

Oracle Hospitality Integration Platform

Central Reservation Systems (CRS)



G45836-02
February 2026



Oracle Hospitality Integration Platform Central Reservation Systems (CRS),

G45836-02

Copyright © 2025, 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

Preface

1 Business Context

2 Prerequisites

Recommended Tools	1
Environment	1
Using Oracle Hospitality REST APIs for CRS Integrations	1
Using Business Events for CRS Integrations	2
Getting Started	2

3 Secure Payment Data Processing

Tokenization Workflow	1
-----------------------	---

4 Reservation Management

OPERA Cloud to Central Reservation System	1
Business Events	1
REST APIs	2
Noteworthy Business Logic	2
Workflows	3
Create Reservations in OPERA Cloud	3
Update Reservations in OPERA Cloud	4
Reservation Marked as Canceled or No-Show in OPERA Cloud	5
Reinstate Cancelled or No-Show Reservations in OPERA Cloud	5
Create Shared Reservations in OPERA Cloud	6
Unlink Shared Reservations in OPERA Cloud	7
Central Reservation System to OPERA Cloud	7
APIs	7
Noteworthy Business Logic	8
Workflows	10

Create Reservation in CRS	10
Update Reservation in CRS	11
Cancel Reservation in CRS	12
Create Linked Reservation in CRS	12
Create Joint Reservation in CRS	13
Break Linked Reservations in CRS	14
Break Shared Reservations in CRS	14
Break Joint Reservations in CRS	15

5 Group Management

OPERA Cloud to Central Reservation System	1
REST APIs	1
Noteworthy Business Logic	1
Workflows	2
Create Group Blocks in OPERA Cloud/OHIP	2
Update Group Blocks in OPERA Cloud/OHIP	3
Central Reservation System to OPERA Cloud	4
REST APIs	4
Noteworthy Business Logic	4
Workflows	5
Create Group Blocks (short duration) in CRS	5
Update Group Blocks (short duration) in CRS	7
Create Group Blocks (longer duration) in CRS	8
Update Group Blocks (longer duration) in CRS	10

6 Availability Management

OPERA Cloud to Central Reservation System	1
Business Events	1
Noteworthy Business Logic	1
Workflows	2
Create Restrictions in OPERA Cloud	2
Central Reservation System to OPERA Cloud	3
REST APIs	3
Noteworthy Business Logic	3
Workflows	4
Create Restrictions in CRS	4
Create Sell Limits in CRS	5
Create Hurdles in CRS	5

7 Rate Management

Central Reservation System to OPERA Cloud	1
REST APIs	1
Noteworthy Business Logic	2
Workflows	2
Create Rate Header in CRS	2
Update Rate Header in CRS	3
Create Rate Details (Flat Pricing) in CRS (short duration)	4
Update Rate Details (Flat Pricing) in CRS (short duration)	4
Create Rate Details (Flat Pricing) in CRS (longer duration)	5
Update Rate Details (Flat Pricing) in CRS (longer duration)	5
Create Rate Details (Derived Pricing) in CRS	6
Update Rate Details (Derived Pricing) in CRS	7

8 Inventory Management

OPERA Cloud to Central Reservation System	1
Business Events	1
Noteworthy Business Logic	2
Workflows	2
Update Inventory in OPERA Cloud	2

9 Anti-patterns

Preface

A Central Reservation System (CRS) is used by hotels and other lodging businesses to manage reservations, group bookings, and availability across multiple distribution channels (such as direct bookings, online travel agents, global distribution systems). This implementation guide outlines best practices for integrating a Central Reservation System with OPERA Cloud, addressing the most common use cases and workflows that align with industry standards and expectations.

Audience

This guide is intended for customers and partners who develop applications with the Oracle Hospitality Integration Platform.

Documentation

Oracle Hospitality product documentation is available on the Oracle Help Center at <https://docs.oracle.com/en/industries/hospitality/>.

Revision History

Date	Description of Change
October 2025	Initial publication
February 2026	Small update to title

1

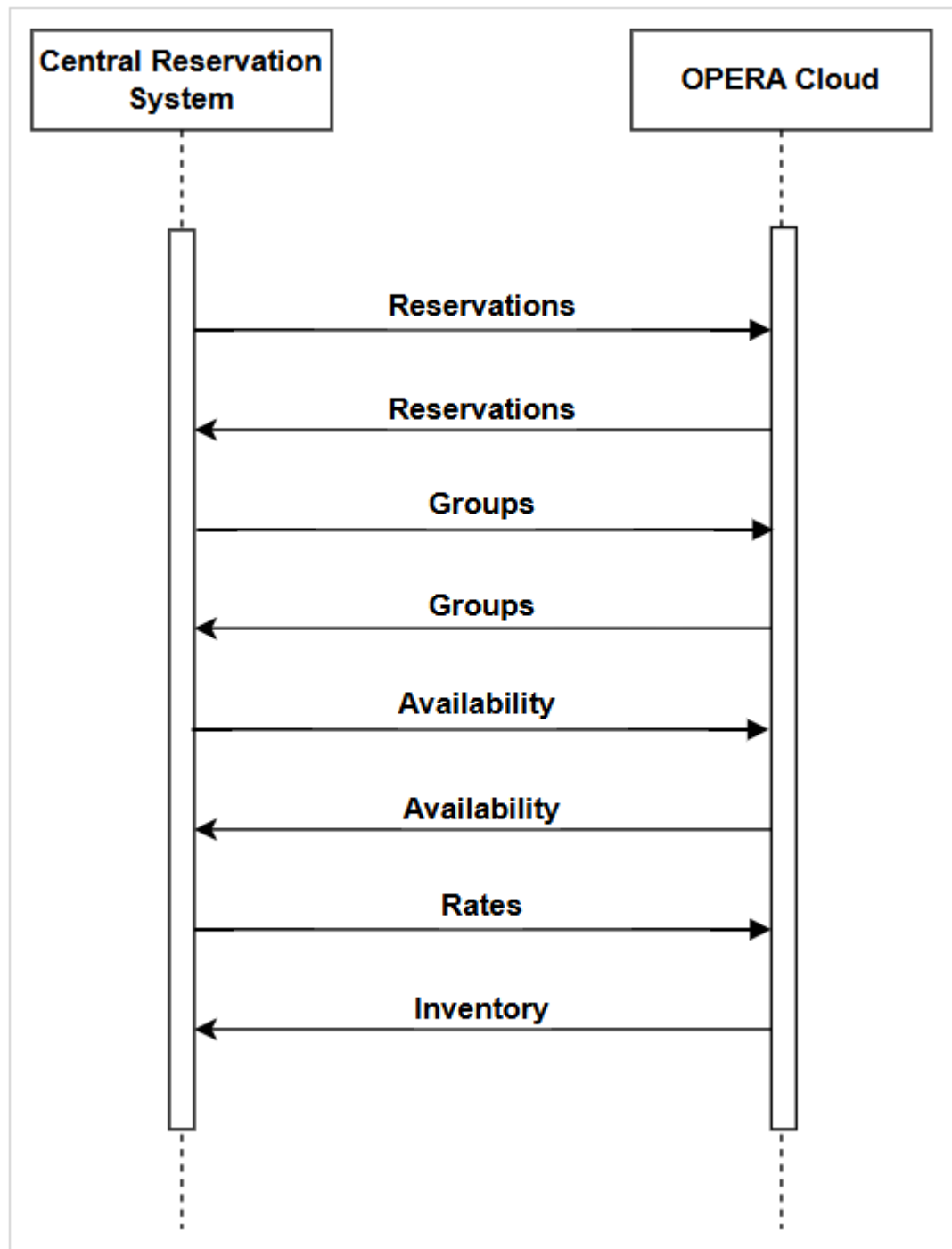
Business Context

A Central Reservation System (CRS) is used by hotels and other lodging businesses to manage reservations, group bookings, and availability across multiple distribution channels (such as direct bookings, online travel agents, global distribution systems). This implementation guide outlines best practices for integrating a Central Reservation System with OPERA Cloud, addressing the most common use cases and workflows that align with industry standards and expectations.

The Business Use Cases covered in this implementation guide are:

- [Secure Payment Data Processing](#)
- [Reservation Management](#)
- [Group Management](#)
- [Availability Management](#)
- [Rate Management](#)
- [Inventory Management](#)

The following diagram outlines the integration flows between CRS and OPERA Cloud, which are detailed in the upcoming sections:



2

Prerequisites

Tools and Environment information required to ensure that you can begin your integration development with Oracle Hospitality Integration Platform (OHIP).

Recommended Tools

Software tools that are useful to follow the steps outlined in this guide.

Tool	Description	Links
Github	A Github repository containing both Oracle Hospitality REST API specifications and accompanying Postman Collections.	Documentation Tool
Postman Collections	In the postman-collections folder in Github, you can download and use the Postman Collections associated to the use cases explained in this guide.	Postman Documentation Tool
GraphiQL	To better visualize the streaming API, a playground application is available that uses a tool called GraphiQL.	Documentation Tool
Node Reference Implementation	A reference implementation for the Streaming API in Node JS.	Tool

Environment

The following environment and modules must be available to perform the Business cases (described above) with the OPERA Cloud REST APIs.

OPERA Cloud Platform Module	Description	Minimum Version
OPERA Cloud Services	The customer must have a subscription to OPERA Cloud Foundation.	Supported with all OPERA Cloud versions.
Oracle Hospitality Integration Platform (OHIP)	For customers, OHIP is included in the subscription to OPERA Cloud Foundation. Partners need a subscription to Oracle Hospitality Integration Cloud Service	The latest released version. OHIP is a single version product.

Using Oracle Hospitality REST APIs for CRS Integrations

When integrating a Central Reservation System (CRS) with OPERA Cloud, Oracle Hospitality REST APIs provide essential capabilities for retrieving and aligning property data. Based on

your integration needs, you can leverage our Property APIs. The Property APIs in OHIP enable the retrieval of key property configuration data from OPERA Cloud. These APIs provide access to information such as room types, rate plans, market codes, and source codes—ensuring the CRS is aligned with the property's setup. Accurate property data supports seamless reservation creation, rate updates, and availability checks.

Using Business Events for CRS Integrations

Business Events - A business event is an event (that is, an update) that happens to a resource, for example, a reservation. You can find more detailed implementation guidelines on using the business events in the [Business Events topic](#) in the OHIP User Guide.

Getting Started

To get started, you must perform the following tasks:

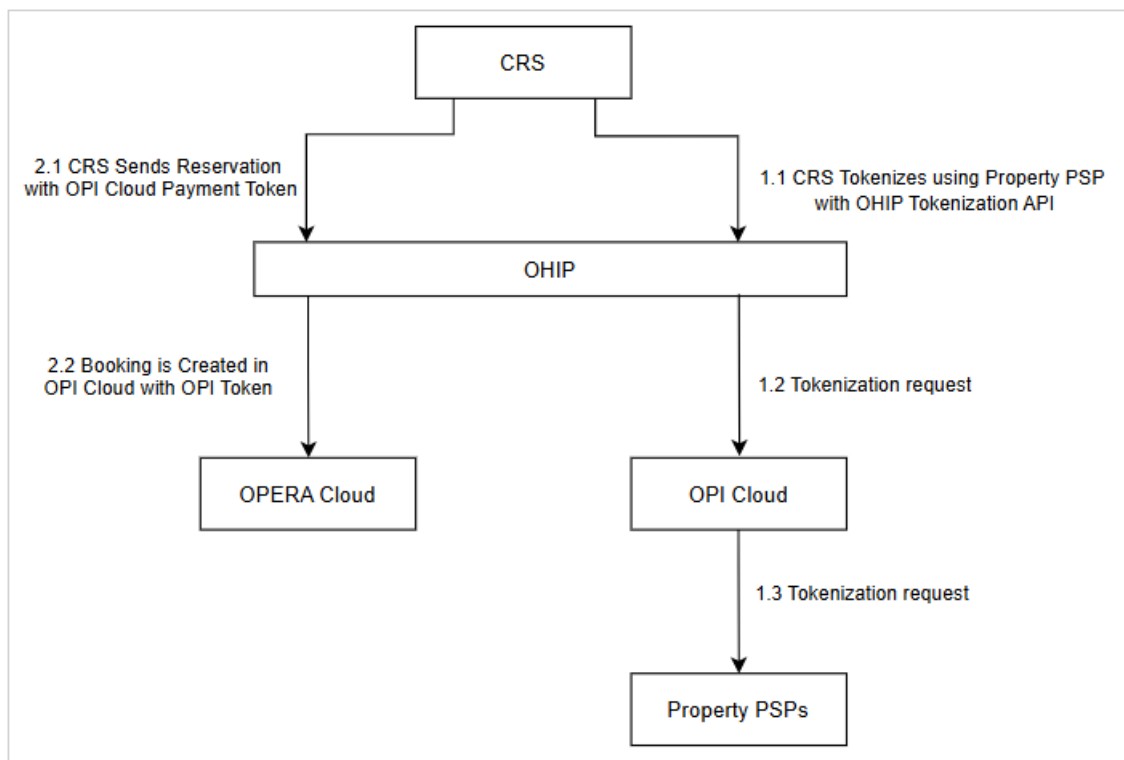
- [Accessing the Oracle Hospitality Integration Platform Developer Portal](#)
- [Creating an Integration User](#)
- [Registering an Application](#)
- [Using the Oracle Hospitality APIs](#)
- [Configuring Business Events](#)

3

Secure Payment Data Processing

Tokenization Workflow

This section outlines how a Central Reservation System (CRS) uses OHIP and OPI Cloud to securely tokenize and transmit payment data using OPI Cloud Payment Tokens. The tokenization workflow ensures that credit card data is processed in a PCI-compliant manner using certified Property Payment Service Providers (PSP).



1.1 CRS Tokenizes Using Property PSP through OHIP Tokenization API

The CRS initiates tokenization by calling the OHIP Tokenization API. This request is forwarded to OPI Cloud, which then routes the tokenization request to the appropriate Property PSP already certified with OPI Cloud.

1.2–1.3 Tokenization Request Propagation

OHIP relays the tokenization request to OPI Cloud. OPI Cloud then communicates with the correct PSP to generate a secure payment token.

2.1 CRS Sends Reservation with OPI Cloud Token

Once tokenization is complete, the CRS sends a reservation to OHIP that includes the OPI Cloud Payment Token.

2.2 Booking Is Created in OPERA Cloud

Using the OPI Token, OHIP creates the booking in OPERA Cloud, associating it with the tokenized payment details.

Refer to the [Reservations Credit Card Payment Methods](#) Implementation Guide for additional information and operation samples.

4

Reservation Management

CRS integrations typically involve both sending reservations to OPERA Cloud and receiving reservations from it. This ensures that reservation data remains synchronized across systems, allowing real-time updates for availability, modifications, and cancellations.

OPERA Cloud to Central Reservation System

When reservations are created, modified, or cancelled in OPERA Cloud, updates are sent to the CRS to ensure that availability and booking status remain consistent across all connected channels.

Business Events

Following are the most commonly used Business Events to send reservation data from OPERA Cloud to a Central Reservation System, though others may also apply:

Module	Business Event	Data Elements	Description
RESERVATION	NEW RESERVATION	Select only the primary key and minimal required fields.	Triggered when a reservation is created.
RESERVATION	UPDATE RESERVATION	Select only the primary key and minimal required fields.	Triggered when a reservation is updated.
RESERVATION	CHECK IN	Select only the primary key and minimal required fields.	Triggered when a reservation is checked in.
RESERVATION	CHECK OUT	Select only the primary key and minimal required fields.	Triggered when a reservation is checked out.
RESERVATION	REVERSE CHECK IN	Select only the primary key and minimal required fields.	Triggered when a checked-in reservation is reversed and returned to reserved status.
RESERVATION	REVERSE CHECK OUT	Select only the primary key and minimal required fields.	Triggered when a checked out reservation is reversed and returned to in-house status.
RESERVATION	CANCEL	Select only the primary key and minimal required fields.	Triggered when a reservation is canceled.

RESERVATION	NO SHOW	Select only the primary key and minimal required fields.	Triggered when a reservation is no-show.
RESERVATION	ROLLBACK CANCEL	Select only the primary key and minimal required fields.	Triggered when a canceled reservation is reinstated and returned to reserved status.
RESERVATION	REACTIVATE NO SHOW	Select only the primary key and minimal required fields.	Triggered when a no-show reservation is reinstated and returned to reserved status.
RESERVATION	JOIN GUEST	Select only the primary key and minimal required fields.	Triggered when two or more individual reservations are combined to make a single share reservation.
RESERVATION	SEPARATE GUEST FROM SHARE	Select only the primary key and minimal required fields.	Triggered when shared reservations are broken and moved back to individual reservations.

