-A", "DeliveryQuantity": 1 },
                { "ProductCode": "product-B", "DeliveryQuantity": 2 }
```

```
            ]
        }
    ]
}
```

5.2. Multi parcel

Single call

```
{
    "OrderId": "100018322",
    "Parcels": [
        {
            "ParcelCode": "100018322-1",
            "Products": [
                { "ProductCode": "product-A", "DeliveryQuantity": 1 }
            ]
        },
        {
            "ParcelCode": "100018322-2",
            "Products": [
                { "ProductCode": "product-B", "DeliveryQuantity": 2 }
            ]
        }
    ]
}
```

5.3. Split parcels

First call (“day 1”)

```
{
    "OrderId": "100018322",
    "Parcels": [
        {
            "ParcelCode": "100018322-1",
            "Products": [
                { "ProductCode": "product-A", "DeliveryQuantity": 1 },
                { "ProductCode": "product-B", "DeliveryQuantity": 1 }
            ]
        }
    ]
}
```

Subsequent call (“day 2”)

```
{
    "OrderId": "100018322",
    "Parcels": [
        {
            "ParcelCode": "100018322-2",
```

```
            "Products": [
                { "ProductCode": "product-B", "DeliveryQuantity": 1 }
            ]
        }
    ]
}
```

5.4. Split parcels – Multi-Hubs

First call (parcel shipped from Hub #1 – code = hub001)

```
{
    "OrderId": "100018322",
    "HubCode": "hub001",
    "Parcels": [
        {
            "ParcelCode": "100018322-1",
            "Products": [
                { "ProductCode": "product-A", "DeliveryQuantity": 1 },
                { "ProductCode": "product-B", "DeliveryQuantity": 1 }
            ]
        }
    ]
}
```

Second call (parcel shipped from Hub #2 – code = hub002)

```
{
    "OrderId": "100018322",
    "HubCode": "hub002",
    "Parcels": [
        {
            "ParcelCode": "100018322-2",
            "Products": [
                { "ProductCode": "product-B", "DeliveryQuantity": 1 }
            ]
        }
    ]
}
```

6. Country Of Origin, Parcel Weight & Dimensions

For Global-e Enterprise, please consult with your account manager about the availability of this feature.
Merchant can send the product country of origin, parcel weight and parcel dimensions to DHL and Aramex via the ship request, according to the data which was provided by the merchant via GSD.

For Global-e Enterprise, if any of the 3 following attributes are not specified, it will be taken from default value of information provided in product catalog as part of ecommerce platform integration:

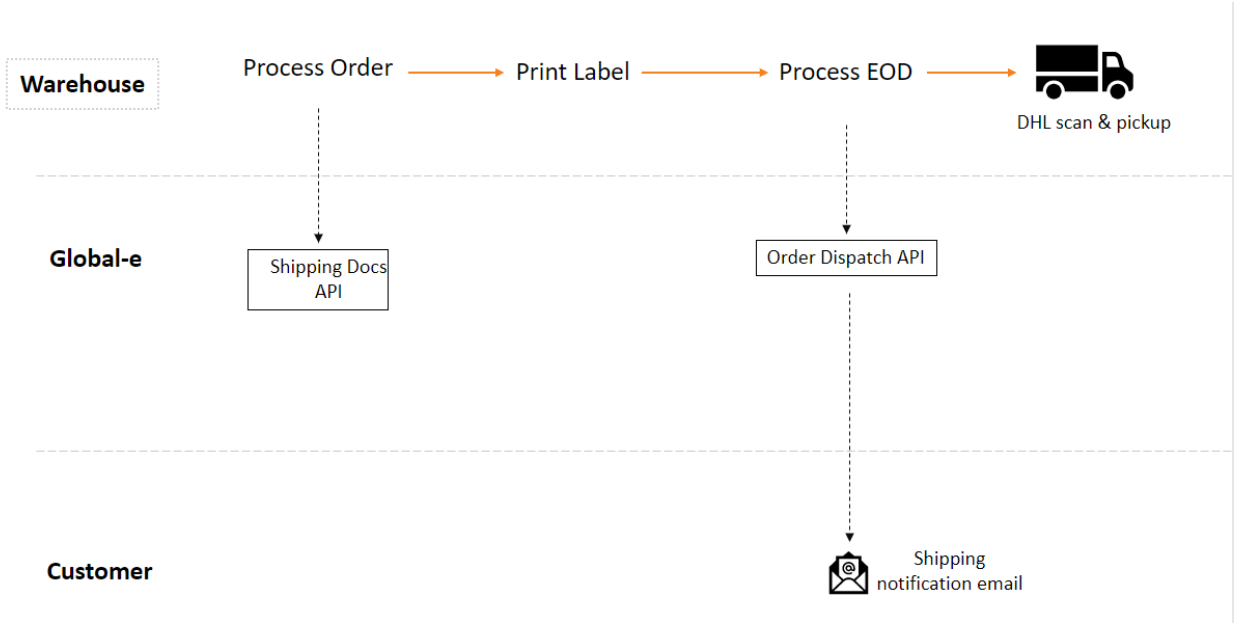
- **Product Country of Origin**
- **Parcel Weight:** For Global-e Enterprise, the weight of the products within the parcel will be calculated according to the parcel weight with a simple average method. Example: For parcel weight 1000 Grams with 2 items, each item will weigh 500 Grams.
