GWD-R, GWD-I or GWD-C
NSI-WG
nsi-wg@ogf.org

John MacAuley, ESnet

December 23, 2014

Network Service Interface

Network Service Agent Description Document

Status of This Document

Grid Working Document (GWD), Recommendation (R).

Copyright Notice

Abstract

This Network Service Interface (NSI) document describes the Network Service Agent (NSA) Description Document that provides syntax for describing metadata for NSAs within the Network Services Framework (NSF) in support of NSA self-description. When used in conjunction with the protocol known as the Document Distribution Service, a mechanism is provided to discover peer NSAs, and their associated interface versions, features, control plane connectivity, and managed networks. Information conveyed in this document allows an NSA to perform basic protocol bootstrapping with minimal configuration by exposing an NSA's identity, enabling version negotiation, and communicating protocol capabilities supported by that NSA.

Contents

1 Introduction

Within the Network Services Framework (NSF) [OGF NSF], the Network Services Agent (NSA) is an entity that manages network service requests. These services can vary in function, and an NSA does not need to offer all of the services defined within a Network. For example, one NSA may offer Connection Services and Topology Services for a specific network, while a second NSA offers Monitoring Services for that same network. In addition, the versions of the services offered can vary from NSA to NSA. The NSA Description Document is a metadata schema designed to enable self-description of all NSI services and associated protocol interfaces offered by these NSA. Other information relating to the NSA itself, such as software version, administrative contacts, location, peering, and managed networks is also defined as part of the meta-data profile.

The NSA Description Document is used in conjunction with the Document Distribution Service (DDS), to support the distribution of information throughout an interconnected network of NSAs.

Such a dynamic meta-data discovery mechanism is an important element of any large-scale distributed system. By making the NSI protocol and its agents more self-descriptive, new features, protocols, or protocol versions can be added to agents within the Network and these can then be discovered by peer agents. As new features come on line, agents supporting the capabilities can discover compatible peer agents, and then negotiate use of these new features, while older versions of agents within the network remain unaffected. Similarly, newer versions of agents can still negotiate features and communicate with older agent versions using mutually supported versions of the protocol as described in the discovered meta-data.

This document defines the base NSA meta-data document schema. The provided examples within this document are informational only and should not be considered normative.

2 Notational Conventions

The key words ‘MUST,” “MUST NOT,” “REQUIRED,” “SHALL,” “SHALL NOT,” “SHOULD,” “SHOULD NOT,” “RECOMMENDED,” “MAY,” and “OPTIONAL” are to be interpreted as described in RFC 2119 [BRADNER], except where the words do not appear in uppercase.

3 NSA Description Document

The NSA Description Document encapsulates descriptive meta-data associated with an NSA. The XML schema types used to define the document format are declared in a separate namespace from the core protocol specification, allowing new versions of the NSA Description Document schema to be introduced without impacting the base description protocol itself. Figure 1 below shows the structure of the NSA Description Document, while Appendix A: NSA Description Document schema contains the full XML schema definition.

The <*nsa*> element is the root element used in all NSA Description Documents. Each NSA Description Document MUST have a single <*nsa*> element describing the subject NSA.

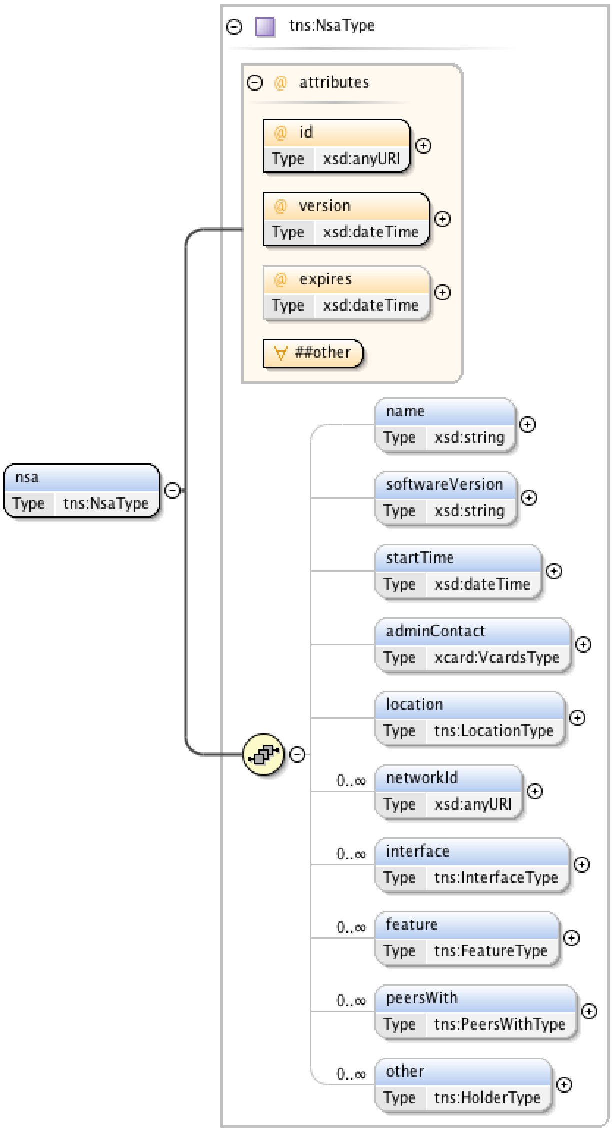


Figure 1 – The NSA Description Document.

