

# Contents

<b>1</b>	<b>About Document</b>	<b>2</b>
1.1	Document Attribute . . . . .	2
1.2	Revision History . . . . .	3
<b>2</b>	<b>Description</b>	<b>3</b>
<b>3</b>	<b>Definitions</b>	<b>3</b>
<b>4</b>	<b>Interface Application Roles</b>	<b>4</b>
<b>5</b>	<b>Interface Details</b>	<b>4</b>
<b>6</b>	<b>Interface Overview</b>	<b>4</b>
6.1	Create SNIRO Optimization Service Instance . . . . .	5
<b>7</b>	<b>Flow Diagrams</b>	<b>6</b>
<b>8</b>	<b>SNIRO Request Handling</b>	<b>6</b>
8.1	SNIRO Create Optimization Request Structure . . . . .	6
8.2	Sample SNIRO Create Optimization Request Structure . . . . .	7
<b>9</b>	<b>SNIRO Response Handling</b>	<b>8</b>
9.1	Success Response to POST, PUT, and DELETE . . . . .	8
9.2	Sample SNIRO Success Response . . . . .	9
9.3	Async Request Status . . . . .	9
9.4	Sample SNIRO Async Request Status . . . . .	9
<b>10</b>	<b>Interface Specification</b>	<b>11</b>
10.1	Common API Definition . . . . .	11
10.1.1	HTTP Request Header . . . . .	12
10.1.2	Request Information Structure . . . . .	12
10.1.3	Placement Information Structure . . . . .	13
10.1.4	Model Information Structure . . . . .	13
10.1.5	Subscriber Information Structure . . . . .	14
10.1.6	VNF Information Structure . . . . .	14
10.1.7	VPN Information Structure . . . . .	15
10.1.8	Service Information Structure . . . . .	15
10.1.9	DHV Service Information Structure . . . . .	15
10.1.10	Demand Information Structure . . . . .	16
10.1.11	Preferred Candidate Information Structure . . . . .	17
10.1.12	Solution Information Structure . . . . .	17

10.1.13 License Information Structure . . . . . 18

10.1.14 Composite Solution Information Structure . . . . . 18

10.1.15 Demand Solution Information Structure . . . . . 18

**11 Error Handling 19**

11.1 Optimization Service Exception . . . . . 20

11.2 Optimization Policy Exception . . . . . 20

**12 Error Codes 21**

**13 Sample Policy Error Message 21**

**14 Sample Optimization Service Error Message 22**

**15 Authentication 22**

15.1 Authorization . . . . . 22

15.2 Encryption . . . . . 23

15.3 Firewall . . . . . 23

**16 Disaster Recovery 23**

**17 Testing Consideration 23**

**18 Related Documents 23**

**19 Acceptance and Approval 23**

19.1 Overview . . . . . 23

19.2 Approvers . . . . . 24

# 1 About Document

## 1.1 Document Attribute

Attribute	Value
Application Name	SNIRO
Application ID	26699
Owner	SNIRO Team
Contact Information	{ankit,sastry}@research.att.com
Services	Placement

## 1.2 Revision History

Author	Date	Version	Description
Ankitkumar Patel	12/04/2016	1	Initial version and review copy

## 2 Description

SNIRO (Service, Network, Infrastructure, and Resource Optimization) is an overlay ECOMP entity that offers optimization as a service. This document describes the SNIRO northbound interface through which a consumer/client can leverage optimization services from SNIRO. This document is referred as Application Interface Design (AID), which communicates the selected interface design to stakeholders and to the design and development teams. The SNIRO NB interface described in this document conforms to the AT&T AID Specifications.

## 3 Definitions

This section defines the nomenclatures used in the context of this AID.

