**Note for for editors: since no document tracking is available in Google docs, please follow these rules:**

**1. Please make sure you are signed in before editing to keep track of editors.  Note this can be automatic if you are logged into gmail or another Google application .**

**2. Please use comments (Insert -> Comment) to add your comments so that we can track who has added a comment.**

**3. If you wish to re-work a sentence, please add the new re-worded sentence  in a comment so that we can see the original.**

***NSI version 1.0 requirement and decision capture***

This document is a sandbox for capturing the requirements and design decisions of the NSI*,*feel free to use the comment text to add new requirements and decisions.

Primary requirements are designated an 'R' number

Design decisions/design requirements are designated a 'D' number.

Items in black font have 'closed' status.  Items in red have 'open' status.

Key words (must, may ..) follow the ietf rfc2119 usage.

<http://www.ietf.org/rfc/rfc2119.txt>.

**Summary of defined words used in this document**

The defined words listed have the meaning assigned in this table.  These follow the more detailed definitions which can be found in this document:  
<http://docs.google.com/Doc?docid=0Ac4ZeyIu2P2YZGhrNHRtdjJfOTNkdzViZndmaw&hl=en>

Defined words are identified using *Capitalized Italics* in the body of this document.

|  |  |
| --- | --- |
| **Connection** | A *Connection*is a conduit that transparently moves user information across a *Network*from an ingress point to an egress point. A *Connection*has a set of properties (for instance, capacity, or authorization, or start time).  These properties, and their allowed range of values, are defined by a*Service Definition*. A *Connection* instance is a particular *Connection*, identified by a *Connection Identifier*. |
| **Connection** **Identifier** | A *Connection Identifier* is a label unique to an NSI interface which can be used to identify a *Connection*for the purposes of request, instantiation and management. |
| **Connection Service** | A *Connection Service* is a service that allows a *Requester NSA* to request and manage a *Connection*from a *Provider NSA* |
| **Control and Management Planes** | The *Control Plane* and/or *Management Plane* are not defined in this document, but follow common usage. |
| **Network** | A*Network*includes all of the transport resources that can be assigned by an NSA, using its local NRM, to a service. |
| **Network  Resource Manager (NRM)** | The *Network Resource Manager* is the agent or function that has ultimate responsibility for authorizing, allocating, and scheduling a network resource.    Each NRM has a one-to-one mapping to an NSA. |
| **Network Service** | A *Network Service* is an abstract notion that must be implemented by a concrete network service agent (NSA). The *Network Service*is defined by the set functional capabilities offered by an NSA.  A *Network Service*can have many types, one of which is *Connection Service.* |
| **Network Service Agent (NSA)** | The *Network Service Agent* is a concrete piece of software that sends and receives NSI *Messages*.  The NSA includes a set of capabilities that allow *Network Services* to be delivered. |
| **Network Service Interface (NSI)** | The NSI is the interface between *Requester*NSAs and *Provider*NSAs.  The NSI defines a set of interactions or transactions between these NSAs to realize a *Network Service*. |
| **NSI Message** | An *NSI Message*is a structured unit of data sent between a *Requester NSA* and a *Provider NSA*. |
| **NSI Message Thread** | An *NSI Message Thread*is a group of messages that belong to both a single service type and a single *Connection*instance. |
| **Path** | A *Path*is an ordered list of *Routing Objects*which describes the route taken by a *Connection*.  a) A Path may be useful for other things than just in association with a Connection, so I do not think a general definition of a Path should reference Conenctions b) I wonder about "routing object"...we don't define routing in NSI, and "routing" is commonly used for conventional IP NLRI advertisements rahter than topology.   I suggest we call them a list of "topology references" or "topology objects" that describe a tour through a graph/topolgy. c) We do not need to rule out use of our paths for intra-domain purposes, and in fact, we may want to allow intra-domain use of the NSI Path object to help construct the connections (a connection will necessarilly need intra-domain resolution...might as well let them use NSI version if it works for them). -Jerry Sobieski 3/14/10 3:35 PM |
| **Requester/ Provider NSA** | An NSA acts in one of two possible roles relative to a particular instance of an NSI.  When an NSA requests a service, it is called a *Requester NSA*. When an NSA realizes a service, it is called a *Provider NSA*. |
| **Routing Object** | A *Routing Object* may include the following transport resources: Topology objects, STP? (*this list is not finalized)*.  might need more discussion of this - I am not sure there isn't more, like Network or Federation that might be included -John Vollbrecht 3/9/10 9:58 AM I am still not certain about network point, it has not been accepted defiition, so I would not use it here yet. -Inder Monga 3/10/10 6:09 AM I agree on both counts.   I assert a routing object could also be another Path, an STP, a name of a topological object,  -Jerry Sobieski 3/14/10 3:44 PM |
| **Service Definition** | The *Service Definition* is the set of attributes associated with connection services (for instance, capacity, or authorization, or start time) and a range of allowed values for these attributes.  Each *Connection*has an associated *Service Definition* instance. Three types or groups of parameters should be considered, regarding RESERVATION services: (i) scheduling/temporal params -when the service starts, ends, etc.; (ii) technology-dependent parameters -like bw, slots, standard used, etc.; and (iii) performance parameters a la SLA/SLS  -Joan Antoni Garcia Espin 3/24/10 4:21 PM |
| **Service Plane** | The *Service Plane*is a plane in which services are requested and managed; these services include the *Network Service.*The *Service Plane* contains set *Network Service Agents* communicating using *Network Service Interfaces*. |
| **Service Topology** | The *Service Topology*resides in the *Service Plane*.  The *Service Topology*describes both the physical resources and their interconnection as well as the non-physical groupings of various components. |
| **Transport Plane** | The *Transport Plane* contains is the set of physical resources that transport user data through the network. |

