

Oracle® Communications Order and Service Management

Dynamic Fulfilment Modelling and Sample Implementation Guide



Release 8.0

G48121-01

January 2026

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

G48121-01

Copyright © 2026, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

About This Content

1 Overview

About OSM	1
About Solution Designer	1
About Capabilities Cartridges	2

2 Managing the Integration Workflow

Role Definitions for End-to-end Solution Workflow	1
End-to-end Solution Workflow	2
Creating Capabilities Cartridge	3
Packaging Capabilities Cartridge	4
Delivering Artifacts to Administrators	4
Configuring and Loading to OCA	4
Importing Capabilities Cartridge into Solution Designer	5
Creating Product Fulfillment Model	5
Publishing the Initiative to Test Workspace	5
Assembling and Deploying the Solution	5
Publishing the Initiative to Production Workspace	5
Capabilities Cartridge Lifecycle Workflow	6
Creating Capabilities Cartridge	6
Updating Capabilities Cartridge	6
Retiring Capabilities Cartridge	6
Solution Lifecycle Workflow	6
Creating the Solution Cartridge	7
Updating the Solution Cartridge	7
Deleting the Solution Cartridge	8

3 Limitations and Restrictions

Limitations and Restrictions	1
DevOps Considerations	1

	Miscellaneous Considerations	1
4	Troubleshooting	
	Accessing Logs	1
	Common Error Scenarios	2
5	About the Sample Integration Reference Solution Package	
	Installing the Emulators	1
	Loading the Capabilities Cartridge	3
	Creating or Upgrading OSM Cloud Native	3
	Installing the Product Model	4
	Importing the Capabilities Manifest File	5
	Updating the Product Model	5
	Creating the Fulfillment Model	6
	Submitting Product Orders	7

About This Content

This document describes how to integrate Oracle Communications Order and Service Management (OSM) with Oracle Communications Service Catalog and Design (SCD). It explains key integration concepts, workflows, and procedures for designing fulfillment solutions and deploying cartridges automatically.

Audience

This guide is intended for administrators, developers, integrators, and solution designers who configure, integrate, or manage OSM and SCD. This guide assumes that you are familiar with both OSM and SCD.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Conventions

The following text conventions are used in this document.

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Overview

Learn about the integration of Oracle Communications Order and Service Management (OSM) with Solution Designer, which is a part of Oracle Communications Service Catalog and Design (SCD) that enables flexibility in the design-time definition of OSM cartridge content. This enables you to use catalog-driven modeling for products, services, bundles, and offerings. This integration allows you to apply dynamic changes to the runtime OSM environment so that new or updated models are immediately enforced. The integration also supports TM Forum (TMF) standard cartridges, aligning with industry standards.

This guide describes the key interactions between OSM and SCD and explains the new, integrated design journey. It also introduces a reference sample that demonstrates common integration scenarios and explains how to use it to address your business requirements. This guide assumes you are familiar with OSM and SCD.

About OSM

Oracle Communications Order and Service Management (OSM) is an order processing system that receives orders from Customer Relationship Management (CRM) systems and manages the fulfillment functions required to complete each order. OSM coordinates tasks across multiple external systems, including billing, inventory, and activation applications. OSM manages both automated and manual fulfillment functions, tracking the overall status and managing any problems that might occur during order processing.

OSM can process orders for various types of services or products and allows you to define custom fulfillment processes for each order type. These processes can include functions like shipping, installation, and service activations, depending on business requirements. OSM monitors every step in the order lifecycle and provides web-based clients for tracking and managing orders.

The OSM architecture includes WebLogic Server-hosted server components, web-based clients for order and task management, utilities, and interfaces for external system communication using Java Message Service (JMS) and Oracle Database. OSM stores both processed orders and the metadata required to create and process new orders.

You implement an order fulfillment process by defining business requirements, specifying the systems OSM must interact with, determining required manual or automated tasks, and modeling the order process in Design Studio. As your business needs change, you can update fulfillment processes or introduce new types of orders.

About Solution Designer

Solution Designer is one of the components of SCD. It centralizes the management of product, service, resource, and network specifications, supporting unified service creation and delivery. This centralization streamlines solution creation, accelerates deployment, and reduces the cost of ownership for operators and system integrators.

Solution Designer enables you to define and manage products, services, resources, and their interfaces. Its role-based design supports rapid solution creation and ongoing modification as business needs evolve. As a cloud-native application, Solution Designer runs on a

containerized architecture and is accessible across devices. It supports typical configuration activities such as modeling, testing, and deployment throughout the solution lifecycle.

During the design process, specialists create and refine product and service specifications, model network and technology components, and enrich capabilities cartridges with fulfillment models. Administrators manage the publishing and release of the completed solution.

Service Catalog and Design is a cloud native, microservices based application that enables you to manage the following applications in Solution Designer:

- **Common Elements:** Define attributes and conversion logic for products, services, and resources.
- **Domains:** Organize specifications by service type or business area.
- **Fulfillment:** Import and view capabilities cartridges and define fulfillment requirements and systems.
- **Infrastructure Specifications:** Model infrastructure and resource roles.
- **Product Specifications:** Define products and commercial characteristics.
- **PSR Models:** Model relationships between product, service, and resource specifications.
- **Publishing Center:** Manage initiatives and lifecycle stages of solution development.
- **Resource Specifications:** Define physical or logical resources such as devices or phone numbers.
- **Service Specifications:** Specify customer-facing and resource-facing services.

About Capabilities Cartridges

It is often seen that many parts of a cartridge are relatively unchanging, while others change often over the course of the cartridge's life in a production environment. Some examples of the former would be the order template, the orchestration functions, processes and tasks. Some examples of the latter would be product-to-service mappings, product specifications, decomposition rules, granularity rules, and other dynamic orchestration settings. A capabilities cartridge aims to package up only the relatively unchanging parts of a cartridge, and offers these parts as building blocks.

A capabilities cartridge is therefore, a packaging concept. It is not a new cartridge type but instead relates to the way a set of OSM cartridge content is packaged for delivery into the dynamic design process. Business users and technical experts can use the Solution Designer web application, in addition to Design Studio, to help create and refine OSM cartridges. See Compatibility Matrix in *OSM Compatibility Matrix* for the supported software versions of OSM and SCD.