- **Entity:** An element with distinct and independent existence. An entity can be a human, computer hardware, or computer software.
- **Event:** An entity generated action or occurrence recognized and/or handled by software.
- **Transaction:** A sequence of information exchange and event handling as a unit for the purpose of handling a request.
- **Consumer:** An entity that initiates a transaction. The consumer role remains constant throughout a transaction.
- **Provider:** An entity that performs operations for an initiated transaction. The provider role remains constant throughout a transaction.
- **Interface:** A point where two entities interact and exchange information.
- **On-Demand:** An interface between a consumer and a provider in which at an independent transaction granularity, which can be handled instantaneously.
- **Batch:** An interaction between a consumer and a provider in which more than one transactions are handled in a batch on a periodic basis.
- **Object-Oriented:** An interface that is implemented with CORBA or EJB technologies.

- **Tag-Based:** An interface that uses text based messages where the fields are delimited with tags (e.g XML, FCIF). Web services are not considered Tag-based interfaces.
- **Web-Service:** An interface that uses RESTful APIs (payload=JSON).

## 4 Interface Application Roles

Application Name	Application ID	Role
SNIRO	26699	Provider
MSO	22881	Consumer

## 5 Interface Details

- Interface Type
  - Batch
  - On-Demand
- Interface Technology
  - Object-Oriented
  - Tag-Based
  - Web Services
  - Database
  - JMS
  - Other

## 6 Interface Overview

SNIRO offers optimization services to consumers. All the service request invocations are inherently handled in an asynchronous manner by SNIRO. As SNIRO gets a request for optimization, SNIRO responds synchronously to the consumer request. The HTTP synchronous response to a request normally indicates whether or not the request has been accepted by SNIRO. A consumer is expected to provide a callback URL in the request to which SNIRO will post notifications and results.

The Consumer-SNIRO interface is of type asynchronous REST. A consumer can make a RESTful call to SNIRO. Once SNIRO accepts the call, SNIRO returns 202 HTTP Accepted

response on the connection established by a consumer. Once results are ready, SNIRO can follow up by one or more asynchronous result notifications to the callback URL sent by a consumer in the original request. SNIRO includes the state of a request (requestState element) in the content of a callback. Based on the content of this requestState element (i.e. "InProgress"), a consumer should determine whether it must wait for another callback from SNIRO. On the other hand, if SNIRO cannot accept the call, SNIRO returns 400 HTTP Bad Request response. In case, if SNIRO cannot find results in the requested time interval, SNIRO returns "timeOut" notification in the content of a callback. Refer to the SNIRO Response Handling Section for potential responses returned by SNIRO to a consumer.

SNIRO interface is a RESTful Webservice.

Operation Action	HTTP Verb	Resource URI
Create Request	POST	/sniro/api/v1/placement

In addition, asynchronous callbacks are supported. The callback URL must be provided in a request.

Operation Action	HTTP Verb	Resource URI
Request Status Notification	POST	{callbackURL}

## 6.1 Create SNIRO Optimization Service Instance

1. A consumer sends a create optimization request with a callback URL to SNIRO.
2. SNIRO validates the request and auto-generates a request ID.
3. SNIRO responds HTTP 202 Accept synchronous response if the request is valid and also sends the request ID, otherwise, SNIRO responds HTTP 400 reject synchronous response with description.
4. SNIRO executes an optimization engine relevant to the request at the back-end.
5. SNIRO periodically posts the state of the request in the content of a callback.
6. SNIRO posts final results in the content of a callback provided by the consumer.

