NGI UC for Availability Orchestration

Use Case Specification

Author: Nguyen Van Binh

Document version: 1.3

Document Date: 2015-June-15

Approvals

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Approved By | Signed | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 2014-Oct-07 | 0.0a | Initial Draft | Nguyen Van Binh |
| 2014-Oct-22 | 0.0b | Updates from 1st peer review | Nguyen Van Binh |
| 2014-Nov-05 | 0.0c | Additional updates to align with the System BDD for Availability Request - Response | Nguyen Van Binh |
| 2014-Nov-14 | 0.0d | Updates from 2nd peer review | Nguyen Van Binh |
| 2014-Dec-01 | 0.0e | Added Exception Flow for requested date that is beyond Subscriber Parameters Max\_Sell\_Date.  Added Step for Availability business rules to apply before routing/conx building (e.g Restricted Flights) and business rules to apply after determining allocation availability (e.g Hide RBD).  Added Rules for routing/connection building to consider Traffic Restriction and to exclude Host-Only Flights when requester is non-host.  Other additional updates | Nguyen Van Binh |
| 2015-Mar-03 | 1.0 | Version change; dropped into SVN | Richard Winstel |
| 2015-Mar-24 | 1.1 | Multiple updates for cascading availability; determine allocation availability; change the application of business rules on "Hide RBD" and "Inhibit Sell" to optimize processing; and other updates. | Nguyen Van Binh |
| 2015-Apr-16 | 1.1a | Added AVAIL.24g story for Synchronise Segment notes (*earlier INV.24a originally written by Andrey Golovachev & further updated by Fereidoun Hamisi*) | Krisna Pawan |
| 2015-Apr-28 | 1.2 | Added AVAIL.24e (set up local view), AVAIL.24f (Apply rules) and AVAIL.24h (Apply segment notes) | Krisna Pawan / Sriram Ananthasivan |
| 2015-Jun-15 | 1.3 | Added AVAIL.24b&c story for applying POO restrictions & Upline\_Downline restrictions to availability & sell requests (included Tim's inputs) | Krisna Pawan |

Contents

1. Overview 4

**1.1** **Brief Description / Background** 4

**1.2** **Story Mapping** 4

2. Actors 4

**2.1** **Primary Actors** 4

**2.2** **Secondary Actors** 4

3. References 5

4. General Preconditions 7

5. General Post Conditions 7

6. Basic Flow of Events 8

**6.1** **Basic Flow 1 – Process Availability Request and Response Orchestration** 8

**6.2** **Basic Flow 2 – Simulate Availability** 16

7. Alternate Flows 17

**7.1** **Alternate Flow 1 – Determine First Availability** 17

**7.2** **Alternate Flow 2 –Bypass Cascading Availability** 18

**7.3** **Alternate Flow 3 – More Availability** 19

8. Exception Flows 21

**8.1** **Exception Flow 1 – Unable to Recognize Availability Request** 21

**8.2** **Exception Flow 2 – Message Transformation Error** 21

**8.3** **Exception Flow 3 – Availability Request Validation Error** 21

**8.4** **Exception Flow 4 – Non-Host Originator Requests for OA Availability** 22

**8.5** **Exception Flow 5 – Non-Host Originator not Eligible for Interactive Availability Request** 22

**8.6** **Exception Flow 6 – No Actionable Segment** 23

**8.7** **Exception Flow 7 – Request Availability Beyond Maximum Sell Date** 23

**8.8** **Exception Flow 8 – No More Routings** 24

9. Sub Flows 24

**9.1** **Sub Flow 1 – Determine Availability on SITA Hosted OA Segments** 24

**9.2** **Sub Flow 2 – Generate Routings for Specific Flight Request** 25

**9.3** **Sub Flow 3 – Determine Segment Note** 27

10. Special Requirements 28

10.1 Subscriber parameter 28

10.2 Message Transformation 28

10.3 Partner Agreements 29

10.3.1 Internal OA Agreement 29

10.3.2 External OA Agreement 29

10.3.3 External GDS Agreement 30

10.3.4 SITA Marketplace Air 30

10.3.5 Interactive and Direct Access Availability and Sell Session Timeout control 30

10.4 Document Updates 30

11. Assumptions, Issues & Future Use Case Considerations 31

12. Additional Information 31

13. Business Data Definitions 32

14. BDD Supporting Business Rules 32

15. Use Case Elaboration 33

15.1 AVAIL.24e: Setup client side cache of MICT records with Hide RBD and Inhibit Sale rules 33

15.1.1 Summary 33

15.1.2 Detail 33

15.1.3 Acceptance 34

15.2 AVAIL.24f – Apply RBD Hide & Inhibit Sale rule during Availability determination 35

15.2.1 Summary 35

15.2.2 Detail 35

15.2.3 Acceptance Tests 35

15.2.4 Non Functional Requirements 37

15.3 AVAIL.24g – Synchronize Segment Notes 38

15.3.1 Summary 38

15.3.2 Detail 38

15.3.3 Acceptance 42

15.4 AVAIL.24h – Apply segment notes to availability response 45

15.4.1 Overview 45

15.4.2 Detail 45

15.4.3 Acceptance Tests 45

15.4.4 Non Functional Requirements 47

15.5 AVAIL.24b – Apply POO (Point-of-Origin) restrictions to Availability & Sell 47

15.5.1 Overview 47

15.5.2 Detail 47

15.5.3 Acceptance Tests 48

15.6 AVAIL.24c – Apply Upline/Downline restrictions to Availability & Sell 51

15.6.1 Summary 51

15.6.2 Background 51

15.6.3 Detail 53

15.6.4 Acceptance Tests 53

NGI UC for Availability Orchestration

# Overview

## **Brief Description / Background**

When HIAS receives an availability request, based on the specific information provided in the request such as point of sale, request type, requested routing etc the system invokes the related use case(s) to process and build the availability response.

This use case orchestrates the processes pertaining to handling availability requests and responses in HIAS.

## **Story Mapping**

|  |  |  |  |
| --- | --- | --- | --- |
| Story | Story Tag | Flow | Description |
| Avail.24b | Apply POO (Point-of-Origin) restrictions to Availability & Sell | - | Apply POO in booking limit bucket on sell, and limit availability on POO on availability request. |
| Avail.24c | Apply Upline/Downline restrictions to Availability & Sell | - | The booking limit template defines upline/downline limits, based on market (not just within a flight) - also exposes the minimum booking calculation. |
| Avail.24ef | Applying availability Hide & Sell rules (includes caching) | Basic Flow 1 | Distribution of rules within WLS tier & Apply hide rules within the availability determination process. |
| Avail.24g | Synchronize flight segment notes in cache |  | Synchronize S&S managed segment notes; store in Grid |
| Avail.24h | Apply flight segment notes | Basic Flow 1; Sub Flow 3 | Add segments notes to availability responses. |
|  | **More stories will be added** |  |  |

# Actors

## **Primary Actors**

Basic Flow 1: Availability Enquirer

Basic Flow 2: HIAS Inventory Analyst

## **Secondary Actors**

Horizon Messaging

S&S

SITA Airfare Shop

SITA E-Commerce

GDS/OA

# References

| Ref. ID | Reference | Notes |
| --- | --- | --- |
| [1] | NGI UC for Determine Allocation Availability | An existing Use Case that describes the processes that determine allocation availability on subscriber's designated flights.  That UC needs to be updated in order to be aligned with the orchestration defined in this UC. |
| [2] | NGI UC for Cascade Availability | An existing document that describes the requirements for cascading availability requests to external systems (1G and OA) in order to get real-time availability for OA flights.  That UC need to be updated in order to be aligned with the orchestration defined in this UC |
| [3] | HF121 EDIFACT Interface Specification (V1.0) | This is the existing Specs documentation to support HIOD EDIFACT interface. The document covers specific flight availability (MSG 46) only. In order to support full availability orchestration, additional interface specification must be made for city pair (daily flight) availability (MSG 44). |
| [4] | HF121 NGI Availability Interface Specification V1.1 | Current version is mainly to support availability in HIOD. Additional updates are required to support HIAS availability. |
| [5] | NGI UC for Simulate Availability | Existing Use Case that defines the requirements for simulate availability.  Updates required to add more options to simulate various availability request data elements as defined in System BDD for Availability Request – Response.  Also needs to add options to include Draft Schedule and Draft Routing Rules in Simulate Availability |
| [6] | NGI UC for Determine Availability on SITA Hosted OA Segments | This is a NEW Use Case that defines the requirements for determination of availability on other SITA hosted airlines' segments to return to a Subscriber's request. |
| [7] | NGI UC for Manage and Process Product Display Order | The dropped UC that defines the requirements for manage Subscriber's business rules for product display order and system processing to enforce those rules. |
| [8] | System BDD for POS Value | The dropped BDD that defines the POS element. No update required. |
| [9] | System BDD for Availability Request - Response | The NEW BDD that defines data elements for availability requests and availability responses in Horizon.  Updates may be required to support outbound availability requests from HIAS to external systems (e.g. in SITA Marketplace Air) |
| [10] | MSG UC for Receive EDIFACT Messages | Messaging UC for receive PADIS EDIFACT messages |
| [11] | MSG UC for Send EDIFACT Messages | Messaging UC for send PADIS EDIFACT messages |
| [12] | [TBD] UC for Transform EDIFACT PAOREQ/PAORES Messages | This UC does not exist at time of writing.  The transformation of EDIFACT PAOREQ and PAORES messages is required in Horizon to support HIAS receives availability request and response back to GDS and OA via interactive link. The transformation is also needed for HIAS in cascading availability request to external GDS and OA systems.  Architects will determine the domain that transforms EDIFACT PAOREQ/PAORES messages. |
| [13] | [TBD] UC for Transform Direct Access Terminal Emulation Request – Response | This UC does not exist at time of writing. The transformation of Direct Access Terminal Emulation is required in Horizon to support HIAS receives availability request and response back to GDS and OA based on existing Direct Access Terminal Emulation links.  Architects will determine the domain that transforms Direct Access Terminal Emulation request/response. |
| [14] | [TBD] UC for Process Direct Access Terminal Emulation Request | This UC does not exist at time of writing. Horizon needs to support Direct Access Terminal Emulation requests from and respond to GDS's and OA as the mainframe does.  Architects will determine the domain that handles Direct Access Terminal Emulation processing. |
| [15] | NGI UC for Generate Subscriber Routings | Existing Use Case that defines the requirements and processes to build routings for a Subscriber based on routings business rules.  Updates required ensuring generation of segments and connections for OA and also interlining routings between the host airline and OA's.  Updates are also required to support the requirement to exclude subscriber's flight ranges in routings built for Alliance OA partners' request for Direct Access and Interactive Availability. Please refer to RES UCS 13-0063 for the base requirement. |

# General Preconditions

* The airline is a Leg/Segment Allocation and Availability (HIAS2) Subscriber or later.
* Actor is authorized to perform the functionality in this use case

# General Post Conditions

* None

# Basic Flow of Events

## **Basic Flow 1 – Process Availability Request and Response Orchestration**

|  |  |
| --- | --- |
| Specific Preconditions | * An availability request has been received |