    - Note: Always provided in Grams in GSD.
- **Parcel Dimensions:** For Aramex we will be providing the parcel dimensions according to the first parcel even for consolidated shipment (in the current implementation, the dimensions exist at shipment level and not per parcel).
    - Note: Always provided in CM in GSD.

```
{
    "OrderID": "{{OrderId}}",
    "Parcels": [
        {
            "ParcelCode": "Order123-P1",
            "Length": 37.5,
            "Width": 24,
            "Height": 13,
            "Weight": 2340,
            "Products": [
            {
                  "ProductCode": "ProductA",
                  "DeliveryQuantity": 1,
                  "OriginCountryCode": "FR"
                }
                {
                  "ProductCode": "ProductB",
                  "DeliveryQuantity": 1,
                  "OriginCountryCode": "IL"
                }
                {
                  "ProductCode": "ProductB",
                  "DeliveryQuantity": 1,
                  "OriginCountryCode": "US"
                },
            ]
        }
    ]
}
```

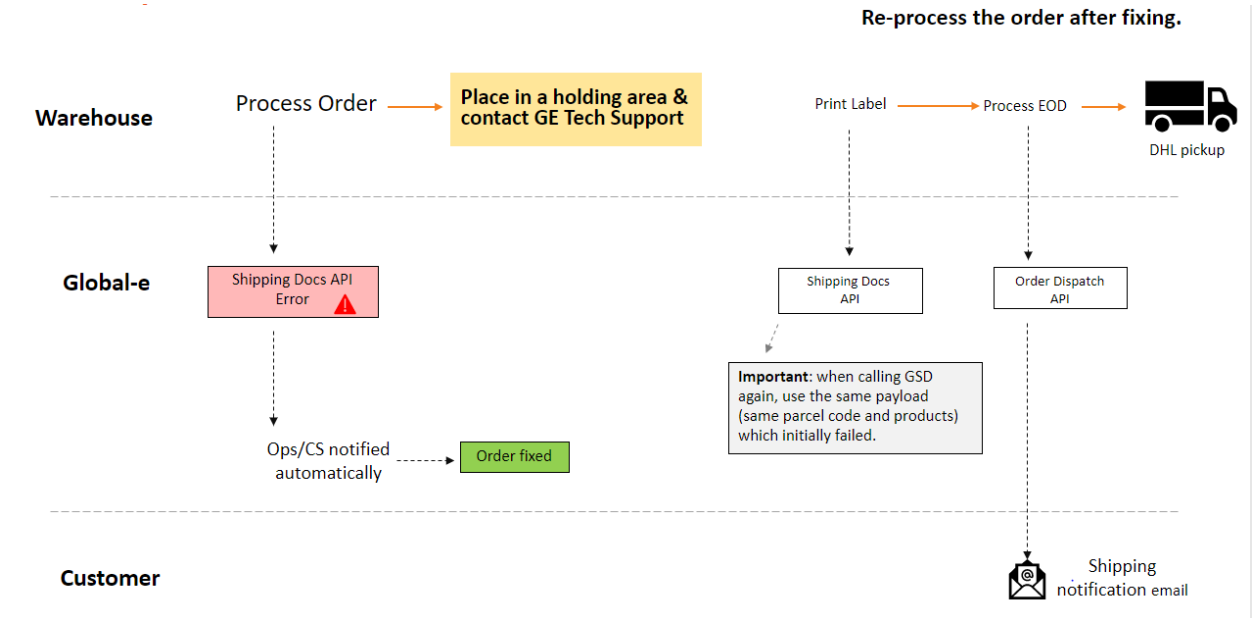
7. Error Handling

7.1. Processing Exception Scenarios

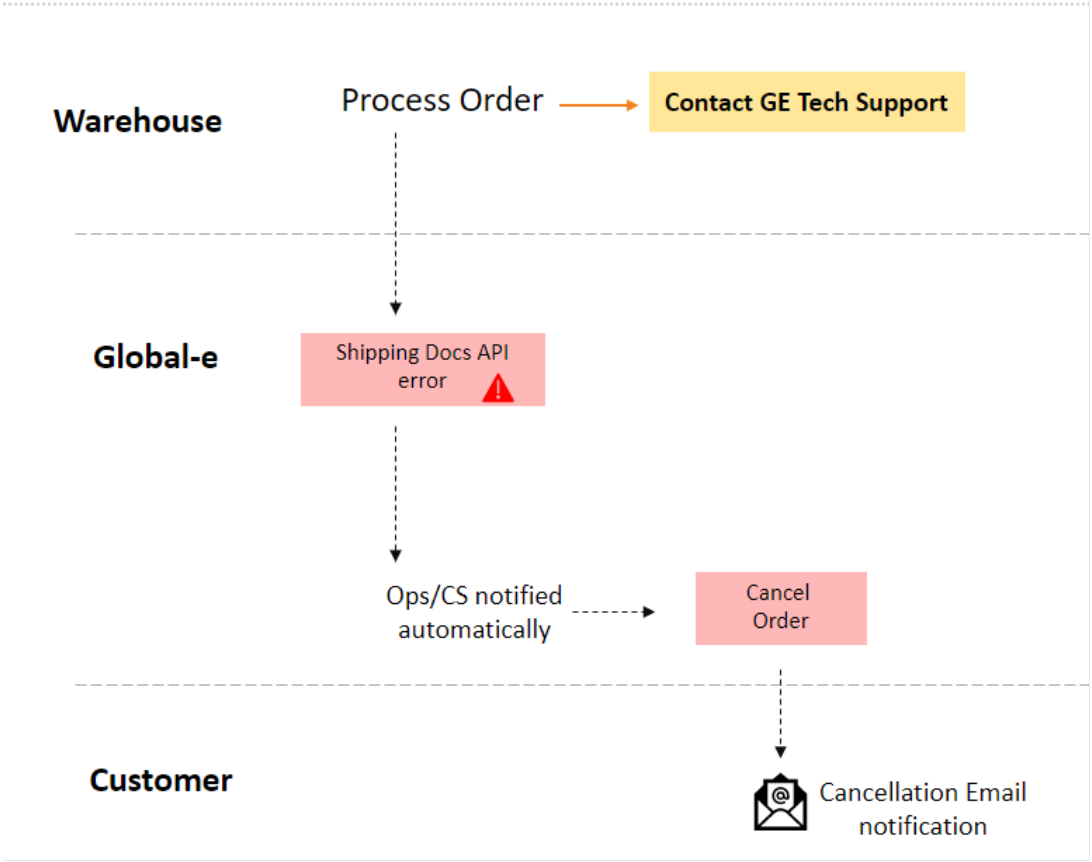
7.1.1. Normal Operation – No issues



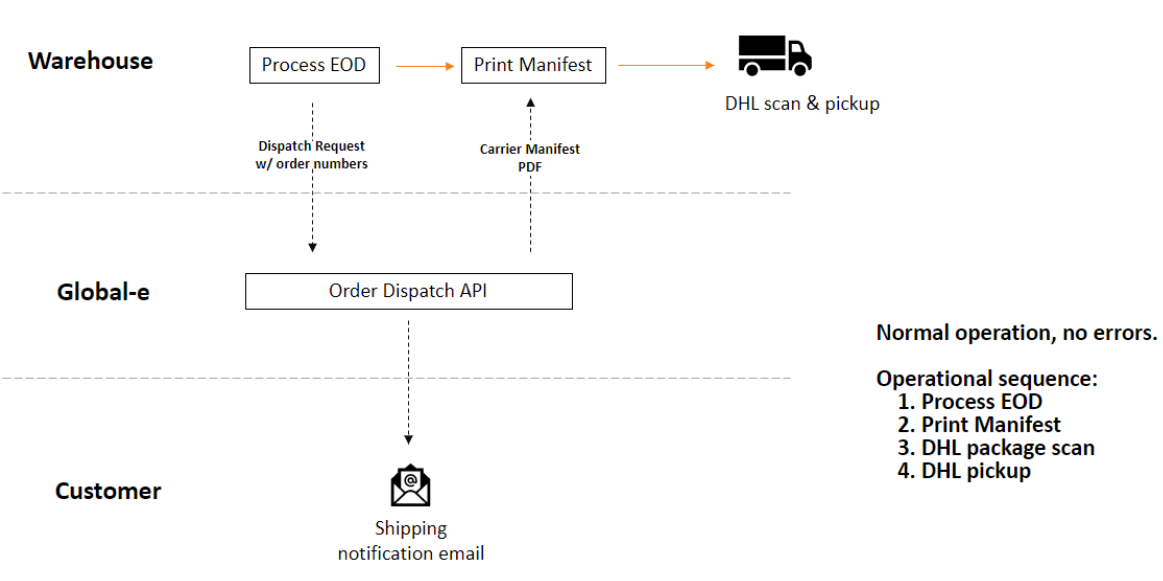
7.1.2. With error handling



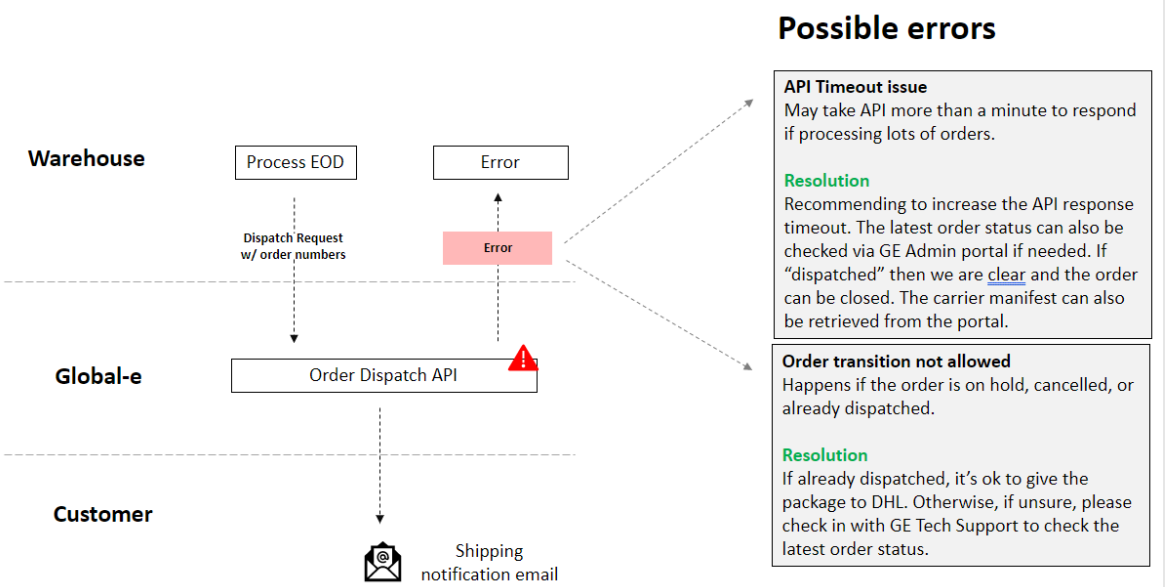
7.1.3. Scenario – Order to be cancelled after failed API request



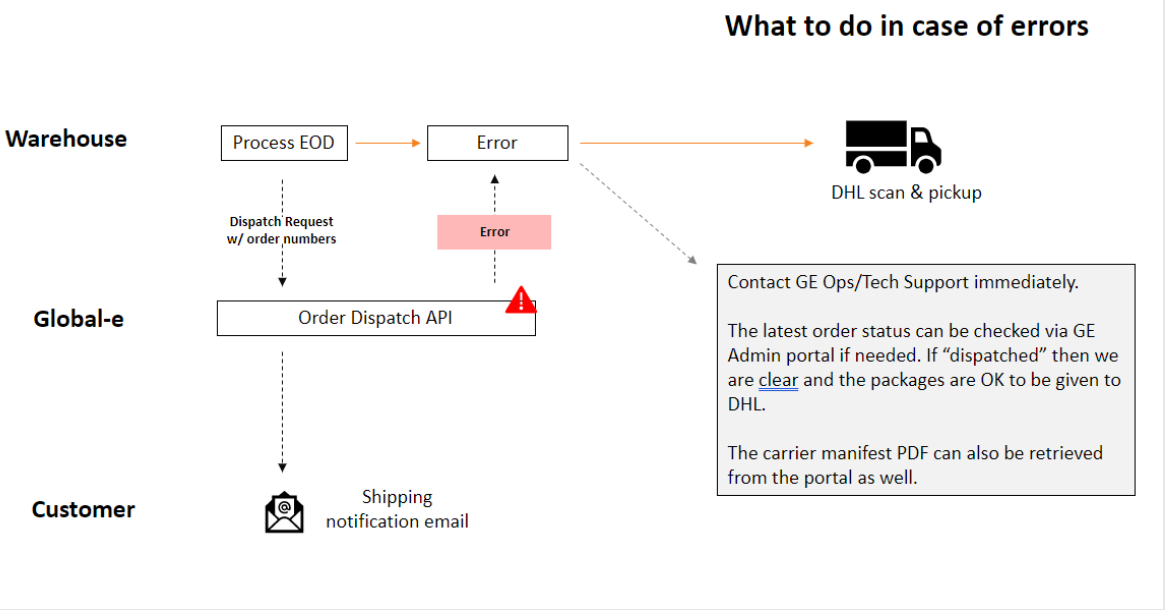
7.1.4. Scenario – End of day manifesting



7.1.5. Scenario – End of day manifesting with possible errors



7.1.6. Scenario – End of day manifesting with possible errors and error handling



7.2. GetShippingDocuments Error Codes

Error Code	Error Text in response	Description
A100	Invalid information schema.	Relevant for technical errors. For example, wrong JSON object structure
A200	Could not retrieve documents and/or process order. Either order is in wrong status or partial information provided.	Relevant for logical errors. For example, order in wrong status