The following XML is an example NSA Description Document for a fictitious NSA with globally unique identifier "*urn:ogf:network:example.com:2013:nsa:vixen*".

```xml
<?xml version="1.0" encoding="UTF-8"?>
<tns:nsa xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xcard="urn:ietf:params:xml:ns:vcard-4.0"
  xmlns:tns="http://schemas.ogf.org/nsi/2014/02/discovery/nsa"
        id="urn:ogf:network:example.com:2013:nsa:vixen"
        version="2014-01-04T18:13:51.0Z"
        expires="2014-01-04T18:13:51.0Z">
    <name>Example NSA</name>
    <softwareVersion>ExampleNsa-Version-1.0</softwareVersion>
    <startTime>2014-01-01T18:13:51.0Z</startTime>
    <adminContact>
        <xcard:vcard>
            <xcard:uid>
                <xcard:uri>http://www.example.com/santa.claus/santa.asc</xcard:uri>
            </xcard:uid>
            <xcard:prodid><xcard:text>OGF Example Maker // EN</xcard:text></xcard:prodid>
            <xcard:rev><xcard:timestamp>20080424T195243Z</xcard:timestamp></xcard:rev>
            <xcard:kind><xcard:text>individual</xcard:text></xcard:kind>
            <xcard:fn><xcard:text>Saint Nicholas</xcard:text></xcard:fn>
            <xcard:n>
                <xcard:surname>Claus</xcard:surname>
                <xcard:given>Santa</xcard:given>
                <xcard:suffix>Saint</xcard:suffix>
            </xcard:n>
            <xcard:tel><xcard:text>+1 555-555-5555</xcard:text></xcard:tel>
            <xcard:email>
                <xcard:text>santa.claus@theworkshop.example.com</xcard:text>
           </xcard:email>
        </xcard:vcard>
    </adminContact>
    <location>
        <name>Santa's Workshop</name>
        <longitude>0.0000</longitude>
        <latitude>90.0000</latitude>
        <altitude>10</altitude>
        <address>
            <xcard:pobox>0001</xcard:pobox>
            <xcard:ext></xcard:ext>
            <xcard:street>1 Top of the world boulevard</xcard:street>
            <xcard:locality>Polar Ice Flows</xcard:locality>
            <xcard:region>The North Pole</xcard:region>
            <xcard:code>CA</xcard:code>
            <xcard:country>Canada</xcard:country>
        </address>
    </location>
    <networkId>urn:ogf:network:example.com:2013:network:theworkshop</networkId>
    <networkId>urn:ogf:network:example.com:2013:network:candycaneforest</networkId>
    <interface>
        <type>application/vnd.ogf.nsi.dds.v1+xml</type>
        <href>https://nsa.example.com/dds</href>
        <describedBy>https://nsa.example.com/dds?wadl</describedBy>
    </interface>
    <interface>
        <type>application/vnd.ogf.nsi.topology.v1+xml</type>
        <href>https://nsa.example.com/topology.xml</href>
    </interface>
    <interface>
        <type>application/vnd.ogf.nsi.cs.v2.provider+soap</type>
        <href>https://nsa.example.com/connectionProvider</href>
        <describedBy>https://nsa.example.com/connectionProvider?wsdl</describedBy>
    </interface>
    <interface>
        <type>application/vnd.ogf.nsi.cs.v2.requester+soap</type>
        <href>https://nsa.example.com/connectionRequester</href>
        <describedBy>https://nsa.example.com/connectionRequester?wsdl</describedBy>
    </interface>
    <feature type="org.ogf.nsi.cs.v2.role.aggregator"/>
    <feature type="org.ogf.nsi.cs.v2.role.uPA"/>
    <feature type="org.ogf.nsi.cs.v2.commitTimeout">120</feature>
    <!-- The following peersWith element describes a control plane peering with
         an aggregator. -->
    <peersWith role="RA">urn:ogf:network:example.com:2013:nsa:dasher</peersWith>
    <peersWith role="PA">urn:ogf:network:example.com:2013:nsa:dasher</peersWith>

    <!-- The following peersWith element describes a control plane peering with
         a uPA. -->
    <peersWith role="RA">urn:ogf:network:example.com:2013:nsa:dancer</peersWith>

    <!-- The following peersWith element describes a control plane peering with
         a uRA. -->
    <peersWith role="PA">urn:ogf:network:example.com:2013:nsa:prancer</peersWith>
</tns:nsa>
```

The remainder of this section defines the XML types used within the NSA Description Document schema.

The schema elements marked as ‘M’ in Section 3 of this recommendation MUST be implemented. The schema elements marked as ‘O’ in Section 3 of this recommendation MAY be implemented.

3.1 NsaType

The ***NsaType*** definition models the primary meta-data elements of an NSA. The *id* attribute of the NSA MUST be globally unique as this is the primary identification key used across all NSAs for discovery. This *id* is referred to as the NSA identifier in the NSI Connection Service (CS) protocol documents [OGF NSI-CS].

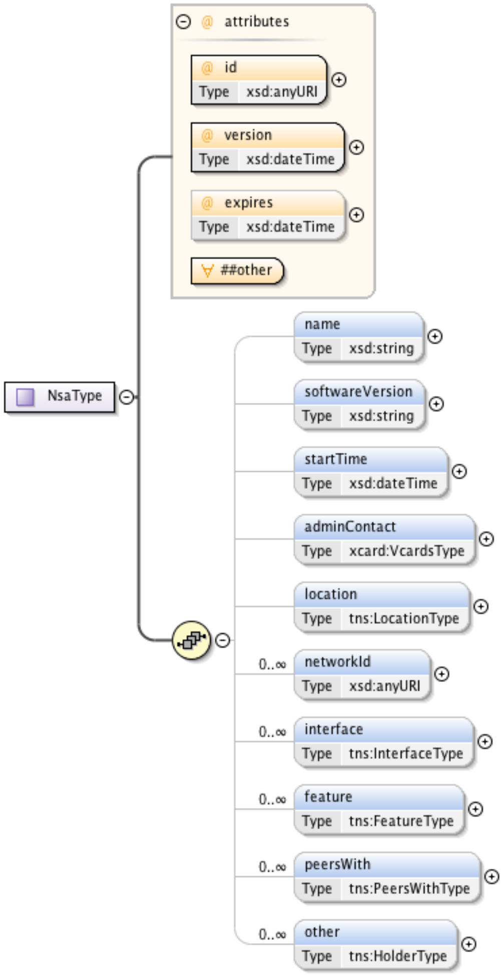


Figure 2 – NsaType.