REST APIs

Following are the most commonly used REST APIs for retrieving reservation data from OPERA Cloud to CRS:

API Name	Data Direction	Description
getReservation	OPERA Cloud to CRS	Retrieve details of a given reservation.
putReservation	CRS to OPERA Cloud	Update Reservation: CRS confirmation number will be sent back to OPERA through the reservation modification call.

Noteworthy Business Logic

Business Data	Comments
Reservation Status	CRS must align with statuses supported by OPERA Cloud: - Reserved - the reservation is confirmed but the guest has not yet checked in. - In House - the guest has checked in and is currently staying at the property. - Checked Out - the guest has completed their stay and checked out of the property. - No Show - the guest did not arrive and the reservation was not canceled. - Cancelled - the reservation was canceled before the arrival date. Please note that invalid statuses may cause rejection.
CRS Confirmation Number	This is a unique identifier assigned by the CRS. If sent, the CRS Confirmation Number will be stored in OPERA Cloud.

OTA/PNR Reservation ID	External references (such as OTA confirmation numbers) can be included but should be passed in the appropriate field for cross-referencing.
Travel Agent Profile	CRS might support multiple travel agents per reservation. These will include primary and secondary travel agents. OPERA Cloud allows only one travel agent profile; the secondary agents from CRS are mapped as the source profiles.
Loyalty Data	Includes membership program code, ID, and level. Please note that the data is stored at the reservation level.
Guest Count by Type	OPERA Cloud supports only two guest types: Adult and Child. Please note that this must be mapped correctly in CRS as there is no default conversion logic.
Ages of Guests	If sent, age information will be stored in OPERA Cloud. (Optional).
Non-Room Products	Non-room products in OPERA Cloud refer to items or services sold alongside a room reservation but not part of the physical room itself. These are supported as part of packaged rate or as add-ons. Product codes must match between CRS and OPERA Cloud. Please note that no defaults are applicable.
Special Request Codes	The special request codes must be mapped correctly between CRS and OPERA Cloud.
Day Use Reservations	Reservations with same-day arrival and departure are flagged appropriately. CRS stores the reservation with the departure date set to one day after the arrival date and marks it with a day-use flag.
Reservation Cancellations	The OPERA cancelation number can be stored in the CRS reservation record.
Multi-Rate Plan Bookings	OPERA supports booking reservations with multiple transient rate plans.
Room Number	Can be provided in the reservation payload. If not provided, OPERA Cloud will assign room number during check-in.
Comments and Comment Types	Comments should be structured to preserve important data if overwritten. Comments in modifications will overwrite previous comments.
Link Types (Share with / Joint)	Both “Share with” and “Joint (Travel With)” link types are supported by OPERA Cloud. Please ensure correct formatting in payload.
Non-Guaranteed Reservations	Supported by OPERA Cloud and can be communicated through CRS. Appropriate configuration is required in both systems.
Reinstating Reservations	Only reservations canceled or marked as no-show within one day after arrival can be reinstated by CRS.
Payment Sub-Type	To be mapped using a specific UDF (such as UDF#XX). Please note that the mapping logic must be predefined.
Redemption Transaction ID	Award number can be passed to OPERA Cloud as a form of payment or loyalty redemption.
Error Logging in CRS	Any functional or data errors encountered during processing should be logged within the CRS system for traceability and issue resolution.

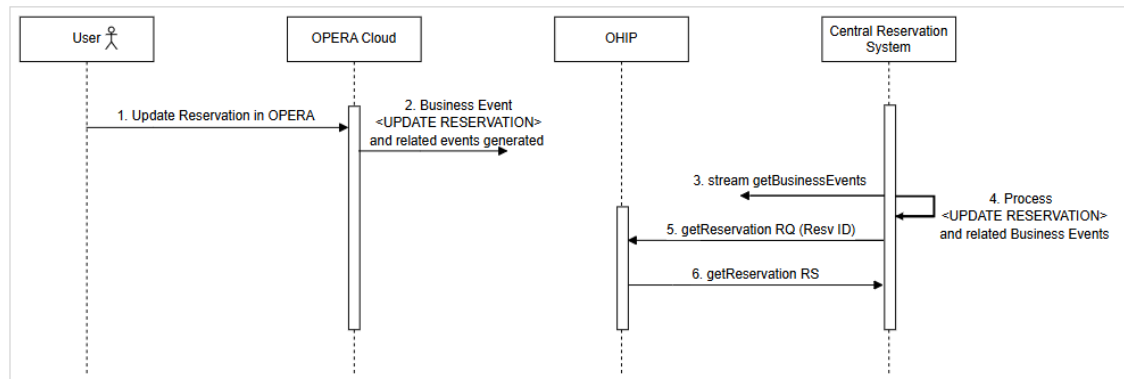
Workflows

Create Reservations in OPERA Cloud

When a reservation is created directly in OPERA Cloud - either by front desk staff, through a direct channel, or through another integrated system - the reservation data can be published to

the CRS through Business Events and Publishing configuration. This ensures that reservations remain synchronized between the CRS and OPERA Cloud.

The diagram and detailed steps below illustrate the end-to-end create reservation flow from OPERA Cloud to the Central Reservation System.

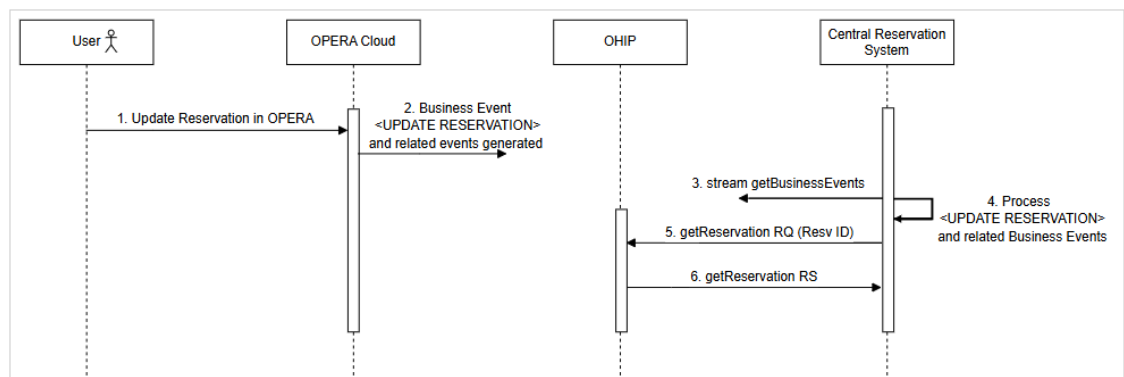


1. User creates a reservation in OPERA Cloud.
2. OPERA Cloud generates a NEW RESERVATION Business Event.
3. CRS can stream the generated Business Events from OPERA Cloud using `getBusinessEvents`.
4. CRS identifies processes the NEW RESERVATION event.
5. CRS sends a `getReservation` request to OHIP.
6. OHIP responds with the full reservation details.
7. CRS confirmation number is sent back to OHIP using the `putReservation` call.
8. OHIP successfully updates reservation and responds with success.

Update Reservations in OPERA Cloud

When a reservation is modified within OPERA Cloud - whether through the front desk, a direct integration, or another connected system - a Business Event is generated and sent to the Central Reservation System (CRS). This allows the CRS to capture and reflect the updated reservation details promptly.

The diagram and detailed steps below illustrate the end-to-end update reservation flow from OPERA Cloud to the Central Reservation System.

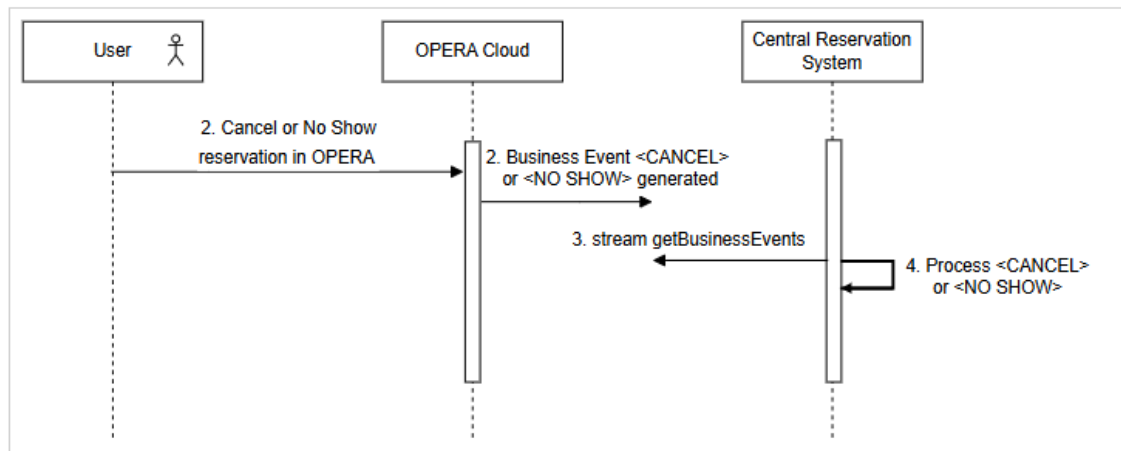


1. User modifies a reservation in OPERA Cloud.
2. OPERA Cloud generates an UPDATE RESERVATION Business Event along with related Business Events (CHECK IN, CHECK OUT, REVERSE CHECK IN, REVERSE CHECK OUT).
3. CRS can stream the generated Business Events from OPERA Cloud using `getBusinessEvents`.
4. CRS identifies and processes the UPDATE RESERVATION and related Business events (CHECK IN, CHECK OUT, REVERSE CHECK IN, REVERSE CHECK OUT).
5. CRS sends a `getReservation` request to OPERA Cloud.
6. OPERA Cloud responds with the full reservation details.

Reservation Marked as Canceled or No-Show in OPERA Cloud

When a reservation is either canceled or marked as a no-show in OPERA Cloud - whether manually by front desk staff or automatically through system processes - a Business Event is generated and made available to the Central Reservation System (CRS). This event ensures that the CRS is notified of the change in reservation status, allowing it to update its records, release inventory, and reflect accurate availability across connected booking channels.

The diagram and detailed steps below illustrate the end-to-end cancel/no show reservation flow from OPERA Cloud to the Central Reservation System.



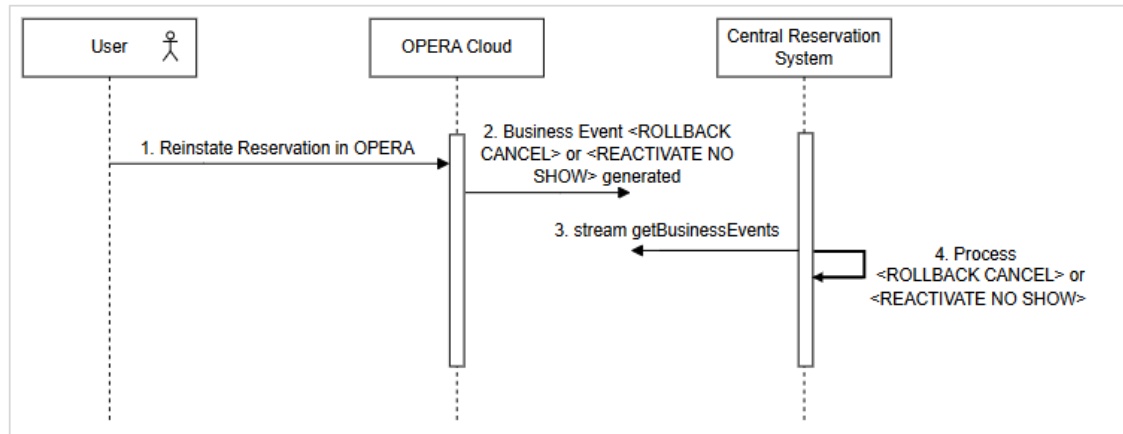
1. User cancels or marks a reservation as no-show in OPERA Cloud.
2. OPERA Cloud generates a CANCEL or NO SHOW Business Event.
3. CRS can stream the generated Business Events from OPERA Cloud using `getBusinessEvents`.
4. CRS identifies and processes the CANCEL or NO SHOW event.

Reinstate Cancelled or No-Show Reservations in OPERA Cloud

When a previously canceled or no-show reservation is reinstated in OPERA Cloud - typically by front desk staff or through a connected system - a Business Event is triggered and made available to the Central Reservation System (CRS). This ensures the CRS is informed of the

reinstatement, allowing it to restore the reservation, adjust inventory, and maintain accurate availability and booking records across all integrated distribution channels.

The diagram and detailed steps below illustrate the end-to-end reinstate reservation flow from OPERA Cloud to the Central Reservation System.

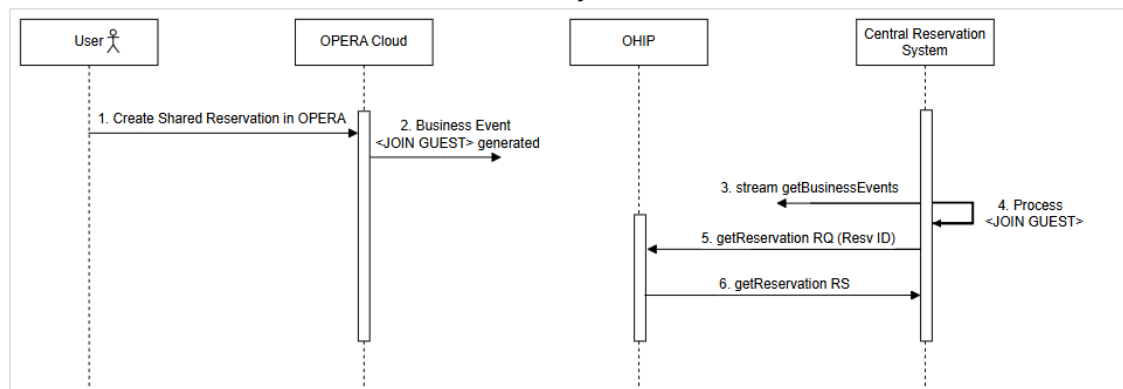


1. User reinstates a cancelled or no-show reservation in OPERA Cloud.
2. OPERA Cloud generates a Business Event: ROLLBACK CANCEL or REACTIVATE NO SHOW.
3. CRS can stream the generated Business Events from OPERA Cloud using `getBusinessEvents`.
4. CRS identifies and processes the ROLLBACK CANCEL or REACTIVATE NO SHOW event.

Create Shared Reservations in OPERA Cloud

A Shared Reservation involves multiple guests staying in the same room, each with an individual reservation record. When a shared reservation is created in OPERA Cloud, a Business Event is triggered for each individual reservation in the share group. These events are then made available to the Central Reservation System (CRS), allowing it to accurately capture and process all related reservations.

The diagram and detailed steps below illustrate the end-to-end create shared reservation flow from OPERA Cloud to the Central Reservation System.



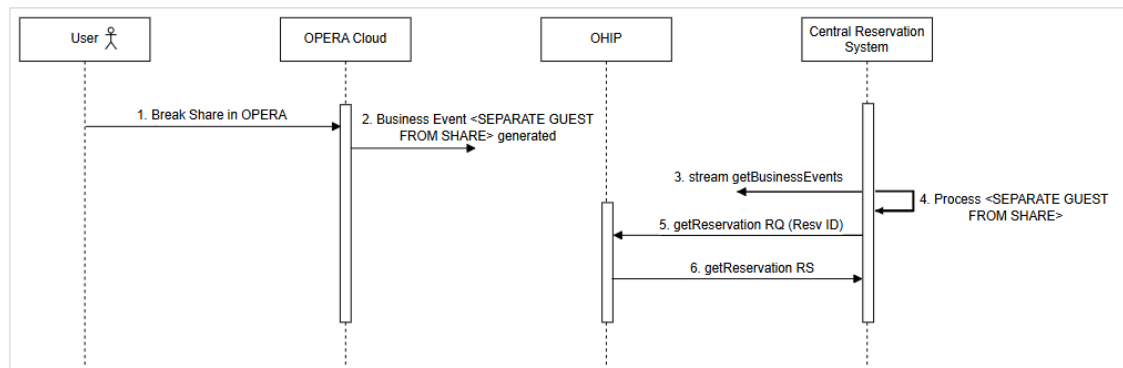
1. User creates a shared reservation in OPERA Cloud.

2. OPERA Cloud generates a JOIN GUEST Business Event.
3. CRS can stream the generated Business Events from OPERA Cloud using `getBusinessEvents`.
4. CRS identifies and processes the JOIN GUEST event.
5. CRS sends a `getReservation` request to OHIP.
6. OHIP responds with the full reservation details.

Unlink Shared Reservations in OPERA Cloud

When a shared reservation is unlinked (broken) in OPERA Cloud, a Business Event is triggered for each affected reservation. This event is made available to the Central Reservation System (CRS), ensuring that the CRS can update its records to reflect the change. By processing the share break event, the CRS can reassign room occupancy, adjust guest details, and maintain accurate inventory and reservation status across all channels.