| Activity | Alternate / Exception Flow Name |
| --- | --- |
| 1. System determines the Request Format of availability request, using Horizon Messaging validation (Services - TBD)   The below Request Formats are to be determined and routed to HIAS:   * Internal Availability Request (canonical form) * Direct Access Terminal Emulation request (TBD) * EDIFACT PAOREQ message * OTA Schema availability search message (New Distribution Capability - NDC) (TBD - from external systems, for Future development?) * Other proprietary availability request formats | **Exception Flow 1 – Unable to Recognize the Availability Request Format** |
| 1. When the request format is NOT an Internal Availability Request message, system normalizes the request by invoking the Horizon service(s) for message transformation using:   System BDD for Availability Request – Response (**Ref [9]**), and,  For EDIFACT PAOREQ:   * MSG UC for Receive EDIFACT Messages * MSG UC for Transform EDIFACT Messages * NGI EDIFACT Interface Specification (**[Ref 3]**)   For Direct Access Terminal Emulation:   * ??? UC for Process Direct Access Terminal Emulation Request (TBD) * ??? UC for Transform Direct Access Terminal Emulation Request – Response (TBD)   For OTA Schema:   * ??? UC for Receive OTA Messages (TBD) * ??? UC for Transform OTA Flight Availability Request – Response (TBD)   For Other Proprietary Availability Message :   * TBD   *Note: It is assumed that the applications or systems that sends availability requests to HIAS in their proprietary formats will be responsible for transformation of the availability requests to and response from HIAS into Horizon Internal Availability Request (canonical form)* | **Exception Flow 2 – Message Transformation Error** |
| 1. System validates and determines all the data elements of the availability request using the **System BDD for Availability Request – Response** (**Ref [9]**)   The below essential information in the request shall be used to consider the actions to take:   * Requester System (S&S or Internal requesters such as AFS and SEC, or OA or GDS) * Request Originator (not necessarily be the same as the system or application that actually send the request to HIAS ):   + Internal Host: when the request was originated by a host carrier user   + Internal OA: when the request was originated by another SITA hosted carrier   + External GDS: when the request was originated by a GDS and the POS User Type is not “A” (Airline) (**Ref [8]**)   + External OA: when the request was originated by an OA system or by a GDS with the POS User Type “A” (Airline) (**Ref [8]**)   *Note: Upon a Sell request originated by a GDS or External OA, S&S sends a Search Type of Determine Availability for Sell to HIAS. In such a case, the Request Originator is not S&S.*   * Search Type   + Standard Availability Request   + First Availability Request   + Determine Availability for Sell   + Determine Availability for Waitlist Sell   + Determine First Availability from Type-B Sell Failure   + More Availability * Direct Connect (Bypass) Indicator   + M = SITA Marketplace Air Bypass   + A = Partner Access on OA Bypass   + Blank = No bypass (Default) * Flight Specific Indicator (Y or N) * City Pair Origin and Destination * Departure Date or Date Range * Preferences: Routing type (Nonstop/Direct/Connection), Sort option (biased/unbiased etc), carrier and routing preference (Host only, Host & OA, specific OA only) etc * Journey data (booked segments) * Award Availability Indicator * Request Format (internal, PAOREQ, Direct Access Terminal Emulation etc) | **Exception Flow 3 – Availability Request Validation Error**  **Exception Flow 4 – Non-host Originator Requests for OA Availability**  **Alternate Flow 1 – Determine First Availability (with date window)**  **Alternate Flow 2 –Bypass Cascading Availability**  *(when Direct Connect Bypass Indicator is either M or A – Note: SMA Bypass is only valid with Internal Host originator )*  **Alternate Flow 3 – More Availability**  **Exception Flow 7 – Request Availability Beyond Maximum Sell Date** |
| 1. System generates and records the Session Reference ID for the request **(Ref [9])** |  |
| 1. When the Request Originator is NOT Internal Host, system validates the Partner Agreements:  * Checks if it is an eligible partner for interactive availability request (OA/GDS with Interactive Agreement or Selective Query settings.   *Note: Internal OA’s – Other SITA Hosted Airlines – are defined as either eligible partners for interactive availability request or not, using HA Agreement(s).*   * In case of GDS/OA requesting Award Availability, check if the GDS/OA Request Originator is eligible for Award Availability Request. * In case of OA requesting interactive availability on host code share operating flight (specific flight request), check if the OA Request Originator is eligible for inbound code share interactive availability. | **Exception Flow 5 – Non-Host Originator not Eligible for Interactive Availability Request** |
| 1. Using the data elements of the availability request, system applies the Availability and Sell business rule for each of the rule Action Types that affect routing and connection building, as below. When at least 1 business rule is selected, system determines, from the action details of the rule, the additional controls for building routings and connections for the request.  * Restrict Flights rule: system shall determine the restricted flight or flights that are not used in building routings and connections for the request, by invoking the **NGI UC for Restrict Flights in Availability**. * Other Action Type(s) - TBD (reserved for future requirements) | **Exception Flow 6 – No Actionable Segment** (when no actionable segment exists)  **Sub Flow 2 – Generate Routings for Specific Flight Request** |
| 1. When Flight Specific Indicator in the request is Y (Yes), system builds the Response Structure for the request by generating the routings based on the origin and destination pair(s), journey data, if exists, and the specific flight segments, using **Sub Flow 2 – Generate Routings for Specific Flight Request.**   *Rule: Restricted flight(s), if determined from Step 6 above, and any other additional controls, if any (for Future business rules), shall be part of the inputs to generate routings*   * When any of the requested flight or flights is part of the List of Restricted Flights determined by the **NGI UC for Restrict Flights in Availability**: for each of such flight(s), system returns response "Invalid Flight" and marks as non-actionable segment.   System joins Step 11. | **Exception Flow 6 – No Actionable Segment** (when no actionable segment exists)  **Sub Flow 2 – Generate Routings for Specific Flight Request** |
| 1. When the Flight Specific Indicator in the request is N (No), system invokes the **NGI UC for Generate Subscriber Routings.**   *Rule: Search Preferences on routing types must be used in building routings.*  *Rule: Journey data, if exists, shall also be used to build routings.*  *Rule: Routings for every date within the Date Range, if requested, must be built.*  *Rule: Traffic Restrictions must be applied in building routings and connections.*  *Rule: Restricted flight(s), if determined from Step 6 above, and any other additional controls, if any (for Future business rules), shall be part of the inputs to build routings and connection,*  *Rule: When the request originator is not Internal Host, the Orchestration shall provide an additional input for routings/connection building service to exclude subscriber (host) flight segments flagged as Host-Only Flight.*  *Note: when Search Type is “More Availability”, system already joined Alternate Flow 3 from Step* 3*, so Step* 8 *and* 9 *are skipped.* | **Exception Flow 6 – No Actionable Segment** (when no actionable segment exists) |
| 1. System sorts the routings by invoking the NGI UC for Manage and Process Product Display Order (**Ref [7]**) |  |
| 1. System builds the Response Structure for the request by selecting the first or next group of routings.   Rules:   * The number of routings in each group shall be either the Number of Routings to Returns provided in the request or the value of the **Subscriber Parameter** *Max\_Routings\_to\_Return*, whichever is lower. * Only routings with at least an actionable segment are selected. * RBD's for each segment in the routing(s) are loaded from schedule. * RBD's with Traffic Restriction Code "A" ("No Traffic Right" a.k.a. "No Local Boarding" at RBD level) are removed from the host segments. * When a host segment in the routing contains more than one leg with different configuration or RBD's, system breaks down the host segment into segments with like configuration and RBD's. * For each routing with at least a host segment, system selects the most specific Availability and Sell business rule actions "Hide RBD" and "Inhibit Sell", using the System BDD for MICT.   + When an Availability and Sell business rule action "Hide RBD" is selected for a routing, system removes the RBD's defined in the business rule action details from the host segments in the routing.   + When an Availability and Sell business rule action "Inhibit Sell" is selected for a routing, system assigns the availability status "C" (Closed) for the RBD's defined in the business rule action details on all the host segments in the routing. * When the request does not contain specific RBD(s) or cabin preference,   + If the request is NOT for Award Availability     - When the request originator is not Internal Host, system removes RBD's that are flagged as Host-Only RBD from host segments.   + If the request is for Award Availability     - For host segments system loads only the Award RBD’s.     - For OA segments, system keeps the segments with no RBD. *(Note: Award RBD's on OA segments are not published, thus not being disclosed to the Subscriber)* * When the request contains specific RBD(s) or cabin preference,   + If the request is NOT for Award Availability     - System removes the routings with OA and Host segments that do not contain any of the specified RBD(s) or cabin(s).     - For those segments that contain the specified RBD's or cabin(s), system removes all other non-specified RBD's and cabins.     - When the request originator is not Internal Host, system removes RBD's that are flagged as Host-Only RBD from host segments.   + If the request is for Award Availability     - System removes the routings with Host segments that do not contain any of the specified RBD(s) or cabin(s).     - For those Host segments that contain the specified RBD's or cabin(s), system removes all other non-specified RBD's and cabins.     - For OA segments, system loads the specified RBD's regardless of whether the specified RBD's are present in their schedule. * Any routing with host segment(s) that do not have any RBD is not selected in the Response Structure.   *Note: Award RBD’s are assigned to RBD on host Flights by Schedule Control Table (SCT) rules for “RBD Special Purposes”* | **Exception Flow 6 – No Actionable Segment** (when no actionable segment exists)  **Exception Flow 8 – No More Routing** (when Search Type is More Availability) |
| 1. System determines, from the Response Structure:  * Host operating actionable segments * Host marketing (Non-SITA OA operated) actionable segments * Host marketing (SITA Hosted OA operated) actionable segments * Non-SITA OA actionable segments * SITA Hosted OA actionable segments (segments of other airlines that are hosted in HIAS) |  |
| 1. When at least a host operating or host marketing (OA operated) actionable segment exists in the Response Structure, system gets availability information for the host operating/marketing actionable segments using:   **System BDD for Availability Request – Response (Ref[9])**  **NGI UC for Determine Allocation Availability on Host Flights (Ref [1])**   * Any RBD's determined as "Inhibit Sell" (by applying Availability and Sell business rule) are assigned with availability status "C" (Closed) in the Response Structure and are excluded from the RBD's to be determined allocation availability on host segments.   *Note: The UC for Determine Allocation Availability determines Allocation availability on host operating segments, host Block Space Code Share marketing segments and host Free-Sell Code share marketing segments. Availability for Host Free Sell Code Share marketing segments determined at this step is based on AVS or AVA status is stored in HIAS.* |  |
| 1. When at least a host Free Sell Code Share Marketing - SITA Hosted OA Operated actionable segment exists, system queries the Code Share Agreements and Partner Agreement to determine the eligibility for cascading Seamless availability to the SITA Hosted OA operating partner for every of such segments.   A host Free Sell Code Share Marketing segment is eligible for cascading availability to SITA Hosted OA Operating Partner only when all the below conditions meet:   * Cascading availability is selected for Code Share in the Partner Agreement for both the requesting subscriber and the OA Operating subscriber.   *Note: Seamless Availability on SITA Hosted OA Flight is done internally in HIAS per Sub Flow 1. Subscriber Parameter Partner\_Access does not control SITA internal cascading availability.* |  |
| 1. When at least a SITA Hosted OA actionable segment exists, AND/OR   When at least a host Free Sell Code Share Marketing (SITA Hosted OA Operated) actionable segment exists and is eligible for Code Share Seamless Availability,  For each SITA Hosted OA carrier of those segments,  System gets availability information for the SITA Hosted OA airline segment using **Sub Flow 1 - Determine Availability on SITA Hosted OA Segments**  *Note: POS rules on the OA SITA airline must apply*  Rule: If Award Availability Indicator is Y, the indicator must be included in the outbound request | **Sub Flow 1 - Determine Availability on SITA Hosted OA Segments** |
| 1. When at least a Non-SITA OA actionable segment exists, for each of those segments, system assigns the stored AVS or AVN status for each RBD on all the OA flight segments. If no AVS or AVN status can be located for an RBD, system assigns the status "Z" (No Knowledge about availability) for that RBD.   System marks all the Non-SITA OA actionable segments as candidates for cascading availability, when and only when the below conditions meet:   * The Request Originator is Internal Host, and * The Subscriber Parameter Partner\_Access is set to Yes, and * The Search Type is one of the below:   + - Standard Availability Request     - More Availability   *Note: "First Availability Request" is not allowed in cascading availability. Other Search Types such as "Determine Availability for Sell" do not cause a cascading availability for Non-SITA OA segments*. |  |
| 1. When there is at least a host Free Sell Code Share Marketing actionable segment operated by a non-SITA OA carrier, system marks all the host Free Sell Code Share marketing segments operated by Non-SITA OA's as candidates for cascading availability, when and only when the below conditions meet:  * The Subscriber Parameter Partner\_Access is set to Yes, and * The Search Type is one of the below:   + - Standard Availability Request     - More Availability |  |
| 1. When there are non-SITA OA actionable segments and/or host Free Sell Code Share Marketing actionable segments operated by non-SITA OA's that are marked as candidates for cascading availability, system invokes the **NGI UC for Cascade Availability** to get seamless availability for those segments  * When there are host Free Sell Code Share marketing segment(s) that need to be processed with Cascade Seamless Availability, system combines the Host marketing segment(s) operated by the OA and the OA actionable segments of the same OA carrier in a Partner Access session. * If Award Availability Indicator is Y, the indicator must be included in the outbound request * System records the Availability Source Indicator and Session ID's received from the external system(s) for the successfully cascaded segments. * Any RBD's determined as "Inhibit Sell" (by applying Availability and Sell business rule) are assigned with availability status "C" (Closed) on host Free Sell Code Share marketing segments in the Response Structure.   System waits until all Cascade Availability sessions complete. | **NGI UC for Cascade Availability** |
| 1. When there is no host actionable segment in the response structure, system joins Step 22. |  |
| 1. For each routing containing at least 1 host segment, system selects the most specific Availability and Sell business rule using the System BDD for MICT for each of the Action Types that adjust the availability, as below:  * Adjust Availability (Future) * TBD (reserved for Future requirements) |  |
| 1. System applies the selected business rule action for each of the business rule Action Types to each of the routings. |  |
| 1. System loops to Step 19 until all routings have been processed. |  |
| 1. For each segment in each routing, determines additional information for the response  * System gets Equipment (Aircraft Type), Change of Equipment, if any, Smoking Indicator, and Meal code for every RBD from the schedule. * System determines Segment Notes for Host Segments by invoking Sub Flow 3 | Sub Flow 3 – Determine Segment Note |
| 1. System adjusts the availability response with the below rules:  * When Quota Sales applies to the GDS/OA requesters on Host actionable segments, system sets the seats availability values that are greater than quota sales values to the quota sale values. * When Quota Sales applies to the OA Code Share Marketing Originator on Host actionable Code Share Operating segments, system sets the seats availability values that are greater than quota sales values on those segments to the quota sale values. * When the Request Originator is Internal Host AND when the number of requested seats exists in the request (availability request or sell request), system sets any seats availability values, on host or OA actionable segments, that are greater than the number of requested seats to the availability status "A" (Available). |  |
| 1. System removes all journey data segments from the response structure.   When Flight Specific Indicator in the request is N (No), system removes all routings with at least a Host actionable segment in error as returned by the **NGI UC for Determine Allocation Availability on Host Flights (Ref [1])** |  |
| 1. When the Flight Specific Indicator in the request is N (No), system initiates the session timer.   *Note: The session timer is for determined if the availability session is still valid for a follow on "More Availability" request or has been expired.* |  |
| 1. System returns the availability response structure to the requester with the appropriate format.  * When the Request Format is EDIFACT PAOREQ, system invokes Messaging transformation services (MSG UC for Transform EDIFACT Messages - TBD) and send the response to the requester (MSG UC for Send EDIFACT Message – TBD) * When the Request Format is Direct Access Terminal Emulation, system invokes Messaging transformation services (**??? UC for Transform Direct Access Terminal Emulation Request – Response (TBD))** and send the response to the requester (**??? UC for Process Direct Access Terminal Emulation Messages (TBD)** * When the Request Format is OTA Schema, system invokes Messaging transformation services and sends the response to the requester. |  |
| 1. Flow ends. |  |

|  |  |
| --- | --- |
| Specific Post Conditions | * System has determined and returned availability response |

