# NSI v2.0: Agreed Features, 9th October

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Topic** | | **Proposer** | | **Decision point** | **Include in 2.0? Decision** | **Benefits** | | **Risks** |
| State machine | | various | | As per SM Skype version 10 (or v4 see discussion on MDL fault handling) | Yes \* |  | |  |
| WSDL | | various | | WSDL as per: <http://code.google.com/p/ogf-nsi-project/source>  ogf\_nsi\_v2.Chicago | yes \* |  | |  |
| Modify | | JM | | NSI v2.0 support a Modify command  See WSDL: <http://code.google.com/p/ogf-nsi-project/source> | Yes \* |  | |  |
| Modify | | JM | | Modify allows the following to be modified: endtime, bandwidth, service attributes as defined in SD | Yes \* |  | |  |
| Modify | | Various | | Modify can only be performed by the same uRA that originally created the reservation | Yes \* |  | |  |
| Modify | | various | | The uRA assigns the reservation version number. | Yes \* |  | |  |
| Discovery service | | JM | | NSI will have a web service for NS interface protocol version discovery | Yes \* | Group will work through an appropriate definition.  Action to investigate into DCN defined mechanism. | |  |
| Discovery service | | JM | | Discoverable: NSI CS version, topology service version, Discovery Service version. Capabilities – what operations are supported, which role supported uRA, PA aggregator etc. Endpoint discovery | Yes \* |  | |  |
| Discovery service | | JM | | As per WSDL : <http://code.google.com/p/ogf-nsi-project/source>  And document by John MacAuley (location?) | Yes \* |  | |  |
| Error handling | | JM | | Supports nested error handling (errors from child NSAs) | yes \* | Nested errors make better error handling possible. Makes it possible to see what actually goes wrong.  Helpful in path finding to know which resources need to be removed from the next request. | | Higher complexity. Can get error codes for unknown NSAs |
| Error handling | | JM | | Should we add an error code hierarchy -  group errors in classes: connection error, security error, topology error, internal error… | Yes \* | Easier readability. | |  |
| Error handling | | JM | | List of valid exception codes - these are not part of WSDL, it is a living list – exception codes are defined in code Google | Yes \* |  | |  |
| Error handling | |  | | Decision on how to aggregate child errors and which to propagate up the tree is not documented – this is based on local policy |  |  | |  |
| Error handling | |  | | If a child returns and error then it is obliged to forward some error up-stream. |  |  | |  |
| Error handling | | JM | | WSDL now includes NSA id with the error code as an optional attribute. | Yes \* | Will permit requester to identify the NSA(s) that failed the request. | | Requester NSA might not know about the NSA generating the error.  The NSA hiding the existence of the NSA generating the error can remove it’s ID from the list and add it’s own. |
| forcedEnd | |  | | ForcedEnd is only issued to indicate up-stream that a fatal error has occurred. | Yes \* |  | |  |
| Protocol versioning | | JM | | Protocol versioning is supported using SOAP, WSDL XML namespaces | Yes \* | Already done. | |  |
| WSDL | | JM | | Moved header information from body into SOAP header.  This would include: correlationId, replyTo, reqNSA, provNSA and security attributes | yes \* | Simplifies compiler generated and simplifies WSDL definitions. | |  |
| EROs | | IM/ CG | | Support for EROs  As per GR’s STP summary doc  <https://forge.ogf.org/sf/go/doc16514?nav=1> | yes \* | With the agreement to expose internal topologies and manage sub-networks. Also with the intention of this protocol being used by network administrators, constraining the path with intermediate points is essential functionality | | NONE |
| Topology | | TK | | Internal topology proposal:  domain internal topology is represented using NML with a relationship between SDPs with NML edge ports | yes \* |  | |  |
| Topology | | JS | | Support unidirectional connections as request option? | Yes \* |  | |  |
| STPs | | GR | | As per GR’s STP summary doc  <https://forge.ogf.org/sf/go/doc16514?nav=1> | yes \* |  | |  |
| STPs | | TK | | ‘reference STPs’ are described in the topology and are used in connection requests.  The connection confirmation return ‘STP instance’ | Yes \* |  | |  |
| Topology service | | JvdH | | Topology service is as per proposal from JvdH:  <https://docs.google.com/document/d/1HIo7uQl7DbTe_y-cnPOrqDkspoRBaTKVVrrldURurTk/edit> | Yes \* |  | |  |
| Topology service | | JvdH | | The Topology service will define a common way to map STPs to NML. Note: SDP is a mapping between STPs, so it conceptual and has no identifier in NML. | Yes \* |  | |  |
| Topology service | | TM | | The data model is independent of underlying network technology: PBB, EoMPLS, | Yes \* |  | |  |
| Topology service | | IM | | The control plane reachability is explicitly defined in the topology using the ‘peers-with’ attribute. This describes existing peerings between NSAs. | Yes \* | Without this, none of the above features will work. NSA pathfinding is important to do to effectively request (in tree mode or tunnelling) services from those domains | | Are there any security risks in advertising the trust topology? |
| ERO | GR | | NSI allows ‘internal’ STPs to be added to an ERO?  Note: unresolved issue around orientation | | Yes |  |  | |
|  | |  | |  |  |  | |  |
| Firewall | | JM | | Provider NSAs (PA) must be publicly accessible through the firewall in v2.0. | yes \* | Decision was they need to be publicly accessible. | | We are investigating firewall safe access methods. |
| Security | | IM | | Security as per proposal: <https://forge.ogf.org/sf/go/doc16515?nav=1> | Yes \* | TLS/SSL for message transport encryption. Mutual authentication for non-NSA client identities that translate into authentication. WS-Security for authz with tokens and attributes. | |  |
|  | |  | |  |  |  | |  |
|  | |  | |  |  |  | |  |
|  | |  | |  |  |  | |  |