The diagram and detailed steps below illustrate the end-to-end break shared reservation flow from OPERA Cloud to the Central Reservation System.



1. User breaks a shared reservation in OPERA Cloud.
2. OPERA Cloud generates a SEPARATE GUEST FROM SHARE Business Event.
3. CRS can stream the generated Business Events from OPERA Cloud using `getBusinessEvents`.
4. CRS identifies and processes the SEPARATE GUEST FROM SHARE event.
5. CRS sends a `getReservation` request to OHIP.
6. OHIP responds with the full reservation details.

Central Reservation System to OPERA Cloud

The CRS communicates reservation details to OPERA Cloud whenever a booking is made through external sources, ensuring real-time updates at the property level.

APIs

Following are the most commonly used REST APIs for sending reservation data from a CRS to OPERA Cloud:

API Name	Data Direction	Description
postReservation	CRS to OPERA Cloud	Create Reservation
putReservation	CRS to OPERA Cloud	Update Reservation
postCancelReservation	CRS to OPERA Cloud	Cancel Reservation
postCombineShareReservations	CRS to OPERA Cloud	Share Reservations
deleteShareReservation	CRS to OPERA Cloud	Break Share Reservations
deleteReservationPackage	CRS to OPERA Cloud	Remove Packages from a Reservation
postReservationLinks	CRS to OPERA Cloud	Link Reservations
deleteReservationLinks	CRS to OPERA Cloud	Break Linked Reservations

Noteworthy Business Logic

Business Data	Comments
Default Values	The postReservation message includes a data element called reservationNotification - To minimize the chance of reservation failures, it is recommended to send reservationNotification=Y - When reservationNotification=Y, OPERA Cloud applies default values if configured in OPERA Controls for any of the following data elements that are missing or not recognized in the message payload: – Cancellation Reason Code – Guarantee Code – Market Code – Payment Method – Rate Plan Code – Room Type – Source Code
Reservation Status	CRS must align with statuses supported by OPERA Cloud: - Reserved (The reservation is confirmed but the guest has not yet checked in) - Cancelled (The reservation was canceled before the arrival date) Please note that invalid statuses may cause rejection.
CRS Confirmation Number	This is a unique identifier assigned by the CRS. If sent, the CRS Confirmation Number will be stored in OPERA Cloud.
OTA/PNR Reservation ID	External identifiers (such as OTA confirmation numbers) can be included but should be passed in the appropriate field for cross-referencing.

Profile Matching	- Any profile passed in the request payload is created if it does not exist in OPERA Cloud. - If a profile passed in the request payload exists in OPERA Cloud, it is matched using the following rules - Guest Profile * OPERA Profile ID * Loyalty Number * Client ID - Company Profile * OPERA Profile ID * Company ID - Travel Agent Profile * OPERA Profile ID * IATA Number - Source Profile * OPERA Profile ID * Client ID
Travel Agent Profile	CRS supports up to two travel agents per reservation, a primary and a secondary. OPERA Cloud allows only one travel agent profile; the secondary agent from CRS is mapped as the source profile.
Loyalty Data	Includes membership program code, ID, and level. Please note that the data is stored at the reservation level as opposed to storing at profile level.
Guest Count by Type	CRS must align with OPERA's support for only two guest types: Adult and Child. <i>Please note that this must be mapped correctly in CRS as there is no default conversion logic.</i>
Ages of Guests	If sent, age information will be stored in OPERA Cloud.
Non-Room Products	Non-room products in OPERA Cloud refer to items or services sold alongside a room reservation but not part of the physical room itself. These are supported as part of packaged rate or as add-ons. Product codes must match between CRS and OPERA Cloud. <i>Please note that no defaults are applicable.</i>
Special Request Codes	The special request codes must be mapped correctly between CRS and OPERA.
Day Use Reservations	Reservations with same-day arrival and departure are flagged appropriately. CRS stores the reservation with the departure date set to one day after the arrival date and marks it with a day-use flag.
Cancellation Reason Code	Applied when a reservation is canceled. <i>Note: The Cancellation Reason Code is mandatory in OPERA Cloud, but optional in CRS.</i>
Multiple Rooms of Same Type	CRS supports booking multiple rooms of the same type under one reservation.
Multi-Rate Plan Bookings	A reservation may include multiple transient rate plans. However, they are supported and processed as part of a single booking flow.
Room Type Code	Room type codes in CRS must match OPERA Cloud configuration. Default values should be defined in OPERA Controls to ensure consistency.
Room Number	Room inventory in CRS must align with the room unit configuration defined in OPERA Cloud. Once a guest has checked in, CRS will not update or change the assigned room number.

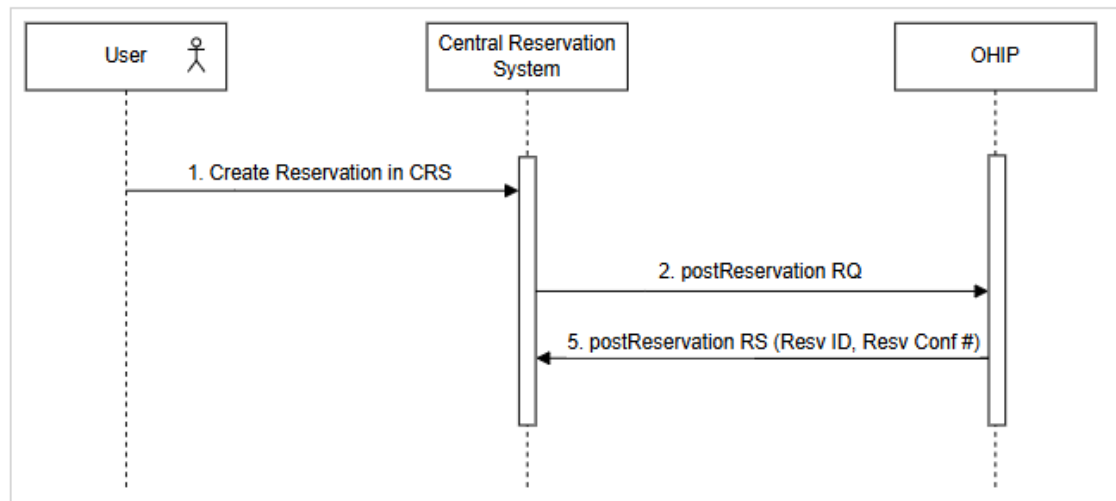
Rate Plan Code	Used to determine pricing, availability, and package inclusions. Must match OPERA Cloud configuration. Note: A default rate plan should be configured in OPERA to prevent reservation failures if not provided.
Comments & Comment Types	Comments should be structured to preserve important data if overwritten. Comments in modifications will overwrite previous comments.
Link Types (Share with / Joint)	Both “Share with” and “Joint (Travel With)” link types are supported by OPERA Cloud. Please ensure correct formatting in payload.
Non-Guaranteed Reservations	Supported by OPERA Cloud and can be communicated through CRS. Appropriate configuration is required in both systems.
Reinstating Reservations	Only reservations canceled or marked as no-show within one day after arrival can be reinstated by CRS.
Payment Sub-Type	To be mapped using a specific UDF (such as UDF#XX). Please note that the mapping logic must be predefined.
Redemption Transaction ID	Award number can be passed to OPERA Cloud as a form of payment or loyalty redemption.
Payment / Credit Card Info	CRS sends only tokenized payment information. OPERA Cloud enforces high security standards by preventing the storage or transmission of clear text credit card data. Note: A default payment method should be configured in OPERA to minimize processing failures when tokens are missing.

Workflows

Create Reservation in CRS

Creating a reservation in CRS triggers the transfer of booking data to OPERA Cloud. This ensures that the property has visibility into the upcoming stay, allowing teams to prepare for arrival and allocate resources accordingly.

The diagram and detailed steps below illustrate the end-to-end create reservation flow from Central Reservation System to OPERA Cloud.

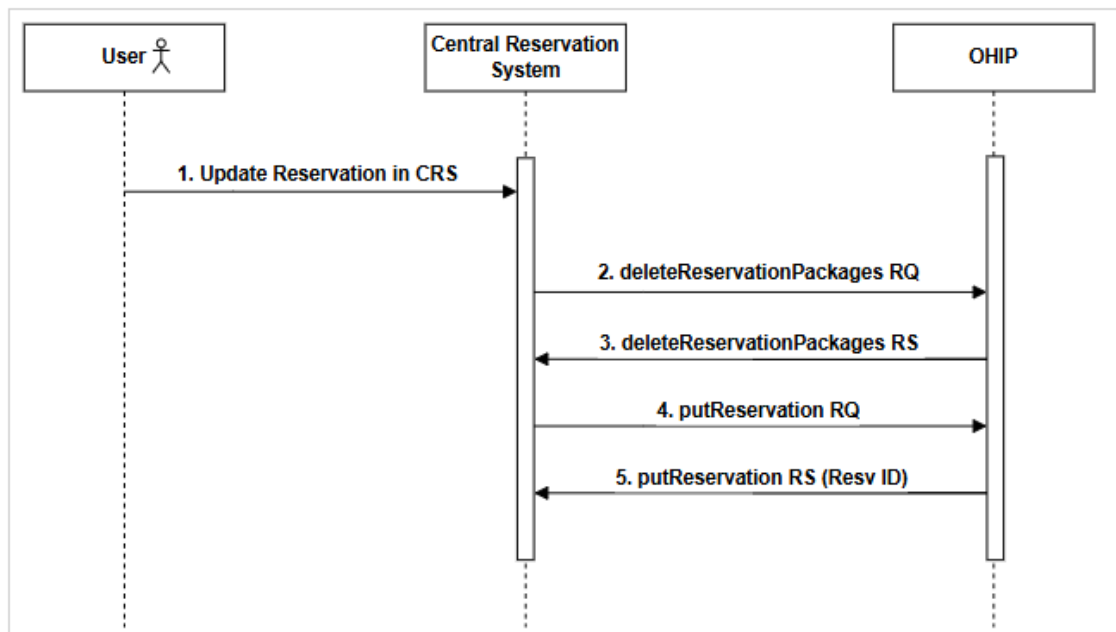


1. User creates a reservation in the CRS.
2. CRS sends a postReservation request to OHIP containing the full reservation data.
3. OHIP returns a successful 201 response with a Reservation ID and Reservation Confirmation Number, which are then stored in CRS. OHIP automatically creates profiles in OPERA Cloud based on guest, company, or travel agent details provided in the reservation.

Update Reservation in CRS

Any reservation changes made in CRS, whether minor or major, are reflected in OPERA Cloud in near real time. This ensures the front desk always has up-to-date information for guest service, inventory planning, and reporting.

The diagram and detailed steps below illustrate the end-to-end update reservation flow from Central Reservation System to OPERA Cloud.

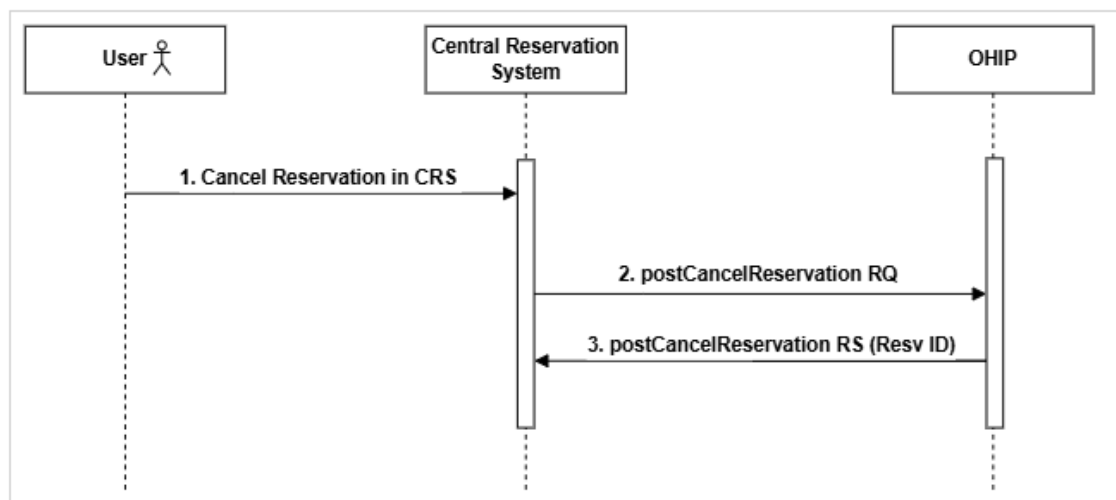


1. User updates a reservation in the CRS.
2. Non-room products (included as packaged rates or add-ons) are deleted and re-added during reservation modifications. CRS sends a deleteReservationPackages request to OHIP to remove any package related information.
3. OHIP returns a 201 response to indicate successful deletion of the packages.
4. CRS sends a putReservation request to OHIP containing the reservation updates.
5. OHIP returns a successful 201 response with the Reservation ID of the updated reservation. OHIP automatically creates profiles in OPERA Cloud based on guest, company, or travel agent details provided in the reservation.

Cancel Reservation in CRS

When a reservation is canceled in the CRS, the update is immediately reflected in OPERA Cloud. This ensures that room availability is released and the property's systems remain accurate for planning, reporting, and guest communication.

The diagram and detailed steps below illustrate the end-to-end cancel reservation flow from Central Reservation System to OPERA Cloud.



1. User cancels a reservation in CRS.
2. CRS sends a postCancelReservation request to OHIP for the reservation cancellation.
3. OHIP returns a 201 response with a Reservation ID to indicate successful cancelation of the reservation.

Create Linked Reservation in CRS

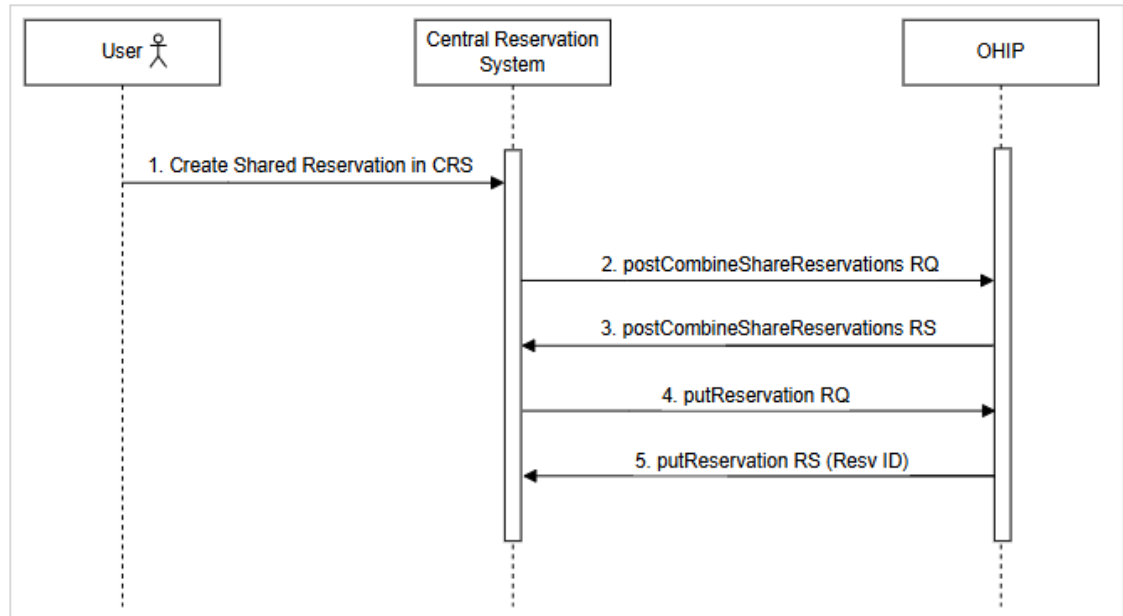
A linked reservation can be of two types:

- **Shared Reservation:** Multiple guests sharing the same room, each with their own reservation record.
- **Joint Reservation:** Separate reservations linked for guests traveling together but staying in different rooms.

When linked reservations - such as shared or joint - are created in the CRS, each related booking is sent to OPERA Cloud with the appropriate linkage details. This ensures the property can manage room assignments, billing preferences, and guest expectations for connected reservations seamlessly.