## 7 Flow Diagrams

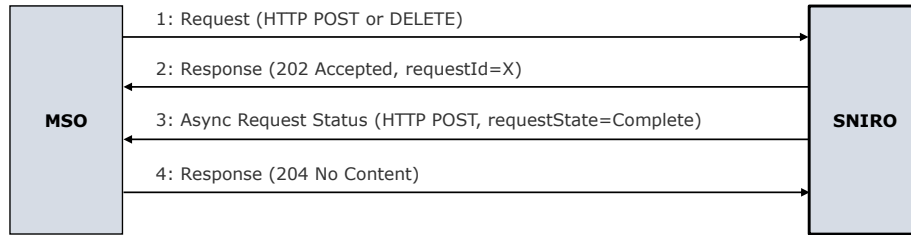


Figure 1: Successful Create Request

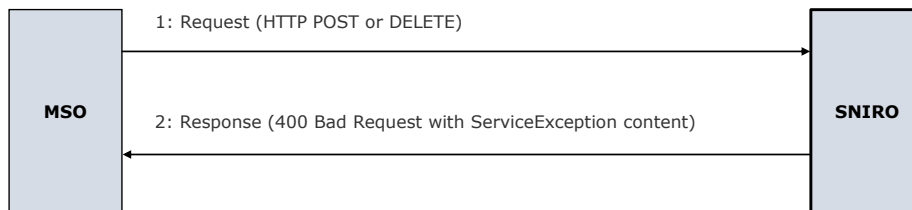


Figure 2: Error Prior to Sending a Synchronous Response

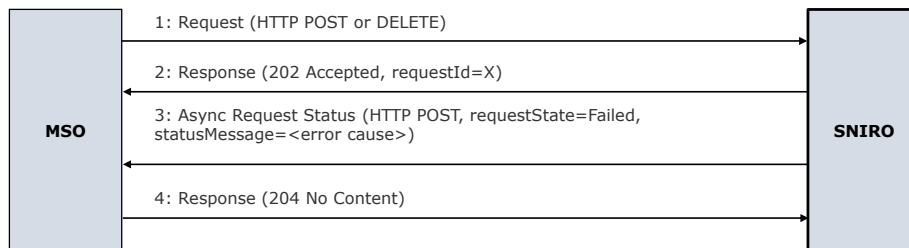


Figure 3: Error After Sending a Synchronous Response

## 8 SNIRO Request Handling

### 8.1 SNIRO Create Optimization Request Structure

SNIRO NB interface is expecting the following parameters as a part of a create placement optimization request.

Field Name	Tag Name	MOC	Format	Val	Notes
Create Request	createReq	M	Container		Create Request information.
{					Start of information structure.
Request Info{}	requestInfo	M	Container		SNIRO meta-info
Placement Info{}	placementInfo	O	Container		Placement info
}					End of information structure.

## 8.2 Sample SNIRO Create Optimization Request Structure

```
{
  "requestInfo": {
    "requestId": "xxx-xxx-x",
    "sourceId": "mso",
    "optimizer": "placement",
    "numSolutions": 1,
    "timeout": 600,
    "callbackUrl": "https://mso:5000/callback"
  },
  "placementInfo": {
    "modelInfo": {
      "modelType": "service",
      "modelInvariant": "258bd39d-97eb-44f5-a917-6ad6133047ca",
      "modelVersionId": "",
      "modelName": "VSRX",
      "modelVersion": "1.0",
      "modelCustomizationId": "asd"
    },
    "subscriberInfo": {
      "globalSubscriberId": "8310000058863",
      "subscriberName": "CocaCola",
      "subscriberCommonSiteId": "DALTX0101",
      "ucpeHostName": "USOSTCDALTX0101UJZZ11"
    },
    "policyId": ["distanceToLocationPolicy", "aicAttributePolicy"],
    "vnfInfo": {
      "vnfType": "VHNF",
      "vnfPartNumber": "24326",
      "nominalThroughput": "100",
      "vnfSoftwareVersion": "1.2",
      "vnfManagementOption": "asd"
    },
    "vpnInfo": { "vpnId": "vpn5335", "pvcId": "asd" },
    "serviceInfo": {
      "dhvServiceInfo": {
        "serviceInstanceId": "service4759347",
        "serviceType": "DHV",
        "e2evpnkey": "vpn-121212",
        "dhvSiteEffectiveTransportBandwidth": 100,
        "dhvIPSecTransportBandwidthUp": 50,
        "dhvIPSecTransportBandwidthDown": 100,
        "dhvIPSec2TransportBandwidthUp": 50,
        "dhvIPSec2TransportBandwidthDown": 100,
        "dhvVendorName": "velocloud"
      }
    },
    "demandInfo": [
      {
        "resourceInstanceType": "vHNPportalaaS",