Parameters

The <*nsa*> element is defined by the complex type ***NsaType*** that has the following parameters (M = Mandatory, O = Optional):

Parameter	M/O	Description
id	M	The globally unique NSA identifier for this resource.
version	M	The version of this NSA Description Document based on the date and time the entry was created at the source NSA. This attribute can be used to compare two versions of the document for equality (same version) or to determine the new and older versions through date comparison.
expires	O	The date that this version of the document expires, after this date it should no longer be considered valid. If this value is not present then the document should be considered to have no expiry date.
anyAttribute	O	Permit inclusion of attributes from other namespaces for flexible extension without needing to update this schema definition.
name	O	A descriptive name for this NSA. This value is typically used for display purposes.
softwareVersion	O	A descriptive string describing the NSA software type and version. This value will allow a peer NSA to adapt behaviors to specific versions of an NSA when required.

startTime	O	The time at which this NSA last started to provide uninterrupted service. This is effectively the last restart time of the NSA. A peer discovering a change in this value can initiate recovery procedures.
adminContact	O	A list of zero or more administrative contacts associated with this NSA.
location	O	The physical location of the logical NSA entity. This can be the location of the server hosting the NSA, or some other location related to the service being offered.
networkId	O	A list of zero or more network identifiers for which this NSA is providing the listed service interfaces and features. These network identifiers can be mapped into network topology using the <*Topology*> element's *id* attribute to determine the network resources being managed by this NSA.
interface	O	A list of zero or more NSI interfaces supported by the NSA.
feature	O	A list of zero or more features supported by the NSA. An NSA feature is a piece of metadata that describes a specific capability offered by this NSA, or a configuration value for this NSA, that is not specifically defined by an independent element definition within this document.
peersWith	O	A list of zero or more NSA identifiers enumerating peer NSA that have had a trusted control plane relationship provisioned with this NSA. Each entry in this list represents a trusted unidirectional relationship with the direction described by the "role" attribute associated with the peersWith element.
any element	O	Provides a flexible mechanism allowing additional elements to be provided from other namespaces without needing to update this schema definition.

3.2 VcardsType

The *adminContact* field of the ***NsaType*** definition uses the standard vCard XML Representation [RFC 6351]. The ***VcardsType*** supports a list of *vcards* that can be used to fully model administrator contact information. Due to the size of the structure it will not be reproduced here.

3.3 LocationType

The ***LocationType*** definition models the location elements of an NSA. A Location is a reference to a geographical location or area for the NSA.

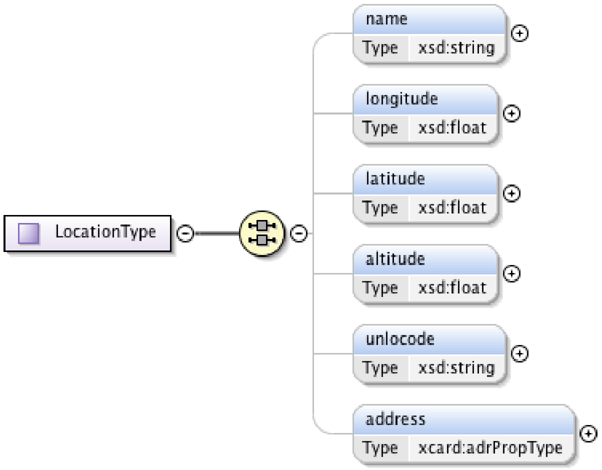


Figure 3 – LocationType.

Parameters

The <*location*> element is defined by the complex type ***LocationType*** that has the following parameters (M = Mandatory, O = Optional):

Parameter	M/O	Description
name	O	A human readable string naming this location.
longitude	O	The longitude of the NSA in WGS84 coordinate system (in decimal degrees).
latitude	O	The latitude of the NSA in WGS84 coordinate system (in decimal degrees).
altitude	O	The altitude of the NSA in WGS84 coordinate system (in decimal meters).
unlocode	O	The UN/LOCODE location identifier for the NSA location.
address	O	The address of the NSA location specified in vCard address format.

3.4 InterfaceType

The ***InterfaceType*** definition models an NSA protocol interface. This type encapsulates the meta-data needed to determine the version, protocol end-point, and schema associated with a specific NSA interface.

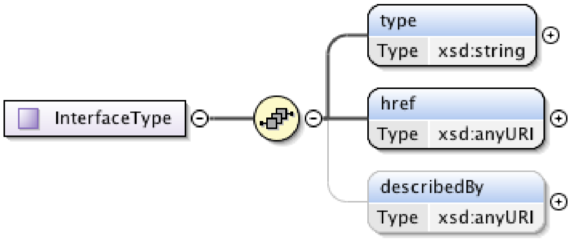


Figure 4 – InterfaceType.

Parameters

The <*interface*> element is defined by the complex type ***InterfaceType*** that has the following parameters (M = Mandatory, O = Optional):

Parameter	M/O	Description
type	M	The unique string identifying the type and version of the NSA interface. Application Internet media types (Content-types) are used to identify the NSI interface, version, and supported encoding type.
href	M	Contains the protocol endpoint for the interface identified in this interface reference.
describedBy	O	This attribute contains a reference to the WSDL or WADL file corresponding to this interface's version (if available).