4.1. Create Shared Reservation in CRS

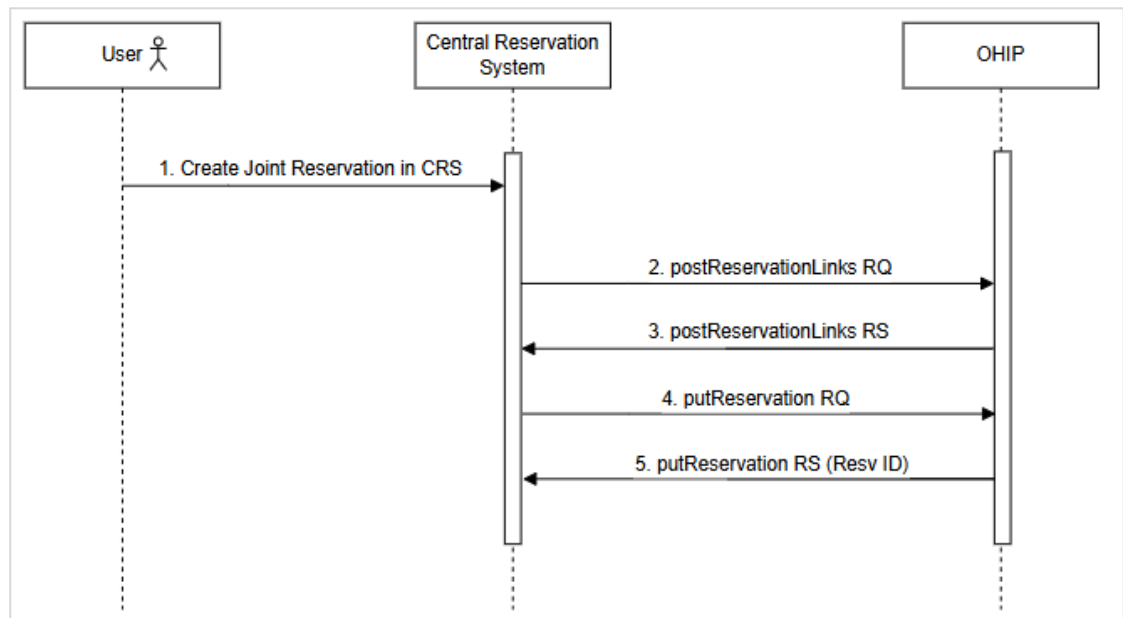
The diagram and detailed steps below illustrate the end-to-end create shared reservation flow from Central Reservation System to OPERA Cloud.



1. User creates a shared reservation in CRS.
2. CRS sends a postCombineShareReservations request to OHIP.
3. OHIP returns a 201 response to indicate successful creation of a shared reservation.
4. CRS sends a putReservation request to OHIP containing the reservation share updates.
5. OHIP returns a successful 201 response with the Reservation ID of the updated reservation.

Create Joint Reservation in CRS

The diagram and detailed steps below illustrate the end-to-end create joint reservation flow from Central Reservation System to OPERA Cloud.



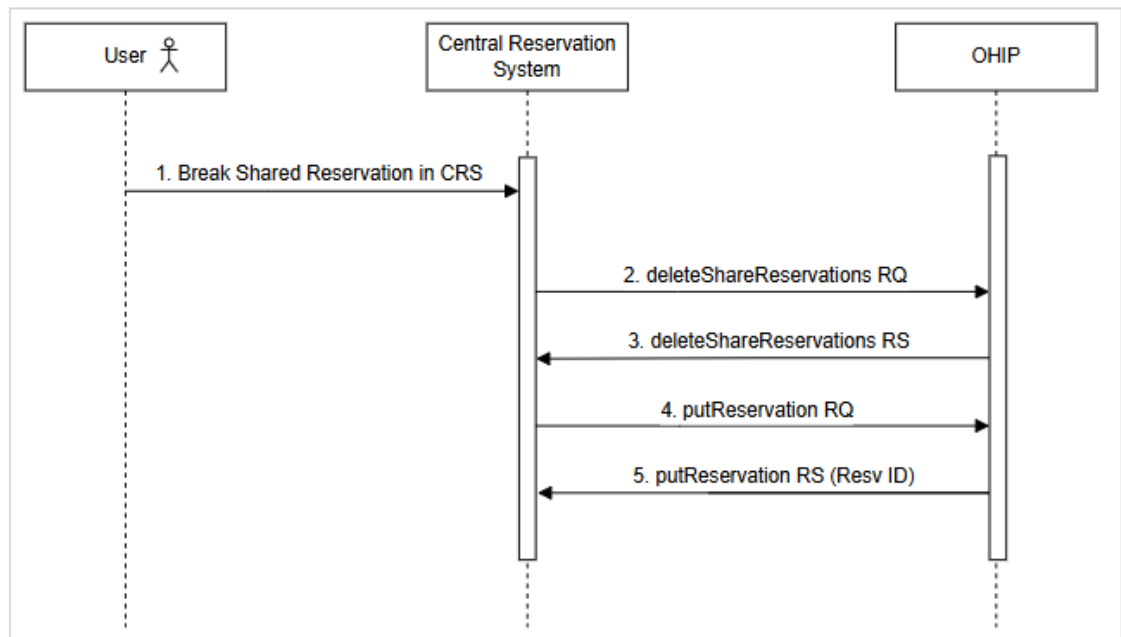
1. User creates a joint reservation in CRS.
2. CRS sends a postReservationLinks request to OHIP.
3. OHIP returns a 201 response to indicate successful creation of a joint reservation.
4. CRS sends a putReservation request to OHIP containing the joint reservation updates.
5. OHIP returns a successful 201 response with the Reservation ID of the updated reservation.

Break Linked Reservations in CRS

When linked reservations are unlinked (broken) in the CRS, the update is reflected in OPERA Cloud to remove the association between the bookings. This ensures that the property's systems reflect accurate room assignments and guest relationships, allowing for independent check-in, billing, and stay management. Similar to the creation of a linked reservation, you can unlink both shared and joint reservations in CRS.

Break Shared Reservations in CRS

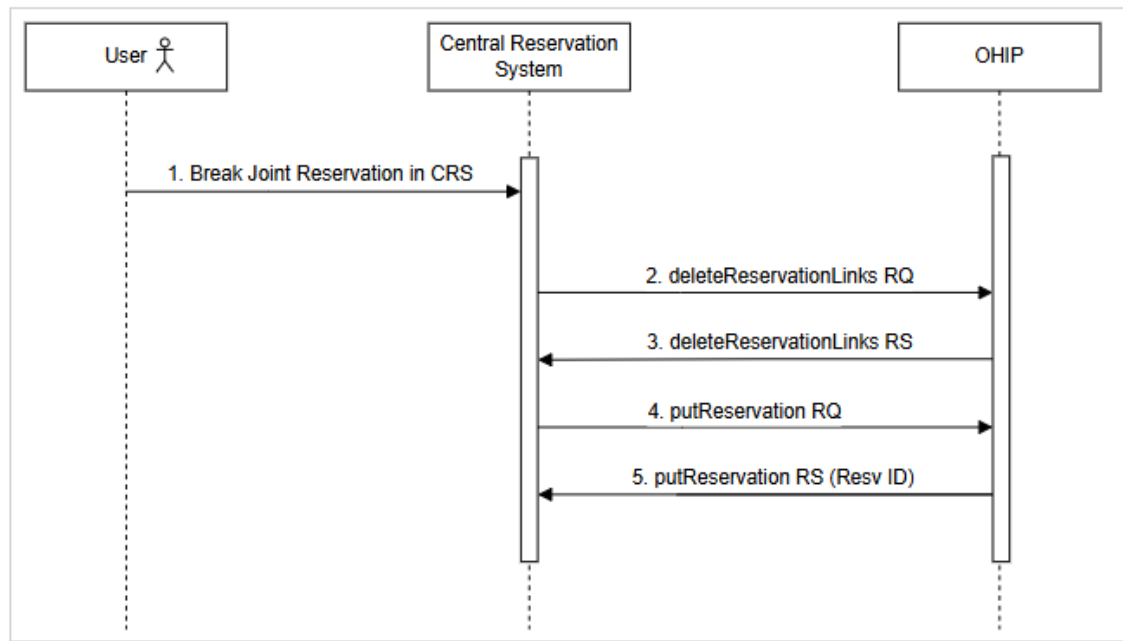
The diagram and detailed steps below illustrate the end-to-end unlink shared reservation flow from Central Reservation System to OPERA Cloud.



1. User breaks a shared reservation in CRS.
2. CRS sends a deleteShareReservations request to OHIP.
3. OHIP returns a 201 response to indicate successful unlinking of a shared reservation.
4. CRS sends a putReservation request to OHIP containing the shared reservation unlinking updates.
5. OHIP returns a successful 201 response with the Reservation ID of the updated reservation.

Break Joint Reservations in CRS

The diagram and detailed steps below illustrate the end-to-end unlink joint reservation flow from Central Reservation System to OPERA Cloud.



1. User breaks a joint reservation in CRS.
2. CRS sends a deleteReservationLinks request to OHIP.
3. OHIP returns a 201 response to indicate successful unlinking of a joint reservation.
4. CRS sends a putReservation request to OHIP containing the joint reservation unlinking updates.
5. OHIP returns a successful 201 response with the Reservation ID of the updated reservation.

5

Group Management

A group block is an allocation of rooms for multiple room types, reserved for a specific date period, that does not require you to create individually named reservations. Group Management CRS integrations typically involve sending and receiving group information from OPERA Cloud. This ensures that group data remains synchronized across systems, allowing real-time updates for room allocations, modifications, and cancelations.

OPERA Cloud to Central Reservation System

When group blocks are created, modified, or cancelled in OPERA Cloud, updates are sent to the CRS to ensure group details and room assignments remain consistent across all connected channels.

REST APIs

Following are the most commonly used REST APIs for retrieving group block data from OPERA Cloud to CRS:

API Name	Data Direction	Description
getChangesByDateTime	OPERA Cloud to CRS	Retrieve list of updated blocks.
getBlock	OPERA Cloud to CRS	Retrieve the details of a specific block.
putBlock	CRS to OPERA Cloud	Update Reservation: CRS confirmation number will be sent back to OPERA through the group block modification call.

Noteworthy Business Logic

Business Data	Comments
---------------	----------

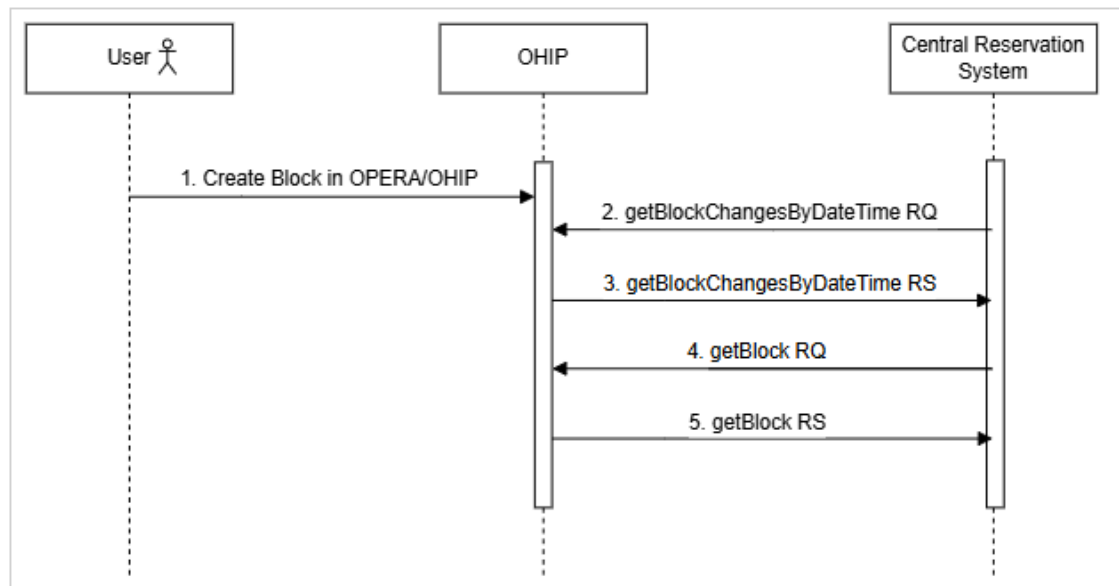
Group Status	CRS must align with statuses supported by OPERA Cloud: 1. Prospect - A preliminary or potential group booking entered to hold space, yet without a formal agreement or commitment. 2. Tentative - A group booking that has expressed intent to reserve but has not yet finalized the agreement; space is held pending confirmation. 3. Definite - A confirmed group booking with a signed agreement or strong commitment; inventory is firmly allocated to the group. 4. Cancelled - A block that was previously created but is now officially canceled; inventory held for the group is released back to general availability. <i>Please note that invalid statuses may cause rejection.</i>
Group Deletions	Group block deletions are handled as group cancelations.
Elasticity of Blocks	You can set an Elastic flag in the Central Reservation System to indicate flexibility in block booking dates. Consequently, when a booking is made outside of the core group dates, the auto-extend functionality will be automatically triggered from OPERA Cloud.
Per Product pricing	With Per Product pricing, the flat price per room type is communicated as a single-person rate.

Workflows

Create Group Blocks in OPERA Cloud/OHIP

When a group block is created directly in OPERA Cloud or OHIP - either by front desk staff, through a direct channel, or through another integrated system - the block data can be published to the CRS. This ensures that group blocks remain synchronized between the CRS and OPERA Cloud.

The diagram and detailed steps below illustrate the end-to-end create group block flow from OPERA Cloud to the Central Reservation System.

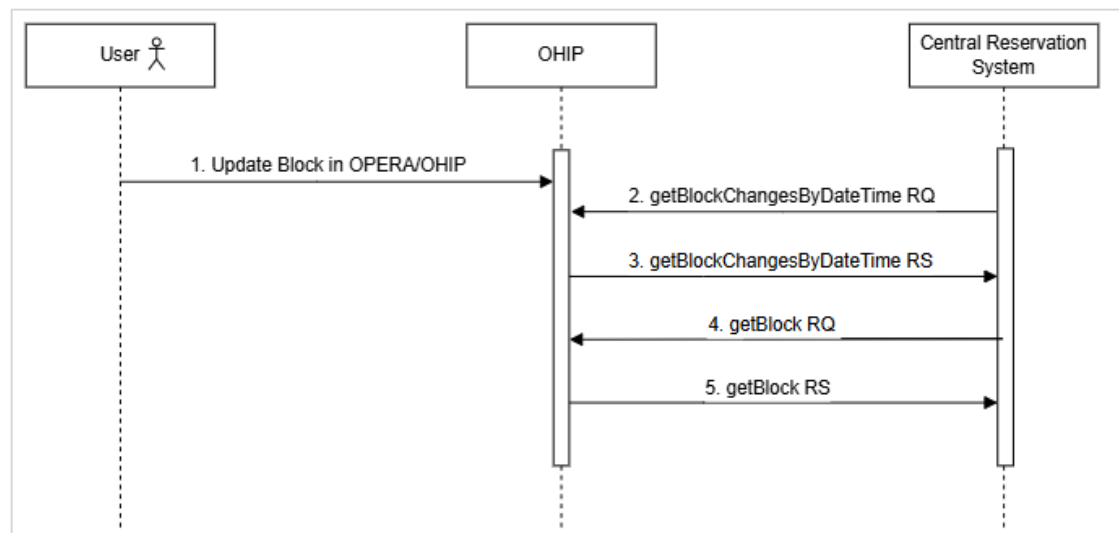


1. User creates a block directly within the OPERA Cloud User Interface or through OHIP.
2. CRS sends a `getBlockChangesByDateTime` request to OHIP to retrieve updates.
3. OHIP responds with a list of changed blocks based on the requested time window.
4. CRS sends a `getBlock` request to OHIP to retrieve the full block details.
5. OHIP responds with the complete block data.

Update Group Blocks in OPERA Cloud/OHIP

When a group block is modified directly in OPERA Cloud or OHIP, the block data can be published to the CRS. This ensures that group blocks remain synchronized between the CRS and OPERA Cloud.

The diagram and detailed steps below illustrate the end-to-end modify group block flow from OPERA Cloud to the Central Reservation System.



1. User updates a block directly within the OPERA Cloud User Interface or through OHIP.
2. CRS sends a getBlockChangesByDateTime request to OHIP to retrieve updates.
3. OHIP responds with a list of changed blocks based on the requested time window.
4. CRS sends a getBlock request to OHIP to retrieve the full block details.
5. OHIP responds with the complete block data.

Central Reservation System to OPERA Cloud

The CRS sends updates to OPERA Cloud whenever group bookings are created, modified, or canceled, ensuring that group information remains accurate and consistent across all connected systems.