**1. NSI requirements/decisions relating to all network services**

Section 1 includes only requirements and design decisions that are relevant to ALL *Network Services*.

**1.1 Primary requirements**

The primary requirements described in this section have been clearly identified by the organizations participating in the NSI working group as functions necessary to deliver the *Network Services*.

R1.1.1 The NSI **must**provide a single comprehensive interface framework which supports *Network Services*to be established between multiple NSI compliant providers.

R1.1.2 TheNSI **must**be extensible to allow more than one type of *Network Services*to be supported. (of which version 1.0 will support *Connection Services*)

R1.1.3 The NSI **must**be able to support a *Connection Service*.

R1.1.4 The NSI **must**ensure confidentiality of messages between NSAs.

R1.1.5 The NSI **must**ensure integrity of messages between NSAs.

R1.1.6 The NSI protocol **must**support a method that allows the requester and provider NSAs to authenticate each other.

R1.1.7 The NSI protocol **must**allow authorization of all service messages according to local administrative policy.

R1.1.8 The NSI **must**support asynchronous processing of service requests.  Specifically, processing of one connection request shall not block processing of another request unless the two requests common resource dependency.

R1.1.9 The Service Plane connectivity **must**not be assumed to be the same as the Transport Plane connectivity.

R1.1.10 The NSI architecture **must**not force services requests to originate from any specific NSA. (i.e. this is different to MPLS UNI which only allows connections requests that terminate on the node associated with the UNI)

**Possible future requirements, not included in version 1.0:**

R1.1.11 The NSI **must** have modularity to support a computing resource service.

R1.1.12 The NSI **must**implement a notification service that informs interested and authorized agents of service related events.

R1.1.13 The NSI **must**be able to support the reseller model of operation.

I think this should be required from the start.  I don't think it is that hard.  It is what is done by GENI and ORCA to create aggregates that are used by brokers that are then reallocated to slices.  -John Vollbrecht 3/9/10 10:05 AM I think in cases like this, the advertised edge ports change  to reflect the connection being resold. In other word, the new link that is being resold now gets advertised by its new edge port rather than its old one(which is now an internal port and not advertised) -Gigi K-E 3/10/10 8:47 AM

Essentially a futures broker...the resource owner never takes actual delivery of the connection, just the reservation, and then he allocates the resources to other connection requests which (ultimately) get actually instantiated. -Guy Roberts 2/26/10 2:43 PM