## **Basic Flow 2 – Simulate Availability**

|  |  |
| --- | --- |
| Specific Preconditions | * System has received a request for the Simulate Availability |

| Activity | Alternate / Exception Flow Name |
| --- | --- |
| 1. System receives a request for Simulate Availability |  |
| 1. System performs Step 3 to Step 25 for **Basic Flow 1** | Note: When it is a First Availability Simulate, system invokes **Alternate Flow 1 – Determine First Availability** then resumes in this flow. |
| 1. System returns the availability response in HIAS GUI for Simulate Availability |  |
| 1. Flow ends |  |

|  |  |
| --- | --- |
| Specific Post Conditions | * System has determined and returned availability response in Simulate Availability GUI |

# Alternate Flows

## **Alternate Flow 1 – Determine First Availability**

This Flow describes the processing for First Availability requests from S&S or GDS/OA, or for a determination of first availability upon a unable Type-B sell transaction that is requested by S&S.

|  |  |
| --- | --- |
| Specific Preconditions | * System has performed:   + Step 3 for Basic Flow 1 * Search Type is First Availability Request or Determine First Availability from Type-B Sell Failure * The Subscriber Parameters *First\_Availability\_Window\_Start* and *First\_Availability\_Window\_End* have been defined * The Subscriber Parameters *First\_Availability\_TypeB\_Sell\_Window\_Start* and *First\_Availability\_ TypeB\_Sell\_Window\_End* have been defined |

| Activity | | Alternate / Exception Flow Name |
| --- | --- | --- |
| 1. When the Request Originator is NOT Internal Host,  * If the Search Type is not Determine First Availability from Type-B Sell Failure, system performs Step 5 of **Basic Flow 1** to determine the eligibility for Interactive Availability.   When the Request Originator is not eligible for Interactive Availability, system switches to **Exception Flow 5 – Non-Host Originator not Eligible for Interactive Availability Request**  When the Request Originator is eligible for Interactive Availability, system continues to the next step.   * If the Search Type is Determine First Availability from Type-B Sell Failure, system continues to the next step.   When the Request Originator is Internal Host, system continues to the next step. | |  |
| 1. System generates and records the Session Reference ID for the request (Ref [9]) | |  |
| 1. System selects the requested date as the Date for Processing | |  |
| 1. System perform Step 8 for **Basic Flow 1** to build routings for the Date for Processing | |  |
| 1. System continues to perform Step 9 to Step 25 for Basic Flow 1 to determine availability on the routings of the Date for Processing | |  |
| 1. When seats available in the requested RBD, if specified in the request, or in at least an RBD, if no RBD is specified in the request,   System joins Step 9. | |  |
| 1. System sets the Date for Processing to the next date in the First Availability Window  * Rule: System searches forward from the requested date + 1 first, and then continues searching backward from the requested date - 1 (??) * When Search Type is First Availability request, the Subscriber Parameters *First\_Availability\_Window\_Start* and *First\_Availability\_Window\_End* shall be used. * When Search Type is Determine First Availability for Type-B Sell Failure, the Subscriber Parameters *First\_Availability\_TypeB\_Sell\_Window\_Start* and *First\_Availability\_ TypeB\_Sell\_Window\_End* shall be used. | |  |
| 1. System loops to Step 4. | |  |
| 1. System returns the availability response to the requester.  * When seats available, as determined in Step 6,the response contains only the first routing with seats are available. * When there is no available routings determined, system returns “No First Available” response. * When this Alternate Flow was invoked by **Basic Flow 2 – Simulate Availability**, system returns the availability response in the Simulate Availability GUI. * When this Alternate Flow was invoked by **Basic Flow 1 – Process Availability Request and Response Orchestration**, system rejoins **Step 26** for **Basic Flow 1**. | |  |
| Specific Post Conditions | * First Availability response has been returned | |

## **Alternate Flow 2 –Bypass Cascading Availability**

Horizon system must provide cascading availability to GDS/OA. Responsibility for cascading requests to GDS/OA will be determined. In any cases, there should not be duplicate processes/services in Horizon that transforms host requests into and send outbound EDIFACT PAOREQ messages.

|  |  |
| --- | --- |
| Specific Preconditions | * System has performed:   + Step 3 for Basic Flow 1 * Direct Connection Bypass Indicator is M (SITA Marketplace Air Bypass) or A (Partner Access Bypass) * The request originator is Internal Host * Search Type is Standard Availability Request |

| Activity | | Alternate / Exception Flow Name |
| --- | --- | --- |
| 1. System determines the request eligibility for Bypass cascading availability  * When the Direct Connect Bypass Indicator is "M": by querying the SITA Marketplace Air Agreement. The request is eligible for Bypass Cascading to SITA Marketplace Air (SMA) if and only if   + The Subscriber Parameter Partner\_Access is set to Yes   + The Subscriber Parameter Marketplace\_Air is set to Yes   + The request originator is Internal Host and belongs to the SMA Allowable Offices.   + The request is not for Award Availability * When the Direct Connect Bypass Indicator is "A": by querying the Partner Access Agreement with the OA. The request is eligible for Bypass Cascading to Partner Access if and only if   + The Subscriber Parameter Partner\_Access is set to Yes   + The OA allows Bypass cascading.   + The Request Originator is eligible for Cascade Availability processing   + When the request is for Award Availability, the OA allows cascading Award Availability   + When the request is not for Award Availability, the OA allows cascading normal published services. | |  |
| 1. When the request is not eligible for Bypass cascading, flow ends. | |  |
| 1. System generates and records the Session Reference ID for the availability request (Ref [9]) | |  |
| 1. System Invokes **NGI UC for Cascade Availability – Basic Flow 1 –Bypass Cascading Availability** | |  |
| 1. When system received the OA Partner or Travelport Session ID and "More Availability Token", as returned from Step 4, system attaches the OA Partner or Travelport Session ID and "More Availability Token" to the Session Reference ID of the availability request, as generated in Step 3. System logs the Session Reference ID for the request. | |  |
| 1. System returns the availability response, including the Session reference ID. | |  |
| 1. Flow ends. | |  |
| Specific Post Conditions | * Availability response from the OA or GDS Partner has been received and returned to the requester, | |

## **Alternate Flow 3 – More Availability**

This Flow describes the processing for a More Availability request

|  |  |
| --- | --- |
| Specific Preconditions | * System has performed:   + Step 3 for Basic Flow 1 * Search Type is More Availability * The Flight Specific Indicator in the request is N (No) |

| Activity | | Alternate / Exception Flow Name |
| --- | --- | --- |
| 1. System determines the Session Reference ID of the previously returned availability response and the elapsed time since the previous timer initiation for the session (**Ref [9])**. | |  |
| 1. When the Previous Session Reference ID is Blank or does not match with any logged Session Reference ID's, system returns the response "Invalid Request". Flow ends. | |  |
| 1. When the session timer has expired, system returns the response "Session expired ". Flow ends. | |  |
| 1. When the Previous Session Reference ID contains a "More Availability Token" for Marketplace Air or Partner Access Bypass Cascading Availability  * If the "More Availability Token" in the Previous Session Reference ID does not match any "More Availability Token" from the log, system returns error "Session expired" and Flow ends. * If the "More Availability Token" in the Previous Session Reference ID matches a "More Availability Token" from the log, system invokes **NGI UC for Cascade Availability - Basic Flow 3 (More Availability).** System returns the response and flow ends.   *Note: The NGI UC for Cascading Availability logged the "More Availability Token" for a Bypass Cascading Availability session. The "More Availability Token" is cleared from log when the session timer reaches the session timeout in order to inhibit "More Availability" for an expired session.* | |  |
| 1. System resets the timer for the session and rejoins Step 10 for Basic Flow 1. | |  |
| Specific Post Conditions | * More availability has been determined or timeout response has been returned. | |

# Exception Flows

## **Exception Flow 1 – Unable to Recognize Availability Request**

|  |  |
| --- | --- |
| Specific Preconditions | * System has performed:   + Step 1 for Basic Flow 1 |

| Activity | | Alternate / Exception Flow Name |
| --- | --- | --- |
| 1. System is unable to recognize the availability request type (non-supported) | |  |
| 1. System logs error and returns the appropriate error message | |  |
| 1. Flow ends | |  |
| Specific Post Conditions | * Errors message has been logged and returned | |

## **Exception Flow 2 – Message Transformation Error**

|  |  |
| --- | --- |
| Specific Preconditions | * System has performed:   + Step 2 for Basic Flow 1 |

| Activity | | Alternate / Exception Flow Name |
| --- | --- | --- |
| 1. System receives an error response from the corresponding Transformation service | |  |
| 1. System logs error and returns the appropriate error message | |  |
| 1. Flow ends | |  |
| Specific Post Conditions | * Errors message has been logged and returned | |

## **Exception Flow 3 – Availability Request Validation Error**

|  |  |
| --- | --- |
| Specific Preconditions | * System has performed:   + Step 3 for Basic Flow 1 |

| Activity | | Alternate / Exception Flow Name |
| --- | --- | --- |
| 1. System detects a validation error | |  |
| 1. System logs error and returns the appropriate error message | |  |
| 1. Flow ends | |  |
| Specific Post Conditions | * Errors message has been logged and returned | |

## **Exception Flow 4 – Non-Host Originator Requests for OA Availability**

|  |  |
| --- | --- |
| Specific Preconditions | * System has performed:   + Step 3 for Basic Flow 1 |

| Activity | | Alternate / Exception Flow Name |
| --- | --- | --- |
| 1. System detects that it is an External GDS or External OA Originator or Internal OA Originator requests for an OA specific availability | |  |
| 1. System logs error and returns the appropriate error message | |  |
| 1. Flow ends | |  |
| Specific Post Conditions | * Errors message has been logged and returned | |

## **Exception Flow 5 – Non-Host Originator not Eligible for Interactive Availability Request**

|  |  |
| --- | --- |
| Specific Preconditions | * System has performed:   + Step 5 for Basic Flow 1 |

| Activity | | Alternate / Exception Flow Name |
| --- | --- | --- |
| 1. System detects that the request originator is NOT Internal Host, AND one of the below conditions:  * It is NOT an eligible partner for interactive availability request (OA/GDS with Interactive Agreement or Selective Query settings), OR * In case of GDS/OA requesting Award Availability, the Originator is NOT eligible for Award Availability Request, OR * In case of External OA or Internal OA requesting interactive availability on host code share operating flight (specific flight request), the Originator is NOT eligible for inbound code share interactive availability on the Subscriber. | |  |
| 1. System logs error and returns the appropriate error message | |  |
| 1. Flow ends | |  |
| Specific Post Conditions | * Errors message has been logged and returned | |

## **Exception Flow 6 – No Actionable Segment**

|  |  |
| --- | --- |
| Specific Preconditions | * System has performed:   + Step 7 or Step 8 or Step 10 for Basic Flow 1 |

| Activity | | Alternate / Exception Flow Name |
| --- | --- | --- |
| 1. System is unable to find an actionable segments (not part of journey data) to match the request | |  |
| 1. System logs error and returns the appropriate error message indicating “No Routings” | |  |
| 1. Flow ends | |  |
| Specific Post Conditions | * Errors message has been logged and returned | |

## **Exception Flow 7 – Request Availability Beyond Maximum Sell Date**

|  |  |
| --- | --- |
| Specific Preconditions | * System has performed:   + Step 3 for Basic Flow 1 |

| Activity | | Alternate / Exception Flow Name |
| --- | --- | --- |
| 1. System detects that the requested date is beyond the Maximum Sell Date defined by Subscriber Parameter Maximum\_Sell\_Date | |  |
| 1. System returns "No Routing" response. | |  |
| 1. Flow ends | |  |
| Specific Post Conditions | * Errors message has been logged and returned | |