REST APIs

Following are the most commonly used REST APIs for retrieving group block data from CRS to OPERA Cloud:

API Name	Data Direction	Description
postBlock	CRS to OPERA Cloud	Create a new group block.
putBlock	CRS to OPERA Cloud	Update an existing group block.
postBlockRestrictions	CRS to OPERA Cloud	Apply restrictions to a group block.
getBlockStatusCodes	CRS to OPERA Cloud	Retrieve list of status codes for group block.
putBlockStatusCode	CRS to OPERA Cloud	Update status of a group block.
putBlockAllocationRange	CRS to OPERA Cloud	Update allocation records of an existing group block across multiple dates and room types.
putBlockAllocation	CRS to OPERA Cloud	Update room type allocations and rates for a specific block for a shorter date range.
startBlockAllocationProcess getBlockAllocationProcessStatus getBlockAllocationProcessInfo	CRS to OPERA Cloud	Asynchronous workflow to add block room type allocations and rates for a specified block, typically across a large date range.

Noteworthy Business Logic

Business Data	Comments
---------------	----------

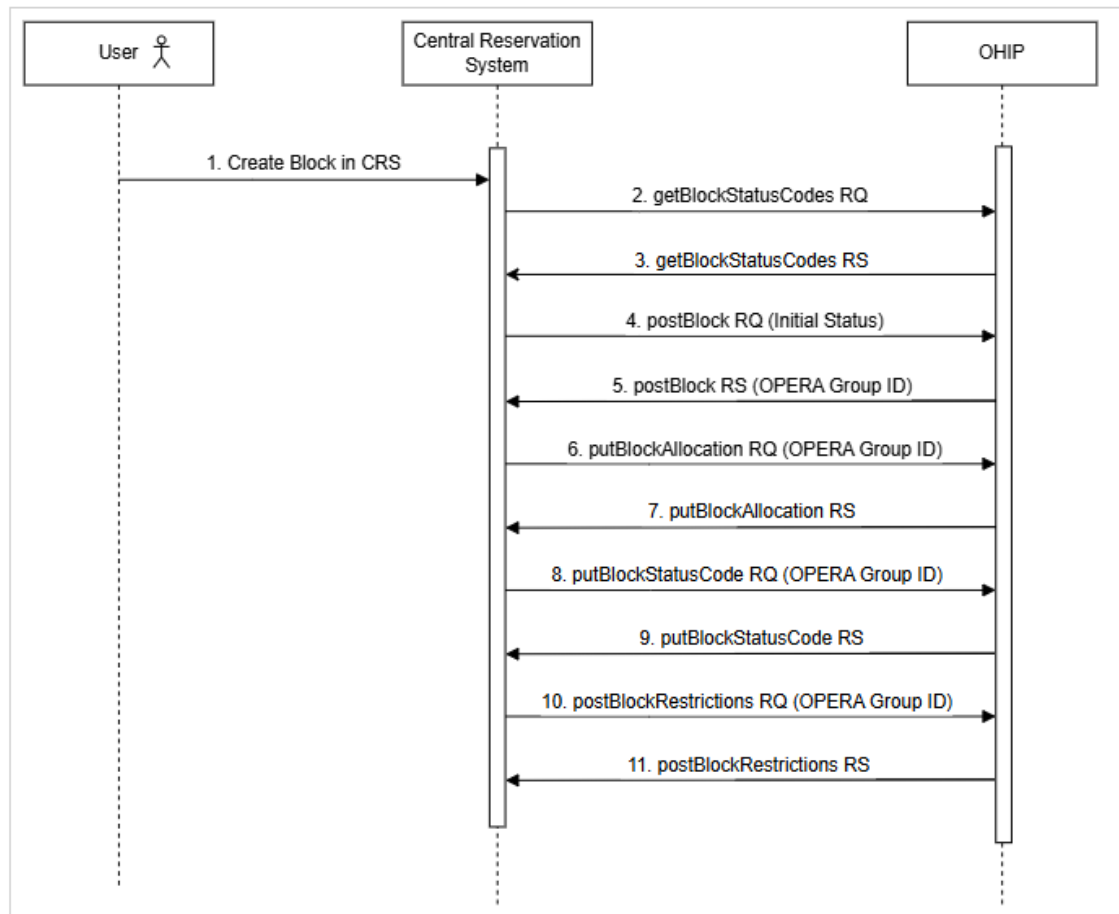
Group Status	CRS must align with statuses supported by OPERA Cloud: 1. Prospect - A preliminary or potential group booking entered to hold space, yet without a formal agreement or commitment. 2. Tentative - A group booking that has expressed intent to reserve but has not yet finalized the agreement; space is held pending confirmation. 3. Definite - A confirmed group booking with a signed agreement or strong commitment; inventory is firmly allocated to the group. 4. Cancelled - A block that was previously created but is now officially canceled; inventory held for the group is released back to general availability. Please note that invalid statuses may cause rejection.
Group Deletions	Group block deletions are handled as group cancellations.
Elasticity of Blocks	You can set an Elastic flag in the Central Reservation System to indicate flexibility in block booking dates. Consequently, when a booking is made outside of the core group dates, the auto-extend functionality will be automatically triggered from OPERA Cloud.
Per Product pricing	With Per Product pricing, the flat price per room type is communicated as a single-person rate.
Contracted Rooms	Contracted Rooms are supported in OPERA Cloud. This represents the total number of rooms that have been agreed upon and guaranteed in the group contract. This is the committed room block, regardless of whether individual reservations have been picked up yet.
Deducted Rooms	Contracted Rooms are supported in OPERA Cloud. This represents the number of rooms that are deducted from general inventory when the group block is created. These rooms are held exclusively for the group and are not available for general sale unless released.
Occupancy pricing	The OPERA Control "OCCUPANCY SPLIT PER ROOM TYPE" must be turned off for counters to be communicated from CRS to OPERA Cloud.

Workflows

Create Group Blocks (short duration) in CRS

Creating a short-term group block (typically less than a week) in the CRS updates OPERA Cloud through OHIP, ensuring the property can manage group inventory, room allocations, and operational planning in real time.

The diagram and detailed steps below illustrate the end-to-end create group block flow from the Central Reservation System to OHIP.

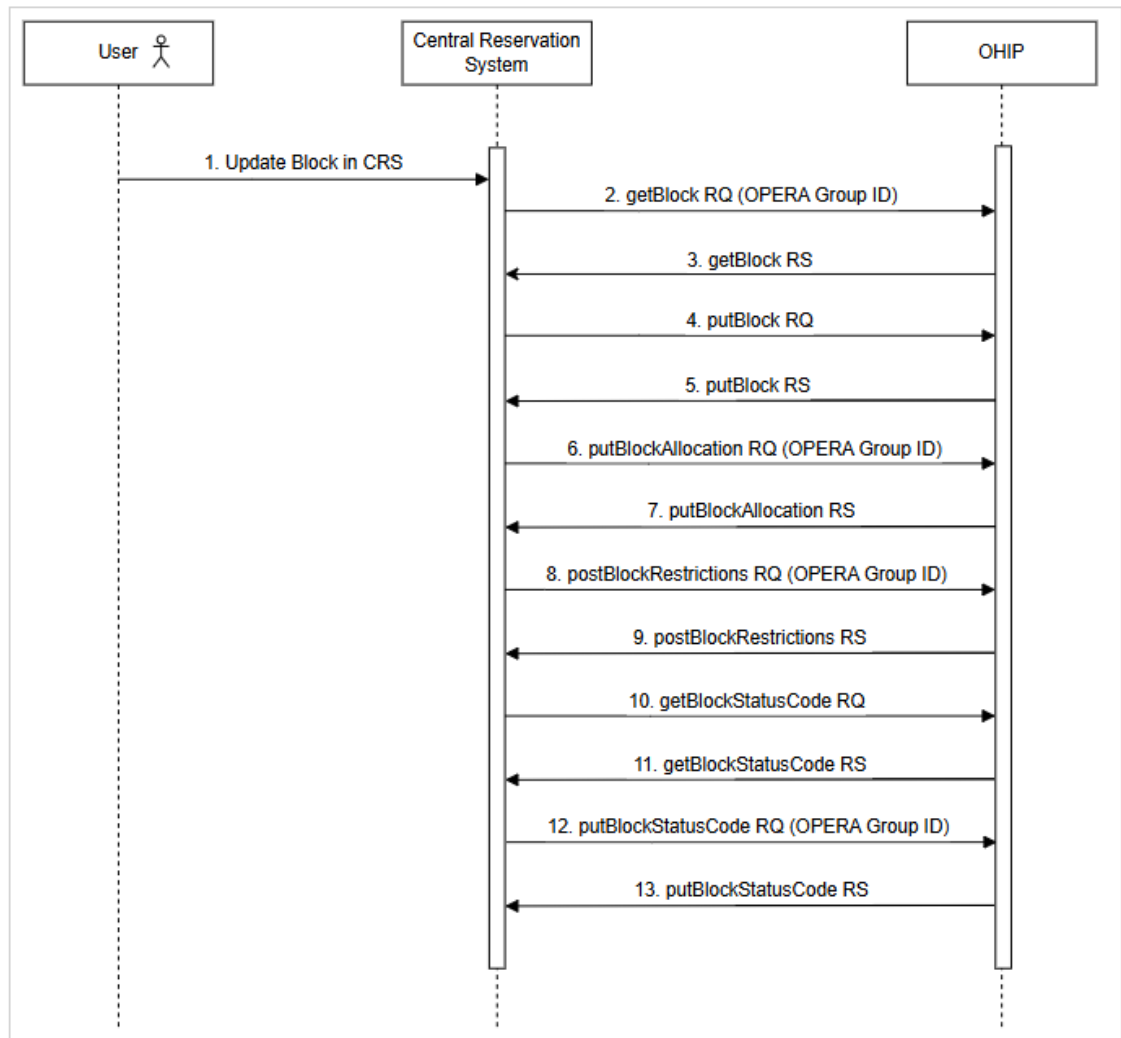


1. User creates a short-term (less than a week) block in the CRS.
2. CRS sends a `getBlockStatusCodes` request to OHIP to retrieve the initial block status and available status workflow.
3. OHIP responds with a list of available block status codes.
4. CRS sends a `postBlock` request to OHIP to create the block with the initial status configured in OPERA Cloud.
5. OHIP returns a successful 201 response containing the OPERA Group ID. This ID will be used for all future group block communications with OPERA Cloud.
6. CRS defines room type allocations by sending a `putBlockAllocation` request for each inventory type and date range, using the values set at the group level.
7. OHIP returns a success message upon assigning the room type allocations.
8. CRS sends a `putBlockStatusCode` request to OHIP to update the block's status until it aligns with the current status in CRS.
9. OHIP returns a successful 201 response to indicate successful updates of the block status.
10. CRS sends a `postBlockRestrictions` request to OHIP to apply any necessary restrictions (e.g., minimum stay).
11. OHIP returns a 201 success response indicating that the restrictions have been applied.

Update Group Blocks (short duration) in CRS

Modifying a short-term group block (typically less than a week) in the CRS triggers an update to OPERA Cloud through OHIP, providing real-time visibility into group changes.

The diagram and detailed steps below illustrate the end-to-end update group block flow from the Central Reservation System to OHIP.



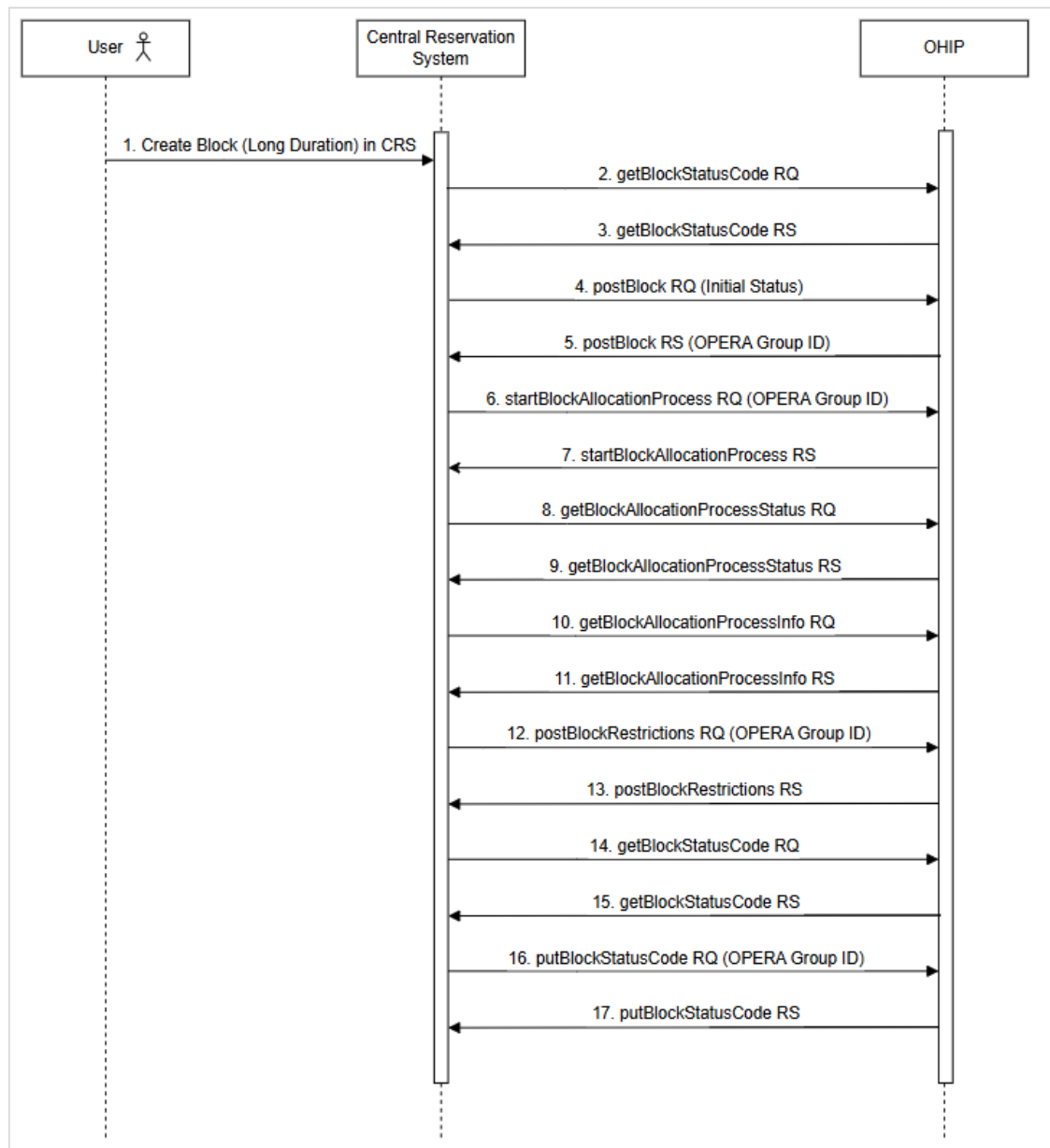
1. User updates a short-duration block (less than a week) in the CRS.
2. CRS sends a getBlock request to OHIP using the OPERA Group ID to retrieve the current block details.
3. OHIP responds with the full block information.
4. CRS sends a putBlock request to OHIP to update the block header details.
5. OHIP returns a successful 201 response confirming the block header update.
6. CRS sends a putBlockAllocation request to OHIP to update the room type allocations for the block.
7. OHIP responds with a successful 201 response confirming the room type allocations update.

8. CRS sends a postBlockRestrictions request to OHIP to apply or update any group block restrictions (such as minimum stay requirements).
9. OHIP responds with a successful 201 response confirming the application of the restrictions.
10. CRS sends a getBlockStatusCode request to OHIP to retrieve the current block status codes.
11. OHIP responds with a list of valid block status codes.
12. CRS sends a putBlockStatusCode request to OHIP to update the block status until it aligns with the current status in CRS.
13. OHIP returns a successful 201 response confirming the block status update.

Create Group Blocks (longer duration) in CRS

Creating a long-term (typically more than a week) group block in the CRS updates OPERA Cloud through OHIP, enabling the property to manage extended bookings and forecast inventory needs accurately.

The diagram and detailed steps below illustrate the end-to-end update group block flow from Central Reservation System to OHIP.