I think this is the notion that someone could request a connection without taking actual delivery of the instantiation, and then- being the allocated owner of the resources - could sub allocate them to real users who would in fact take delivery of the connection instance. -Jerry Sobieski 2/26/10 10:08 AM

There can be two ways: authority delegation and certificate delegation. The former is the reseller model, I think. Or, if John is thinking of an agent to reserve (or secure) some resources first, and then split and "resell" them, it is a completely different concept. I do not think this is "must".-Tomohiro Kudoh 2/26/10 11:44 PM

This is an interesting notion that deserves some serious thinking, but its not crucial for V1.0, so that why I thought it should be here on the Future Features List -Jerry Sobieski 2/26/10 10:15 AM

**1.2  NSI/NSA**

This section includes a set of agreed design statements and/or requirements regarding the NSI architecture. Note that these are generic to all types of *Network Service*, not just the *Connection Service*.

D1.2.1 Each NSA must have a identifier that distinguishes it from all other NSAs on the *Service Plane* and allows access.

D1.2.2 The *Network Service Interface* is used to communicate between *Network Service Agents*.

D1.2.3 The *Connection Service*is a type of *Network Services*.

D1.2.4 The only *Network Service*described in version 1.0 of the NSI is the *Connection Service*.

D1.2.5 The *Network Service Agent* (NSA) is the agent that is used to request and manage services.

D1.2.6 The NSA **must**implement a mechanism to maintain synchronization of the calendar clocks between reachable NSAs to an a pre-agreed accuracy. (This might for example be NTP)

D1.2.7 The NSA **must**implement mechanisms to ensure that if the NSA calendar clocks drift within their bounds, no timing deadlocks/conflicts can occur. (This could for example require a hold-off time between reservations)

D1.2.8 The NSA **must**implement a mechanism to hold the calendar and service state in long term memory to recover from an NSA re-start.

**1.3  NRMs**

This section includes a set of agreed design statements and/or requirements regarding using the NSAs for managing local resources.

D1.3.1 The *Network Resource Manager* (NRM) is the functionality within an NSA that provides control over transport resources internal to the associated *Network*.

D1.3.2 Each transport resource **must**be managed by exactly one NRM.

D1.3.3 A *Network Resource Manager (*NRM) **must**be authoritative for all devices under its control for all service plane semantics applied to those devices.

D1.3.4 T Each *Network Resource Manager* (NRM) **must**be associated with exactly one NSA.

D1.3.5 An NSA is not required to contain an NRM.

**1.4  NSI Messages - General**

This section includes a set of agreed design statements and/or requirements regarding NSI messages.

D1.4.1 An *NSI Message*is a datagram style payload that canbe broken into smaller sizes by the underlying layers for transport.  Note: transport refers here to the transport of the NSI protocol.

D1.4.2 The *Provider NSA* **must**handle an incoming NSI *Message*completely, returning at a minimum either a confirmation or rejection of the request in its entirety.

D1.4.3  Authentication of the source of all messages may be done on a message by message basis or with an authenticated session.

D1.4.4  The NSI interface **must**have a mechanism to detect the interface state going down.

D1.4.4a  The NSA **must** have a method of recovering to a known state after the NSI interface is restored after a going down.

D1.4.4b  The NSA **must**have a mechanism to either: (a) validate alignment of the requester and provider states after the NSI interface has gone down and been restored, (b) tear down the connection and re-provision in the case of misalignment of states.

D1.4.6 An *NSI Message* **must**include an addressing field to allow identification of the destination NSA and  its associated network

D1.4.7a An *NSI Message* **must** allow identification of the Network Service type. (eg Connection Service, Topology Service, etc).

D1.4.7b An *NSI Message* **must** allow identification of the Connection service version (v1.0, v2.0 etc)

D1.4.8 An *NSI Message* **must** allow identification of the Network Service Instance. (e.g. a connection identifier).