For example, the first <*interface*> element below identifies the proposed NSI Document Distribution Service Version 1 [OGF NSI-DS] XML encoded representation. The <*type*> element describes the specific version of the DDS interface, as well as the media encoding used on the interface. The <*href*> element provides the protocol endpoint used to access this interface. The optional <*describedBy*> element provides a reference to the meta-data document formally describing the interface. In this case, a WADL document is available describing the description REST interface.

```
<interface>
    <type>application/vnd.ogf.nsi.dds.v1+xml</type>
    <href>https://nsa.example.com/dds</href>
    <describedBy>https://nsa.example.com/dds?wadl</describedBy>
</interface>
```

This second entry also defines an interface for the proposed NSI Document Distribution Service Version 1, but instead of XML, this is a definition for a JSON representation:

```
<interface>
    <type> application/vnd.ogf.nsi.dds.v1+json</type>
    <href>https://nsa.example.com/dds</href>
    <describedBy>https://nsa.example.com/dds?wadl</describedBy>
</interface>
```

It is also possible to define an <*interface*> element that does not contain the <*describedBy*> element. This can be used in situations where dynamically discovering the interface description is not required or available:

```
<interface>
    <type> application/vnd.ogf.nsi.topology.v1+xml</type>
    <href>https://nsa.example.com/topology.xml</href>
</interface>
```

3.5 FeatureType

The **FeatureType** definition is a simple type value pair used to model an NSA feature within the network. This type is left underspecified so that external values can be defined as additional features as protocol interfaces are introduced.

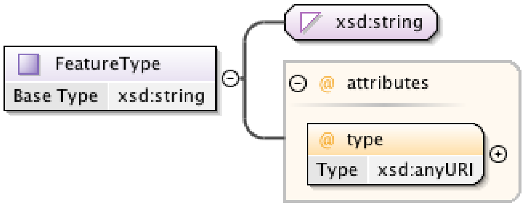


Figure 5 – FeatureType.

Parameters
The <*feature*> element is defined by the simple type ***FeatureType*** that has the following parameters (M = Mandatory, O = Optional):

Parameter	M/O	Description
type	M	Identifies the type of role modeled by the supplied value.
value	O	The optional string value associated with the type.

An NSA feature is a piece of metadata attached to the NSA Description Document that describes a specific capability offered by that NSA, or configuration value on that NSA that needs to be communicated to requester agents within the network. These features may be specifically associated with an instance of NSI protocol interface on the NSA, or may be associated with the NSA entity itself.

Features associated with a specific NSI protocol, such as the NSI Connection Service, which take advantage of this <feature> element, should define their values in that protocol's specification document, while more generic features can be captured in this document as they are defined.

We formally define three values to model an NSA's CS "role" within the network as shown below:

```
<feature type="org.ogf.nsi.cs.v2.role.aggregator"/>
```

An NSA MUST include in its Description Document a <*feature*> element of this type if the NSA is performing an aggregator NSA role as defined in the NSI CS Version 2 specification. Presence of this <*feature*> element type communicates the NSA's willingness to perform reservation path finding and CS protocol message forwarding through to connected peers on the control plane. In addition, the NSA must populate all control plane peered NSA using the <*peersWith*> element. This will allow a remote NSA to determine control plane paths to this aggregator, and control plane reachability through the aggregator to other networks.

```
<feature type="org.ogf.nsi.cs.v2.role.uPA"/>
```

An NSA MUST include in its Description Document a <*feature*> element of this type if the NSA is performing uPA NSA role as defined in the NSI CS Version 2 specification. In addition, the NSA must populate all control plane peered NSAs using the <*peersWith*> element. This will allow a remote NSA to determine control plane paths to the uPA.

```
<feature type="org.ogf.nsi.cs.v2.role.uRA"/>
```

An NSA MUST include in its Description Document a <*feature*> element of this type if the NSA is performing uRA NSA role as defined in the NSI CS Version 2 specification. In addition, the NSA must populate all control plane peered NSAs using the <*peersWith*> element.

An NSA entity will typically support a single NSI CS role, however, multi-function NSA supporting all three features is possible. In this case, all three roles would be defined in the NSA Description Document instance for that NSA.

The previous <*feature*> definitions had no value associated with the feature type. As an example to illustrate a type/value pair, it is possible to model the NSI CS 2.0 reservation commit timeout value for an NSA as follows:

```
<feature type="org.ogf.nsi.cs.v2.commitTimeout">120</feature>
```

3.6 PeersWithType

The **PeersWithType** definition is modeling the NSI Connection Services RA->PA unidirectional relationship between an NSA and its peer. This type consists of an NSA role enumeration (RA or PA), as well as an anyURI value holding the NSA identifier of the remote peer NSA of this relationship. This information, in combination with the NSA role type feature information, can be used to build a directed graph of control plane connectivity for the purpose of routing Connection Service reservation messages to a destination NSA.

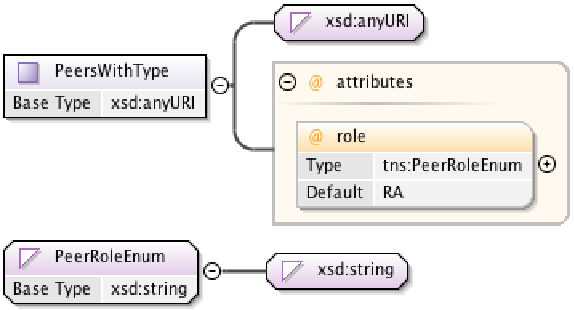


Figure 6 – PeersWithType.

<u>Parameters</u>

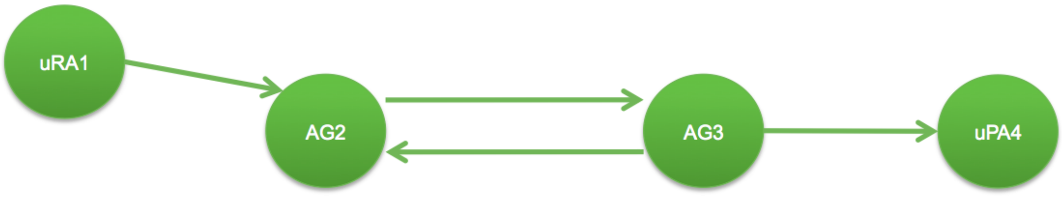
The <*peersWith*> element is defined by the simple type ***PeersWithType*** that has the following parameters (M = Mandatory, O = Optional):

Parameter	M/O	Description
role	O	Identifies the source NSA role (RA or PA) in the peering relationship. If a role value is not provided
value	M	The NSA identifier of the remote peer NSA that is the target of this peering relationship.