A300	There is at least one SKU that was not part of the original order. Please use only SKUs which were originally ordered.	Relevant for cases in which the fulfilled SKUs contain an SKU not originally ordered (and SKU validation is turned on)
A400	Either there are no valid SKUs to place in parcels or parcels could not be created.	Relevant for cases in which ALL fulfilled SKUs were not originally ordered and if we were to amend there will be no item to put in parcels (and SKU validation is turned off)
A500	SMB API returns an error	There was an error generating a label via the Global-e SMB Bridge

7.3. DispatchOrders Error Codes

Error Code	Error Text in response	Description
B100	There are no orders valid for dispatch.	Relevant for dispatch requests that do not contain any valid order for dispatch
B200	Could not determine the destination hub. Either none or more than one hub was found.	Relevant for consolidated shipment dispatch requests that contain orders associated with more than one hub
B300	Number of outer boxes cannot be greater than the number of parcels. Total number of parcels is %Number of Parcels% and the number of outer boxes is %Number of outer boxes%	
B400	Could not create manifest	Relevant when the hub manifest cannot be created
B500	Could not create AWB	Relevant when the consolidated AWB cannot be created
B600	Configuration error. Cannot determine source hub. Please contact Global-e TechSupport	There is a configuration error on GE side, more than one merchant hub is configured
B700	Unknown failure	Relevant for general errors and unhandled exceptions
B800	Hub Code is required	Merchant is configured to used multiple hubs per order but did not send hub code
B001	Order ID not found	Refers to Order ID provided by the merchant but it does not exist
B002	Order cannot be dispatched. Order and/or parcels are in wrong status.	Possible reasons: - Order AWB generation has failed - Order yet to be processed with GetShippingDocuments - Order already dispatched
B003	Order must be dispatched as part of a consolidated shipment	For consolidated post orders
B004	Order must NOT be dispatched as part of a consolidated shipment	For NON consolidated post orders

8. Void Parcel API (recommended)

This API is currently not supported for Shopify Markets Pro.