When cartridge design spans both Design Studio and Solution Designer, the OSM Cartridge Assembler (OCA) microservice merges content and handles deployment to OSM. This process, known as Dynamic Cartridge Assembly, supports distributed cartridge design by integrating contributions from both business-focused and technical users.

Capabilities cartridges support agile solution development, but have some restrictions:

- You can use only TMF cartridges to generate a capabilities cartridge **.cpar** file.
- The cartridge must host a TMF 622 specification; TMF 641 is not yet supported.
- The target Design Studio OSM server version must be 8.0 or later.
- You can deliver only a single capabilities cartridge for dynamic design. However, multiple versions of the cartridge, identified by solution name, are supported.

See "About Capabilities Cartridges Scope " in *OSM Modeling Guide* for detailed information.

You must adhere to the guidelines for generating capabilities cartridges. See "Prerequisites" in *Design Studio Modeling OSM Orchestration* for the guidelines about working with capabilities cartridge.

2

Managing the Integration Workflow

Learn about managing the OSM and SCD integration workflow that contains the end-to-end user journeys for designing, deploying, and managing solutions. This chapter covers the detailed processes for developing capabilities cartridges, integrating fulfillment models, managing solution lifecycle stages, and ensuring operational readiness in production environments.

See the following topics:

- [End-to-end Solution Workflow](#)
- [Capabilities Cartridge Lifecycle Workflow](#)
- [Solution Lifecycle Workflow](#)

Role Definitions for End-to-end Solution Workflow

Use roles to control access and permissible operations in OSM and Solution Designer. Create user accounts and assign roles to grant access according to job responsibilities.

[Table 2-1](#) lists the roles are required for the end-to-end solution workflow, along with their responsibilities and the applications they use.

Table 2-1 Roles for End-to-end Solution Workflow

Roles	Description	Applies To
OSM Developer	<ul style="list-style-type: none">• Create Capabilities Cartridge, new fulfillment patterns, fulfillment functions, and fulfillment systems.• Package artifacts using new packaging process.• Deliver packaged artifacts to OSM DevOps Admin.	Design Studio
Product Specialist	<ul style="list-style-type: none">• Select capabilities cartridge to model against initiative.• Define and manage the product specifications.• Define and manage product fulfillment model.	Solution Designer

Table 2-1 (Cont.) Roles for End-to-end Solution Workflow

Roles	Description	Applies To
Service Domain Specialist	<ul style="list-style-type: none"> • Configure Product to Customer Facing Services mapping together with parameter mappings. • Define the structure and models for customer services. • Define how the services are designed and delivered to the customers. 	Solution Designer
Fulfillment Specialist	<ul style="list-style-type: none"> • Configure the fulfillment patterns. • Configure Product to fulfillment pattern mapping. • Define routing rules. • Define execution level. 	Solution Designer
Service Catalog Admin	<ul style="list-style-type: none"> • Receive notification from OSM DevOps Admin regarding capabilities cartridge manifest. • Import capabilities cartridge manifests into SCD. • Manage workspace lifecycle. • Publish initiative to Test or production workspace. 	Solution Designer
OSM DevOps Admin	<ul style="list-style-type: none"> • Configure OSM runtime instances. • Communicate with service catalog admin regarding capabilities cartridge manifest. • Load capabilities cartridges into OCA. • Manage OSM runtime instance lifecycle. • Configure CNTK instance specification. • Deploy and undeploy cartridges. 	OSM

End-to-end Solution Workflow

The end-to-end workflow of Solution Designer and OSM starting from creating a capabilities cartridge, to deploying the solution cartridge in the OSM run-time instance is as follows:

- 1. Create Capabilities Cartridge:** The OSM Developer develops and tests the capabilities cartridge in Design Studio. See [Creating Capabilities Cartridge](#) for more information.
- 2. Package Capabilities Cartridge:** The OSM developer packages artifacts in Design Studio. See [Packaging Capabilities Cartridge](#) for more information.
- 3. Deliver Artifacts to Administrators:** The OSM developer delivers the capabilities cartridge to OSM DevOps Administrator. The OSM DevOps Administrator delivers the

capabilities cartridge manifest to the service catalog administrator. See [Delivering Artifacts to Administrators](#) for more information.

4. **Configure and Deploy to OSM Cartridge Assembler (OCA):** The OSM DevOps administrator performs the following:
 - a. Configures the OSM instance with connectivity details for a single Solution Designer instance.
 - b. Loads the capabilities cartridge into OCA.See [Configuring and Loading to OCA](#) for more information.
5. **Import the Capabilities Cartridge in Solution Designer:** The service catalog administrator imports the capabilities manifest into Solution Designer. See [Importing Capabilities Cartridge into Solution Designer](#) for more information.
6. **Create the Product Fulfillment Model:** In Solution Designer, a product specialist enriches the capabilities cartridge by defining the product fulfillment model. A Fulfillment Specialist defines the routing rules in the product fulfillment model. See [Creating Product Fulfillment Model](#) for more information.
7. **Publish to Test Workspace:** The service catalog administrator manages the initiative lifecycle and publishes it to a test workspace. When the initiative is published, SCD metadata is sent to the OCA. See [Publishing the Initiative to Test Workspace](#) for more information.
8. **Assemble and Deploy the Solution:** OSM Cartridge Assembler (OCA) then assembles and deploys the cartridge to the OSM instance. If the deployment is across the functional testing, acceptance testing, and production environments, the same major version is deployed to the database, although the final digit (representing the publish number) may vary between environments. An example of cartridge naming convention, for instance, employs a format such as '1.0.0.592', in which '1.0.0' signifies the version of the capabilities cartridge and '592' designates the publish number of the deployed cartridge. See [Assembling and Deploying the Solution](#) for more information.
9. **Publish to Production Workspace:** After the deployed solution is tested against OSM runtime, the service catalog administrator transitions the initiative through its lifecycle and finally, publish the initiative to the production workspace. After the publish operation is successful, the cartridge is deployed in the configured OSM production instance and OSM runtime processes new orders using the newly deployed solution. See [Publishing the Initiative to Production Workspace](#) for more information.