The **PeerRoleEnum** definition is an enumeration with the following two values:

- A value of “RA” implies the <*peersWith*> element represents an RA->PA relationship with the local NSA in the RA role and the target peer in the PA role (AG or uPA).
- A value of “PA” implies the <*peersWith*> element represents an RA->PA relationship with the local NSA in the PA role and the target peer in the RA role (AG or uRA).

Figure 7 below shows a simple example control plane interconnection and their associated <*peersWith*> elements.



uRA 1:
<peersWith role=“RA”>AG2</peersWith>

AG3:
<peersWith role=“PA”>AG2</peersWith>
<peersWith role=“RA”>AG2</peersWith>
<peersWith role=“RA”>uPA4</peersWith>

AG2:
<peersWith role=“PA”>uRA1</peersWith>
<peersWith role=“RA”>AG3</peersWith>
<peersWith role=“PA”>AG3</peersWith>

uPA4:
<peersWith role=“PA”>AG3</peersWith>

Figure 7 – Example <peersWith> elements.

In the above example we see a single uRA (uRA1), two aggregator NSA (AG2, and AG3), and one uPA (uPA4). The following can be noted from the diagram:

- uRA1 has a single <*peersWith*> element describing its RA role with AG2. AG2 has a similar <*peersWith*> element describing its PA role with uRA1. Based on these two entries we get a single unidirectional link from uRA1 to AG2 when building the control plane graph.
- uPA4 has a single <*peersWith*> element describing its PA role with AG3. AG3 has a similar <*peersWith*> element describing its RA role with uPA4. Based on these two entries we get a single unidirectional link from AG3 to uPA4 when building the control plane graph.
- AG2 and AG3 have two additional <*peersWith*> elements describing their RA and PA roles in the bidirectional peering relationship.

The following additional rules apply to the use of the <*peersWith*> element:

- Two AG with a bidirectional peering MUST each contain a <*peersWith*> element to model the bidirectional relationship. Unidirectional relationships are allowed between AG.
- A uPA MUST describe any peering relationships with <*peerWith*> elements with a role set to “PA”.
- Feature type of AG+uPA has an implicit <*peersWith*> for itself with an RA->PA relationship from the AG to the uPA.
  Feature type of uRA+AG has an implicit <*peersWith*> for itself with an RA->PA relationship from the uRA to the AG.

3.7 HolderType

The **HolderType** definition is a simple holder type for inclusion of elements and attributes from external namespaces.

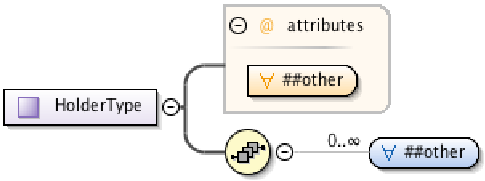


Figure 8 – HolderType.

4 Interface Versioning

Versioning within the NSI suite of protocols utilizes Internet Assigned Numbers Authority (IANA) MIME Media Types as a standard mechanism for distinguishing between releases of each protocol. For the REST protocol specifications based on HTTP these media types are used directly in the protocol via the *Content-Type* and *Accept* header parameters, whereas for SOAP the values are used in a protocol version field. Which of the two mechanisms are used is left up to the protocol profile itself. For example, the current NSI CS 2.0 profile [OGF NSI-CS] utilizes SOAP over HTTP as a transport that has a standard MIME Media Type of “application/soap+xml”. We have created a custom Media Type for the NSI CS 2.0 SOAP profile to distinguish this protocol, however, it is only used in the *protocolVersion* field of the SOAP header and not the *Content-Type* field of the HTTP header that still remains “application/soap+xml”.
Table 1 below enumerates the MIME Media Types defined for versions of the NSI protocol suite, and the specific protocol interface role the NSA supports. An NSA will populate values in the interface elements (*InterfaceType*) of their NSA Description Document.

Version	Interface	MIME Media Type
NSI CS version 1.0	Provider	"application/vnd.ogf.nsi.cs.v1.provider+soap"
NSI CS version 1.0	Requester	"application/vnd.ogf.nsi.cs.v1.requester+soap"
NSI CS version 1.1	Provider	"application/vnd.ogf.nsi.cs.v1-1.provider+soap"
NSI CS version 1.1	Requester	"application/vnd.ogf.nsi.cs.v1-1.requester+soap"
NSI CS version 2.0	Provider	"application/vnd.ogf.nsi.cs.v2.provider+soap"
NSI CS version 2.0	Requester	"application/vnd.ogf.nsi.cs.v2.requester+soap"
NSI Topology version 1.0	Provider	"application/vnd.ogf.nsi.topology.v1+xml"
NSI Topology version 2.0	Document	"application/vnd.ogf.nsi.topology.v2+xml"
NSA Description Document version 1.0	Document	"application/vnd.ogf.nsi.nsa.v1+xml"
NSI Document Distribution Service version 1.0	Requester/ Provider	"application/vnd.ogf.nsi.dds.v1+xml"

Table 1 – NSI CS protocol version MIME Media Types.

5 Optionality

An NSA Description Document SHOULD be created and made available via the NSA Document Distribution Service to all interested NSA within the interconnected network. However, the NSA Description Document contains information that would typically be considered configuration information, or may have been previously hard coded within NSA implementations, so there can be situations where this information MAY be manually provisioned an NSA. The following statements are made to help guide implementations.

An Ultimate Provider NSA (uPA) MUST participate in the NSA Document Distribution Service and make available an NSA Description Document describing the available interfaces, capabilities, and networks managed. The uPA will only participate in the provider role within the NSA Document Distribution Service as it does not contain a requester component, and therefore will not need to discover documents from other NSAs. This will allow peer NSA (Requester roles) to dynamically bootstrap communications.