## **Exception Flow 8 – No More Routings**

|  |  |
| --- | --- |
| Specific Preconditions | * System has performed:   + Step 10 for Basic Flow 1 * Search Type is More Availability |

| Activity | | Alternate / Exception Flow Name |
| --- | --- | --- |
| 1. System is unable to find an actionable segments (not part of journey data) to match the request | |  |
| 1. System logs error and returns the appropriate error message indicating “No More Routings” | |  |
| 1. Flow ends | |  |
| Specific Post Conditions | * Errors message has been logged and returned | |

# Sub Flows

## **Sub Flow 1 – Determine Availability on SITA Hosted OA Segments**

|  |  |
| --- | --- |
| Specific Preconditions | * System has performed:   + Step 14 for Basic Flow 1 |

| Activity | | Alternate / Exception Flow Name |
| --- | --- | --- |
| 1. System queries the Agreement with the SITA Hosted OA to determine the Availability Type between the Subscriber airline and the SITA hosted OA. | |  |
| 1. When the Availability Type is Interactive, system invokes the **NGI UC for Determine Seamless Availability on Other Host Airline flight – (TBD)** with the SITA OA segments or host free-sale marketing segment(s) operated by the SITA OA as the input. | |  |
| 1. When the Availability Type is NOT Interactive, OR   When there is an error or timeout response,  System assigns the stored AVS/AVN status for the input segments. | |  |
| 1. System resumes in the invoking Flow/Step. | |  |
| Specific Post Conditions |  | |

## **Sub Flow 2 – Generate Routings for Specific Flight Request**

|  |  |
| --- | --- |
| Specific Preconditions | * System has performed:   + Step 7 for Basic Flow 1 |

| Activity | | Alternate / Exception Flow Name |
| --- | --- | --- |
| 1. System detects that the Specific Flight Indicator in the request is set to “Y” (Yes) | |  |
| 1. System creates the Response Structure by building the routings between each of the requested City Pairs Origin – Destination using  * Journey data (booked segments), if exists, and the provided specific flight segments. * Subscriber’s connection rules   Rules:   * Only routings with at least an actionable segment are selected. * System stops building routings when the number of routings is greater than the value of the **Subscriber Parameter** *Max\_Routings\_to\_Return*. Flight segments remaining in the request are truncated. * RBD's are loaded from schedule for all actionable segments. * For each routing with at least a host segment, system selects the most specific Availability and Sell business rule actions "Hide RBD" and "Inhibit Sell", using the System BDD for MICT.   + When an Availability and Sell business rule action "Hide RBD" is selected for a routing, system does NOT load the RBD's defined in the business rule action details for the host segments in the routing.   + When an Availability and Sell business rule action "Inhibit Sell" is selected for a routing, system assigns the availability status "C" (Closed) for the RBD's defined in the business rule action details on all the host segments in the routing. * When the request does not contains specific RBD(s),   + If the request is for Award Availability,     - For host segments system loads only the Award RBD’s from the schedule. RBD's determined as "Hide RBD" are excluded. Any host segment(s) that do not contain any Award RBD's are marked as non-actionable and assigned with the response "Invalid Flight" in the Response Structure..     - For OA segments, system keeps the segments with no RBD. *(Note: Award RBD's on OA segments are not published, thus not being disclosed to the Subscriber)*   + If the request is not for Award Availability,     - For all segments, system loads RBD's from the schedule. RBD's determined as "Hide RBD" are excluded on host segments.     - For host segments containing more than 1 leg with different RBD's, system loads only the common RBD's.     - When the request originator is not Internal Host, system removes RBD's that are flagged as Host-Only RBD from host segments.     - Host segment(s) that do not have any RBD are marked as non-actionable and assigned with the response "Invalid Flight" in the Response Structure. * When the request contains specific RBD(s),   + If the request is for Award Availability,     - For host segments system loads only the specified RBD's that are Award RBD’s. RBD's determined as "Hide RBD" are excluded. Host segments that do not contain any matched Award RBD's are marked as non-actionable and assigned with the response "Invalid Class" in the Response Structure.     - For OA segments, system loads the specified RBD's. Segments that do not contain any of the specified RBD's in their schedule are retained in the Response Structure.   + If the request is not for Award Availability,     - System loads the specified RBD's that exist in the schedule for all segments. RBD's determined as "Hide RBD" are excluded for host segments.     - For host segments containing more than 1 leg with different RBD's, system loads only the common RBD's.     - Host and/or OA segments that do not contain any of the specified RBD's in their schedule are marked as non-actionable and assigned with the response "Invalid Class" in the Response Structure.     - When the request originator is not Internal Host, system removes RBD's that are flagged as Host-Only RBD from host segments. Host segment(s) that do not have any RBD are marked as non-actionable and assigned with the response "Invalid Flight" in the Response Structure.   *Note: Award RBD’s are assigned to RBD on host Flights by Schedule Control Table (SCT) rules for “RBD Special Purposes”* | |  |
| 1. System invokes **UC for Assign Married Segment** in order to determine potential marriage units between the segments in routings   *(Note: this Step is for FUTURE consideration).* | |  |
| 1. System returns to the invoking Flow/Step | |  |
| Specific Post Conditions |  | |

## **Sub Flow 3 – Determine Segment Note**

*Note: this Sub Flow is carried over from the old version of the NGI UC for Determine Allocation Availability dated back in 2012. For more elaborated information for this Flow, please refer to the mentioned UC version.*

|  |  |
| --- | --- |
| Specific Preconditions | * System has performed:   + Step 22 for Basic Flow 1 |

| Activity | | Alternate / Exception Flow Name |
| --- | --- | --- |
| 1. System checks if the availability request has come from host (request originator is Internal Host). If not, then the system proceeds to step 4 below. | |  |
| 1. System checks cache for matching Segment Notes with “Availability Display” usage option for those operational segments that may have matching notes. | |  |
| 1. System returns to the Flow/Step where it was invoked. | |  |
| 1. System checks cache for matching Segment Notes with "Non-Host Interactive" usage option for those operational segments that may have matching notes. | |  |
| 1. System includes found segment notes for the segment into availability response. | |  |
| 1. System resumes in the invoking Flow/Step. | |  |
| Specific Post Conditions | * System has found applicable segment notes and included them into availability response. | |

# Special Requirements

# Subscriber parameter

* *Max\_Routings\_to\_Return:* Parameter that defines the maximum number of routings to return in each availability response.
* *First\_Availability\_Window\_Start* and *First\_Availability\_Window\_End* : Parameters that define date range to search for first availability, which are the number of days before and the number of days after the date specified in a First Availability request, or today's date when the request does not specify any date.
* *First\_Availability\_TypeB\_Sell\_Window\_Start* and *First\_Availability\_ TypeB\_Sell\_Window\_End* : Parameters that define date range to search for first availability when a type-B sell is failed due to no seats available, which are the number of days before and the number of days after the requested date.

*Note: Other subscriber parameters for availability processing and generate routings are defined in the corresponding Use Case(s).*

# Message Transformation

Horizon must provide the transformation services to transform the below type of availability requests into HIAS canonical form and to transform HIAS responses into the message formats corresponding to the requests received:

* 1. EDIFACT PAOREQ (request) and PAORES (response) from GDS and OA. This is to support inbound interactive availability request.
  2. HIAS canonical form for availability request into EDIFACT PAOREQ (request from HIAS to external system) and PAORES (response from GDS and OA) into HIAS canonical form for availability response. This is to support outbound interactive availability request, for example in SITA Marketplace Air use.
  3. Direct Access Terminal Emulation inbound request and outbound response. This is to support GDS/OA direct access request sent into Horizon.
  4. Direct Access Terminal Emulation outbound request and inbound response. This is to support subscriber's direct access requests sent out to OA's.

*Note: The need for Outbound Direct Access Terminal Emulation will be determined based on specific customer's requirement on a particular OA that cannot support Direct Access Interactive.*

* 1. Inbound OTA Schemas availability requests into HIAS canonical form and vice versa.

For other availability request formats (proprietary), it is assumed that the application or system that relies on the proprietary format shall be responsible for the transformation.

# Partner Agreements

### Internal OA Agreement

Agreement between HIAS Hosted carriers must be in place in order to define the below availability controls

* 1. The Availability Type: Inbound or Outbound or Both
     + AVS or
     + AVN or
     + Interactive
  2. Scope for Interactive Availability Type: (Allowed or Disallowed; Default or On Request only)
     + Award Availability
     + Code share flights only
     + All published flights or selected flights or flight /date range

### External OA Agreement

Agreement between the Subscriber and non-SITA OA must be in place in order to define the below availability controls

1. The Availability Type: Inbound or Outbound or Both
   * + AVS or
     + AVN or
     + Interactive
     + Direct Access
2. Scope for Interactive Availability Type: (Allowed or Disallowed; Default or On Request only)
   * + Award Availability
     + Code share flights only
     + All published flights or selected flights or flight /date range
3. Allowable Booking Offices for Outbound Interactive Availability (Partner Access): The booking offices that are allowed to get Partner Access cascading availability, for each of the Interactive Availability types (Award Availability, Code Share, Published)

### External GDS Agreement

1. The Availability Type: (Inbound only)
   * + AVS or
     + AVN or
     + Interactive (Allowed or Disallowed)
     + Direct Access (Allowed or Disallowed)
2. Scope for Interactive Availability Type:
   * + Award Availability: IATA/In-house POS's that are allowed to request Award Availability on host flights
     + All published flights or selected flights or flight /date range

### SITA Marketplace Air

1. Scope for Interactive Availability Type:
   * + Service allowed or disallowed
     + Selected OA's or All OA's to cascade availability
     + Selected OA flights or flight range and/or date range (??)
2. Allowable Booking Offices for SITA Marketplace Air: The booking offices that are allowed to get SITA Marketplace Air cascading availability.

### Interactive and Direct Access Availability and Sell Session Timeout control

The above Agreements must provide the capability to control session timeout for both inbound and outbound requests.

Subscriber parameters may be used to define the default interactive availability and/or sell session timeout.

# Document Updates

Section 3 **– Reference** lists all documents related to this Use Case. Please refer to the "Notes" column that outlines the requirements to update each of the related documents.

# Assumptions, Issues & Future Use Case Considerations

**Assumptions**

* SITA will continue to support Direct Access Terminal Emulation in Horizon.
* Outbound interactive availability messaging shall be handled by a common service that will be used by both HIAS and S&S.
* S&S always forwards the POS information as received from other systems to HIAS in availability requests.
* S&S always forwards journey data as received from other systems or as exists in CJ to HIAS in availability requests.

**Issues**

* As time of writing, it is not known about which domain will generate direct connect availability requests such as outbound direct access, SITA Marketplace Air Bypass etc to other systems. There is a question about whether S&S should send request to HIAS for cases that do not involve HIAS actions such as direct connect availability to Alliance or Partner Access or SMA Bypass.
* Does it required for HIAS to support Calendar Availability OR will Airfare Shop handle it?

**Future Considerations**

* Enhanced Availability Data Messages (AIRIMP 4.9)

# Additional Information

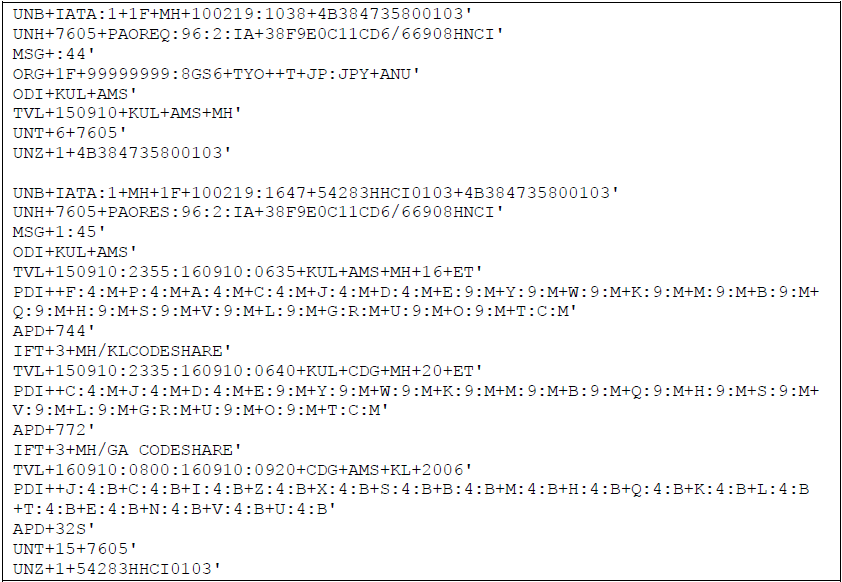
Notes from Cliff

Here is how I see it to work the best.