# NSI v2.0: Dropped Features

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Item** | **Proposer** | **Decision point** | **Include in 2.0? Decision** | **Benefits** | **Risks** |
| Simplification | JM | For simplification of the NSI protocol, do we require congruency between data plane and control plane? | No \* | Simpler breakout of reservation and makes chaining possible. | Probably won’t match real world. |
| Simplification | JM | For simplification of the NSI protocol, do we use NSI to tunnel reservation request be sent to head-end NSA before processing? | No \* | This allows users/NSA remote from the source endpoint network to issue a chain request to source endpoint’s  NSA. | Need to introduce new signalling plane message tunnelling /routing feature. |
| Simplification | JM/ IM | For simplification of the NSI protocol, do we remove support for the tree model, and do chain model only? | No \* | Permits only one mode of routing, and with #1, will not require reservations on NSA not in the data plane path. | If we can’t assume #1 then we can’t support chain only. |
| Simplification | JM | For simplification of the NSI protocol, should we by policy require the ultimate requester to send a connection request to head-end NSA only? | No \* | Argument is that if the end user has permission to utilize the resource in the head-end network then they should be able to talk directly to the head-end NSA to request the reservation. | Although may remove reservation state machines from some NSA that are not involved in the data plane, without #1 to permit chaining to data plane only NSA, then this does not provide value. |
| Topology | JS | Should we re-define NSI topology to be based unidirectional STPs only. Both uni and bi directional are supported | No \* | method for mapping to unidirectional concept in NML needs to be proposed |  |
| Service type | TM | Should we support the commercial requirement (#1) for requests for point-to-multipoint networks. | No \* | NSI will be used by commercial network in the future. For commercial network, the multipoint network is essential. |  |