An Aggregator NSA (AG) MUST participate in the NSA Document Distribution Service and make available an NSA Description Document describing the available interfaces and capabilities (AG do not directly manage network resources). An AG contains both a requester and provider component, so SHOULD support the requester roles within the NSA Document Distribution Service. This will allow peer NSA (Requester roles) to dynamically bootstrap communications with the AG's provider role, and allow the AG to dynamically bootstrap communications with its peers in the requester role.

An Ultimate Requester NSA (uRA) may participate in the NSA Document Distribution Service as a requester to bootstrap communications with an AG or uPA, but will never participate as a provider, so does not need to make an NSA Description Document available. The uRA may decide not to participate in the NSA Document Distribution Service, but instead choose to statically provision all information required to bootstrap communications with the target AG or uPA.

6 Security Considerations

This document describes the information modeled within the NSA Description Document, but does not define the specific mechanism that is used by an NSA to get access to all documents within the network. The NSA Description Document MUST be verifiable (e.g. the agent MUST be able to determine that the contents of the NSA Description Document were not altered during delivery). It is also required that the exchange of documents between NSA is secured to the level of other protocols within the NSI protocol suite such as the NSI Connection Services protocol and its use of TLS. This security MUST include authentication, authorization, and confidentiality. How this is achieved is out of scope for this document.

7 Glossary

Aggregator NSA (AG)	The Aggregator NSA is a Provider Agent that acts as both a requester and provider NSA. It can service requests from other NSA, perform path finding, and distribute segment requests to child NSA for processing.
Connection Service (CS)	The NSI Connection Service is a service that allows an RA to request and manage a Connection from a PA. See [OGF NSI-CS].
Document Distribution Service (DDS)	The Document Distribution Service is a protocol for the exchange and propagation of NSA meta data between NSA throughout the interconnected control plane. The NSA Description document is an example of information exchanged using this protocol.
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 RAs and PAs. The NSI defines a set of interactions or transactions between these NSAs to realize a Network Service.
Network Services Framework (NSF)	The Network Services framework describes an NSI message-based platform capable of supporting a suite of Network Services such as the Connection Service and the Topology Service. See [OGF NSF].
NSA Description Document	The NSA Description Document encapsulates descriptive meta-data associated with an NSA such as all NSI services and associated protocol interfaces offered by the NSA.
NSI Topology	The NSI Topology defines a standard ontology and a schema to describe network resources that are managed to create the NSI service. The NSI Topology as used by the NSI CS (and in future other NSI services) is described in [OGF NSI-TS].
Requester/Provider Agent (RA/PA)	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 Agent (RA). When an NSA realizes a service, it is called a Provider Agent (PA). A particular NSA may act in different roles at different interfaces.
NSI Service Definition	A document describing the service offered by an NSA and it's underlying network. A network can offer multiple services, and therefore, have multiple Service Definitions defined.
Simple Object Access Protocol (SOAP)	SOAP is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks.
Ultimate PA (uPA)	The ultimate PA is a Provider Agent that has an associated NRM.
Ultimate RA (uRA)	The Ultimate RA is a Requester Agent is the originator of a service request.
XML Schema Definition (XSD)	XSD is a schema language for XML. See [W3C XSD]

eXtensible Markup Language (XML) XML is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

8 Contributors

John H. MacAuley, ESnet, macauley@es.net
Henrik Thostrup Jensen, NORDUnet, htj@nordu.net
Guy Roberts, GÉANT Association, guy.roberts@dante.net

9 Intellectual Property Statement

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights, which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director.

10 Disclaimer

This document and the information contained herein is provided on an “As Is” basis and the OGF disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

11 Full Copyright Notice

12 References

[BRADNER] Scott Bradner. Key Words for Use in RFCs to Indicate Requirement Levels, RFC 2119. The Internet Society. March 1997. http://tools.ietf.org/html/rfc2026

[RFC 6350] Simon Perreault. vCard Format Specification RFC 6350 (Standards Track), August 2011. URL http://tools.ietf.org/html/rfc6350.

[RFC 6351] S. Perreault. xCard: vCard XML Representation RFC 6351 (Standards Track), August 2011. URL http://tools.ietf.org/html/rfc6351.

[OGF NSF] Guy Roberts, et al. “OGF Network Service Framework v2.0”, Group Working Draft (GWD), candidate Recommendation Proposed (R-P), January 28, 2014.

[OGF NSI-CS] Guy Roberts, et al. “OGF NSI Connection Service v2.0”, Group Working Draft (GWD), candidate Recommendation Proposed (R-P), January 12, 2014.

[OGF NSI-TS] Jeroen van der Ham, GWD-R-P Network Service Interface Topology Representation, Group Working Draft (GWD), candidate Recommendations Proposed (R-P), January 2013.

[OGF NSI-DS] John MacAuley, et al. “Network Service Interface Document Distribution Protocol v1.0”, Group Working Draft (GWD), candidate Recommendation Proposed (R-P), February 18, 2014.

[OGF NML] OGF GFD.206: Network Markup Language Base Schema version 1, http://www.gridforum.org/documents/GFD.206.pdf

[W3C XSD] W3C XML “Schema Definition Language (XSD) 1.1 Part 2: Datatypes”, http://www.w3.org/TR/xmlschema11-2/#anyURI

13 Appendix A: NSA Description Document schema

The following XML schema forms a normative part of this recommendation and is captured from the *ogf_nsi_nsa_description_v1_0.xsd* schema file that is the official source for the NSA Description Document schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
The OGF takes no position regarding the validity or scope of any intellectual
property or other rights that might be claimed to pertain to the implementation
or use of the technology described in this document or the extent to which any
license under such rights might or might not be available; neither does it
represent that it has made any effort to identify any such rights.  Copies of
claims of rights made available for publication and any assurances of licenses
to be made available, or the result of an attempt made to obtain a general
license or permission for the use of such proprietary rights by implementers or
users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights,
patents or patent applications, or other proprietary rights which may cover
technology that may be required to practice this recommendation.  Please
address the information to the OGF Executive Director.