* S&S makes AV call to HIAS
* HIAS creates display and within display one airline is RSS and other Partner Access
* HIAS cascades the request to 1G (RSS) and another to Partner Access
* HIAS receive a response from both
* HIAS returns AV display to S&S with indicators of RSS and Partner Access on each availability line.
* S&S makes a sale on RSS availability and ITAREQ goes direct to 1G(RSS) This starts the session
* S&S request a seat map (SMPREQ) on previously sold RSS airline. SMPREQ goes directly to 1G. This usually is a one off and will not be on session established by ITAREQ.
* S&S closes PNR and HWPREQ goes directly to 1G. This closes the session

This does not take into account the bypass input to go directly to 1G(RSS). I do not think HIAS should be a middle man in this request and the city pair request (PAOREQ MSG 44) should go direct to 1G (RSS).

* Sample PAORES and PAOREQ message



# Business Data Definitions

**Ref [9]**

# BDD Supporting Business Rules

None

# Use Case Elaboration

# AVAIL.24e: Setup client side cache of MICT records with Hide RBD and Inhibit Sale rules

### Summary

This story describes the need to setup a client side cache (local view in Weblogic server nodes) of MICT records with Hide RBD and/or Inhibit sale rules from MICT rules cache available in the GigaSpaces data grid.

### Detail

This task can be done concurrently with AVAIL.24f, which applies Hide RBD and/or Inhibit sale rules to the availability response returned by the Inventory enquirer.

To enable Inventory Enquirer service to rapidly access the Hide RBD and Inhibit sale rules for applying the rules, a local view (cache) of MICT records with Hide RBD and Inhibit sale rules needs to be set up in the WebLogic server nodes.

The local view must consist of MICT records that match all the following filter conditions

1. The MICT record's Action element contains either RBDInhibitSaleAction element or RBDHideAction element as in the XPaths below.

../MICTRecord/Action/RBDInhibitSaleAction

../MICTRecord/Action/RBDHideAction

1. The MICT record should be in 'Active' state

../MICTRecord/Status is 'Active'

1. The MICT record's document status is 'Active'

../{DocumentHeader}/AdministrativeRecord/DocumentStatus

Where an MICT record matches all the above filter criteria, the whole MICT record is loaded in the local view.

As the MICT records with Hide RBD and/or Inhibit sale rules are available in the local view of the WebLogic node, retrieval of these MICT rules from the local view should not take more than 1 millisecond.

In an effort to monitor the adding and updating of MICT records that are loaded in the local view, explicit logging is required

* Whenever an MICT record is updated in the view.
* Whenever a new MICT record is added to the local view after startup.

If there is a provision to use call-back methods or triggers, these mechanisms must be used for the logging activity.

### Acceptance

Definition of done requires that all test scenarios are satisfied; furthermore it is expected that the development team will identify additional scenarios to add to the test pack.

| **Sr.** | **Scenario** | **Action** | **Prerequisite** | **Post Condition** |
| --- | --- | --- | --- | --- |
|  | State of the local view at startup. | Query the local view for MICT records that were loaded after startup. | MICT records are available in the GigaSpaces data grid including those that are 'Active' and with Hide RBD and Inhibit sale rules. | The local view in the WebLogic node must contain MICT records that are 'Active' and with Hide RBD and Inhibit sale rules.  This should be verified using the log files. |
|  | Is an MICT record in the local view up to date with the same MICT record in GigaSpaces data grid? | Query the local view for an MICT record. | Update the MICT record (being tested) using GUI/SOAP UI resulting in updating of MICT rules cache in GigaSpaces data grid. | The MICT record in the local view must match with the MICT record in MICT rules cache in GigaSpaces data grid.  The updating of the MICT record in the local view should not take more than 100 milliseconds.  This should be verified using the log files. |
|  | Does an MICT only get updated when it is active? | Query the local view for an MICT record. | Update an MICT record with status as 'Inactive' using GUI/SOAP UI resulting in updating of MICT rules cache in GigaSpaces data grid. | The MICT record should not be available in the local view because only 'Active' MICT records are available in the local view. |
|  | Does the view survive a switch from primary to back up in the space? | Query the local view for a previously queried MICT record. | Update the MICT record (being tested) using GUI/SOAP UI resulting in updating of MICT rules cache in GigaSpaces data grid.  Switch off the primary space. | The local view must contain the latest copy of the MICT record obtained from the back up space (within 100 milliseconds). |

# AVAIL.24f – Apply RBD Hide & Inhibit Sale rule during Availability determination

### Summary

This story adds the capability to apply availability and business sell rules while returning availability response.

Prior to taking up this task, ensure development of INV.04pq – MICT rule for RBD Hide and Inhibit from Sale is developed. Caching of availability and sell business rules (AVAIL.24e) can done together with this task.

*Note: This UC is not final and therefore recommend offshore do not refer to any of its flows, while delivering this task.*

### Detail

The rules apply only on host segments. Following MICT rules are applied:

1. Hide RBD
2. Inhibit Sale

Availability & sell business rules are maintained in operational Weblogic. Availability information is maintained in availability space within the grid. The rules could be applicable to the segment for which availability has been sought or to the POS (request originator). After fetching the availability information, system amends the response structure depending on the rules as below:

* When Business Rule for Hide RBD applies, the corresponding RBDs/Cabins are not displayed on the flight/routing
* When Business Rule for Inhibit Sell applies, system assigns “C" status (Closed – Request Closed) to such RBDs.
* **The rules need to be checked irrespective of whether the request is for a booking or group or waitlist.**

To deliver this task, existing process of constructing availability response for simulate availability is adequate. Please note the Availability calculation in not affected by this.

System prevents inventory adjustment (Sell/Cancel) on RBDs that meet the above criteria for their corresponding POS (which will be addressed in INV.25def)

### Acceptance Tests

Definition of done requires that all test scenarios are satisfied; furthermore it is expected that the development team will identify additional scenarios to add to the test pack. Demo has to be done through the GUI.

**Test Data**

1. Following schedules/ICRs in FY configuration exists in the system

|  |  |  |  |
| --- | --- | --- | --- |
| # | Flight # | Route | RBDs |
| 1 | 100 | LHR-DEL | FPYBMHRSTQLV |
| 2 | 110 | LHR-FRA-DEL | FPYBMHRSTQLV |
| 3 | 200 | LHR-FRA | FPYBMHRSTQLV |
| 4 | 210 | FRA-DEL | FPYBMHRSTQLV |

Ensure connection exists between LHR-FRA + FRA-DEL flight

1. Following MICTs for Hide RBD / Inhibit Sell exist.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Rule | POS Group | Flight Scope | Condition |
| 1 | **Hide RBDs**: RBDs – L, V | **Ireland POS:** POS Location – IE (Country) | **Country – Country**: GB - IN | Flight Type: **Direct** |
| 2 | **Inhibit Sell**: RBDs – H, S | **Ireland POS:** POS Location – IE (Country) | **Country – Country**: GB - IN | Flight Type: **Direct** |
| 3 | **Hide RBDs**: RBDs – T, Q | **Ireland POS:** POS Location – IE (Country) | **Country – Country**: GB - IN | Flight Type: **Non Stop** |
| 4 | **Hide RBDs**: Cabin – F | **-** | **Airport – Airport**: LHR-DEL | Flight Type: **Connect** |

**Note**: If both Hide and Inhibit rule exists for same RBD for same period and matching conditions, then Hide rule takes precedence, since it also prevents sale on the RBD.

| **Sr.** | **Test** | **Prerequisite** | **Scenario** | **Post Condition** |
| --- | --- | --- | --- | --- |
|  | Availability Search for POS to whom rule apply | ICRs exist in the grid with Hide RBD and Inhibit sale applicable as above (#1 ,#2 & #3). | POS User to whom the rules apply, tries to check availability on LHR-DEL. | System hides the RBDs and displays ‘C’ against applicable RBDs for the POS in the Availability response (as shown below) |
|  | Availability Search for POS without any rule | As above | POS User to whom the rules don’t apply, tries to check availability. | System returns all RBDs for the POS in the Availability response (as shown below) |
|  | Test for location based MICT, regardless of POS | Add Rule #4 to above | Multiple users try to check availability between LHR-DEL | System does not display F cabin for 200+210 connection , regardless of POS |
|  | Test for effective from / to dates | MICT rule exists but is effective on future date. | User tries to check availability on a segment where MICT rule is yet to apply. | System displays all availability data on that segment (since MICT is still not applicable) |
|  | Departure date in future | MICT rule exists with rule applicable for flights with future Departure date | User tries to check availability on a segment for future departure date on which MICT is applicable | System applies the corresponding availability rules and displays only those RBDs on which rules don’t apply |
|  | Group & waitlist (dependent on Avail.26) | As in A1 | POS User, tries to seek availability for group or waitlist on RBDs on which the rules apply | System returns groups / waitlist availability as per rule (ie Hides RBDs & displays ‘C’ for inhibit sale RBDs) |

**Availability Response**

When No rules exist (normal availability)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Flt.No** | **Org** | **Dest.** | **Stops** | **Availability** |
| 1 | 100 | LHR | DEL | 0 | F9 P9 Y9 B9 M9 H9 R9 S9 T9 Q9 L9 V9 |
| 2 | 110 | LHR | DEL | 1 | F9 P9 Y9 B9 M9 H9 R9 S9 T9 Q9 L9 V9 |
| 3 | 200  210 | LHR  FRA | FRA  DEL | 0  0 | F9 P9 Y9 B9 M9 H9 R9 S9 T9 Q9 L9 V9  F9 P9 Y9 B9 M9 H9 R9 S9 T9 Q9 L9 V9 |

When rules #1, #2 & #3 apply

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Flt.No** | **Org** | **Dest.** | **Stops** | **Availability** |
| 1 | 100 | LHR | DEL | 0 | F9 P9 Y9 B9 M9 HC R9 SC ~~T9 Q9 L9 V9~~ |
| 2 | 110 | LHR | DEL | 1 | F9 P9 Y9 B9 M9 HC R9 SC T9 Q9 ~~L9 V9~~ |
| 3 | 200  210 | LHR  FRA | FRA  DEL | 0  0 | F9 P9 Y9 B9 M9 H9 R9 S9 T9 Q9 L9 V9  F9 P9 Y9 B9 M9 H9 R9 S9 T9 Q9 L9 V9 |

~~T9 Q9~~ means that RBD is not visible

When all rules apply (incl. #4)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Flt.No** | **Org** | **Dest.** | **Stops** | **Availability** |
| 1 | 100 | LHR | DEL | 0 | F9 P9 Y9 B9 M9 HC R9 SC ~~T9 Q9 L9 V9~~ |
| 2 | 110 | LHR | DEL | 1 | F9 P9 Y9 B9 M9 HC R9 SC T9 Q9 ~~L9 V9~~ |
| 3 | 200  210 | LHR  FRA | FRA  DEL | 0  0 | ~~F9 P9~~ Y9 B9 M9 H9 R9 S9 T9 Q9 L9 V9  ~~F9 P9~~ Y9 B9 M9 H9 R9 S9 T9 Q9 L9 V9 |

Note: ~~F9 P9~~ is not visible for all POS. But T, Q, L & V is not visible only for **Ireland POS**

**Analysis**

|  |  |  |
| --- | --- | --- |
| Flight options | Rules that apply | Effect on availability |
| 100 (Direct & non-stop) | #1, #2 & #3 | **Hide RBDs**  - T, Q (since this is a non-stop flight) &  - L, V (since this is also a direct flight)  **Inhibit Sell**  - H, S (since this is a direct flight) |
| 110 (Direct) | #1 & #2 | **Hide RBDs**  - L, V (since this is a direct flight, *but not non-stop*)  **Inhibit Sell**  - H, S (since this is a direct flight) |
| 200+210 (Connection) | #4 | **Hide RBDs**  - Cabin F ie F,P RBDs (since this is a connection) |

### Non Functional Requirements

The impact on availability performance should be less than 1 ms per segment if there are applicable availability and sell business rules.

# AVAIL.24g – Synchronize Segment Notes

### Summary

Segment Note is a subscriber-specific free text which can be included into availability or sell response for segments of host marketed flights. It can provide any kind of extra information for the recipient, for example about codeshare flight, traffic restrictions, group booking restrictions, wet/dry lease, etc. Today segment notes are not widely used in the mainframe.

Segment notes will be managed in Sales & Services, and they will also apply segment notes to sell responses. However HIAS needs to apply them to availability response and will fetch them from SnS. The cache will be kept up to date using a synchronization framework based on receiving a trigger from the Notification Broker.

SnS user will define scope for application of a segment note by specifying range of flight numbers and/or travel dates and/or arrival/departure city-pairs. For detailed description of fields please refer to S&S “UC for Manage Flight Segment notes” [1].

This story uses the following terminology:

|  |  |
| --- | --- |
| Scheduled Segment | Active segment of standard or ad hoc schedules, defined for a schedule period. Cached in Active Scheduled Segments Coherence cache within Operational WLS. |
| Operational Segment | ICR segments returned in availability response, flown on specific date. |

*Note: This task has nothing to do with the flows mentioned in this UC and therefore we recommend offshore do not refer to any of its flows, while delivering this task*.