D1.4.9 An *NSI Message* **must**include a mechanism to associate it with an *NSI Message Thread* to allow differentiation of message streams associated with simultaneous and asynchronous service functions occurring between pairs of NSAs

D1.4.10 NSI *Messages* **must**include a mechanism to ensure that ordering is maintained in a NSI *Message Thread.*  Note: NSI message ordering just does not prevent asynchronous operation of the service protocol.  Just that message sequence is preserved across the secure session. -Jerry Sobieski 2/26/10 9:52 AM

**2. Connection service v1.0**

Section 2 includes only requirements and design decisions that are relevant to v1.0 of the *Connection Service.*

**2.1 Connection Service general**

This section includes a set of agreed design statements and/or requirements regarding *Connection Services.*

D2.1.1 A *Connection*request is initiated by a *Requester*NSA and implemented by a *Provider*NSA.

D2.1.2 The *Connection Service***must** provide unidirectional point-to-point *Connections*. (these are the only types of *Connections*supported by the *Connection Service* in NSI v1.0)

D2.1.3 The *Connection Service***must**provide for reserving connection services at some point in the future. (advanced reservation)

D2.1.4 The *Connection Service***must** provide the ability to request connection services to be reserved and provisioned near-immediate time.

**Advanced/future requirements:**

D2.1.4 The *Connection Service****must***allow other forms of *Connections*(apart from unidirectional point-to-point) to be requested in future versions, these may include bidirectional or point-to-multipoint Connections. I.e requires fields to identify the  *Connection Service* is of type v1.0: unidirectional, pt-to-pt.

**2.2 Connection Messages General**

D2.2.1 A *Connection*message **must**include a *Connection Identifier*that is at a unique within the context of the two NSAs participating in this *Connection* Request.

D2.2.2  The *Connection Identifier* **must**be resolvable for identifying the complete end-to-end *Connection*.

D2.2.3  *Connection Identifiers* are not required to be globally unique by the NSI *Connection Service*, however the *Connection Identifier* fields should be sufficiently large to support globally unique identifiers where desired.

D2.2.4 The Provider NSA **must**be responsible for allocating the *Connection identifier.*

***2.3 Connection Messages Primitives and Responses***

D2.3.1 The*Connection Service***must** support a 'Reserve' function.  This will signal the *Connection*reservation state to be instigated.

D2.3.2 The Connection 'Reserve' function **must** support a response to indicate that the Provider has either failed to reserve or has moved from reserving to scheduled state.

D2.3.3 The Connection Service **must** support a 'Provision' function .  This will signal the Provider connection state to transit from scheduled to provisioning.

D2.3.4 The Connection 'Provision' function **must** support a response to indicate that the Provider has either failed to provision or has moved from provisioning to in-service state.

D2.3.5 The*Connection Service* **must** support a 'Cancel' function .  This will allow a Connection reservation or a provisioned service to be removed.

D2.3.6 The Connection 'Cancel' function **must** support a response to indicate that the Provider has either failed to cancel or has completed de-provisioning of the connection.

D2.3.7 The *Connection Service* must support a 'Query' function .  This will allow he state of a Connection to be found by the Requester NSA. (this is to allow requester and provider state machines to align?)

D2.3.8 The *Connection*Service **must** support a 'List' function. This will allow the the Requester to find a list of all *Connection*service instances that have been created by the Requester.

***2.4 Connection Requests***

D2.4.1 A *Connection*request must include a *Service Definition*.

D2.4.2  A*Connection*request **must**allow the requester to nominate the start and end time of the*Connection.*(for details of the service definition see the service definition section of this document)

D2.4.3 A*Connection*request **must**include a fully or partially complete*Path*of the*Connection.*(for details of the Path object see the Path section of this document)

***2.5 Service Definitions***

D2.5.1 The *Service Definition****must***specify the Connection attributes and their ranges.

D2.5.2 Service Definitions **must**be based on templates produced by operators or federations of operators.

D2.5.3 Service Definitions template **must**include all parameters associated with the relevant service type.

D2.5.4 A sample OGF-NSI Service Definition template will be provided by NS

***2.6 Advanced and immediate reservations***

D2.6.1 A *Connection*request **must**include a flag to identify the category of reservation. (i.e. advanced reservation, immediate reservation) (do we support immediate reservation in v1.0?)