This document and the information contained herein is provided on an "As Is"
basis and the OGF disclaims all warranties, express or implied, including but
not limited to any warranty that the use of the information herein will not
```

```
infringe any rights or any implied warranties of merchantability or fitness
for a particular purpose.

Copyright (C) Open Grid Forum (2009-2012). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and
derivative works that comment on or otherwise explain it or assist in its
implementation may be prepared, copied, published and distributed, in whole or
in part, without restriction of any kind, provided that the above copyright
notice and this paragraph are included on all such copies and derivative works.
However, this document itself may not be modified in any way, such as by removing
the copyright notice or references to the OGF or other organizations, except as
needed for the purpose of developing Grid Recommendations in which case the
procedures for copyrights defined in the OGF Document process must be followed,
or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by
the OGF or its successors or assignees.

Open Grid Forum NSI NSA Discovery Document v1.0.

Description: This is the NSA Discovery Document schema defined for use in the
OGF NSI Discovery Service v1.0. Comments and questions can be directed to the
mailing list group mailing list (nsi-wg@ogf.org).
-->
<xsd:schema targetNamespace="http://schemas.ogf.org/nsi/2014/02/discovery/nsa"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xcard="urn:ietf:params:xml:ns:vcard-4.0"
    xmlns:tns="http://schemas.ogf.org/nsi/2014/02/discovery/nsa"
    version="1.0">
    <xsd:annotation>
        <xsd:appinfo>ogf_nsi_discovery_nsa_v1_0.xsd 2014-12-01</xsd:appinfo>
        <xsd:documentation xml:lang="en">
            This is an XML schema document describing the NSA element of the
            OGF NSI Interface Discovery Protocol v1.0.  There is a corresponding
            document providing a description of the RESTful service definition
            and protocol specific types.

            Within the NSI reference architecture the Network Services Agent
            (NSA) is an entity that offers network services.  These services
            can be varied in functionality, and an NSA does not need to offer
            all services defined within a network.  For example, one NSA may
            offer Connection Services and Topology Services for a specific
            network, while a second NSA offers Monitoring Services for that
            same network.  In addition, the versions of the services offered
            can vary from NSA to NSA.  The NSI Discovery Protocol is a metadata
            service designed to enable self-description of all NSI services
            and associated protocol interfaces offered by these NSA.

            The NSI Discovery schema allows an NSA to describe the
            interfaces and versions of interfaces that it supports.  Through
            the REST API access methods defined, an NSA can dynamically
            discover interfaces and capabilities supported by a peer NSA,
            perform protocol version negotiation based on the supplied
            metadata, identify protocol endpoints, and bootstrap peer
            communications with minimal configuration.

            This document encapsulates the types used to model meta-data
            associated with an NSA.  By defining these type in a separate
            namespace, it is hoped that new versions of the meta-data will not
            impact the base discovery protocol.
        </xsd:documentation>
    </xsd:annotation>

    <!-- Import additional standard name spaces. -->
    <xsd:import namespace="urn:ietf:params:xml:ns:vcard-4.0"
        schemaLocation="xCard.xsd"/>

    <!-- *****************************************************************
```

```
         *                      XML element types                       *
         ***************************************************************** -->

    <!-- NSA resource definition. -->
    <xsd:element name="nsa" type="tns:NsaType" />

    <!-- *****************************************************************
         *                        XML base types                        *
         ***************************************************************** -->

    <xsd:complexType name="NsaType">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                This is the type definition for meta-data associated with an
                NSA resource.

                Attributes:

                id - The globally unique NSA identifier for this resource.

                version - The version of this NSA resource based on the date
                and time the entry was created at the source NSA.  This
                attribute can be used to compare two versions of the document
                for equality (same version) or to determine the new and older
                versions through date comparison.

                expires - The date this version of the document expires and should
                no longer be used.

                anyAttribute - Permit inclusion of attributes from other namespaces
                for flexible extension without needing to update this schema
                definition.

                Elements:

                name - A descriptive name for this NSA resource.  This value is
                typically used for display purposes.

                softwareVersion - A descriptive string describing the NSA software
                type and version.   This value will allow a peer NSA to adapt
                behaviors to specific versions of an NSA when required.

                startTime - The time at which this NSA last started to provide
                uninterrupted service.   This is effectively the last restart
                time of the NSA.  A peer discovering a change in this value
                can initiate recovery procedures.

                adminContact - A list of zero or more administrative contacts
                associated with this NSA.

                location - The physical location of the logical NSA resource.
                This can be the location of the server hosting the NSA, or
                some other location related to the service being offered.

                networkId - A list of zero or more network identifiers for which
                this NSA is providing the listed service interfaces and
                features.  These network identifiers can be mapped into network
                topology to determine the network resources being managed by
                this NSA.

                interface - A list of zero or more service interfaces supported
                by the NSA.

                peersWith - A list of zero or more NSA entries enumerating the
                peer NSA that have set up a trusted control plane relationship
                with this NSA.  Each entry in this list represents a trusted
                unidirectional relationship with the direction described by the
                "role" attribute associated with the peersWith element.
                other - Provides a flexible mechanism allowing additional elements
                to be provided from other namespaces without needing to update
                this schema definition.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="name" type="xsd:string" minOccurs="0" />
            <xsd:element name="softwareVersion" type="xsd:string" minOccurs="0" />
            <xsd:element name="startTime" type="xsd:dateTime" minOccurs="0" />
            <xsd:element name="adminContact" type="xcard:VcardsType" minOccurs="0" />
            <xsd:element name="location" type="tns:LocationType"  minOccurs="0" />
            <xsd:element name="networkId" type="xsd:anyURI" minOccurs="0"
                maxOccurs="unbounded" />
            <xsd:element name="interface" type="tns:InterfaceType" minOccurs="0"
                maxOccurs="unbounded" />
            <xsd:element name="feature" type="tns:FeatureType" minOccurs="0"
                maxOccurs="unbounded" />
            <xsd:element name="peersWith" type="tns:PeersWithType" minOccurs="0"
                maxOccurs="unbounded" />
            <xsd:element name="other" type="tns:HolderType" minOccurs="0"
                maxOccurs="unbounded" />
        </xsd:sequence>
        <xsd:attribute   name="id"      use="required" type="xsd:anyURI" />
        <xsd:attribute   name="version" use="required" type="xsd:dateTime" />
        <xsd:attribute   name="expires" use="optional" type="xsd:dateTime" />
        <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>

    <xsd:complexType name="LocationType">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                This is a type definition modeling the location of an NSA.  A
                Location is a reference to a geographical location or area for
                the NSA.