### Detail

#### Synchronization of Segment Notes with SnS

HIAS will maintain distributed Coherence *SegmentNotesCache* in Operational Weblogic cluster to be used by *InventoryEnquirer* when applying notes to availability (which is a subject of separate story AVAIL.24h). Segment notes are subscriber-specific; however one cache will be used for all subscribers. The cache will be kept up to date using a synchronization framework based on receiving a trigger from the Notification Broker. HIAS receives these notifications by subscribing to the Notification Broker’s three topics:

* IAS/SEGNOTECREATEEVENT
* IAS/SEGNOTEUPDEVENT
* IAS/SEGNOTEDELEVENT

Since generation of Availability response is a performance-critical operation, **read-through must not be used**. Instead the cache should work in asynchronous refresh mode with refresh being triggered by a dedicated task in Scheduled Jobs Manager (SJM), which in turn will trigger *OperationalTrigger* operation of HIAS *FlightInventoryService* with Topic=SegmentNotes and Operation=UpdateCache. Cache will not expire.

When the *FlightInventoryService* receives a request to update the *SegmentNotesCache*, it should invoke the *FindFlightSegmentNote* operation of the *Manage* SnS service (for *FlightSegmentNoteManager*) to retrieve the segment notes.

* There is a concept of “Display Options” in S&S segment notes which denotes the type of transaction to which a particular segment note should be applied. HIAS will only apply segment notes with display option for “AVAD-Availability Display” or “PAOR-PAORES Response”, therefore only notes with such display options need to be extracted from SnS. To find the segment notes with that particular option set the *FindFlightSegmentNote* should be used.
* HIAS will request SnS to provide notes which have been changed, added or deleted since the last update. Only the data which got changed is required to be refreshed in cache.

#### Populating Cache on System Startup

Given scarcity of notes, Active Scheduled Segments Cache in Operational WLS must be populated prior to populating *SegmentNotesCache* during the system startup. This is to minimise the number of requests for matching a note with a scheduled segment – see below. As part of startup each Weblogic instance must attach to the cache.

#### Pre-caching Segment Notes

It is anticipated that subscriber-managed segment notes will be sparse across flights. In order to take advantage of this assumption and minimise impact on availability performance, the system should not check each and every **operational segment** (ICR segment flying on a specific date) for applicability of a segment note when responding to availability request.

Instead, a new *SegmentNoteInd* indicator needs to be added to *ActiveScheduledSegments* Coherence cache within operational WLS cluster. This indicator should be set to True for a **host** **scheduled segment** if there is at least one flight date matching at least one segment note from *SegmentNotesCache*. This indicator must only exist in cache and must not be persisted in Schedules DB.

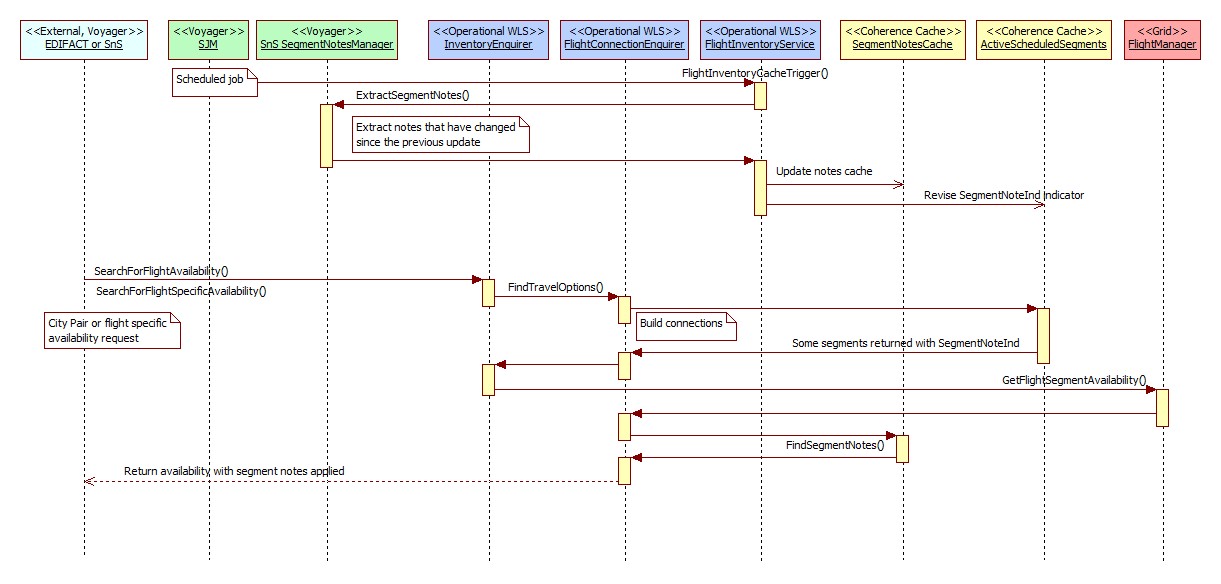
The following paragraph is for information only and outlines application of segment notes which will be developed as a separate story AVAIL.24h.

When returning travel options to *InventoryEnquirer, FlightConnectionEnquirer* will indicate those built from **scheduled segments** with *SegmentNoteInd*. In turn, *InventoryEnquirer* prior to **returning** availability for such travel options (and only for them!) will invoke internal lookup interface of *SegmentNotesCache* for notes matching the **operational segments** of the returned travel options. This way the impact on availability performance is kept to minimum.

Please note that memory footprint within Weblogic must be kept to a minimum, hence *SegmentNoteInd* has to be Boolean. In theory HIAS could store IDs of all notes applicable to scheduled segment and thus reduce lookup scope when applying notes to operational segments in availability response. However, it is deemed that given the scarcity of notes, the memory considerations will outweigh this potential benefit.

#### Sequence Diagram

The upper part of the diagram below presents the high level picture of AVAIL.24g story and the lower part illustrates AVAIL.24h.

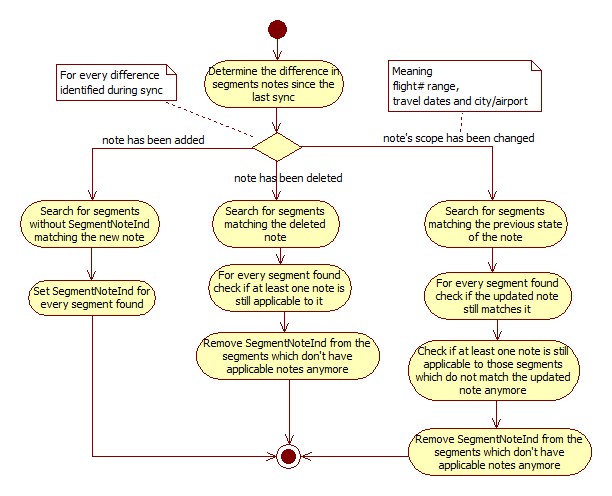


#### Updating *ScheduledSegmentsCache* during Synchronization of Segment Notes

During propagation of new Active host Schedule Segments into Operational WLS cache, the system should search within *SegmentNotesCache* for matching segment notes and set *SegmentNoteInd* if there is at least one match. For matching rules please refer to “Matching Notes with Segments” section below.

Given the scarcity of notes, during population of *SegmentNotesCache* on the system startup HIAS should, on the contrary, search for cached segments matching every cached note and mark those matching at least one note with *SegmentNoteInd*. This is why the cache of notes should be populated after the cache of segments in the startup sequence. Prior to initial population of Segment Notes cache all the existing *SegmentNoteInd* indicators should be cleared.

Likewise, during synchronization of segment notes HIAS should identify which segment notes have been added, modified (**note text** changes are not considered for the purposes of this comparison!) or deleted and then revise cached schedule segments in accordance with the diagram below.



#### Matching Notes with Segments

ALL of the following conditions must be met in order for a note to match a segment or vice versa (blank elements of a note’s scope should be considered as applicable to all segments). The same logic is used for operational and scheduled segments.

* Marketing airline code of the segment matches the Note’s airline code.
* Schedule Period of the segment intersects with the Note’s date range, and there is at least one operational **departure** within the intersection (one operational **arrival** – in case **only** Offpoint Station is specified within City/Airport).

If finding notes for operational segments during application to availability, actual departure/arrival date of a segment should be matched against the Note’s date range.

For example, if a flight segment operates on Mondays, Wednesdays and Fridays between 1st and 30th of June 2015 departing from JFK and arriving to LHR the next day:

JFK(01.06)-LHR(02.06)  
JFK(03.06)-LHR(04.06)  
JFK(05.06)-LHR(06.06)  
…

The following notes would be applicable:  
Travel Date 2-4, Boarding at JFK, Offpoint LHR (because there is departure on 3rd)  
Travel Date 1, Boarding at JFK, Offpoint not specified (because there is departure on 1st)  
Travel Date 1-30, Arrival into LHR (because there are numerous arrivals within this range)  
  
And the following note would not be applicable:  
Travel Date 3, Arrival into LHR (because there are no arrivals on 3rd)

* Note’s Flight Number (whether it is a range or a single number) matches the segment’s flight number (matching of flight number suffixes may be a subject of future story)
* Note’s Boarding City/Airport matches departure station of the segment’s first leg (Locations Manager data should be used to match cities with airports – see story ARCH.19)
* Note’s Offpoint City/Airport matches arrival station of the segment’s last leg.

This logic should be used in both directions: for finding segments matching a note as well as for finding notes matching a segment.

#### Assumptions

* SnS will apply segment notes to sell responses.
* SnS will be able to return only segment notes that have changed since the last update in order to minimise payload. (Including notes that have been deleted, which can be flagged as inactive.) Otherwise HIAS would have to determine the delta between the existing cache and the entire set of notes returned by SnS.
* Auto-generated (“canned”) segment notes will not be applied by HIAS on the grounds that availability response already contains the required information (codeshare operating airline and traffic restriction). Transformation of these data elements into human-readable text presentation (IFT) will be addressed in future stories.

### Acceptance

Definition of done requires that all test scenarios are satisfied; furthermore it is expected that the development team will identify additional scenarios to add to the test pack.

Integration with actual SnS service is part of definition of done, however to demonstrate some of the acceptance tests below mock service can be used. Test data in tables below or similar should be used for acceptance testing.

**Table 1**: Segment Notes initially existing in SnS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Flight number** | **Travel date** | **City-Pair** | **Boarding City/Airport** | **Offpoint City/Airport** | **Usage option** |
| N1 | 1300-1499 | - |  | - | - | PAORES Response |
| N2 | 1400-1499 | - |  | - | JFK | Availability display |
| N3 | 100-200 | 01.10.2015-03.10.2015 |  | - | LHR | Availability display |
| N4 | 148 | - |  | - | - | PAORES Response |
| N5 | 1-9999 | - |  | - | - | Agent Sell Response, ITARES Response |
| N6 | - | - | KUL-LHR | - | - | Availability display |

**Table 2**: Schedules in HIAS DB (active status and weekly frequency is assumed for all):

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Flight #** | **Origin** | **Destination** | **Start Date** | **End Date** | **Arrival date variation** | **DOW** | **SegmentNoteInd as set in Cache after startup** |
| S1 | 1330 | ATL | JFK | 01.09.2015 | 31.10.2015 | - | 1234567 | True (matches note #1) |
| S2 | 1480 | ORD | JFK | 01.09.2015 | 31.10.2015 | - | 1234567 | True (matches notes #1 and #2) |
| S3 | 144 | JFK | LHR | 01.09.2015 | 31.10.2015 | +1 | 1030500 | True (matches note #3) |
| S4 | 146 | JFK | LHR | 01.09.2015 | 31.10.2015 | +1 | 0204060 | True (matches note #3) |
| S5 | 148 | JFK | LHR | 01.09.2015 | 31.10.2015 | +1 | 0204060 | True (matches notes #3 and #4) |
| S6 | 329 | KUL | LHR | 01.09.2015 | 31.10.2015 | - | 1234567 | True (marches note #6) |
| S7 | 432 | PKC | SVO | 01.10.2015 | 31.10.2015 | - | 0000067 | - |

**Note: MCT should allow connecting S2 to S3 and S1 to S5**.

| **Nr.** | **Test** | **Pre-condition** | **Action** | **Post-condition** |
| --- | --- | --- | --- | --- |
|  | Population of SegmentNotes cache on startup | The system is not running. Both Active Scheduled Segment and Segment Notes caches are empty.  Segment notes from Table 1 above exist in SnS, and their service is up and running. Schedule periods from the Table 2 above exist in HIAS database. | Initialize startup sequence. | Both Active Scheduled Segments cache and Segment Notes cache are populated in accordance with table above.  Note #N5 is not applicable to HIAS and is NOT synchronized.  *SegmentNoteInd* is set for some segments in accordance with Table 2.  Need to demonstrate that segment notes with XML markup and other special characters do not cause any issues during synchronization (i.e. </> and other characters from SnS UC [1] section 15.2 must be properly escaped where needed) |
|  | Delete segment note | Test #1 has been completed.  For the purposes of Demo cache refresh interval should be short. | Delete note #N1. | The following impact can be observed on Segment Notes cache after the cache update has been triggered by SJM task:  Segment #S2 retains *SegmentNoteInd*, while segment #S1 no longer has it. |
|  | Modify segment note | Test #1 has been completed. | Change Travel Date of note #N3 to 03.10.2015-04.10.2015 | Segments #S4 and #S5 retain *SegmentNoteInd*, while segment #S3 no longer has it. |
|  | Add segment note | Test #1 has been completed. | Add a new segment note with the following details:  Flight #: 100-400  Travel date: 01.10.2015-05.10.2015  Usage option: PAORES Response | *SegmentNoteInd* is set for segment #S5 and is not set for segments #S3, #S4 & #S6 |
|  | End to end test | Segment note(s) exist in SnS but not all in HIAS. | HAIS receive a trigger from Notification Broker that new segment notes are available | Segments notes in HIAS are synched with those in SnS both showing the same population |

# AVAIL.24h – Apply segment notes to availability response

### Overview

This story describes application of cached segment notes to the availability response covering Sub flow 3 of this use case. Please refer to AVAIL.24g “Synchronize segment notes” story description for general description of segment notes and their caching. AVAIL.24g story will be developed prior to AVAIL.24h.