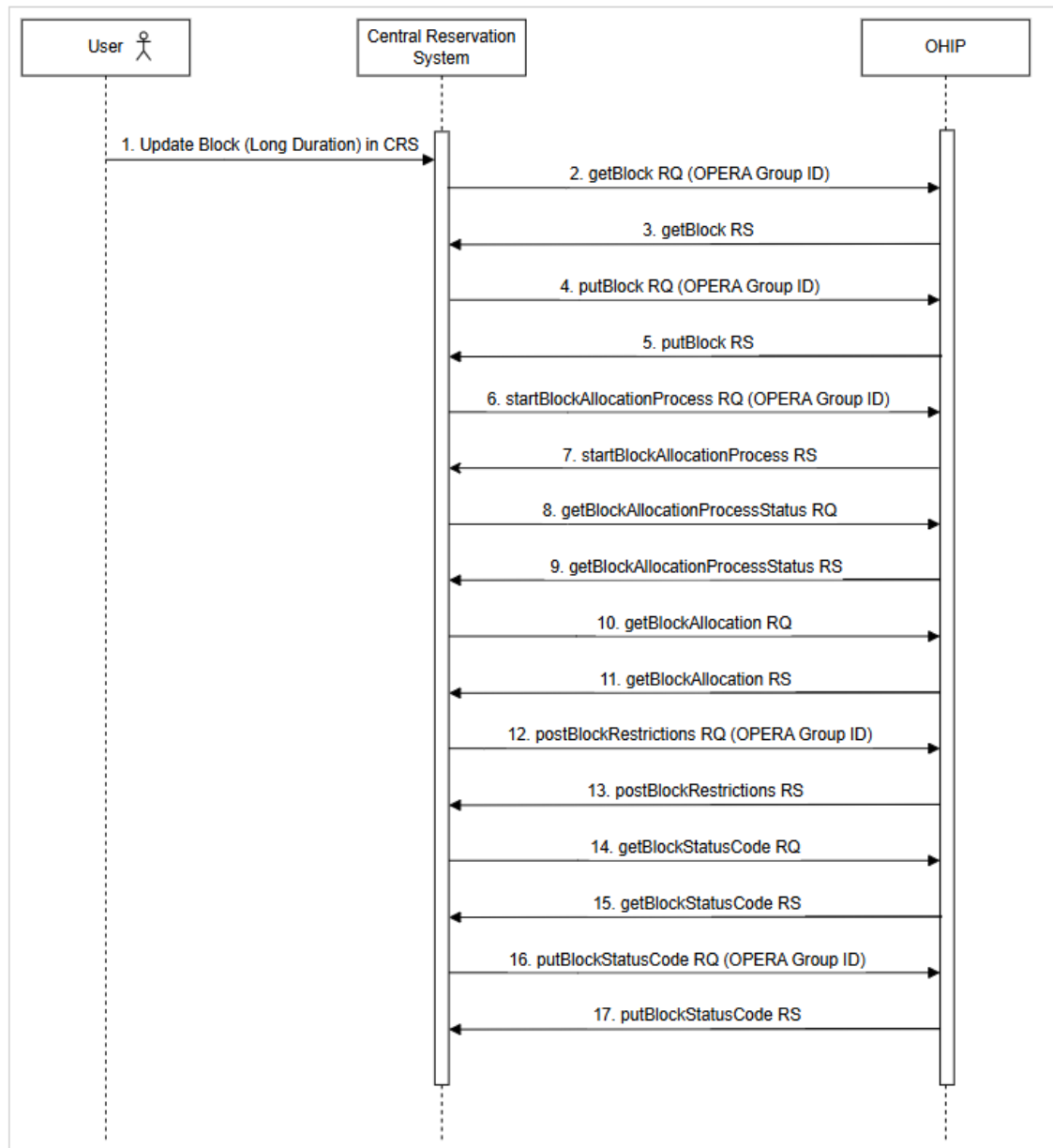
1. User creates a long-duration (more than a week) block in the CRS.
2. CRS sends a `getBlockStatusCode` request to OHIP to retrieve the initial block status and available status workflow.
3. OHIP responds with a list of available block status codes.
4. CRS sends a `postBlock` request to OHIP to create the block with the initial status configured in OPERA Cloud.
5. OHIP returns a successful 201 response with the OPERA Group ID. This ID will be used for all future group block communications with OPERA Cloud.
6. CRS sends a `startBlockAllocationProcess` request to OHIP using the OPERA Group ID to initiate the asynchronous update of block room type allocations and rates to a specified Block.
7. OHIP responds with a process ID indicating that the allocation process has started.

8. CRS sends a `getBlockAllocationProcessStatus` request to OHIP to check the progress of the allocation process.
9. OHIP responds with the current status of the block allocation process.
10. CRS sends a `getBlockAllocationProcessInfo` request to OHIP to retrieve the final status of the asynchronous update initiated in step 6 above.
11. OHIP responds with the final status of the allocation process, confirming whether the block allocations were successfully completed or if further action is required.
12. CRS sends a `postBlockRestrictions` request to OHIP using the OPERA Group ID to apply any applicable group restrictions (such as minimum stay).
13. OHIP returns a 201 success response confirming that restrictions have been applied.
14. CRS sends a `getBlockStatusCode` request to OHIP to recheck the current available block status codes.
15. OHIP responds with the updated list of status codes.
16. CRS sends a `putBlockStatusCode` request to OHIP to update the block's status until it aligns with the current status in CRS.
17. OHIP returns a successful 201 response to indicate successful updates of the block status.

Update Group Blocks (longer duration) in CRS

Updating a long-term (typically more than a week) group block in the CRS updates OPERA Cloud through OHIP, enabling the property to manage extended bookings and forecast inventory needs accurately.

The diagram and detailed steps below illustrate the end-to-end update group block flow from the Central Reservation System to OHIP.



1. User updates a long-duration (more than a week) block in the CRS.
2. CRS sends a getBlock request to OHIP using the OPERA Group ID to retrieve the current block details.
3. OHIP responds with the full block information.
4. CRS sends a putBlock request to OHIP to update the block header details.
5. OHIP returns a successful 201 response confirming the block header update.
6. CRS sends a startBlockAllocationProcess request to OHIP using the OPERA Group ID to initiate the asynchronous update of block room type allocations and rates to a specified Block.
7. OHIP responds with a process ID indicating that the allocation process has started.
8. CRS sends a getBlockAllocationProcessStatus request to OHIP to check the progress of the allocation process.
9. OHIP responds with the current status of the block allocation process.

10. CRS sends a `getBlockAllocationProcessInfo` request to OHIP to retrieve the final status of the asynchronous update initiated in step 6 above.
11. OHIP responds with the final status of the allocation process, confirming whether the block allocations were successfully completed or if further action is required.
12. CRS sends a `postBlockRestrictions` request to OHIP using the OPERA Group ID to apply any applicable group restrictions (such as minimum stay).
13. OHIP returns a 201 success response confirming that restrictions have been applied.
14. CRS sends a `getBlockStatusCode` request to OHIP to recheck the current available block status codes.
15. OHIP responds with the updated list of status codes.
16. CRS sends a `putBlockStatusCode` request to OHIP to update the block's status until it aligns with the current status in CRS.
17. OHIP returns a successful 201 response to indicate successful updates of the block status.

6

Availability Management

Availability enables you to manage your room inventory by providing a detailed view of all available and sold rooms at your hotel. Some of the tasks you can perform include defining conditions for stay restrictions, setting room sell limits, and searching for and viewing room availability.

Availability Management in CRS integration allows OPERA Cloud to reflect real-time room availability, restrictions, and sell limits across all connected channels. It enables the property to control how and when rooms can be sold by defining rules such as minimum/maximum stay, closed-to-arrival, and overbooking thresholds.

OPERA Cloud to Central Reservation System

Restrictions are rules that control rate and room availability based on specific conditions such as stay dates, length of stay, or arrival patterns.

Availability is primarily used to communicate stay restrictions between OPERA Cloud and the CRS. When restrictions such as minimum stay, closed-to-arrival, or closed-to-departure are configured in OPERA Cloud, they are published to the CRS to ensure consistent booking rules across all sales channels.

Business Events

Business Events

Module	Business Event	Data Elements	Description
RATE	RATE RESTRICTIONS	Select only the primary key and minimal required fields	Triggered when stay restrictions are updated for a rate plan in OPERA Cloud.

Noteworthy Business Logic

Business Data	Comments
Configuration in CRS	Rate Buckets, Room Buckets, and Inventory Types in CRS must match their corresponding OPERA Cloud configurations, or message processing will fail.
Advance Booking	Supported across all levels (house, room type, room bucket, rate plan, rate category).
Closed	Can be configured at any level; indicates room or rate is not available for sale.
Open	Used to explicitly override Closed restrictions at any level.

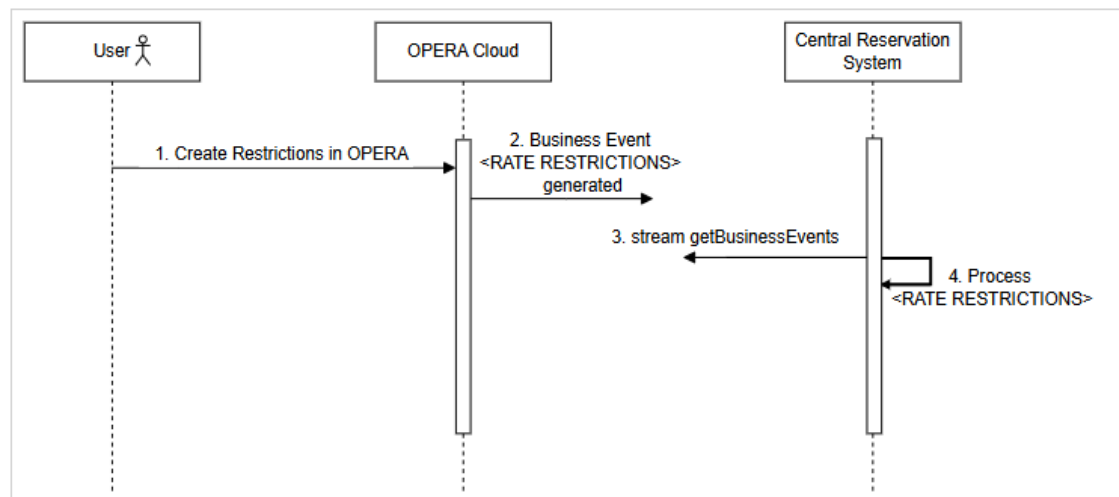
Closed to Arrival	This prevents arrivals on specific dates while allowing stay-throughs. Please note that this is supported at all levels.
Open to Arrival	Overrides Closed to Arrival settings to allow arrivals on specific dates.
Closed to Departure	Prevents departures on selected dates. Please note that this is supported at all levels.
Open to Departure	Allows departures when Closed To Departure is otherwise configured
Minimum / Maximum Length of Stay	Defines how long guests must stay (minimum) or can stay (maximum). Please note that this is supported at all levels.
Minimum / Maximum Stay Through	Controls length of total stay across a date range. Please note that this is supported at all levels.

Workflows

Create Restrictions in OPERA Cloud

When new stay restrictions are configured in OPERA Cloud, they are published to the CRS through Business Events. This ensures that selling conditions remain consistent across all distribution channels, preventing invalid bookings and supporting accurate availability control.

The diagram and detailed steps below illustrate the end-to-end set Restrictions flow from OPERA Cloud to CRS.



1. User sets restrictions in OPERA Cloud.
2. OPERA Cloud generates a RATE RESTRICTIONS Business Event.
3. CRS can stream the generated Business Events from OPERA Cloud using getBusinessEvents.
4. CRS identifies and processes the RATE RESTRICTIONS Business Event.

Central Reservation System to OPERA Cloud

When restrictions, oversell limits, or hurdle rates are configured in the CRS - either manually or through an integrated RMS - they are communicated to OPERA Cloud using designated APIs. This ensures that OPERA Cloud reflects the most current availability controls, supporting consistent booking rules and optimized revenue strategies across all channels.

REST APIs

Following are the most commonly used REST APIs for retrieving restrictions from CRS to OPERA Cloud:

API Name	Data Direction	Description
putSellLimitsByDateRange	CRS to OPERA Cloud	Create sell limits by date range
postRestriction	CRS to OPERA Cloud	Create stay restrictions on various levels
postHurdleRates	CRS to OPERA Cloud	Create hurdle rates
putHurdleRates	CRS to OPERA Cloud	Update existing hurdle rate

Noteworthy Business Logic

Business Data	Comments
Configuration in CRS	Rate Buckets, Room Buckets, and Inventory Types in CRS must match their corresponding OPERA Cloud configurations, or message processing will fail.
Advance Booking	Supported across all levels (house, room type, room bucket, rate plan, rate category).
Closed	Can be configured at any level; indicates room or rate is not available for sale.
Open	Used to explicitly override Closed restrictions at any level.
Closed to Arrival	This prevents arrivals on specific dates while allowing stay-throughs. Please note that this is supported at all levels.
Open to Arrival	Overrides Closed to Arrival settings to allow arrivals on specific dates.
Closed to Departure	Prevents departures on selected dates. Please note that this is supported at all levels.
Open to Departure	Allows departures when Closed To Departure is otherwise configured.
Minimum / Maximum Length of Stay	Defines how long guests must stay (minimum) or can stay (maximum). Please note that this is supported at all levels.
Minimum / Maximum Stay Through	Controls length of total stay across a date range. Please note that this is supported at all levels.

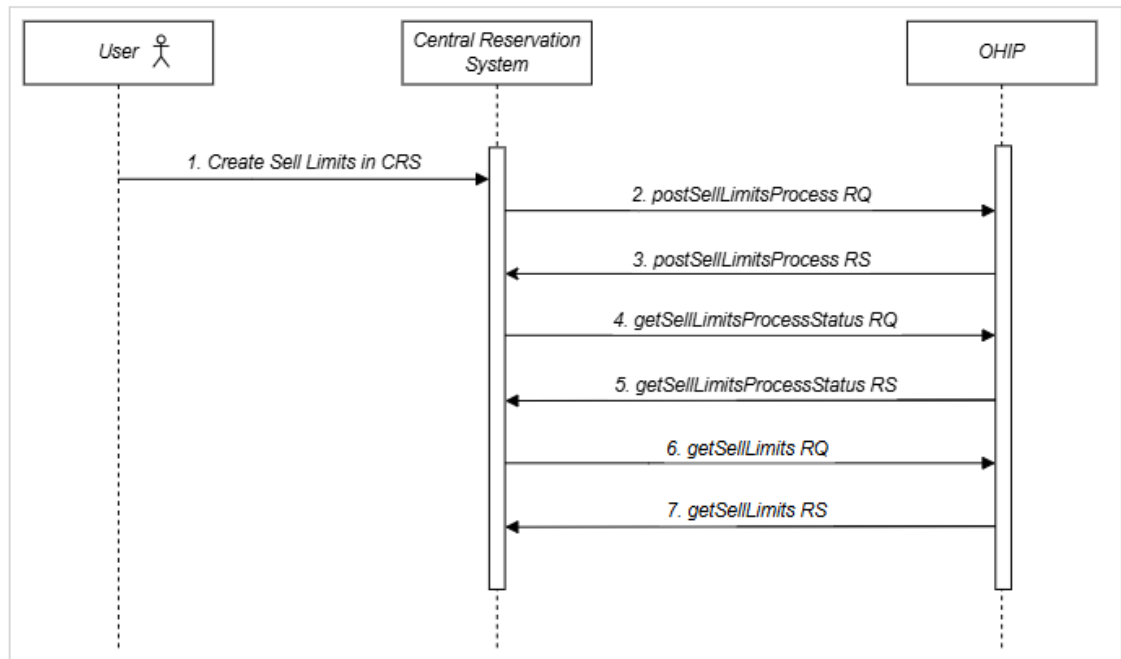
Date-based Hurdle Point for a Yield Class	Supported. Please note that the yield Class in CRS must match the Yield Category in OPERA Cloud for message processing.
Dynamic Date-based Hurdle Point for a Yield Class	Supported. Please note that the yield Class in CRS must match the Yield Category in OPERA Cloud for message processing.

Workflows

Create Restrictions in CRS

Restrictions are rules that control rate and room availability based on specific conditions such as stay dates, length of stay, or arrival patterns. When stay restrictions are created in the CRS, they are sent to OPERA Cloud through OHIP APIs.

The diagram and detailed steps below illustrate the end-to-end create restriction flow from Central Reservation System to OPERA Cloud.