D2.6.2  The start time of a *Connection*request **may**accept the value 'ASAP' in the case of  'immediate' connection reservation types.

D2.6.3  The end time of a *Connection*request **may**accept the value 'Infinite' to indicate an undetermined end time.

D2.6.4 Provisioning of the Connection **must**be initiated in one of 2 ways: automatic or explicit.

D2.6.5 A*Connection*request **must**include a flag to identify if provisioning is automatically or explicitly initiated.

D2.6.6 In explicit mode, provisioning **must**be initiated by a message from the Requester NSA.

D2.6.7 If the connection provisioning is automatically provisioned, the Requester NSA **must** not send a connection request with a start-time sooner than Now+RequesterGuardTime.

.

D2.6.8 A *Provider*NSA when scheduling a *Connection*request **may**perform a guard time test on the start time as follows: If the start time is sooner than Now+ProviderGuardTime then the *Connection*request will be rejected. (this is a path-finding constraint)

D2.6.9 The guard times will be agreed based on local policy and managed by the NSA.

D2.6.10 The start time will refer to the In-service time (not the provisioning start time), the Provider NSA is responsible for trying to meet this time.

**2.7 Connection Service states**

D2.7.1 A *Connection*service can be in any one of the following states: Reserving, Scheduled, Provisioning, In-service, Releasing.

D2.7.2 In the 'Reserving' state, the connection request has been received and the Provider NSA is waiting for resources to be negotiated to meet the service request

D2.7.3 In the 'Scheduled' state, the resources have been found and reserved, but the connection start time has not begun.

D2.7.4 In the 'Provisioning' state, the connection start time has begun and the provisioning process is ongoing.

D2.7.5 In the 'In-service' state, the provisioning process has been completed and the Requester NSA has been notified.

D2.7.6 In the “Releasing” state, the cancel primitive has been received by the Provider NSA and the de-provisioning process is ongoing.

**2.8 Path decisions/requirements**

D2.8.1 A *Path***must**include at a minimum two *Routing Objects*that identify the ingress and egress points for the *Connection.*

D2.8.2 A*Path***may**include intermediate *Routing Objects.*

D2.8.3 A *Connection***must**transit the *Routing Objects* in the order they appear in the *Path*.

D2.8.4 The direction of the *Path*and the order which a *Connection Service* is forwarded are independent.

D2.8.5 A *Routing Object***must**be one of the following: a) a reference to a topological location (STP),   b) a reference to a named *Path* object.   A named *Path* will not be visible to unauthorized agents, so it must be associated with an authorization policy. I don't understand this -John Vollbrecht 3/9/10 10:14 AM

D2.8.6 An STP is a symbolic reference, i.e. it is comprised of a parsable alphanumeric string containing two  components: 1) a *Network*identifier string in the higher order portion, and 2) a local portion that identifies an STP within that *Network*.

D2.8.6 Where the *Path*of a parent request is segmented into multiple child requests the *Path*of each child request **must**include an ingress and egress Routing Object belonging to the destination Network.

D2.8.7 Where the *Path* of a parent request is segmented into multiple child requests the *Path*of the child request **may**optionally contain other *Routing Objects* belonging to the parent request.

D2.8.8 Where the *Path*of a parent request is segmented into multiple child requests the *Path*of the child request **may** optionally contain locally computed *Routing Objects*.

D2.8.9 Where the *Paths*of multiple child responses are aggregated into a single parent response the *Path*of the parent response **must**contain the ingress and egress *Routing Objects* of the original request.

D2.8.10 Where the *Path*of multiple child responses are aggregated into a single parent response the *Path*of the parent response **may**contain the other *Routing Objects* from each child response.

D2.4.5 through D2.4.10 need to be reviewed and agreed by the group  -Guy Roberts 3/10/10 5:36 PM

**Advanced/future requirements:**

R2.8.11 A Path may include transport resources that are excluded (i.e. may not be transited by the path). Note For NSI v1.0 we consider the *Connection*request parameters as described by the partially (or fully) completed service request as the only constraints on the path.   -Jerry Sobieski 2/26/10 10:02 AM

**2.9 Path-finding design**

D4.9.1 *Network*-internal path-finding is out-of-scope for the NSI.