### Detail

#### Finding Segment Notes

When returning travel options to *InventoryEnquirer*, *FlightConnectionEnquirer* will indicate **scheduled segments** that can potentially have a segment note applied by setting the *SegmentNoteInd*. In turn, *InventoryEnquirer* prior to returning availability for such travel options should invoke internal lookup interface of *SegmentNotesCache* for all notes matching the **operational segments** of the returned travel options. For matching rules please refer to “Matching Notes with Segments” section of AVAIL.24g story.

IAS is concerned with segment notes of two usage options: “Availability Display” (for host availability requests) and “PAORES Response” (for all other availability requests). This should be done independently for every segment and O&D in the response (as the response may contain segment marketed by different carriers).

#### Returning Segment Notes

*InventoryEnquirer* should include text of matching segment notes into availability response for corresponding segments within the following element:

*tns:ShowFlightAvailability/tns:OriginDestination/inv:ODOptions/ inv:ODOption/inv:Segment/inv:SegmentNote*

Messaging will transform this XML element into IFT (Interactive Free Text) element of EDIFACT PAORES (subject of future stories).

More than one segment note can be applicable to a segment. The schema allows the maximum of 10 segment notes to be returned, which should be sufficient. However if more than 10 notes are applicable, then first 10 should be returned. Currently there are no requirements for ordering them within availability response.

#### Skipping Segment Notes

The requestor will be able to ask not to include segment notes into availability response. If the following element on the request is set to True, then InventoryEnquirer should not attempt to find matching segment notes:  
*tns:SearchForFlightAvailability/tns:SearchInfo/inv:OmitSegmentNotesInd* (for City Pair availability)

*tns:SearchForFlightSpecificAvailability/tns:SearchInfo/inv:OmitSegmentNotesInd* (for Flight-specific)

#### Assumptions

* More than one segment note can be applicable to an operational segment.

### Acceptance Tests

Definition of done requires that all test scenarios are satisfied; furthermore it is expected that the development team will identify additional scenarios to add to the test pack. See acceptance tests of story AVAIL.24g for test data referenced below.

| **Nr.** | **Test** | **Pre-condition** | **Action** | **Post-condition** |
| --- | --- | --- | --- | --- |
|  | Return Segment Notes in response to Host availability request | Test #1 from story AVAIL.24g has been completed. | Perform **host** availability requests:  - city-pair for ORD-LHR on 02.10.2015  - flight-specific for JFK-LHR on 29.09.2015  - city-pair for KUL-LHR on 01.10.2015 | For city-pair request note #N2 is returned for ORD-JFK segment (flight 1480) and note #N3 for JFK-LHR segment (flight 144).  For flight-specific request flights 146 and 148 are returned with no notes (note #N3 is not effective on 29.09.2015)  For city-pair request on KUL-LHR,note #N6 is returned (flight 329).  Need to demonstrate that the system is capable of returning segment notes with XML markup and other special characters (e.g. </> % ) |
|  | Return Segment Notes in response to regular availability request | Test #1 from story AVAIL.24g has been completed. | Perform **regular** availability request (city-pair) for ATL-LHR on 03.10.2015 | Note #N1 is returned for ATL-JFK segment (flight 1330) and note #N4 is returned for JFK-LHR segment (flight 148). |
|  | Request to omit segment notes | Test #1 from story AVAIL.24g has been completed. | Same as in test A2 above, but set *OmitSegmentNotesInd* to True | Connection of flight 1330 and 148 is returned as in test A2 above, but without segment notes. |
|  | Cache is not present. | Test #1 from story AVAIL.24g had been completed.  Segment Notes cache was not loaded. | Same as in test A2 above. | Connection of flight 1330 and 148 is returned as in test A2 above, but without segment notes. |
|  | Return Segment Notes in response to Host +OA segment availability request | Test #1 from story AVAIL.24g has been completed. | Request availability on a Host + OA segment (which contains segment notes) | Segments notes are returned for both Host segment & OA segments in the availability response. |

### Non Functional Requirements

The impact on availability performance should be less than 1 ms per segment if there are up to 10 applicable segment notes,

There will be no recordable impact to response time for segments without *SegmentNoteInd*.

There will be no recordable impact to response time for requests with *OmitSegmentNotesInd.*

# AVAIL.24b – Apply POO (Point-of-Origin) restrictions to Availability & Sell

### Overview

HIAS currently returns availability response based on the POS. However point-of-origin (POO) restrictions are not applied to the responses. This task enables applying POO restriction to availability and inventory adjustment responses after evaluating the entire journey data that comes in the request.

Point of origin refers to the first departure station of a passenger’s journey (itinerary), irrespective of which airline they are travelling.

### Detail

When HIAS receives an availability request from internal or external source*,* the POS data is included in the request (*SearchForFlightAvailability/* ***POS*** or *SearchForFlightSpecificAvailability/* ***POS***). The POO information has to be inferred from the journey data (provided in *BookedSegments*). It is the departure airport of the first segment in the itinerary. If no booked segment data is given then origin station of request is considered as the POO (*OriginDestinationList/* ***OriginCode***). Again first origin code must be taken as POO when multiple *OriginDestinationList* exists in request. It is assumed that the request is always according to order of itinerary.

For Sell requests, the POS & POO information are provided in the *AdjustInventory* request (AdjustInventory/**POS**, AdjustInventory/**BookedSegments**/Segment/DepartureAirport & AdjustInventory/**Segments**/ Segment/ DepartureAirport). To determine POO, same logic as above applies.

POO restrictions are specified in the POS template (*POSTemplate/POSItems/POSItem/****POOLocation***). Every POS template is assigned to a Booking Limit bucket.

System must determine the POO of the request, and from that determine the POS/POO Template Groups that apply to the request/requestor and check whether any restrictions apply to it. If applicable then availability as per the corresponding booking limit bucket is returned in the response. If no restrictions apply then normal availability is returned. If POO restrictions apply on multiple booking limit buckets, then the most restricted availability is returned.

Likewise for Sell requests, depending on the POO (POS Template Group associated with the request) restrictions inventory adjustment in corresponding booking limit bucket takes place.

To deliver this task, existing process of constructing availability response for simulate availability is adequate. Booking limit sell functionality already exists and is addressed in INV.66f1 and INV.66a2.

**For all request - booking or group or waitlist, the restrictions have to be checked and applied.**

**POO\_Routing Subscriber Parameter**

A new subscriber parameter, POO\_Routing with values ‘Online’ & ‘Trip’ is introduced. The parameter is intended to identify if the Trip O&D (with OA segments) or the On-Line O&D (contiguous segments that is marketed by the host) is to be checked against the type value when determining if the booking limit applies.

#### Assumptions

* We assume booked segments will be included in the internal and external request that HIAS receives.
* Only valid connections will be received for processing availability and system will ignore/reject invalid connections prior to seeking availability.

### Acceptance Tests

Definition of done requires that all test scenarios are satisfied; furthermore it is expected that the development team will identify additional scenarios to add to the test pack. Demo has to be done through the GUI.

#### Test Data

Following schedules/ICRs in FY configuration exists in the system

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Type | Flight # | Route | RBDs |
| 1 | OA\* | BA100 | DUB-LHR | FYBM |
| 2 | Host operated | AI110 | LHR- DEL | FPYBM |
| 3 | Host marketed | AI9200 | DEL-KUL | FPYBM |

\* BA schedule is for representation purpose only and need not actually exist in HIAS.

Following POS for POO restrictions exist to ICRs.

|  |  |  |  |
| --- | --- | --- | --- |
| # | POS Group | Applied on Booking Limit\* | Assigned to (Dressed) |
| 1 | **Ireland POS:** POO location – IE (Country) | **Ireland BLB**: RBD- B, M; BAU - **4** seats | AI9200 |
| 2 | **UK POS:** POS location – GB (country) | **UK BLB**: RBD- Y, B, M; BAU - **8** seats | AI9200 |

\*values for GAU, WAU & MinAU assumed to be zero (0)

**Note:**

Assume passenger routing is **DUB-LHR-DEL-KUL**, then

1. ‘Trip’ check will include OA segments. Hence POO for ‘Trip’ is DUB.
2. ‘Online’ check is for Host segments only (incl.marketed). Hence POO for ‘Online’ is LHR.

#### Availability tests

| **Sr.** | **Test** | **Prerequisite** | **Scenario** | **Post Condition** |
| --- | --- | --- | --- | --- |
|  | Availability Search where **POO rule apply** | ICRs as per above test data exists. POO\_Routing – **Trip** | **UK POS user** makes a request to seek availability for AI9200. (**Note**: Request includes ‘Booked segments’ on BA100 and AI110 i.e. itinerary is DUB-LHR-DEL-KUL) | System returns availability on AI9200 – F9 P9 **Y8 B4 M4** (most restricted)\*  Note: Here both rules apply and the most restricted between them is returned |
|  | Availability Search where **POO rule** **does not apply** | ICRs as per above test data exists. POO\_Routing – **Trip** | **UK POS user** makes a request to seek availability for AI9200. (**Note**: No booked segments included) | System returns availability on AI9200 - F9 P9 **Y8 B8 M8** \*  Note: Here rule #1 does not apply. Only rule #2 applies. |
|  | Availability Search where **POO rule apply** | ICRs as per above test data exists. POO\_Routing – **Trip** | **Normal user** makes a request to seek availability for AI9200. (**Note**: Request includes ‘Booked segments’ on BA100 and AI110) | System returns availability on AI9200 – F9 P9 Y9 **B4 M4**\*  Note: Here only ‘Ireland POS’ rule (#1) applies |
|  | Availability Search where **No rule**s apply | ICRs as per above test data exists. POO\_Routing – **Trip** | **Normal user** makes a request to seek availability for AI9200. (**Note**: No booked segments included) | System returns normal availability on AI9200 for **all RBDs (9 seats)**  Note: No rule applies |
|  | Availability Search by different Subscriber Parameter (POO rule doesn’t apply) | ICRs as per above test data exists. POO\_Routing – **Online** | **UK POS user** makes a request to seek availability for AI9200. (**Note**: Request includes ‘Booked segments’ on BA100 and AI110) | System returns availability on AI9200 - F9 P9 **Y8 B8 M8**\*  Note: Here rule #1 does not apply since OA segment is not included in check. Only rule #2 applies |
|  | Availability Search by different Subscriber Parameter (No rule applies) | ICRs as per above test data exists. POO\_Routing – **Online** | **Normal user** makes a request to seek availability for AI9200. (**Note**: Request includes ‘Booked segments’ on BA100 and AI110) | System returns normal availability on AI9200 for **all RBDs (9 seats)**  Note: No rule applies, since OA segment is not included in check |
|  | Group & Waitlist (dependent on Avail.26) | As above. Test for both subscriber parameters | User makes a request for group or waitlist availability | Systems returns groups and waitlist availability based on POO restriction |

#### \* Other RBDs are displayed with 9 seats (since no restrictions apply on them)

#### Inventory Adjustment (Sell) tests

| **Sr.** | **Test** | **Prerequisite** | **Scenario** | **Post Condition** |
| --- | --- | --- | --- | --- |
|  | Adjust Inventory where **POO rule apply** | ICRs as per above test data exists. POO\_Routing – **Trip** | **UK POS user** makes a sell request on AI9200. (**Note**: Request includes ‘Booked segments’ on BA100 and AI110 i.e. itinerary is DUB-LHR-DEL-KUL) | System permits inventory adjustment on AI9200 (for B & M for request up to **4 seats,** based on most restricted BLB)  Note: Both rules apply and sell is allowed until corresponding BL SSC=4. For other RBDs, system permits normal sell. |
|  | Inventory adjust where **POO rule** **does not apply** | ICRs as per above test data exists. POO\_Routing – **Trip** | **UK POS user** makes a sell request on AI9200. (**Note**: No booked segments included) | System permits inventory adjustment on AI9200 (for Y, B & M for requests up to **8 seats**)  Note: Here only rule #2 applies. Sell request is allowed until its BL SSC=8. For other RBDs, system permits normal sell.. |
|  | Inventory adjust where **POO rule apply** | ICRs as per above test data exists. POO\_Routing – **Trip** | **Normal user** makes a sell request on AI9200. (**Note**: Request includes ‘Booked segments’ on BA100 and AI110) | System permits inventory adjustment on AI9200 (for B & M for request up to **4 seats**)  Note: Only #1 rules applies and sell is allowed until its BL SSC=4. For other RBDs, system permits normal sell. |
|  | Inventory adjust where **No rule**s apply | ICRs as per above test data exists. POO\_Routing – **Trip** | **Normal user** makes a sell request on AI9200. (**Note**: No booked segments included) | System permits inventory adjustment on AI9200 for **all RBDs**  Note: No rule applies |
|  | Inventory adjust by different Subscriber Parameter (POO rule doesn’t apply) | ICRs as per above test data exists. POO\_Routing – **Online** | UK POS user makes a sell request on AI9200. (**Note**: Request includes ‘Booked segments’ on BA100 and AI110) | System permits inventory adjustment on AI9200 (for Y, B & M for requests up to **8 seats**)  Note: Here rule #1 does not apply since OA segment is not included in check. Only rule #2 applies. |
|  | Inventory adjust by different Subscriber Parameter (No rule applies) | ICRs as per above test data exists. POO\_Routing – **Online** | Normal user makes a sell request on AI9200. (**Note**: Request includes ‘Booked segments’ on BA100 and AI110) | System permits inventory adjustment on AI9200 for **all RBDs**  Note: No rule applies |
|  | Group & Waitlist (dependent on Avail.26) | As above. Test for both subscriber parameters | User makes group or waitlist sell request | Systems actions sell request for groups or waitlist based on POO restriction |

### Non Functional Requirements

The impact on availability performance should be less than 1 ms per segment.