Creating Capabilities Cartridge

To create a capabilities cartridge:

1. Provision a dedicated OSM cloud native instance for testing to avoid overlaps with other environments. See *Preparing for OSM Cartridge Assembler* in *OSM Cloud Native Deployment Guide* for more information.
2. Develop the capabilities cartridge in Design Studio. You can include new fulfillment patterns, functions and system integrations as required by your solution. See *Generating a Capabilities Cartridge Using the Wizard* in *Design Studio Modeling OSM Orchestration* for more information. Add a test model to this in order to get a deployable solution cartridge to aid in cartridge development and testing.

Key activities include:

- Defining technical entities, behaviors, and orchestration logic.
- Implementing complex business rules and integrations.

- Defining a set of test data so that the orchestration logic can be validated through runtime order processing.
 - Repeating the building and deployment cycle until testing confirms correct cartridge processing.
3. Test the solution cartridge by following the steps:
 - a. Deploy the solution cartridge to the OSM cloud native test instance.
 - b. Run unit and functional tests to validate expected behavior.
 - c. Diagnose and fix defects found during testing.
 - d. Iterate until the cartridge meets all technical and functional acceptance criteria.

Packaging Capabilities Cartridge

Package the capabilities cartridge after validation is complete. Use the packaging approach designed to enable business users to operate within Solution Designer. See Capabilities Cartridge Build in *OSM Modeling Guide* for information on generating the capabilities cartridge.

The following two key artifacts are generated in the capabilities cartridge:

- **.cpar file**: An archive of reusable OSM content that is exposed in Solution Designer. This is not a deployable artifact and must be loaded in OCA that is configured with the connectivity details of the Solution Designer instance. The file is used to create an image that is loaded into the OSM Cartridge Assembler Microservice. See Installing OSM Cartridge Assembler (OCA) for Integration with Solution Designer in *OSM Cloud Native deployment guide* for more information on configuring the OCA and loading the `.cpar` file to OCA.
- **capabilitiesManifest.json file**: Describes the reusable components that are made available in the `.cpar` file. This file must be imported into Solution Designer.

Delivering Artifacts to Administrators

The generated artifacts are delivered to the administrators:

1. The OSM developer delivers the `.cpar` file to the OSM DevOps administrator for OSM instance configuration per specification.
2. The OSM DevOps in turn deliver the capabilities cartridge (**capabilitiesManifest.json**) manifest to the Service Catalog administrator.

The OSM Developer must collaborate with the OSM DevOps administrator and Service Catalog administrator to resolve any questions or issues and confirm all dependencies are satisfied.

Configuring and Loading to OCA

To configure and load the capabilities cartridge to OCA, the OSM DevOps Administrator:

1. Creates a new cpar container image using the Image Builder tool based on the `.cpar` file. The details of the new image are then added to the project specification file for OSM cloud native. For more information, see Building the Capabilities Cartridge Image.
2. Configures the OSM instance with connectivity details for a single Solution Designer instance. For more information, see [Configuring the Instance Specification](#).
3. Starts the OSM instance or upgrades it, if it is already running, to enforce the new `.cpar` file in OCA runtime.

Importing Capabilities Cartridge into Solution Designer

The Service Catalog administrator imports the **capabilitiesManifest.json** file into Solution Designer, which provides access to fulfillment entities, fulfillment patterns, fulfillment functions, fulfillment systems, order item properties, and granularities. In Solution Designer, you can build a fulfillment solution using the entities from a capabilities cartridge. See Importing Capabilities Cartridges in *Solution Designer User's Guide* for more details on importing capabilities cartridge into Solution Designer.

Creating Product Fulfillment Model

You can enrich the capabilities cartridge by creating a product fulfillment model and associating the model with an initiative. The fulfillment specialist creates the product fulfillment model in Solution Designer. This involves adding product specifications, mapping them to CFSSs, providing parameter mapping, defining target fulfillment patterns, and creating routing rules for order decomposition. See Creating Product Fulfillment Models using Guided Mode in *Solution Designer User's Guide* for more information on creating a product fulfillment model.

Publishing the Initiative to Test Workspace

The product fulfillment model containing fulfillment patterns, fulfillment functions, fulfillment systems, and routing rules is associated with an initiative. After you complete the product fulfillment model, you must publish the associated initiative to the Test workspace to perform functional testing and then acceptance testing. When the initiative is published to OSM participant in the test workspace, in Solution Designer, the initiative contents are sent to OCA. OCA then uses the initiative content and the capabilities cartridge `.cpar` file (loaded in OCA) to generate an OSM Central Order Management (COM) cartridge, which is deployed to the OSM runtime automatically.

See Publishing Initiatives to OSM Participant in *Solution Designer User's Guide* for information on how to publish the initiative to OSM participant.

Assembling and Deploying the Solution

When the service catalog administrator publishes the initiative to OCA, it also sends the metadata to OCA. OCA then assembles the initiative content and the capabilities cartridge `.cpar` file (loaded in OCA) to generate an OSM Central Order Management (COM) cartridge, which is deployed to the OSM runtime automatically. If the deployment is across the functional testing, acceptance testing, and production environments, the same major version is deployed to the database, although the final digit (representing the publish number) may vary between environments. An example of cartridge naming convention, for instance, employs a format such as 1.0.0.59, in which 1.0.0 signifies the version of the capabilities cartridge and 59 designates the publish ID of the solution cartridge.

Publishing the Initiative to Production Workspace

After functional testing and acceptance testing are successful, in Solution Designer, the Service Catalog administrator releases the initiative to the OSM participant in the production workspace. When the publish operation is successful, the initiative's status transitions to **Released**. OCA generates the production cartridge and deploys it in the OSM production instance. The OSM instance then processes new orders using the deployed solution.

See Publishing Initiatives to OSM Participant for information on how to publish the initiative to OSM participant.

Capabilities Cartridge Lifecycle Workflow

A capabilities cartridge is an important artifact that a OSM Developer develops and hands over to the DevOps admin for deployment. The lifecycle of a capabilities cartridge involves three key stages:

- Create
- Update
- Retire

Creating Capabilities Cartridge

See [Creating Capabilities Cartridge](#) for information on creating capabilities cartridge.

Updating Capabilities Cartridge

The capabilities cartridge may need to be modified to reflect the changes in business requirements.