```

```

        "resourceInstancePriSec": "PRIMARY",
        "pairNumber": "1",
        "preferredCandidateInfo": { "candidateId": "wertwert", "inventoryType": "aai" }
    },
    {
        "resourceInstanceType": "vHNPportalaaS",
        "resourceInstancePriSec": "SECONDARY1",
        "pairNumber": "aai",
        "preferredCandidateInfo": { "candidateId": "", "inventoryType": "wertwert" }
    },
    {
        "resourceInstanceType": "vHNGWaaS",
        "resourceInstancePriSec": "PRIMARY",
        "pairNumber": "1",
        "preferredCandidateInfo": { "candidateId": "wertwert", "inventoryType": "aai" }
    },
    {
        "resourceInstanceType": "vHNGWaaS",
        "resourceInstancePriSec": "SECONDARY",
        "pairNumber": "1",
        "preferredCandidateInfo": { "candidateId": "wertwert", "inventoryType": "aai" }
    },
    {
        "resourceInstanceType": "vVIGaaS",
        "resourceInstancePriSec": "PRIMARY",
        "pairNumber": "1",
        "preferredCandidateInfo": { "candidateId": "wertwert", "inventoryType": "aai" }
    },
    {
        "resourceInstanceType": "vVIGaaS",
        "resourceInstancePriSec": "SECONDARY",
        "pairNumber": "1",
        "preferredCandidateInfo": { "candidateId": "wertwert", "inventoryType": "aai" }
    },
    {
        "resourceInstanceType": "vVIGaaS",
        "resourceInstancePriSec": "PRIMARY",
        "pairNumber": "2",
        "preferredCandidateInfo": { "candidateId": "wertwert", "inventoryType": "aai" }
    },
    {
        "resourceInstanceType": "vVIGaaS",
        "resourceInstancePriSec": "SECONDARY",
        "pairNumber": "2",
        "preferredCandidateInfo": { "candidateId": "wertwert", "inventoryType": "aai" }
    }
}
]
}

```

## 9 SNIRO Response Handling

### 9.1 Success Response to POST, PUT, and DELETE

This is the content returned in response to requests that perform actions, i.e. POST, PUT, and DELETE requests.



Field Name	Tag Name	MOC	Format	Val	Notes
Optimization Request Reference	optimizationReqRef	M	Container		A synchronous response to an optimization request. Start of information structure.
{					
Request ID	requestId	M	String		ID of an optimization request.
}					End of information structure.

## 9.2 Sample SNIRO Success Response

```
{
  "requestId": "xxx-xx-xx"
}
```

## 9.3 Async Request Status

This is the content posted to the callback URL in asynchronous request scenario.

Field Name	Tag Name	MOC	Format	Val	Notes
Async Request Status	asyncRequestStatus	M	Container		The status of async request. Start of information structure.
{					
Request ID	requestId	M	String		The UUID of the optimization request.
Start Time	startTime	M	String		Date and time the request was created in GMT with format, "Wed, 15 Oct 2014 13:01:52 GMT".
Response Time	responseTime	M	String		Date and time the response was created in GMT with format, "Wed, 15 Oct 2014 13:01:52 GMT".
Request Type	requestType	M	Enum	{"create"}	The requested operation type.
Request State	requestState	M	Enum	{"complete", "inProgress", "failed"}	The state of a request.
Status Message	statusMessage	O	String		Additional descriptive status information. For failures, this will contain the error cause. For InProgress notifications, this will contain an informational message.
Percentage Progress	percentProgress	O	Integer		Percentage completed from 0 to 100.
Solution Info{}	solutionInfo	O	Container		Solutions for an optimization request.
}					End of information structure.

