**NSI Connection Service v2.0 to v2.1 delta**

Copyright Notice

Copyright © Open Grid Forum (2008-2014). Some Rights Reserved. Distribution is unlimited.

# New features for v2.1

* Resource availability feedback
  + Return resource availability in *resvFailed* message
* Last modified for queries
  + Timestamp on incident (i.e. state change on reservation)
  + Enable query to ask for events ***ifModifiedSince*** based on incident timestamps
* Error code updates
* Errata updates
  + No schema changes
  + Clarifications
  + Minor behavioral changes
* New discussions
  + ERO exclusions
  + pathTrace (?)

# Differences between v2.0 and v2.1

* Completely backward compatible
  + No schema recompilation
  + Resource availability feedback
    - Use *serviceException variables* ***variable*** (optional) element
    - Ref: OGF\_Resource\_Selection\_v2-jhm.ppt
    - String based, more verbose and rigid
  + Last modified queries
    - Use NSI-header to put ***ifModifiedSince*** (optional) element for request
    - Use NSI-header to put ***lastModified*** (optional) element for response
* Schema recompile with backwards compatible behavior
  + No change to name space
  + Cleaner implementation
  + Resource availability feedback
    - Create *serviceException variables* ***feedback*** (optional) element
    - Ref: “Error Handling” email sent 30 June 2015 from John MacAuley ([macauley@es.net](mailto:macauley@es.net))
    - Provides more flexibility to provide feedback (e.g. integer, EROs, etc)
  + Last modified queries
    - Add ***ifModifiedSince*** element to *QueryType*
    - Add ***lastModified*** element to *QuerySummaryConfirmedType*, *QueryRecursiveConfirmedType*

# Specific Changes

## Section 3.2 Explicit Routing Object

The following sentence has been added to the second paragraph of section 3.2

“Also note that STP at either end of an SDP can be used to uniquely identify the SDP to transit. Both STPs in a single SDP are not required in the ERO, and in fact, only a single one should be specified.”

## Section 4.3.1 Reservation State Machine

The Reservation State Machine has been modified to allow the rsvTimeout to be propogated by an Aggregator. The text has been revised to clarify the operation of the RSM.

## Section 4.3.1 Change to Explanation of RSM

The following text has been added to this section:

While a reservation is being modified the <*reservationState*> reflects the current state of the modification even though the <criteria> represents the last committed version.

## Section 4.5 Provisioning Sequence

In section 4.5 diagrams 7 and 8 have been updated to remove the ambiguity aruound the term "In service". This could have been assumed to mean the dataplane is activated. However the intent (with the decoupling of the PSM and dataplane activation) is to indicate that the "In service" is a primer for the dataplane to be activated (if it wasn't already so).

## Section 5.3.2 Message checks.

A diagram has been added to section 5.3.2 to show an Example of SOAP fault translation to NSI failed message.

## Section 6.1.3: Correlation Ids and Failure Recovery

The aggregator has been amended to include an additional explanation of howThe *reserveFailed* message from child NSAs are to be aggregated before passing up the tree.  Change the coordinator functionality to get all *failed* messages back to the requester instead of just the first one as defined in the pseudo code now.

## Section 6.1.5 Per reservation information elements.

The following bullet has been added to this section: If an RA receives a Connection request with a *startTime* in the past, this should be treated as ‘now’. The RA should not change the *startTime* and keep it as part of the record of the reservation

## Section 8.4.9 text removed

The following text has been removed from Section 8.4.10 dataPlaneStateChange message elements:

"The originating connectionId and uPA are provided in separate elements to maintain the original context generating the data plane state change. The timeStamp is populated by the originating PA and propagated up the tree untouched by intermediate NSA."

Explanation: This text problem was caused by the author copying the text from section 8.4.9 (errorEvent) into section 8.4.10 then modifying.  The quoted text has been removed as it is not applicable to the dataPlaneStateChange.

## Section 8.5.1.16 QueryFailedTyperemoved

The *QueryFailedType* type is not relevant, this was left in by mistake. The following section has been removed: 8.5.1.16 QueryFailedType

## Section 8.5.1.30: updated versioning number to start with 1

Sections 6.1.6 and 8.5.1.30 are currently in conflict.  I believe I wrote 8.5.1.30 when we were using "0" as a special version number to show the currently uncommitted first reservation criteria.  This decision was later changed to what is described in 6.1.6.  We need to update the text in 8.5.1.30 to better reflect reality.

6.1.6 Reservation Versioning Information (page 26)

To support the modification of reservations, the notion of versioning has been introduced to identify the instance of a reservation over its lifetime.

* + Versioning MUST be used as follows:
  + Version numbers are integer values ≥ 0 (zero)
  + Version numbers are assigned by the RA when a reservation request (i.e. NSI\_rsv.rq) is made to a PA
  + If a version number is not specified in an NSI\_rsv.rq, it is assumed to be 0 (zero) regardless of whether the request is theinitial or a subsequentrequest.
  + An NSI\_rsv.rq with a version number ≤ the (highest) current committed reservation version number will result in a failed request and an appropriate error

8.5.1.30 ReservationRequestCriteriaType (page 89)

Type definition for a reservation and modification request criteria. Only those values requiring change are specified in the modify request. The version value specified in a reservation or modify request MUST follow versioning rules as defined in section 6.1.6.

## Appendix B: Error Messages and Best Practices

The error message details have been moved and now belong in a separate error codes document. The reasoning behind this is that we expect the error codes to change based on implementation experience. Error codes can be updated easily without needing to re-issue the CS document

## Appendix D: Formal Statement of Coordinator

**NSI\_rsv.fl(Conn\_ID, Corr\_ID)** /\* ***from child NSA*** \*/

if request\_list(Conn\_ID, Corr\_ID).Status != fail then

{

set request\_list(Conn\_ID, Corr\_ID).Status = fail

send res.fl(Corr\_ID, Ver) to RSM(Conn\_ID)

The aggregator description needs an additional explanation of howThe *reserveFailed* message from child NSAs are to be aggregated before passing up the tree.  Change the coordinator functionality to get all *failed* messages back to the requester instead of just the first one as defined in the pseudo code now.

## Appendix E: Service-Specific Schema

In Section 19.6 Reservation request, the example *reserve* request XML message for a bidirectional service has been updated.

## Appendix G: ERO

Appendix G has been added to bring in relevant text from the old document draft-gwdi-macauley-nsi\_ero-v03 to the CS document.