1. OSM Developer updates the content in Design Studio and generates updated capabilities artifacts. See [Generating a Capabilities Cartridge Using the Wizard](#) for more information.
2. The updated `.cpar` file is passed to OSM DevOps Administrator and the capabilities manifest JSON file is passed to the Service Catalog Administrator.
3. The OSM DevOps Administrator creates a new image based on the updated `.cpar` file. The CNTK specification file for the OSM cloud native instance is also updated with the details of the new image, replacing the previous version. See [Building the Capabilities Cartridge Image](#) for more information.
4. Service Catalog Administrator imports the updated JSON file (capabilities cartridge) in Solution Designer.

Retiring Capabilities Cartridge

You may need to retire a Capabilities Cartridge when it is no longer needed due to service deprecation or redundancy.

1. The DevOps admin removes all entries related to the retiring Capabilities Cartridge from the cloud native toolkit specification file for the OSM cloud native instance to decommission the cartridge.
2. The retired capabilities cartridge is no longer active in the OSM environment. Retiring the cartridge is essential to maintain a clean environment.

Solution Lifecycle Workflow

The lifecycle of the solution cartridge in an OSM environment involves multiple users in creating, updating, and deleting these cartridges. The detailed workflow includes:

- Creating the Solution Cartridge
- Updating the Solution Cartridge

- Deleting the Solution Cartridge

Creating the Solution Cartridge

To create the solution cartridge, initiate the publish process in Solution Designer. The prerequisites are that the capabilities cartridge is created in Design Studio, loaded in OCA, imported into Solution Designer, enriched using product fulfillment model in Solution Designer.

Service catalog administrator initiates a publish in Solution Designer. See Publishing Initiatives to OSM Participant in *Solution Designer User's Guide* for information on publishing the initiative to OSM participant.

During the publish operation, Solution Designer sends the initiative content and the associated capabilities cartridge name with version appended with the publish operation ID to OCA. OCA creates the solution cartridge by merging the loaded capabilities cartridge `.cpar` file and the received initiatives content. OCA then deploys the solution cartridge into the OSM runtime environment.

For every publish operation, a new version of the solution cartridge is created and deployed to OSM environment. When you publish an initiative in Solution Designer, a unique publish number is generated which is used to identify the publish requests within the initiative.

Updating the Solution Cartridge

You may need to update the solution cartridge due to a business requirement such as the need to coordinate with a new type of downstream system (say, a work order management tool) for some types of products. Here you would need to code a new orchestration function to talk to the system. This would be worked into existing and, if needed, new fulfillment patterns.

Note

The updated solution cartridge is effective only for new orders, and does not retroactively affect the in-progress orders.

1. OSM Developer implements the fulfillment pattern changes to address the identified production bug.
Generate a new version of the solution reflecting the fix.
2. OSM developer then packages the cartridge artifacts, the `.cpar` file, and the JSON file.
3. OSM DevOps Administrator loads the `.cpar` into OCA.
4. Service Catalog Administrator imports the new version of capabilities cartridge into Solution Designer.
5. Fulfillment specialist updates the fulfillment model with capabilities cartridge and fulfillment pattern.
6. Service Catalog Admin initiates the publish operation to OSM Participant. OSM participant sends the updated content to OCA which in turn generates the updated solution cartridge and deploys it in OSM instance. Record the new version of the solution cartridge.

Existing orders are not affected by the new solution cartridge version and any in-progress orders will continue to progress using their pre-existing cartridge versions. OSM does not apply changes retroactively; the updated fulfillment pattern affects only new orders.

7. The upstream System (Order Management system) resubmits the stuck in-progress orders after making any necessary adjustments so they are processed under the updated fulfillment pattern. Validate that the orders are processed correctly with the updated fulfillment pattern.

Deleting the Solution Cartridge

OSM DevOps Administrator can undeploy all or selected solution cartridges from the OSM environment using the product specification and the `manage-cartridges.sh` cloud native toolkit script. For more information, see [Working with Cartridges](#).

3

Limitations and Restrictions

This chapter describes the limitations and restrictions that apply to the OSM-SCD integration, including considerations for OSM release 8.0.x and SCD release 8.3.x. It explains the system constraints, supported features, and specific guidelines you must follow during deployment and operation.

Limitations and Restrictions

This topic describes the limitations and restrictions for OSM release 8.0 and SCD release 8.3. Consider the following limitations when working on these releases of OSM and SCD.

- Orders submitted to OSM via the TMF REST API must contain values for all desired product characteristics, even for those that have been modeled with a default value in SCD. Failure to submit orders this way will result in those characteristics being deemed missing or empty.
- You must not model Product Specifications with characteristics that are complex data structures (Feature Groups in Solution Designer).
- You cannot retire a capabilities cartridge that has moved to production.
- You cannot use special characters in the value of a routing rule condition. For example, if "Australia & New Zealand" is one of the enumerated values on a product specification attribute, then you cannot use them in the conditions of a routing rule.

DevOps Considerations

This topic describes the DevOps considerations that you need to keep in mind while working on OSM and SCD.

- SCD Integration Callback: OSM Cartridge Assembler (OCA) uses a single Solution Designer URL for callbacks. OSM retrieves required data through this callback during cartridge assembly. An OSM cloud native instance can only interact with one Solution Designer instance.
- Version Digit Ownership: Version Digit Ownership: During dynamic cartridge assembly, OCA manages the last field of the 4 field cartridge version. This ensures cartridge versions are unique and can be mapped to initiative publishes.
- OSM Workspace: You can connect multiple OSM instances to one Solution Designer instance. Each OSM instance operates in a separate workspace. You can support different stages such as development, quality assurance, and user acceptance testing.

Miscellaneous Considerations

This topic describes the miscellaneous considerations that you need to keep in mind while working on OSM and SCD.

- OCA Cartridge Manifest Exposure: OCA provides a dedicated GET endpoint that lists manifests for each loaded capabilities cartridge. The returned JSON document can contain multiple manifests. However, Solution Designer imports only one manifest at a time.

Therefore, you must edit the JSON file to separate each manifest into its own document before importing, if the OCA is to be used as the source of the capabilities cartridge manifest (instead of direct communication between the Solution Developer and the Catalog Administrator).