The void parcel API aims to provide merchants with an ability to remove a non-shipped, declared parcel, in the cases of a parcel that will not be shipped, or one that needs to be redeclared based on changed shipment requirements.

API call (POST request):
https://connect.bglobale.com/Parcel/VoidParcel?merchantGUID=abcdabcd

Body input:

- **string OrderId (mandatory)** *Order unique identifier*
- **string ParcelCode (mandatory)** *Code used to identify the ParcelCode to void.*

8.1. Example

Request:

```
{
  "OrderId": "#12354", "ParcelCode": "454234"
}
```

Response:

```
{
  "IsSuccess":true, "Errors":[]
}
```

8.2. Error handling

Error Code	Error Text	Additional Information
C100	The parcel cannot be voided due to the status of parcel.	This is the error in case the hub is operated by GE and parcel is in wrong status : Failed Transfer To Shipping, Shipped By Merchant.
C101	The parcel cannot be voided due to the status of parcel.	This is the error in case the hub is operated by Merchant and parcel is in wrong status : Failed

		Transfer To Shipping, Shipped By Merchant, Received In Hub
C103	Internal unknown failure	
C104	The parcel code was not found, or the parcel was already voided	

Appendix – Shopify Markets Pro

Testing Shopify Markets Pro Integration

To do an end-to-end test for Shopify Markets Pro orders, you need to use a Shopify Shop that has Shopify Markets Pro enabled, its recommended that you use a separate test shop. It's also possible to reuse an existing test shop by simply enabling Shopify Markets Pro.

Build a New Shopify Shop

A new developer shop should be created and connected to your software or warehouse management system.

1. This shop should have at least one physical product in its catalog
2. Set up the location of a USA based warehouse and allocate product to it so that orders can be placed
3. You will need to enable Shopify Payments.
4. Be aware that sandbox shops cannot be promoted to production shops

Enable Shopify Markets Pro

1. Contact Shopify support. Let them know you are testing an SMP shipping integration, what your shop's myshopify.com domain is, and some of your existing Shopify merchants. Make it clear you would like to make a sandbox integration – this will allow you to test using test credit cards and test carrier accounts.
2. You should be given access to create an application for Markets Pro. Fill this out to the best of your ability. As this is a sandbox shop, accurate data is not required to be accepted.
3. After creating an application, you will soon get an email indicating that your application has been accepted. Click the link in this email and you will be walked through the process of enabling Shopify Markets Pro
4. You will also receive a *merchantGUID* from Shopify

5. Adding a country picker to your shop template can help streamline the testing international orders, but isn't required

Testing Shopify Markets Pro

You must place an order via standard checkout and not a draft order, created through Shopify Admin.

1. Create a new Incognito window every time you submit a test order to avoid previous addresses being reused in checkout
2. If you added a country picker, you can select the country you want to test.
3. Any physical product may be purchased
4. You will need to provide an accurate international country, city, state and postal code at checkout; name and address can be anything
5. Use the following test credit card data:
    a. Card Number: 4111 1111 1111 1111
    b. Name: Any Name
    c. Expiration: 03/2030
    d. CVC: 737
6. You should see the order in Shopify Admin immediately.
7. The order should also appear in your Warehouse Management system, correctly flagged as either a Shopify Markets Pro order (if it's an international order) or as a standard order (if it's domestic)

Example Test Orders

- Destination France: (75001, Paris)
    - This should be a Shopify Markets Pro order.
    - Ensure that the label is correct, and that a Commercial Invoice (CI) was recorded
    - The label should print at 4"x6"
    - France allows Paperless Trading (PLT) and the CI need not be printed
- Destination Brazil: (Rio de Janeiro, Rio de Janeiro)
    - This should be a Shopify Markets Pro order.
    - Ensure that the label is correct, and that a Commercial Invoice (CI) was recorded
    - The label should print at 4"x6"
    - Brazil is not a PLT country – the CI should be printed at 8.5"x11"
- Destination USA (Domestic):
    - This is not a Shopify Markets Pro order.
    - Ensure that your standard operations for assigning carriers and printing labels still apply to this order