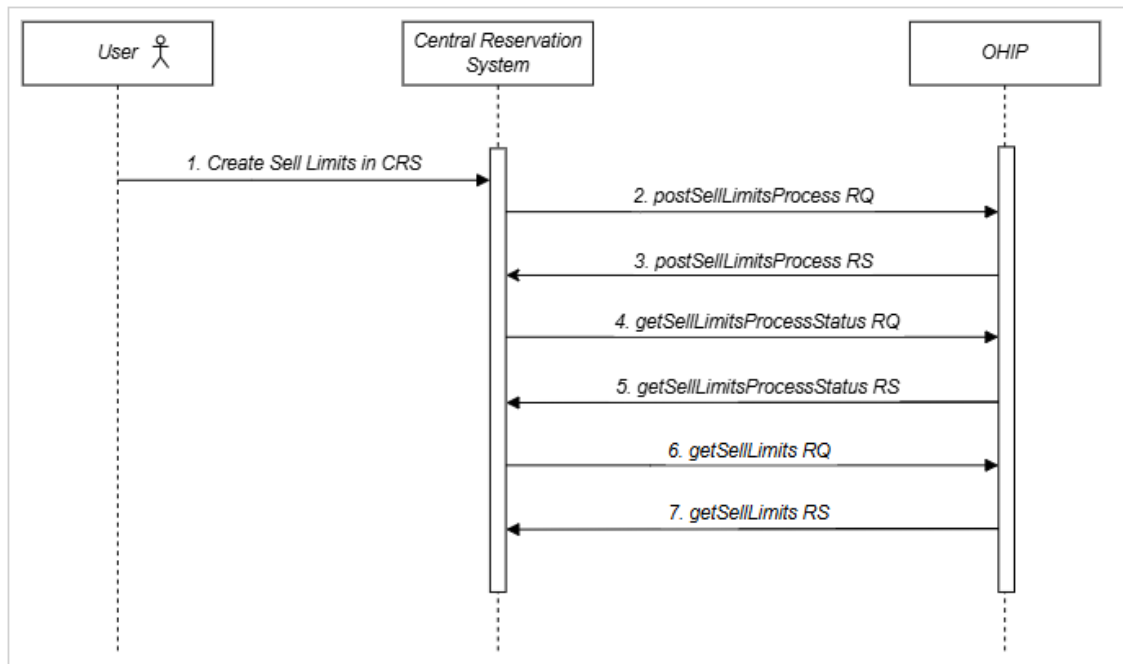


1. User configures stay restrictions in the CRS.
2. CRS sends a postRestriction request to OHIP with the applicable restriction details.
3. OHIP responds with a 201 status, confirming the stay restrictions were successfully applied.

Create Sell Limits in CRS

Sell Limits control how many rooms can be sold beyond or below actual physical inventory. Positive values allow overbooking to account for cancellations, while negative values restrict availability to hold back rooms. When sell limits are created in the CRS, they are sent to OPERA Cloud through OHIP to keep availability aligned across systems.

The diagram and detailed steps below illustrate the end-to-end create sell limits flow from Central Reservation System to OPERA Cloud.



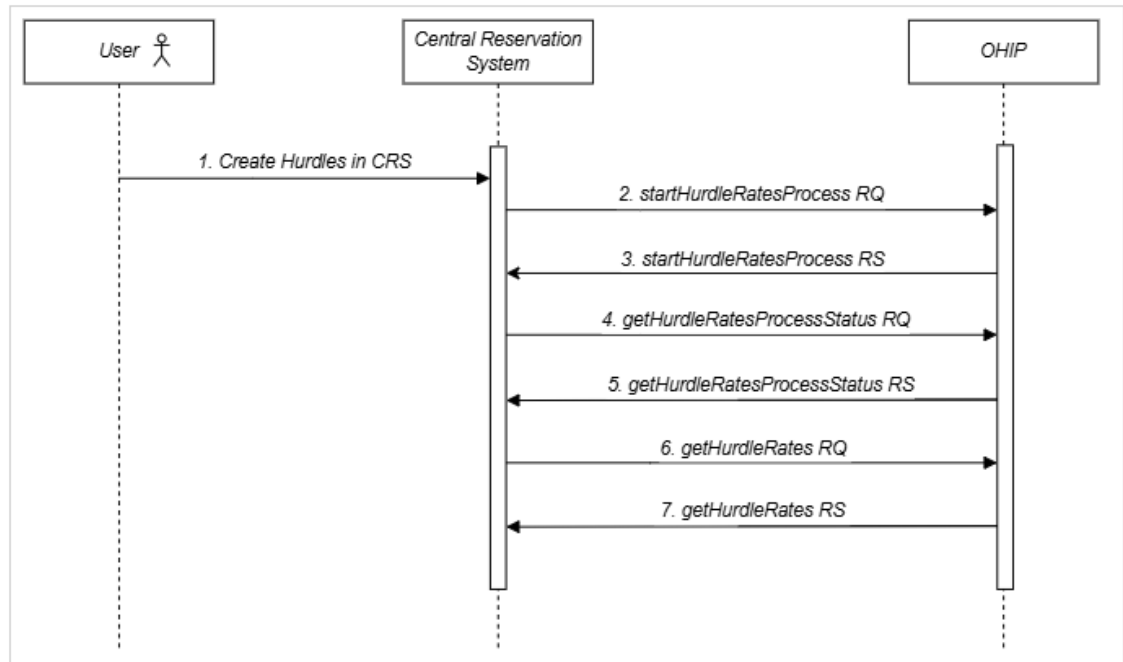
1. User creates Sell Limits in the CRS.
2. CRS sends a `postSellLimitsProcess` request to OHIP to initiate the asynchronous sell limits update process.
3. OHIP responds with a process ID confirming the initiation of the sell limits creation process.
4. CRS sends a `getSellLimitsProcessStatus` request to OHIP to check the progress of the sell limits update process.
5. OHIP responds with the current status of the sell limits processing request.
6. CRS sends a `getSellLimits` request to OHIP to retrieve the final status of the asynchronous sell limits creation initiated in step 2 above.
7. OHIP responds with the final status, confirming successful creation and processing of the sell limits.

Create Hurdles in CRS

Hurdle Rates represent the minimum acceptable price, calculated by the revenue management system based on data from OPERA Cloud. A rate code and room type combination will only be shown in OPERA Cloud if it meets or exceeds this hurdle value. When hurdle rates are created

or updated in the CRS, they are sent to OPERA Cloud via OHIP to enforce dynamic availability controls.

The diagram and detailed steps below illustrate the end-to-end create hurdles flow from Central Reservation System to OPERA Cloud.



1. User creates hurdle rates in the CRS.
2. CRS sends a startHurdleRatesProcess request to OHIP to initiate the asynchronous creation of hurdle rates.
3. OHIP responds with a process ID confirming that the hurdle rates creation process has started.
4. CRS sends a getHurdleRatesProcessStatus request to OHIP to check the progress of the hurdle rates processing.
5. OHIP responds with the current status of the hurdle rates creation process.
6. CRS sends a getHurdleRates request to OHIP to retrieve the final status of the asynchronous hurdle creation initiated in step 2 above.
7. OHIP responds with the final status, confirming successful creation of the hurdle values.

7

Rate Management

Rate plans and pricing information are managed in the Central Reservation System and pushed to OPERA Cloud. CRS serves as the **system of record** for all rate plans and is responsible for receiving pricing updates from a Revenue Management System (RMS) or through manual entry in the CRS user interface.

In OPERA Cloud, rates consist of two main components:

- **Rate Header** – Contains high-level configuration details such as rate plan code, market code, and booking rules.
- **Rate Details** – Include date-specific pricing, occupancy-based rates, and additional charges.

Both components must be sent to OPERA Cloud to successfully create or modify a rate plan. Separate integration flows are required to manage the header and detail sections, and each must be completed to ensure the rate is fully configured in OPERA Cloud.

Rate Plan configuration of Rate Details includes the following pricing strategies:

- **Flat Pricing**: Setting fixed rates for specific dates or seasons.
- **Derived Pricing**: Creating rate plans that derive their pricing from a base rate, allowing for dynamic adjustments.

Please refer to our user [guide](#) on Rate Code Configuration for additional details.

Central Reservation System to OPERA Cloud

When rate plans are created or updated in the CRS - either manually or through RMS - the rate configuration is sent to OPERA Cloud. This ensures that pricing and rate availability remain synchronized between CRS and OPERA Cloud, allowing the property to offer consistent rates across all booking channels.

REST APIs

Following are the most commonly used REST APIs for updating rate data from CRS to OPERA Cloud:

API Name	Data Direction	Description
postRatePlan	CRS to OPERA Cloud	Create new rate headers
putRatePlan	CRS to OPERA Cloud	Update existing rate header
setDailyRatePlanScheduleRange	CRS to OPERA Cloud	Set daily rate plan schedule range

postRatePlanSchedules	CRS to OPERA Cloud	Create Rate Plan pricing schedule
getRatePlanSchedules	CRS to OPERA Cloud	Retrieve Rate Plan schedules
deleteRatePlanSchedules	CRS to OPERA Cloud	Delete Rate Plan schedules
startSetDailyRatePlanSchedulesProcess getSetDailyRatePlanSchedulesProcessStatus getDailyRatePlanSchedules	CRS to OPERA Cloud	Create and update Daily Rate Plan Schedules at scale

Noteworthy Business Logic

Business Data	Comments
Rate Categories	A Rate Category is mandatory in OPERA, and the corresponding rate buckets must be set up in CRS to avoid message failures.
Room Types	The room type codes in CRS must correspond to those configured in OPERA.
Minimum/Maximum Length of Stay	Date-based minimum and maximum length of stay settings are not supported at the rate plan level. Please use restrictions to configure them instead.
Minimum/Maximum Advanced Booking	Date based minimum and maximum advanced booking is not supported on the rate plan. Please use restrictions to configure them instead.
Non-Room products included in rate	Non-room products in OPERA Cloud refer to items or services sold alongside a room reservation but not part of the physical room itself. These must already be configured as a Package code in OPERA prior to Rate Creation/Update
Yield Options	Yield options can be based on the same rate or a different rate.
Commission Amount	Commission can be defined as either a flat amount or a percentage.
Rate Details- Per Product pricing	CRS will define the price for the first adult.
Rate Details - Extra Person Charge	CRS supports a single extra person charge applicable across all guest types. This value will be applied as both the adult and child extra person charge in OPERA.

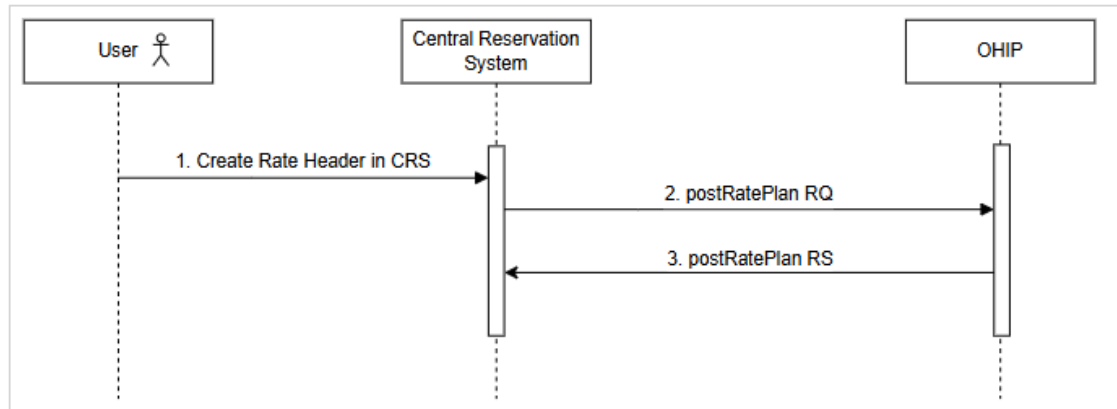
Workflows

Create Rate Header in CRS

When a new rate plan header is created in the CRS - either through the user interface or by an integrated RMS - the rate data is sent to OPERA Cloud through OHIP using designated APIs.

This ensures that OPERA Cloud reflects the latest rate offerings, allowing accurate quoting, availability checks, and booking across all distribution channels.

The diagram and detailed steps below illustrate the end-to-end create rate header flow from CRS to OPERA Cloud.

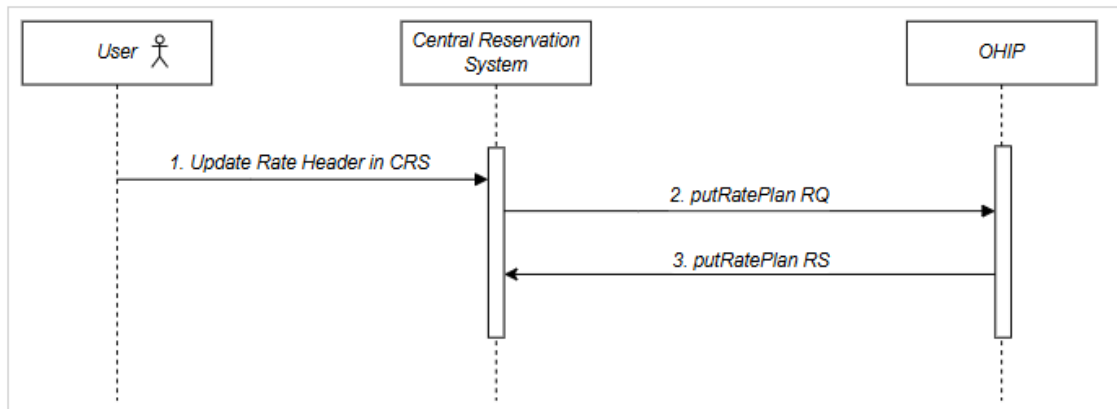


1. User creates a rate header in the CRS.
2. CRS sends a postRatePlan request to OHIP to create Rate Plans for a given property.
3. OHIP returns a successful 201 response confirming the rate header creation.

Update Rate Header in CRS

Similar to the creation of rate headers, you can also update a rate plan header in CRS, the rate data is sent to OPERA Cloud through OHIP using designated APIs.

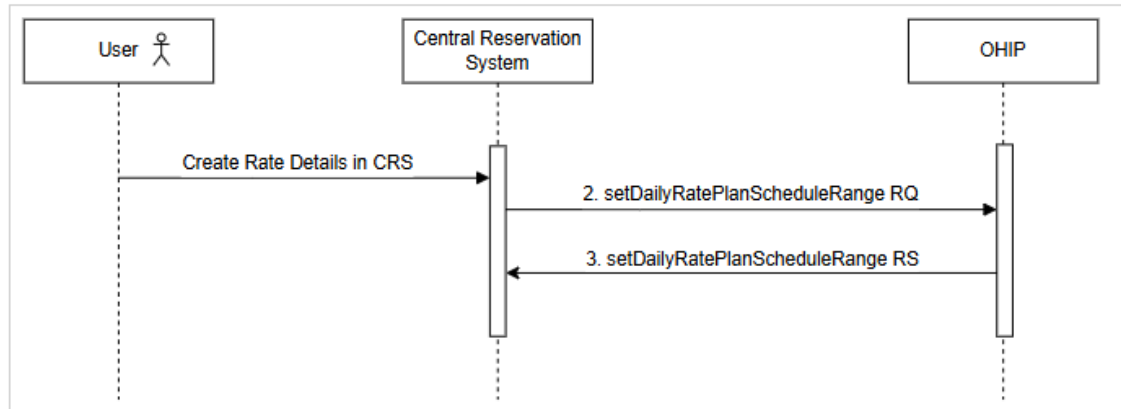
The diagram and detailed steps below illustrate the end-to-end update rate header flow from CRS to OPERA Cloud.



1. User updates a rate header in the CRS.
2. CRS sends a putRatePlan request to OHIP to update Rate Plans for a given property.
3. OHIP returns a successful 201 response confirming the rate header update.

Create Rate Details (Flat Pricing) in CRS (short duration)

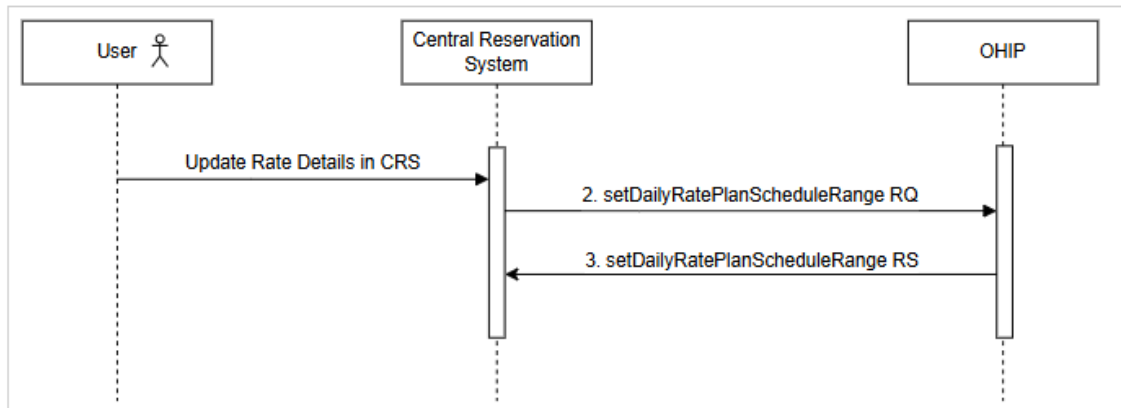
The diagram and detailed steps below illustrate the end-to-end create rate details for short duration flow from CRS to OPERA Cloud.



1. User creates rate details (flat pricing) in the CRS for a short duration (2 weeks or less).
2. CRS sends a `setDailyRatePlanScheduleRange` request to OHIP to create rate details for a given property.
3. OHIP returns a successful 201 response confirming the rate details creation.