D4.9.2 The NSA may make use of the inter-*Network* path-finding which returns a candidate partial (or complete) inter-*Network Path.* This is a candidate path only since the reservation needs first to be confirmed in each domain before reservation.

D4.9.3 The algorithms for inter-*Network* path-finding are out of scope of this document.

D4.9.4 The inputs to theinter-*Network* path-finding function are: an inter-Network *Topology*, a *Path* and the service definition.

**2.10 Connection Segmentation/Aggregation**

In summary, segmentation is the process of segmenting a parent request into child requests.  Local child requests are processed by the local NRM, other child requests are forwarded to another NSA.  When an NSA has segmented a parent request, several responses can come from multiple *Provider*NSAs, in this case the local NSA must aggregate the results before passing them back to the original *Requester*NSA

D2.10.1*Connection Segmentation*is optionally performed by an NSA.

D2.10.2The process of *Connection Segmentation* is where a *Provider NSA* analyses incoming *Connection*requests and may split and forward these as new request.

D2.10.3A *Provider*NSA has to perform*Connection Segmentation*if one or more of the request's *Routing Objects*are reachable but not part of the local network

D2.10.4 *Connection Segmentation*operates on a parent *Connection* request.

D2.10.5 *Connection Segmentation*creates one or more child *Connection* requests.

D2.10.6Local child *Connection* requests are forwarded to the NRM.

D2.10.7 *Connection Segmentation*uses the NSI interface to manage non-local child *Connection* requests

D2.10.8A *Provider* NSA performs*Connection Aggregation*on the response if  *Connection Segmentation*was performed to the request.

D2.10.9  *Connection Aggregation*is the process by which responses to segmented *Connection*requests are recombined to form a single *Connection* response.

D2.10.10 *Connection Aggregation*operates on multiple related child responses to *Connection* requests.

D2.10.11 *Connection Aggregation*creates one parent response to *Connection* requests.

D2.10.12 *Connection Aggregation*gets local child responses from the local NRM.

D2.10.13 *Connection Aggregation*gets non-local child responses from the NSI interface*.*

D2.10.14 The NSA **must**hold state of relationship between parent and child connection request.

**Comments on Segmentation/Aggregation:**

Don't confuse segmentation with Request Handling.  Request handling deals with local and remote networks...A segemetnation operation may not even transit a domain boundary, or even be associated with setting up or managing the connection.    -Jerry Sobieski 2/26/10 2:39 PM

Connection Aggregation  Aggregation does not imply ordering - which in this case we require. (aggregation is often used to sum capacities of parallel links, or to deote an unordered grouping.) In other communication protocols this is refered to as "Segemetnation and Re-assembly" or "Fragmentation and Re-assembly"...I suggest the process we are describing here is "Concatenation.", or "[re]Composition" -Jerry Sobieski 2/26/10 2:49 PM

In point of fact, all the local NSA is doing is computing a response back to the requester - all the engineering of the segments (end point selection et al) is done prior to submitting the sub-request.  IMO, There is no real process of aggregation/composition performed. -Jerry Sobieski 2/26/10 3:00 PM

Aggregation NSA allows allocation of resources in models that have been called chain, tree and reseller.  Chain is the GMPLS model, tree is a hierarchical method, and reseller is a resource delegation method.  The ability to support the three models above create some issues that are not yet resolved including how to define segmented and aggregated connections. JVR

**3. Modelling static inter-network topology**

D 3.1: The model must be able to describe a grouping of network resources that are owned or controlled by a single *Provider*NSA. (this is referred to here as a  *Network*) This may be able to be more generalized to say "the model must support groupings of topological components.-Jerry Sobieski 2/26/10 10:25 AM I think we should call this a network until we come up with a better name -John Vollbrecht 3/9/10 10:16 AM