## 9.4 Sample SNIRO Async Request Status

```
{
  "requestId": "xxx-xx-xx",
  "startTime": "2016-12-13T12:00:00-05:00",
}
```

```
"responseTime": "2016-12-13T12:15:00-05:00",
"requestType": "create",
"requestState": "complete",
"statusMessage": "",
"percentProgress": "100",
"solutionInfo": {
  "placement": [
    [
      {
        "resourceInstanceType": "vHNPortalaaS",
        "resourceInstancePriSec": "PRIMARY",
        "pairNumber": 1,
        "inventoryType": "aicRegion",
        "cloudRegionId": "aic-1-34",
        "cloudOwner": "att_aic",
        "solutionId": 1,
        "vnfHostName": "host-12344",
        "serviceInstanceId": ""
      },
      {
        "resourceInstanceType": "vHNPortalaaS",
        "resourceInstancePriSec": "SECONDARY",
        "pairNumber": 1,
        "inventoryType": "aicRegion",
        "cloudRegionId": "aic-1-34",
        "cloudOwner": "att_aic",
        "solutionId": 1,
        "vnfHostName": "host-12344",
        "serviceInstanceId": ""
      }
    ],
    {
      "resourceInstanceType": "vHNGWaaS",
      "resourceInstancePriSec": "PRIMARY",
      "pairNumber": 1,
      "inventoryType": "aicRegion",
      "cloudRegionId": "aic-1-67",
      "cloudOwner": "att_aic",
      "solutionId": 1,
      "vnfHostName": "host-6746",
      "serviceInstanceId": ""
    },
    {
      "resourceInstanceType": "vHNGWaaS",
      "resourceInstancePriSec": "SECONDARY",
      "pairNumber": 1,
      "inventoryType": "aicRegion",
      "cloudRegionId": "aic-1-67",
      "cloudOwner": "att_aic",
      "solutionId": 1,
      "vnfHostName": "host-6746",
      "serviceInstanceId": ""
    }
  ],
  {
    "resourceInstanceType": "vVIGaaS",
    "resourceInstancePriSec": "PRIMARY",
    "pairNumber": 1,
    "inventoryType": "vnfInstance",
    "cloudRegionId": "vnf-1-89",
    "cloudOwner": "att_aic",
    "solutionId": 1,
    "vnfHostName": "host-6786",
    "serviceInstanceId": "si-6363456"
  }
}
```

```
    "resourceInstanceType": "vVIGaaS",
    "resourceInstancePriSec": "SECONDARY",
    "pairNumber": 1,
    "inventoryType": "vnfInstance",
    "cloudRegionId": "vnf-1-89",
    "cloudOwner": "att_aic",
    "solutionId": 1,
    "vnfHostName": "host-6786",
    "serviceInstanceId": "si-5678"
  },
  {
    "resourceInstanceType": "vVIGaaS",
    "resourceInstancePriSec": "PRIMARY",
    "pairNumber": 2,
    "inventoryType": "vnfInstance",
    "cloudRegionId": "vnf-1-89",
    "cloudOwner": "att_aic",
    "solutionId": 1,
    "vnfHostName": "host-6786",
    "serviceInstanceId": "si-1234"
  },
  {
    "resourceInstanceType": "vVIGaaS",
    "resourceInstancePriSec": "SECONDARY",
    "pairNumber": 2,
    "inventoryType": "vnfInstance",
    "cloudRegionId": "vnf-1-89",
    "cloudOwner": "att_aic",
    "solutionId": 1,
    "vnfHostName": "host-6786",
    "serviceInstanceId": "si-63456"
  }
]
],
"licenseInfo": { "featureGroupId": "fg123445tf23232" }
}
}
```

## 10 Interface Specification

### 10.1 Common API Definition

The API described in this document conforms to the AT&T Service Specifications which dictates the overall messaging structure. Definitions that are common to all APIs conforming to this specification are in this section, including:

- HTTP Request Header
- Request Information Structure
- Placement Information Structure
- Model Information Structure
- Subscriber Information Structure

- VNF Information Structure
- VPN Information Structure
- Service Information Structure
- DHV Service Information Structure
- Demand Information Structure
- Preferred Candidate Information Structure
- Solution Information Structure
- License Information Structure
- Composite Solution Information Structure
- Demand Solution Information Structure

### 10.1.1 HTTP Request Header

The header of a REST call to SNIRO must comply the following requirements.