# NSI v2.0: Features pending agreement

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Item** | **Proposer** | | **Decision point** | | **Include in 2.0? Decision** | **Benefits** | **Risks** |
| Topology service |  | | Do we only allow sharing of topology of NSAs which have an existing peering relationship? | | Follow up needed |  |  |
| Topology service | JM | | Allow sharing of different topologies based on your trust relationship  Answer: | | Follow up needed |  |  |
| State Machine | TK | | Layering of the protocol. Message delivery layer error handling – should this be in the statemachine (SM v10) or in the MDL layer (SM v4)? | | Follow up needed |  |  |
| State Machine | TK | | Work needed to define MDL error cases. | | Follow up needed |  |  |
| Error handling | HJT | | Do we need a separate set of downstream error codes… other solutions? | | Not in version 2.0 |  |  |
| Query | TK | | Do we need an ‘availability query’?  Note: this is like a reservation request, but will only check to see if resources are available, and will not reserve resources.  Change to WSDL: add ‘hold resources’ attribute to reservation request | | Yes |  |  |
| Firewall | JM/HTJ | | Make NSI more NAT/firewall friendly in NSI v2.0? | | no |  |  |
| Firewall | PB | | Should the client NSA be publicly accessible in NSI v2.0? | | Yes |  |  |
| Firewall | TM | | Should we support the commercial requirement (#3) NAT aware interface.  I.e should support non-persistent client requester Agent | | no |  | Why add this as a ‘commercial requirement’ as it is already a known problem. |
| Notification | HTJ | | Which strategy for state updates? | | ? | (depends on protocol choice) |  |
| NSI client | JM | | Should a separate NSI client be defined in NSI v2.0? | | ? | APs will provide investigation towards a simplified interface for clients. | “NSI client” is insufficiently defined. Will it have a smaller state machine? If the client interface does not expose the full NSI functionality, then clients become completely dependent on pathfinding etc., which is not yet defined. |
| Topology | GR | | How do we handle the ‘orientation’ attribute in the case of ‘internal’ STPs in an ERO?  Jeroen: STP on edge of domain use orientation relative to the network (i.e ingress or egress to the Network). The issue arises for bidirectional ‘internal’ STPs.  Jeroen solution: orientation is relative to the NMLnode.  Jerry: corner case, revisit if necessary. | | No change |  |  |
| Pathfinding | TK | | Should it be possible to select any STP as an end point?  Yes we should not explicitly exclude this. | | Yes |  |  |
| Service Definitions | | JS | | Service Definition defines what service characteristics can be requested. |  |  |  |
| Service Definitions | | JS | | Is the concept of an SD a requirement for NSI? | Yes \* |  |  |
| Service Definitions | | TK | | Should SD part of topology, or is it advertised separately via discovery service? |  |  |  |
| Service Definitions | | JS | | Are tech specific attributes in connection request required? | yes \* |  |  |
| Service Definition | | JS | | Should SD be part of the NSI version service? | Yes \* |  |  |
| Service Definition | JS | | Should end-point details be included in service definitions? Eg STPs, end point VLANs etc? | | ? |  |  |
| Service Definition | JS | | Should SD defaults be available for use for path-finding? | | ? |  |  |
| Service Definition | JS | | Should SD reflect network’s performance or performance of lowest common denominator service? Proposal: 2 types of SD: common and network specific. | | ? |  |  |
| Service Definition | JS | | Should access to SD be subject to authentication? | | ? |  |  |
| Service Definition | JS | | How do we ensure that the combination of CS version, Topology version and discovery version are compatible? | | ? |  |  |
| NSA roles | JM | | Should the topology capture the NSA role: Provider, Aggregator etc? This will allow pathfinder to better understand which NSAs can forward (aggregator) and which cannot (ultimate provider).  Change to Topology: add attribute to show NSA type (i.e RA/PA aggregator) | | Yes |  |  |
| Error handling | TK | | Errors are classified as: recoverable and unrecoverable errors.  Note: needs to be added to WSDL  Are ForcedEnd same as unrecoverable error? Should we rely on ForcedEnd only (no need for recoverable/unrecoverable state) | | Yes \* | For the purposes of defining error recovery mechanisms, we should be clear about which errors are recoverable. |  |