D 3.2: The model must be able to describe a grouping of *Networks*. (e.g. a federation of providers with shared policy)  groupings again. -Jerry Sobieski 2/26/10 10:27 AM federation implies trust in my mind  -- is that what is meant here? -John Vollbrecht 3/9/10 10:17 AM

D 3.3: The model must be able to describe resources (ports/points) in a *Network*that are available for connecting to other *Networks*.  (this is a *Network Port*)I think the \*requirement\* is that the model must indicate where topologically networks transport planes interconnect.  We need to come to a vote/consensus on a model to adopt.  This requirement (and many of these) sound more like definitions than requirements. -Jerry Sobieski 2/26/10 10:18 AM  I think this is necessary only for doing pathfinding -  so perhaps we can punt it for the short term -John Vollbrecht 3/9/10 10:18 AM

D 3.4: *Network Ports* must be able to be assigned to the end of a link that is internal to the domain as well as to ports on a device.  In my opinion a *Network Port* on a link requirement needs a use-case -Guy Roberts 2/26/10 3:17 PM  Sounds like a definition rather than a requirement-Jerry Sobieski 2/26/10 10:21 AM

D 3.5: The model must be able to describe an arbitrary number of layers of logical ports within a *Network Port*. Must be able to describe multi-layer networks.

D3.5a. The model must support arbitrary groupings of topological components.-Jerry Sobieski 2/26/10 10:22 AM

D 3.6: The model must be able to describe connectivity between *Networks*. (*Network Points*) See R3.3...dulicate? -Jerry Sobieski 2/26/10 10:28 AM  Points are different to ports.  Ports are th e place where you can connect, Points are the connection of two Ports -Guy Roberts 2/26/10 3:50 PM

D 3.7: The model must be able to describe groups of *Network Points*.  What is the use-case for this? VLANs? If so, is it really necessary for NSI that these are in the same group? -Jeroen van der Ham 2/26/10 3:45 PM I think these can be handled by our "groupings" requirement.  Our model needs more consideration...but I don't think we need a lot of new classes or suposedly new features...this can (in my estimation) be generailed in a simplisitc solution. -Jerry Sobieski 2/26/10 10:28 AM

D 3.8: The resources that make up *Network Points*must have ownership by a clearly identifiable provider. (i.e. resources without NSA ownership are not allowed).  (note: Does this also include the patch cord between providers?) All "resources" in the NSI model \*must\* map to a single authoritative management agent (NSA). -Jerry Sobieski 2/26/10 10:31 AM

D 3.9: The model must allow policy to be assigned to *Network Points*, even where the *Network Points*is wholly or partly made up of passive resources.  Huh?  Generalize this...transport policy should able t be associated with any [all] transport resources, everything else is simply service layer issue...?  If we group resources into a group, network, or federation, it still is transport resources and can have authorization policies.  BTW- what are "passive" resources?-Jerry Sobieski 2/26/10 10:32 AM

**4. Associated functions**

**4.2 NSA lookup service**

D4.1.5 A lookup service may be used by an NSA, this function is out-of-scope for the NSI.

**4.3 Authentication**

D4.1.6 The NSI will provide hooks to make use of 3rd party Authentication.

D4.1.7 Implementation detail for Authentication is out-of-scope for the NSI.

**4.4 Authorization**  
D4.1.8 The NSI will provide hooks to make use of 3rd party Authorization.

D4.1.9 Implementation detail for Authorization is out-of-scope for the NSI.

\*\*\* how do we support authorization in the case of request forwarding?  does authorization need to know the original requester?

\*\*\* is the authorization between adjacent NSAs or should authorization know the original requester and his credentials?

**5. Open issues**  
5.1a STPs, (Jerry/John)

5.1b Inter-Network topology (John)

5.2 Pathfinding

5.3 Calendar Reservation (Tomohiro Kudoh)

5.4 Error Handling (Chin/Inder)

5.5 Service Definitions (Jerry)

5.6 Segr/Aggr of connections

5.7 Impact on state, service setup time

5.8 Agent finding as a subset of Pathfinding

5.9 Trust and Authorization

5.10 Do we need to use the word “OWN”?, Definition of the word ownership. (Joan)