Header	Required	Description
Accept	No	Defines the acceptable format(s) for response content. Only application/json is supported.
Authorization	Yes	Supplies Basic Authentication credentials for the request. If the Authorization header is missing, then an HTTP 400 Invalid Request response is returned. If the string supplied is invalid, then an HTTP 401 Unauthorized response is returned.
Content-Length	No	Number of bytes in the body of the request. Note that content length is limited to 1 MB.
Content-Type	Yes	Determines the format of the request content. Only application/json is supported.

### 10.1.2 Request Information Structure

SNIRO NB interface is expecting the following information structure as a part of every optimization request. <sup>1</sup>

<sup>1</sup>M-Mandatory, C-Conditional, O-Optional

Field Name	Tag Name	MOC	Format	Val	Notes
Request Info {	requestInfo	M	Container		The meta-info for SNIRO. Start of information structure.
Request ID	requestId	M	String		ID of a request.
Callback URL	callbackUrl	M	String		Client URL to use for async responses.
Source Id	sourceId	M	String		The Id of a source originating a request. For 1702, the value is “MSO”.
Optimizer	optimizer	M	Enum	{“placement”}	Optimizer identity.
Num. of Solutions	numSolutions	O	Integer		Number of requested solutions. The default value is 1.
Timeout	timeout	O	Integer		Time in minutes to return a solution.
}					End of information structure.

### 10.1.3 Placement Information Structure

SNIRO NB interface is expecting the following parameters as a part of placement optimization request.

Field Name	Tag Name	MOC	Format	Val	Notes
Placement Info {	placementInfo	M	Container		Information for placement optimization. Start of information structure.
Model Info{}	modelInfo	M	Container		Model metadata.
Subscriber Info{}	subscriberInfo	M	Container		Subscriber data.
Policy ID[]	policyId	M	List of String		Policy Ids associated with a request.
VNF Info{[]}[]	vnfInfo	M	List of Container		The information of VNFs requested for placement optimization.
VPN Info{}	vpnInfo	O	Container		The information of a VPN.
Service Info{}	serviceInfo	M	Container		The information of a service associated with VNFs.
Demand Info{[]}[]	demandInfo	M	List of Containers		The information of demand parameters for placement optimization.
}					End of information structure.

### 10.1.4 Model Information Structure

The following structure captures information related to service models.

Field Name	Tag Name	MOC	Format	Val	Notes
Model Info	modelInfo	M	Container		The meta data of a service model.
{					Start of information structure.
Model Type	modelType	M	Enum	{“service”}	The type of a model.
Model Invariant ID	modelInvariantId	M	String		Invariant UUID for the model name, irrespective of the version, as defined by ASDC (authoritative).
Model Version Id	modelVersionId	M	String		UUID for the model name and version combination, as defined in ASDC.
Model Name	modelName	M	String		Name of the model entity as defined in ASDC (non-authoritative).
Model Version	modelVersion	M	String		Version number of the model entity as defined in ASDC (non-authoritative).
Model Customization Id	modelCustomizationId	O	String		UUID for a customized set of parameters associated with a model in a given service context. Note: there can be multiple customizations per modelVersionId. CCD will not populate this in 1702..
}					End of information structure.

### 10.1.5 Subscriber Information Structure

This structure captures information about a subscriber of a service.