- **Undeploying a Cartridge from OSM:** In non-production environments, you can take down the cloud-native instance, delete and recreate the OSM schema, and restart the instance. This purges the database of all orders and previously deployed cartridges.

Note

You should take this action only when nothing needs to be preserved.

In production environments, where purging the entire database is not an option, you can use the cloud native toolkit to remove specific cartridges. You can add the cartridge entries to the specification file, set the fast-undeployed state, and run synchronization. This process undeploys cartridges with precise control.

- **SCD Publish Number:** When you reinstall SCD as a fresh environment, all the database records are deleted and the SCD publish counter resets. When this happens, you may end up publishing to an OSM instance using an operation ID that was used on this OSM instance previously. Currently you also need to wipe OSM workspaces when the Solution Designer counter resets. This action keeps the operation IDs aligned between the applications.
- **Decomposition Rules:** In Design Studio, you are able to create decomposition rules for a single target order component. This allows different conditions to be used during different stages of decomposition. In Solution Designer, there is only one rule and the configuration applies to two target order components - system and granularity. It is not possible to use different conditions for each of these targets.

4

Troubleshooting

This chapter describes where to find log files, some common error scenarios and how to troubleshoot them.

Accessing Logs

This section describes how to access logs.

Use the following commands to access the OCA logs:

```
kubectl get pods -n project
kubectl logs -n project <oca pod>
```

When the system starts, the log displays the list of capabilities cartridges loaded into this OSM instance.

Note

You can set the OCA log level in the specification files available in the OSM cloud native toolkit if you require more detailed log information.

OSM-SCD Interaction Pattern

For every publish request initiated from Solution Designer, there is a series of interactions between the two applications. OCA logs several key messages for each of these interactions:

- Publish Request

```
[oracle.comms.ordermanagement.endpoints.impl.PublishOperationsEndpointImpl]
- Processing asynchronously for operation id : 10050
```

- Synchronous Response from OCA

```
publishOperations complete successful with code 202
```

- Asynchronous Response from OCA

```
Response posted :class PublishOperationResult {
  taskId: 10316
  result: completed
  message: Deployed Cartridge: TMF_PO_B2C_Solution and version:
1.0.0.0.10050
  errors: null
}
```

- Activate or Rollback Request

```
[oracle.comms.ordermanagement.endpoints.impl.PublishOperationsEndpointImpl]
- Metadata refresh complete. Cartridges activated successfully
```

Common Error Scenarios

This section describes some of the common error scenarios you might face and how to troubleshoot them.

Product Order Stuck in "Incomplete" State

Issue: If you have not configured the emulators properly, the order will not progress to the "Completed" state.

Resolution: Configure the emulators to ensure that order processing completes correctly. For more information, refer to [Installing the Emulators](#).

Connectivity Issues

The following error scenarios may happen if you are facing connectivity issues.

Publish Operation Returns 504 Connection Error

Issue: When you publish an initiative to OCA, it checks if the connection between SCD and OSM is functional. If you receive this error message, it signals an issue with SCD's connection to OSM.

Resolution: The possible resolution for this issue depends on the content in your initiative.

- If the initiative you are publishing defines the OSM participant, then it is possible that the connection details are incorrect. If this is the case, you need to validate the endpoint URL, and the scope and authentication details configured on the Connection entity.
- If you have already validated the connection details, and released the connection initiative, it is possible to see this error when you are publishing a fulfillment model. This usually indicates a transient network connection with OCA. You can do the following:
 1. Retry the publish operation.
 2. If the publish operation fails again, then check the status of the OCA pod in your OSM environment.

```
kubectl get pods -n project
```

If the OCA pod is restarting, then wait until it is running and then retry publishing from Solution Designer. If the OCA pod is not running, you need to investigate and either recreate or upgrade your OSM instance to bring the pod up. If the issue persists after multiple publish attempts, you need to contact Oracle support.

Publish Operation Fails with "java.net.UnknownHostException" Error

Issue: When you publish an initiative to OCA, if the OSM hostname is not added to the host alias, the following error appears in the OSM outcome: `java.net.UnknownHostException: <hostname>`.

```
2025-09-16 12:53:00.165 [kafka-14] INFO o.m.w.s.impl.AsyncMessageProducer -
Submit message: CloudEvent{id='ProcessTaskResponseEvent-10001-10002',
source=http://workspace-manager:8080, type='workspacemanager.publishevent',
data=io.cloudevents.core.data.PojoCloudEventData@fd8cc273,
```

```

extensions={partitionkey=10001}}
2025-09-16 12:53:00.182 [kafka-14] INFO o.m.w.s.impl.AsyncMessageConsumer -
Received event with id: ProcessTaskResponseEvent-10001-10002, source: http://
workspace-manager:8080, type: workspacemanager.publishEvent, data: class
ProcessTaskResponseEvent { class PublishEvent { eventType:
ProcessTaskResponseEvent operationId: 10001 taskId: 10002 } response: class
WsmPublishOperationResult { taskId: 10002 code: WM:09 reason: null message:
java.net.UnknownHostException: kscd.osmtest3.osm.org status: null reference
Error: null content: null errors: null result: failed } }

```

Resolution: To fix this error, in the SCD configuration file, under the workspace-manager, add a host alias section. Then, upgrade the SCD instance.

```

# List of host aliases.
hostAliases: []
# - ip: "127.0.0.1"
# hostnames:
# - "kscd.osmtest3.osm.org"

```

Publish Operation Returns 4xx Status Code Errors

Issue: When you publish an initiative to OCA, a 4xx status code may appear in the OCA log, indicating a publish operation failure. After 30 minutes, the publish attempt fails.