# AVAIL.24c – Apply Upline/Downline restrictions to Availability & Sell

### Summary

Upline / Downline restrictions are limitations that apply to a segment based on the stations (legs) it comprises. They are defined in the Booking Limit template. This task enables applying upline / downline restrictions to availability and inventory adjustment request after evaluating the journey data.

### Background

Upline / downline restrictions are explained through the route network diagrams below.

**Upline Downline**

On FRA-DEL flight, **limited seats** are offered to passengers who connect from upline point JFK and likewise **limited seats** are offered to passengers who travel to downline point HKG.

**Rule:** Departure station can be an upline point, but not a downline point (eg: LHR, CDG, JFK, ATL & **FRA** are upline points for FRA-DEL). Arrival station can be a downline point, but not an upline point (eg: HKG, KUL, SIN, SYD & **DEL** are downline points for FRA-DEL). Assigning booking limits to flights as per above rule is part of training issue.

Note: These restrictions could apply to all requests (ie every POS) or can apply to specific POS groups, if defined in the booking limit template.

#### Subscriber Parameter

This story introduces a new subscriber parameter ‘UpLine\_DownLine\_Station’ with values **‘Flight’** & **‘Routing’**. For ‘Flight’, request is compared to stations in the segment (including legs that make up the segment) and for ‘Routing’ it is compared to stations in any segment in the routings (entire journey).

How upline / downline restrictions apply to availability / sell request is illustrated in the below example. Assume following schedules exist for an airline as per above routing

|  |  |  |  |
| --- | --- | --- | --- |
| # | Flight # | Route | RBDs |
| 1 | AI-9100\* | LHR-FRA | FPYBM |
| 2 | AI-9101\* | CDG-FRA | FPYBM |
| 3 | AI-9102\* | JFK-FRA | FPYBM |
| 4 | AI-9103\* | ATL-FRA | FPYBM |
| 5 | AI-110 | FRA- DEL | FPYBM |
| 6 | AI-201 | DEL-HKG | FPYBM |
| 7 | AI-202 | DEL-KUL | FPYBM |
| 8 | AI-203 | DEL-SIN | FPYBM |
| 9 | AI-204 | DEL-SYD | FPYBM |

\*code share flights

And the booking limit templates that exist are below:

|  |  |  |  |
| --- | --- | --- | --- |
| # | Booking Limit Temp | Restriction | Assigned to all Leg/Seg (dressed) |
| 1 | **JFK BLB:** Upline – JFK (Airport) | RBD- Y, B, M; BAU - **2** seats | AI-110 (FRA-DEL) |
| 2 | **Germany BLB:** Upline – DE (Country) | RBD- Y, B, M; BAU - **8** seats | AI-110 (FRA-DEL) & AI-201 (DEL-HKG) |
| 3 | **India BLB:** Downline – IN (Country) | RBD- B, M; BAU - **4** seats | AI-110 (FRA-DEL) & AI-9102 (JFK-FRA) |
| 4 | **Hong Kong BLB:** Downline – HK (Country) | RBD- B, M; BAU - **3** seats | AI-110 (FRA-DEL) |

Restrictions will apply **only** on the following Availability and Sell city-pair request.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Segment request (City pair)** | **Flight (Leg/Seg)** | **Flight** | | **Routing** | |
| **Y/N** | **BLB** | **Y/N** | **BLB** |
| **JFK – DEL (JFK-FRA + FRA-DEL)** | AI9102/ JFK-FRA (both) | N | - | Y | #3 |
| AI-110/ FRA-DEL (both) | Y | #2, #3 | Y | #1, #2, #3 |
| **JFK – HKG (JFK-FRA + FRA-DEL + DEL-HKG)** | AI9102/ JFK-FRA (both) | N | - | Y | #3 |
| AI-110/ FRA-DEL (both) | Y | #2, #3 | Y | #1, #2, #3, #4 |
| AI-201/ DEL-HKG (both) | N | - | Y | #2 |
| **FRA – HKG (FRA-DEL + DEL-HKG)** | AI-110/ FRA-DEL (both) | Y | #2, #3 | Y | #2, #3, #4 |
| AI-201/ DEL-HKG (both) | N | - | Y | #2 |
| **All other city pair request that include FRA – DEL segment** | AI-110/ FRA-DEL (Both) | Y | #2, #3 | Y | #2, #3 |

### Detail

Journey data is included in the availability and sell request (provided in *Segments* or *BookedSegments* or *OriginDestinationList*) that HIAS receives. System must determine the origin in upline routing and/or destination in downline routing from the request data and check whether any restrictions apply to it (*tns:ShowBookingLimit/tns:BookingLimitTemplate/ inv:BookingLimitItems/ inv:BookingLimitItem/inv:UplineCondition or DownLineCondition*) . If applicable then availability as per the corresponding booking limit bucket is returned in the response. If no restrictions apply then normal availability is returned. In case restrictions apply on multiple booking limit buckets, then the most restricted availability is returned.

Likewise for Sell requests, depending on the restrictions inventory adjustment in corresponding booking limit bucket takes place.

To deliver this task, existing process of constructing availability response for simulate availability is adequate. Booking limit sell functionality already exists (AVAIL.24 & AVAIL.24d) and is further addressed in INV.66f1 and INV.66a2.

**For all request - booking or group or waitlist, the restrictions need to be checked and applied.**

### Acceptance Tests

Definition of done requires that all test scenarios are satisfied; furthermore it is expected that the development team will identify additional scenarios to add to the test pack. Demo has to be done through the GUI.

**Test data provided in section (§15.6.2.1) must be used.** POS template as below exists.

|  |  |  |  |
| --- | --- | --- | --- |
| # | POS Group | Applied on Booking Limit | Exclude/Include |
| 1 | **BA POS:** POS channel – BA (Airline) | **Germany BLB (#2) & India BLB (#3)** | Include |

#### Availability tests

| **Sr.** | **Test** | **Prerequisite** | **Scenario** | **Post Condition** |
| --- | --- | --- | --- | --- |
|  | Availability Search where **restrictions apply** | ICRs with BLBs as per above test data exists. Upline\_Downline – **Flight** | HIAS receives a availability request for city-pair JFK-HKG | * Systems returns all RBDs for AI9102 (since no restrictions apply within this Subs.parameter) * System displays most restrictive availability for AI110 – F9 P9 **Y8 B4, M4** \* (rules #2, #3 apply) * Systems returns all RBD for AI201 (since no restrictions apply within this Subs.parameter) |
|  | Availability Search where **restrictions apply** | ICRs with BLBs as per above test data exists. Upline\_Downline – **Routing** | HIAS receives a availability request for city-pair JFK-HKG | * Systems returns availability for AI9102 – F9 P9 Y9 **B4 M4** \* (since BLB#3 restrictions apply within this Subs.parameter) * System displays most restrictive availability for AI110 –F9 P9 **Y2 B2 M2** \* (all rules apply) * Systems returns availability for AI201 – F9 P9 **Y8 B8 M8** \* (since BLB#2 restrictions apply within this Subs.parameter) |
|  | Availability Search where **restrictions apply for POS** | Add rule to limit restrictions for only certain POS (eg:Airline BA). Re-dressed ICRs with BLBs as per above rule exists. Upline\_Downline – **Flight** | HIAS receives a availability request for city-pair JFK-HKG from this POS | * Systems returns all RBDs for AI9102 (since no restrictions apply for this POS within this Subs.parameter) * System displays most restrictive availability for AI110 – F9 P9 **Y8 B4 M4** \* (rules #2, #3 apply) * Systems returns all RBD for AI201 (since no restrictions apply for this POS within this Subs.parameter) |
|  | Availability Search where **restrictions apply for POS** | Add rule to limit restrictions for only certain POS (eg:Airline BA). Re-dressed ICRs with BLBs as per above rule exists. Upline\_Downline – **Routing** | HIAS receives a availability request for city-pair JFK-HKG from this POS | * Systems returns availability for AI9102 - F9 P9 Y9 **B4 M4** \* (since BLB#3 restrictions apply for this POS within this Subs.parameter) * System displays most restrictive availability for AI110 – F9 P9 **Y8 B4 M4** \* (rules #2, #3 apply) * Systems returns availability for AI201 – F9 P9 **Y8, B8, M8** \* (since BLB#2 restrictions apply for this POS within this Subs.parameter) |
|  | Availability Search where **restrictions don’t apply** | Add rule to limit restrictions for only certain POS (eg:Airline BA). Re-dressed ICRs with BLBs as per above rule exists. Upline\_Downline – **Flight / Routing** | HIAS receives a availability request for city-pair FRA-DEL from normal POS | * All RBDs are displayed on AI-110 (since no restrictions apply to the normal POS) |
|  | Group & Waitlist (dependent on Avail.26) | As above. Test for both subscriber parameters | User seeks group or waitlist availability | Systems returns groups and waitlist availability based on upline / downline restrictions |

\*Other RBDs are displayed with 9 seats (since no restrictions apply on them)

#### Inventory Adjustment (Sell) tests

| **Sr.** | **Test** | **Prerequisite** | **Scenario** | **Post Condition** |
| --- | --- | --- | --- | --- |
|  | Inventory adjust where **restrictions apply** | ICRs with BLBs as per above test data exists. Upline\_Downline – **Flight** | HIAS receives a sell request for following segments:   * AI9102, AI110 & AI201 in **M class** for 6 seats (first request) * AI110 in **M class** – 4 seats (2nd request). Al9102 & AI201 bookings are included in booked segments | First Request  System permits inventory adjustment on AI9102 & AI201 in M class, but rejects request for AI110 (since as per most restrictions only 4 seats for M class is allowed)  2nd Request  System permits inventory adjustment of 4 seats for AI110 in M class. |
|  | Inventory adjust where **restrictions apply** | ICRs with BLBs as per above test data exists. Upline\_Downline – **Routing** | HIAS receives a sell request for following segments:   * AI9102, AI110 & AI201 in **M class** for 9 seats (first request) * AI9102, AI110 & AI201 in **M class** for 2 seats (2nd request) | First Request  System rejects all sell request for AI9102, AI110 & AI201 for M class (since as per restrictions only following seats are allowed in M class for each flight: AI9102- 4 seats, AI110-2 seats & AI201-8 seats)  2nd Request  System accepts all sell request & adjusts inventory for resp. flights as per request. |
|  | Inventory adjust where **restrictions don’t apply** | Add rule to limit restrictions for only certain POS (eg:Airline BA). Re-dressed ICRs with BLBs as per above rule exists. Upline\_Downline – **Flight or Routing** | HIAS receives a sell request from **normal** user for following segments:   * AI9102, AI110 & AI201 in **M class** – 5 seats | System accepts sell request & adjusts inventory in M class for flights AI9102 & AI201 & rejects on AI110 when **'Flight'** (since as per restrictions only 4 seats for M class is allowed)  System accepts sell request on AI201, but rejects sell request on AI9102 & AI110 when **'Routing'** (since as per restrictions only 4 seats for M class is allowed) |
|  | Group & Waitlist Sell (dependent on Avail.26) | As above. Test for both subscriber parameters | HIAS receives group or waitlist sell request | Systems actions sell request for groups or waitlist based on upline / downline restrictions |

### Non Functional Requirements

The impact on availability performance should be less than 1 ms per segment.