Update Rate Details (Flat Pricing) in CRS (short duration)

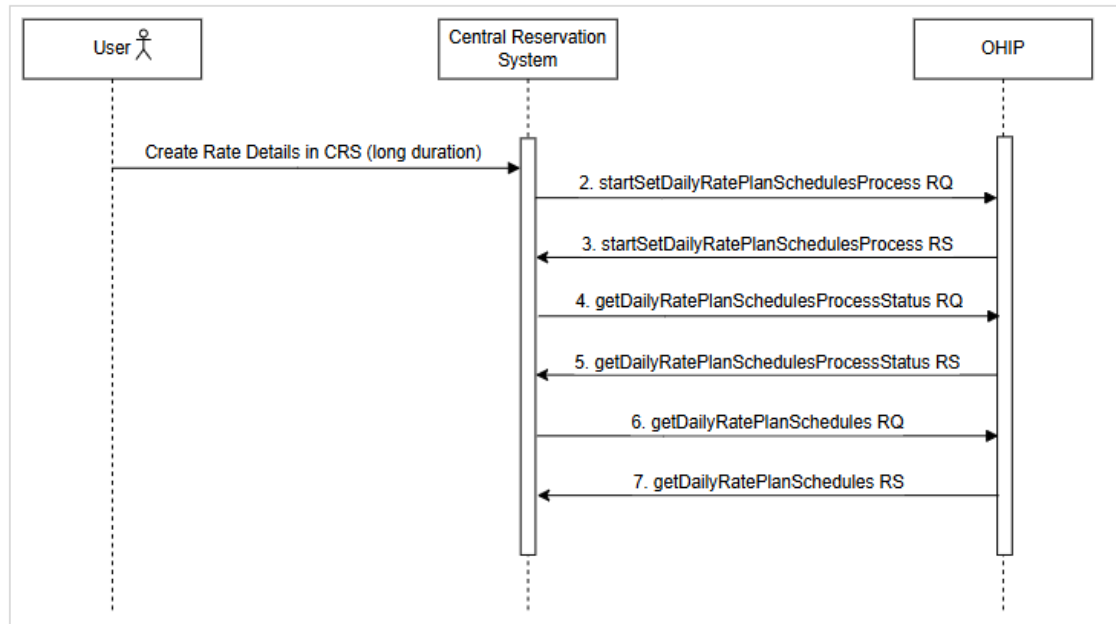
The diagram and detailed steps below illustrate the end-to-end update rate details for short duration flow from CRS to OPERA Cloud.



1. User creates rate details in the CRS for a short duration (2 weeks or less).
2. CRS sends a `setDailyRatePlanScheduleRange` request to OHIP to update rate details for the property.
3. OHIP returns a successful 201 response confirming the rate details update.

Create Rate Details (Flat Pricing) in CRS (longer duration)

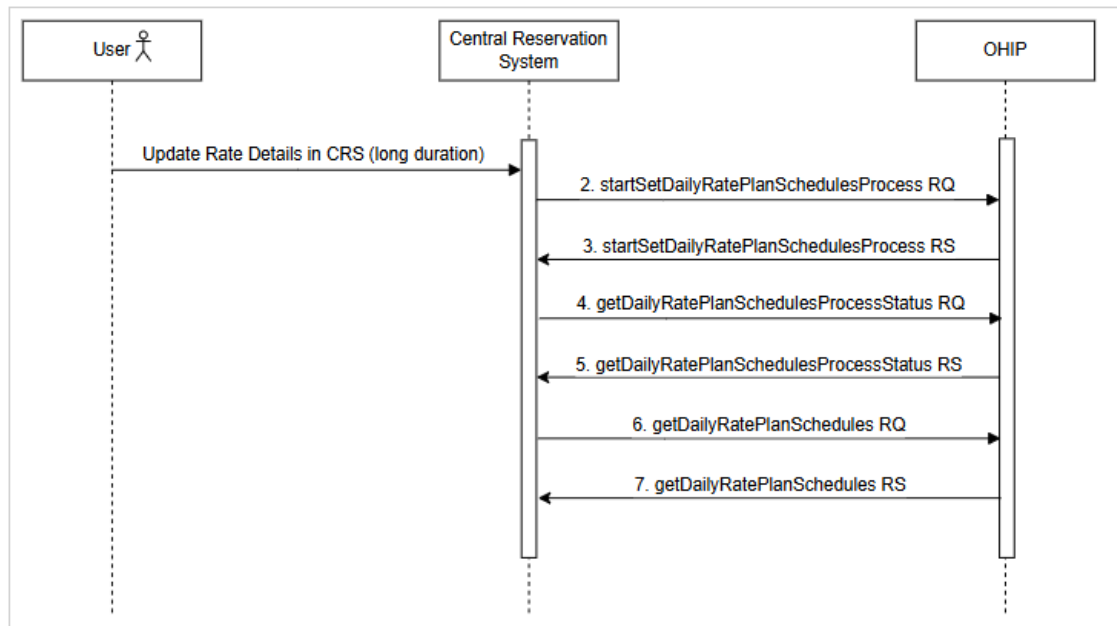
The diagram and detailed steps below illustrate the end-to-end create rate details for long duration flow from CRS to OPERA Cloud.



1. User creates rate details (flat pricing) in the CRS for a longer duration (more than 2 weeks).
2. CRS sends a startSetDailyRatePlanSchedulesProcess request to initiate the asynchronous creation of rate details for a given property.
3. OHIP responds with a process ID confirming the initiation of the rate details creation process.
4. CRS sends a getSetDailyRatePlanSchedulesProcessStatus request to OHIP to check the progress of the rate details creation process.
5. OHIP responds with the current status of the rate plan schedules creation process.
6. CRS sends a getDailyRatePlanSchedules request to OHIP to retrieve the final status of the asynchronous creation of rate details initiated in step 2 above.
7. OHIP responds with the final status, confirming successful creation of rate details for the given duration.

Update Rate Details (Flat Pricing) in CRS (longer duration)

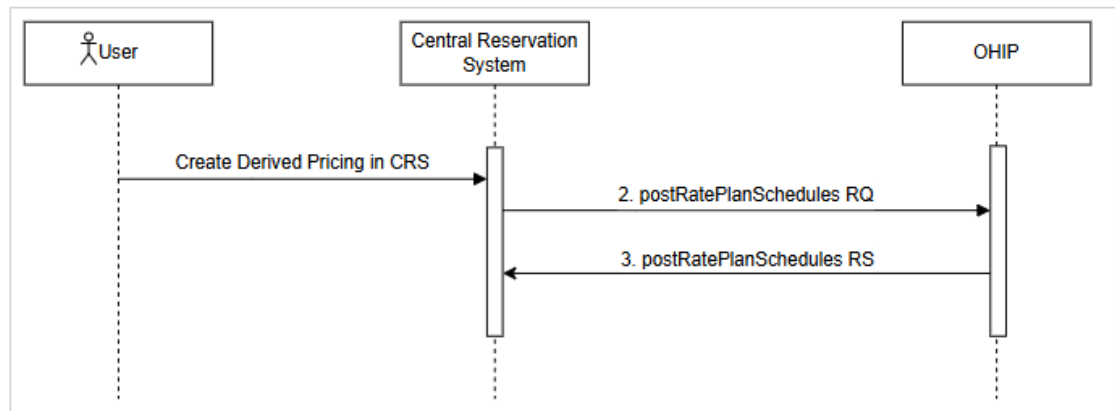
The diagram and detailed steps below illustrate the end-to-end update rate details for long duration flow from CRS to OPERA Cloud.



1. User updates rate details (flat pricing) in the CRS for a longer duration (more than 2 weeks).
2. CRS sends a startSetDailyRatePlanSchedulesProcess request to initiate the asynchronous update of rate details for a given property.
3. OHIP responds with a process ID confirming the initiation of the rate details update process.
4. CRS sends a getSetDailyRatePlanSchedulesProcessStatus request to OHIP to check the progress of the rate details update process.
5. OHIP responds with the current status of the rate plan schedules update process.
6. CRS sends a getDailyRatePlanSchedules request to OHIP to retrieve the final status of the asynchronous update of rate details initiated in step 2 above.
7. OHIP responds with the final status, confirming successful update of rate details for the given duration.

Create Rate Details (Derived Pricing) in CRS

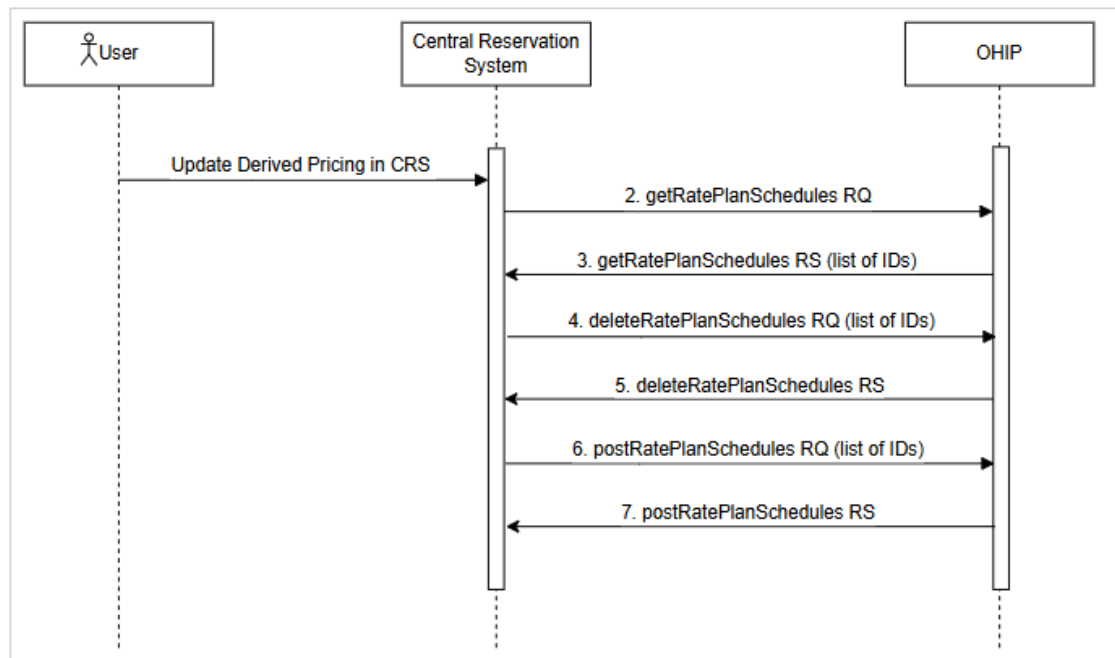
The diagram and detailed steps below illustrate the end-to-end create rate details (derived pricing) from CRS to OPERA Cloud.



1. User creates rate details (derived pricing) in the CRS.
2. CRS sends a postRatePlanSchedules request to OHIP to create rate details pricing schedules.
3. OHIP responds with a 201 confirming the successful creation of rate details pricing schedules.

Update Rate Details (Derived Pricing) in CRS

The diagram and detailed steps below illustrate the end-to-end update rate details (derived pricing) from CRS to OPERA Cloud.



Season IDs in OPERA Cloud are PMS-specific constructs used to group date ranges for rate management (e.g., high, low, shoulder seasons). CRS does not manage Season IDs because they are property-specific configurations in OPERA Cloud, and rate updates from CRS are typically applied using explicit date ranges rather than season-based groupings.

1. User updates rate details (derived pricing) in the CRS. When modifying derived pricing, the season IDs in OPERA Cloud are deleted and reinserted via OHIP to manage rate details.

2. CRS sends a `getRatePlanSchedules` request to OHIP to retrieve the list of rate plan schedules of derived pricing rules.
3. OHIP responds with a successful 200 response with a list of rate plan schedules that were updated.
4. CRS sends a `deleteRatePlanSchedules` request to OHIP to delete the above list of rate plan schedules in OPERA Cloud through OHIP.
5. OHIP responds with a 200 confirming the successful deletion of the rate plan schedules under consideration.
6. CRS sends a `postRatePlanSchedules` request to OHIP with the updated rate plan schedules to insert them in OPERA Cloud through OHIP.
7. OHIP responds with a successful 201 response confirming the insertion of the updated rate plan schedule records in OPERA Cloud through OHIP.

8

Inventory Management

Inventory counts in the Central Reservation System are influenced by current bookings, allocated group blocks, and Out of Order (OOO) designations made in OPERA Cloud. OPERA Cloud does not send full inventory snapshots to the CRS; instead, it communicates only OOO updates based on its configuration.

CRS can support inventory tracking at two levels depending on how the property is configured in the CRS:

- **Room Type Level** (Inventory Type)
- **Room Unit Level** (Inventory Item)

OPERA Cloud to Central Reservation System

Inventory updates in OPERA Cloud are communicated to the Central Reservation System through Business Events when Out of Order (OOO) status is applied to rooms or room types. These updates ensure the CRS accurately reflects current sellable inventory at the property.

Business Events

Following are the most commonly used Business Events for sending inventory updates from Central Reservation System to OPERA Cloud, though others may also apply.

1. Inventory Item (Room Unit) Configuration:

The following Business Events are applicable when the property is configured at **Inventory Item (room number)** level in CRS.

Module	Business Event	Data Elements	Description
HOUSEKEEPING	NEW OUT OF ORDER	Select only the primary key and minimal required fields	Triggered when a room is newly placed Out of Order in OPERA Cloud.
HOUSEKEEPING	UPDATE OUT OF ORDER	Select only the primary key and minimal required fields	Triggered when an existing Out of Order record is modified in OPERA Cloud.

2. Inventory Type (Room Type) Configuration:

The following Business Events are applicable when the property is configured at **Inventory Type (room type)** level in CRS.

Module	Business Event	Data Elements	Description
AVAILABILITY	SUMMARY TOTALS	DATE ROOM TYPE OUT OF ORDER	Triggered when inventory (including Out of Order rooms) is updated at the Room Type level in OPERA Cloud.

Noteworthy Business Logic

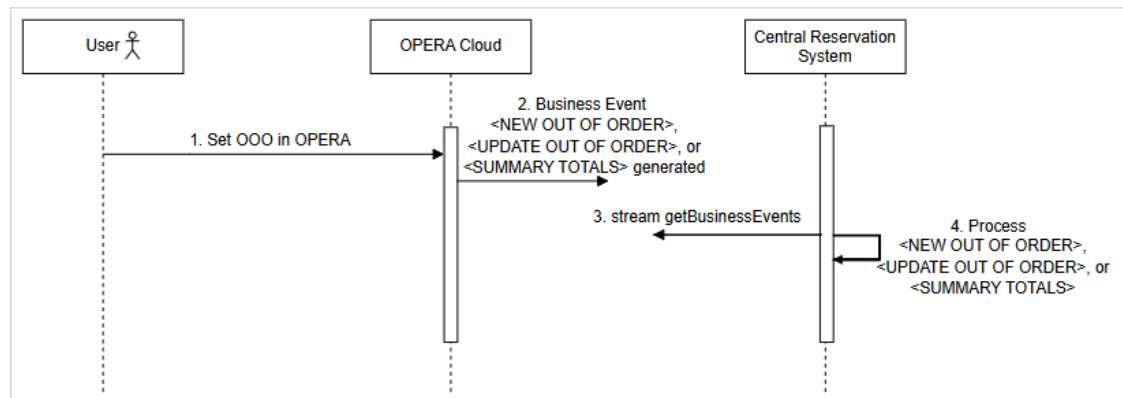
Business Data	Comments
Out of Order (OOO) updates on Inventory Types	Any Out of Order (OOO) updates on physical room types should only be used when Inventory Items are not in use. Please note that the Inventory Type setup in CRS must align with the Room Type configuration in OPERA Cloud.
Out of Order (OOO) updates on Inventory Items	If Inventory Items are configured in the CRS, OOO updates must be sent at that level. Please note that the Inventory Item configuration in CRS must align with the Room Number setup in OPERA Cloud.

Workflows

Update Inventory in OPERA Cloud

When inventory is updated in OPERA Cloud - such as marking a room or room type as Out of Order - Business Events are triggered and published to the CRS based on the property's integration configuration. This ensures that inventory availability remains aligned between OPERA Cloud and the CRS in real time.

The diagram and detailed steps below illustrate the end-to-end update inventory flow from OPERA Cloud to CRS.



1. User sets a room/room type as Out of Order (OOO) in OPERA Cloud.
2. OPERA Cloud generates a NEW OUT OF ORDER, UPDATE OUT OF ORDER or SUMMARY TOTALS Business Event.

3. CRS can stream the generated Business Events from OPERA Cloud using `getBusinessEvents`.
4. CRS identifies and processes the above business events.

9

Anti-patterns

Review the [Best Practices chapter](#) in the OHIP user guide to learn more about functional and non functional and technical anti-patterns.