- If the SCD portion in the OCA secret has the wrong scope, you encounter a 403 Forbidden error.

```

2025-09-17 06:35:46.941 [[0x254e1d46 0x7cc449b2] WebServer socket] [INFO ]
[oracle.comms.ordermanagement.util.impl.RestHttpClientUtil] - Status: 403
2025-09-17 06:35:46.941 [[0x254e1d46 0x7cc449b2] WebServer socket] [INFO ]
[oracle.comms.ordermanagement.util.impl.RestHttpClientUtil] - Reason:
Forbidden
2025-09-17 06:35:46.941 [[0x254e1d46 0x7cc449b2] WebServer socket] [ERROR]
[oracle.comms.ordermanagement.util.impl.SCDClientImpl] - Failed to post
response to http://100.76.140.68:80/scd/api/workspaceOperations/v1/
publishOperationResponses/10005

```

- If the SCD portion in the OCA secret has incorrect client details, you encounter a 401 Unauthorized error.

```

2025-09-17 09:02:45.640 [[0x4ed627ed 0x37348992] WebServer socket] [INFO ]
[oracle.comms.ordermanagement.util.impl.RestHttpClientUtil] - Status: 401
2025-09-17 09:02:45.640 [[0x4ed627ed 0x37348992] WebServer socket] [INFO ]
[oracle.comms.ordermanagement.util.impl.RestHttpClientUtil] - Reason:
Unauthorized
2025-09-17 09:02:45.641 [[0x4ed627ed 0x37348992] WebServer socket] [ERROR]
[oracle.comms.ordermanagement.util.impl.SCDClientImpl] - Failed to post
response to http://100.76.140.68:80/scd/api/workspaceOperations/v1/
publishOperationResponses/10008

```

Resolution: Update the OCA secret `<project>-<test>-oca-credentials` with the correct SCD client or scope details based on the issue you encountered. After that, restart the instance using the `ocams` option.

Runtime Issues

The following error scenarios may happen if you are facing runtime issues.

Order Creation Fails with "**Product Specification Name is not a deployed entity**"

Issue: When the system displays this error, it means that the product specification has not been added to the Fulfillment Model.

Resolution: Ensure that the product specification in the inbound product order exists in the Fulfillment Model included in the cartridge deployment.

Order Creation Fails with "**Value Name is not a valid value for the element Element Name**"

Issue: If the element "*Element Name*" is a lookup field, verify that "*Value Name*" is present in its list of values.

Resolution: Ensure that the value you want to enter for the **Element Name** field is present in the list of accepted values.

Troubleshooting Workspace Publish Failures

This section describes the issues you might encounter when publishing an initiative to the Workspace and how to troubleshoot them:

- If an initiative publish remains in the "In Progress" state for longer than expected, it may time out. The default initiative publish duration is 30 minutes. The system then treats the attempt as a failure. Once this happens, you can reattempt the publish operation. If repeated publishing attempts fail, verify that the configuration for OCA contains the correct connection credentials and settings.
- Occasionally, the system may indicate that publishing to OSM succeeded, but the overall publish failed. This happens very rarely and is caused when the Workspace Manager returns an incorrect task sequence while performing the next task. If this happens, the OSM publish operation returns a successful response, but the overall status remains stuck for the timeout duration, which is 30 minutes. After the timeout, the overall status changes to "Failed" while the OSM status remains "Successful". If this happens, retry the publish operation.

5

About the Sample Integration Reference Solution Package

This chapter introduces the reference solution package that demonstrates the essential concepts in integrating OSM with SCD. The sample artifacts are available in the OSM Software Development Kit (SDK) folder.

The installation process for the Reference Solution Package consists of the following steps, which are performed across OSM and SCD:

1. Setting up OSM
 - a. Installing the emulators
 - b. Loading the capabilities cartridge
 - c. Creating or Upgrading OSM cloud native
2. Setting up SCD
 - a. Installing the product model
 - b. Importing the manifest file
 - c. Updating the product model
 - d. Creating the fulfillment model
3. Submitting product orders

Installing the Emulators

The reference solution uses four emulators to support order processing:

- Billing
- Shipping
- Partner gateway
- Provisioning

The SDK includes these emulator artifacts in the **SDK/Emulators** directory. This directory also provides a readme file that details the setup process. These steps are:

1. Creating the Emulator Images.
2. Creating the Emulator Pods.
3. Configuring the OSM cloud native instance to communicate with the emulators.

In the project specification file, add the following target systems:

Note

The names you add should match the following list, as the cartridge is configured to use them.

```
requiredTargetSystems:
- name: som-emulator
  description: "Emulator for SOM"
- name: shipping-emulator
  description: "shipping portal"
- name: partnergateway-emulator
  description: "Emulator for TMF 622 partner gateway"
- name: po-notification-listener
  description: "622 upstream mock"
- name: billing-emulator
  description: "Emulator for billing"
```

In the instance specification file, provide the authentication for the target systems you added as well as the emulator details:

```
# Define security scheme for target systems enabled with security
# For each security scheme defined kubernetes secret should be created using
# ${CNTK_HOME}/scripts/manage-target-system-credentials.sh script.
securitySchemes:
.....

targetSystems:
  systems:
    po-notification-listener:
      url: http://localhost:8080/mocked/fabric/notification/
      protocol: http
      description: mock system upstream to COM notifications
    som-emulator:
      url: http://<ip>:<port>/orchestration/<project>/<instance>/mock/tmf-api
      protocol: http
      description: som emulator
    shipping-emulator:
      url: http://<ip>:<port>/orchestration/<project>/<instance>/mock/tmf-api
      protocol: http
      description: shipping emulator
    billing-emulator:
      url: http://<ip>:<port>/orchestration/<project>/<instance>/mock/tmf-api
      protocol: http
      description: billing emulator
    partnergateway-emulator:
      url: http://<ip>:<port>/orchestration/<project>/<instance>/mock/tmf-api
      protocol: http
      description: Partner Gateway emulator
```

To validate your deployment, run the following command:

```
kubectl get pods -n project
```

This displays the pods for the four emulators.

Loading the Capabilities Cartridge

The capabilities cartridge is provided as a .cpar archive. Load this cartridge into the OCA microservice when you create or upgrade your OSM cloud native instance.

Navigate to the **SDK/Samples/TMFCartridges/Build Artifacts** directory. The archive is labelled as **RI_COM-2025xxxxx.cpar**.