                Elements:

                name  - A human readable string naming this location.

                longitude - The longitude of the NSA in WGS84 coordinate system
                (in decimal degrees).

                latitude - The latitude of the NSA in WGS84 coordinate system (in
                decimal degrees).

                altitude  - The altitude of the NSA in WGS84 coordinate system (in
                decimal meters).

                unlocode - The UN/LOCODE location identifier for the NSA
                location.

                address - The address of the NSA location specified using the
                vCard address format.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:all>
            <xsd:element name="name"       type="xsd:string"        minOccurs="0" />
            <xsd:element name="longitude"  type="xsd:float"         minOccurs="0" />
            <xsd:element name="latitude"   type="xsd:float"         minOccurs="0" />
            <xsd:element name="altitude"   type="xsd:float"         minOccurs="0" />
            <xsd:element name="unlocode"   type="xsd:string"        minOccurs="0" />
            <xsd:element name="address"    type="xcard:adrPropType" minOccurs="0" />
        </xsd:all>
    </xsd:complexType>

    <xsd:complexType name="InterfaceType">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                Type definition that models an NSA protocol interface.  This
                type encapsulates the meta-data needed to determine the version,
                location, and schema associated with a specific NSA interface.

                Elements:

                type - The unique string identifying the type and version of
                the NSA interface.  Application Internet media types
                (Content-types) are used to identify the NSI interface,
                version, and supported encoding type.  For example, the
                first string below identifies the NSI Interface Discovery
                Protocol Version 1 XML encoded representation, while the
                second string identifies the same protocol and version, but
                the JSON representation:

                type="application/vnd.ogf.nsi.discovery.v1+xml"
                type="application/vnd.ogf.nsi.discovery.v1+json"

                href - This attribute contains the protocol endpoint for the
                interface identified in this interface reference.  For example, the
                following URL provides the protocol endpoint for the interface type
                identified in this interface reference.

                href="https://nsa.ogf.org/discovery"

                describedBy - This attribute contains a reference to the WSDL
                or WADL file corresponding to this interface's version (if
                available).  For example, the following URL provides the
                location for a WADL description of the NSI Interface Discovery
                Protocol Version 1.

                describedBy="https://nsa.ogf.org/discovery/wadl"
            </xsd:documentation>
        </xsd:annotation>
        <xsd:sequence>
            <xsd:element name="type"        type="xsd:string" />
            <xsd:element name="href"        type="xsd:anyURI" />
            <xsd:element name="describedBy" type="xsd:anyURI" minOccurs="0" />
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="FeatureType">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                Type definition for an NSA feature within the network.  This type
                is left underspecified so that external values can be defined
                as additional features and protocol interfaces are introduced.

                As an example we can model the NSA's CS "role" within the network
                as shown below:

                <feature type="org.ogf.nsi.cs.v2.role.aggregator"/>
                <feature type="org.ogf.nsi.cs.v2.role.uPA"/>
                <feature type="org.ogf.nsi.cs.v2.role.uRA"/>

                We could also model the NSI CS 2.0 reservation commit timeout
                value for an NSA:

                <feature type="org.ogf.nsi.cs.v2.commitTimeout">120</feature>

                Attributes:

                type - Identifies the type of role modeled by the supplied
                value.

                value - The optional string value associated with the type.
            </xsd:documentation>
        </xsd:annotation>
        <xsd:simpleContent>
            <xsd:extension  base="xsd:string">
```

```
                    <xsd:attribute  name="type"  type="xsd:string"  use="required"/>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>

        <xsd:complexType name="PeersWithType">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                    Type definition models a unidierctional trusted control plane
                    relationship between an NSA and its peer.

                    Attributes:

                    role - Identifies the directionality of the peering relationship
                    being modeled by the value.  A value of "RA" indicates the NSA
                    represented by the document is performing the RA role (source NSA)
                    in the unidirectional relationship.   A value of "PA" indicates
                    the NSA is performing the PA role (destination NSA) in the
                    unidirectional relationship.

                    value - The NSA identifier of the remote peer NSA modeled by
                    this relationship.
                </xsd:documentation>
            </xsd:annotation>
            <xsd:simpleContent>
                <xsd:extension  base="xsd:anyURI">
                    <xsd:attribute  name="role"  type="tns:PeerRoleEnum"  default="RA"/>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>

        <xsd:simpleType name="PeerRoleEnum">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                    Type enumerating the role of a unidirectional trusted
                    control plane relationship.
                </xsd:documentation>
            </xsd:annotation>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="RA" />
                <xsd:enumeration value="PA" />
            </xsd:restriction>
        </xsd:simpleType>

        <xsd:complexType name="HolderType">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                    A simple holder type for inclusion of elements and attributes
                    from external namespaces.  This separate type is required to
                    get around bugs in Libxml2 library used by command line tools
                    like xmllint.
                </xsd:documentation>
            </xsd:annotation>
            <xsd:sequence>
                <xsd:any namespace="##other" processContents="lax" minOccurs="0"
                    maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:anyAttribute namespace="##other" processContents="lax" />
        </xsd:complexType>
    </xsd:schema>
```