Once you have accessed the archive, do the following:

1. Create a cpar image. Tag and push this image to an appropriate container image repository. For more information about creating a cpar image, see [Creating OSM Cloud Native Images](#).
2. Add the image details to the OSM cloud native toolkit specification files for your OSM cloud native instance. Edit the project specification, adding the image details to the capabilities cartridge section, update the image path based on how the image was pushed:

```
#capabilitiesCartridgeList: []
capabilitiesCartridgeList:
- name: TMF_PO_B2C_Solution
  version: "1.0.0.0.0"
  image: "localhost/tmf_po_b2c_solution-cpar:1.0.0.0.0"
```

Creating or Upgrading OSM Cloud Native

Whether you currently have an OSM cloud native instance running or not, you need to make adjustments to the specification files in the cloud native toolkit. You need to update the specification file to apply configuration that is needed to integrate with SCD. Do the following:

1. Update the OSM cloud native specification files:
 - a. Enable OCA and specify its image in the project specification file.
 - b. Uncomment OCA related annotations in the project specification file.
 - c. Specify the capabilities cartridge list, as described in Step 2, in [Loading the Capabilities Cartridge](#).
 - d. Add the SCD callback URL in the instance specification file.

```
# OSM Cartridge Assembler
#osm-cartridge-assembler: {} # This empty declaration should be removed
if adding items here.
osm-cartridge-assembler:
  # SCD Base URL
  scdBaseUrl: "http://<ip>:8001"
```

2. Create the ocaOidc secret using the `manage-credentials` script in the cloud native toolkit. The first set of OIDC credentials you create should be used for accessing OCA.

Note

You will need this information when you are setting up the connection to Solution Designer.

You will be prompted asking if you want to create credentials for SCD. Choose 1 to create the credentials. This second set of OIDC credentials are needed to enable OCA's callback to SCD.

3. Create the ingress.
4. Create your OSM cloud native instance.

To validate the setup, use the following command:

```
kubectl get pods -n namespace
```

You should see the OCA pod running.

Use the following command to access the logs for the OCA pod:

```
kubectl logs -n namespace <oca pod>
```

Towards the end of the log, you will see a statement showing that the capabilities cartridge is successfully loaded:

```
[main] [INFO ] [oracle.comms.ordermanagement.Main] - Loaded Capability  
Cartridges are -  
[main] [INFO ] [oracle.comms.ordermanagement.Main] - TMF_PO_B2C_Solution/  
1.0.0.0.0
```

The URL to test if the OCA microservice is up and knows about the capabilities cartridge is:

```
http://<instance>.<project>.osm.org:30305/orchestration/<project>/<instance>/  
cartridgeAssembler/v1/capabilitiesManifest
```

Note

Save this URL as you will need it while setting up the connection with Solution Designer.

Installing the Product Model

To install the product model:

1. Navigate to **SDK/Samples/TMFCartridges/ProductModel** and locate the script `productSpecLoader.sh` and the readme file, **README.md**.
2. Review the **README.md** file to find out any prerequisites you need to complete.

- After completing the prerequisite steps, run the script `productSpecLoader.sh` with the initiative ID and the commercial domain ID to load the product specifications into the specified initiative with the specified commercial domain.

```
productSpecLoader.sh $INITIATIVE_ID $COMMERCIAL_DOMAIN_ID
```

Importing the Capabilities Manifest File

To import the manifest file:

- Navigate to **SDK/Samples/TMFCartridges/BuildArtifacts** and locate the capabilities manifest file **RI_COM-2025xxxxxx-capabilitiesManifest.json**.
- Log in to Solution Designer and navigate to the Fulfillment application.
- Click **Import and select**, or drag and drop the manifest file. Click **Import** to import the file.

The tabs at the bottom of the page show you some of the entities that are available in the capabilities cartridge such as Fulfillment Patterns, Fulfillment Functions, and Fulfillment Systems.

Updating the Product Model

The SCD model does not contain certain information that is needed, which you need to add manually. Complete the following steps to add this information:

- Create the following product specifications:
 - Bundled Offering Product Specification
 - Promotional Offering Product Specification
 - Product Offering Pricing Product Specification
 - Wireless Data CPE Product Specification. Within this, add the following commercial parameters:
 - IMEI
 - Type
 - Brand
 - Model (Text)
- Create converters:
 - Navigate to **Common Elements** and select **Converters** from the bottom of the page.
 - Here, add a new converter, called **bandwidth**, and select the type **measurement**.
 - Add the following entries:

Unit of Measure	Value
KB	1
MB	1024
GB	1048576

- Add a second converter, called **roamingAreaMap**, and select the type **Value Map**.
- Add the following entries:

Input	Output
All	ALL
Canada	CA
USA	US
Europe	EU
Schengen Area	SA
Middle East	ME
Australia and New Zealand	ANZ
National	NTL
International	INTL
Asia	AS

3. Use converters on product specification parameters:
 - a. Open the **Data Roaming Product Specification** and navigate to **Parameter Mapping**.
 - b. Here, edit the mappings for Roaming Area to Roaming Zone. Click the icon to add a converter, choose roamingAreaMap and click Save. Repeat this step for the **Voice Roaming Product Specification** and the **Text Roaming Production Specification**.
 - c. Open the **Wireless Data Bandwidth Product Specification** and navigate to **Parameter Mapping**.
 - d. Here, edit the mappings. Click the icon to add the converter for monthly quota to monthly attribute. Select the bandwidth converter and click Save.

Creating the Fulfillment Model

To create the fulfillment model:

1. Navigate to the PSR Models application.
2. Click the ellipses and select **Create Product Fulfillment Model** from the menu that opens.
3. Name your model, and select the Capabilities Cartridge. Click **Continue**. The **Select commercial domain** step opens.
4. Select the **Mobile Domain**. The **Build Model** step opens.
5. In the **Build Model** step, you can configure products and map them to the CFS. You can also update or remove existing mappings as needed. Once you have done that, click **Continue**. The **Configure Parameters** step opens.
6. You do not need to make any changes in the **Configure Parameters** step as the parameters and parameter mappings are made available while importing the product model. Click **Continue**.
7. Define fulfillment by editing the canvas that opens. Add the following Product-Fulfillment mappings.

Product Specification	Fulfillment Pattern
Bundled Offering Product Specification Product Offering Pricing Product Specification	Billing Only
Promotional Offering Product Specification	Non Billing Item
Wireless Handset Product Specification	Shipped Product

Product Specification	Fulfillment Pattern
Digital TV Product Specification	Simple Product
SIM Card Product Specification	Shipped Service
All Remaining Product Specifications	Simple Service

8. Create the following routing rules, exactly as provided below. This will allow you to validate the basics with the existing product orders.

Rule	Product Centric	Function	Condition	Route To	Granularity
Shipping Rule	No	ShipOrderFunction	None	Shipping System	WholeGranularity
Billing Rule	No	FulfillBillingFunction	None	EmulatedBilling	ServiceBundleGranularity
ProvisioningRule	No	ProvisionOrderFunction	None	Emulated_SOM	WholeGranularity
ProdProvRule	Yes	ProductProvisionOrderFunction	Case1: Channel contains "Amazon" Case2: Channel contains "Disney" default:	AmazonPrime Disney PartnerGateway	OrderItemGranularity OrderItemGranularity WholeGranularity

9. In the **Publishing Center**, publish your initiative to your workspace, where a connection with OSM is configured.

Submitting Product Orders

Install the collection of product orders that you can submit to OSM.

1. Open Postman and click **Import**.
2. In the window that opens, navigate to the OSM SDK: **SDK/Samples/TMFCartridges/ProductOrders/RI-COM-2025xxxxx.json**.
3. Set the following variable values for the collection:

- The OIDC scope for the gateway microservice

```
GW_SCOPE : tmf
```

- The OIDC client id for the gateway microservice

```
GW_CLIENT_ID : osm_api
```

- The OIDC secret for the gateway microservice

```
GW_SECRET: OIDC secret
```

- The OIDC token URL for the gateway microservice

```
TOKEN_URL: OIDC token URL
```

- The OSM cloud native project name

```
PROJECT:
```

- The OSM cloud native instance name

INSTANCE :

4. Once you have updated the values for the variables, you can submit an order to OSM gateway for processing.

The orders in the following sections demonstrate various aspects of the default fulfillment model.

Order VPO41010_E_1531

Navigate to the OSM order orchestration UI to validate the Order State, the Version and the Order Transformation.

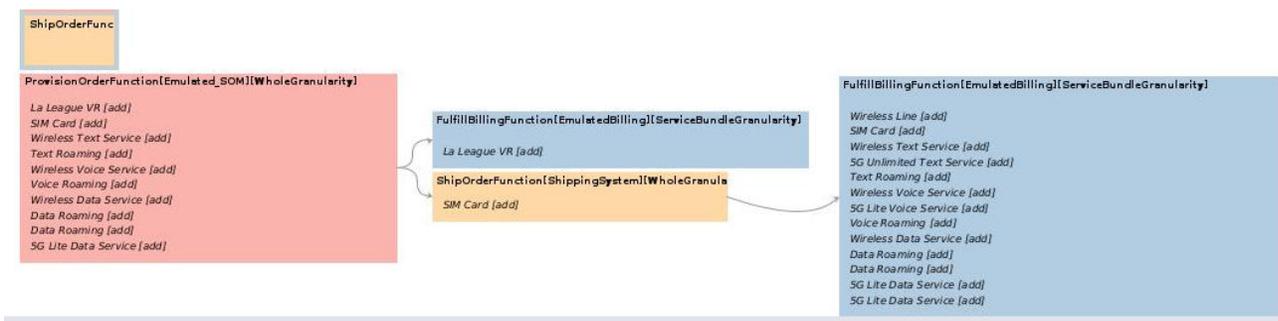
- Order State: The order should be in the **Completed** state. This confirms that your emulators are configured properly and are handling the messages exchanged for the system interactions.
- Version: The cartridge version should match the version seen in SCD after a successful publish action.
- Order Transformation: You should see five Transformed Order Lines. This indicates that five CFS line items were created, using both the Primary and Auxiliary mappings. The Wireless Data CFS line has an example of both the converters.

The Wireless Data Bandwidth PS has a **Monthly Quota** attribute on the inbound order of 20 GB. The transformed line representing the CFS shows the **Monthly Quota** attribute, which is the KB conversion.

Similarly, the **Roaming Area** attribute on the Data Roaming PS is converted from **Europe** in the inbound order to **EU** in the transformed line, representing the CFS.

The transformed lines also show that the **Service Specification Name** and **Id** fields are also auto populated. No explicit mapping of these values is required. The OCA microservice maps these values during cartridge assembly.

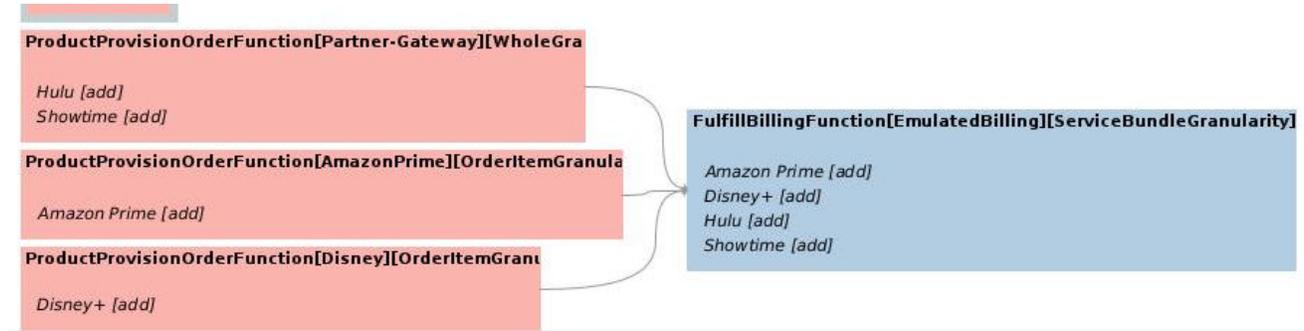
Figure 5-1 VPO41010_E_1531 Order Orchestration Plan



Order VPO41010_E_1204

This order contains four product order items for the Digital TV PS. Each line item specifies a different channel selection, which demonstrates the use of the routing rules you created. If you look at the orchestration plan in the OSM order orchestration UI, you will see the following:

Figure 5-2 VPO41010_E_1204 Order Orchestration Plan



This confirms the routing rule configuration, wherein the channel selection of Amazon Prime goes individually to the Amazon Prime system. Similarly, the channel selection of Disney+ goes individually to the Disney System. All other channel selections go to the Partner-Gateway system at whole granularity, or all together.

Additionally, all lines go together to the Emulated Billing system.