Field Name	Tag Name	MOC	Format	Val	Notes
Subscriber Info	subscriberInfo	M	Container		The information of a subscriber.
{					Start of information structure.
Global Subscriber Id	globalSubscriberId	M	String		A globally unique subscriberId provided by CCD, i.e. the global customer id (can be the same as Subscriber Name as long as its unique in A&AI).
Subscriber Name	subscriberName	O	String		Text name that goes along with the globalSubscriberId.
Subscriber Common Site Id	subscriberCommonSiteId	O	String		Id representing the subscribers location. CCD will populate this field with the Common Site Id received from SDNOM.
UCPE Host Name	ucpeHostName	M	String		Host name of uCPE.
}					End of information structure.

### 10.1.6 VNF Information Structure

The following structure captures the list of VNFs.

Field Name	Tag Name	MOC	Format	Val	Notes
VNF Info	vnfInfo	M	Container		Information on a VNF.
{					Start of information structure.
VNF Type	vnfType	M	Enum	{“vHNF”}	Type of a VNF.
VNF Part Number	vnfPartNumber	M	String		AT&T part number of the VNF.
VNF Nominal Throughput	nominalThroughput	M	String		Nominal Throughput. Specified in Mbps.
VNF Software Version	vnfSoftwareVersion	O	String		Software version of the VNF.
VNF Management Option	vnfManagementOption	O	Enum	{“ATT”, “Customer”}	Entity that manages the VNF.
}					End of information structure.

### 10.1.7 VPN Information Structure

The following structure captures the list of VNFs.

Field Name	Tag Name	MOC	Format	Val	Notes
VPN Info	vpnInfo	M	Container		Information on a VPN.
{					Start of information structure.
VPN Id	vpnId	M	String		Id of a VPN
PVC Id	pvcId	O	String		Id of a PVC.
}					End of information structure.

### 10.1.8 Service Information Structure

This structure captures the information on a service for which placement optimization is requested.

Field Name	Tag Name	MOC	Format	Val	Notes
Service Info	serviceInfo	M	Container		Information related to a service.
{					Start of information structure.
DHV service Info{}	dhvServiceInfo	C	Container		Information of a service if VNFs are associated with DHV.
}					End of information structure.

### 10.1.9 DHV Service Information Structure

This structure captures the information on a service for which placement optimization is requested.

Field Name	Tag Name	MOC	Format	Val	Notes
DHV Service Info	dhvServiceInfo	M	Container		Information related to the DHV service.
{					Start of information structure.
Service Instance Id	serviceInstanceId	M	String		Id of a service instance generated by MSO.
Service Type	serviceType	M	String		Type of service associated with a placement request.
End-to-End VPN Key	e2evpnkey	M	String		End-to-end VPN key for the DHV service.
DHV Site Effective Transport Bandwidth	dhvSiteEffectiveTransportBandwidth	M	Integer		Effective transport bandwidth in Mbps.
DHV IPsec Transport Bandwidth Up	dhvIPsecTransportBandwidthUp	M	Integer		Specified in Mbps. If both IPsec and IPsec2 Transport Bandwidth Up/Down are populated, this is an indication that there are to be 4 vVigs. Otherwise, there are to be 2 vVigs.
DHV IPsec Transport Bandwidth Down	dhvIPsecTransportBandwidthDown	M	Integer		This value is calculated based on DHV Bandwidth and the VLAN id. Specified in Mbps.
DHV IPsec2 Transport Bandwidth Up	dhvIPsec2TransportBandwidthUp	O	Integer		Specified in Mbps.
DHV IPsec2 Transport Bandwidth Down	dhvIPsec2TransportBandwidthDown	O	Integer		Specified in Mbps.
DHV Vendor Name	dhvVendorName	O	String		This value is derived from vHNF ATT Part Number received from SDNOM.
}					End of information structure.

### 10.1.10 Demand Information Structure

This structure captures information on demand parameters for the Placement optimization. If SNIRO finds duplicate demand information structure in a request, SNIRO treats them as two independent demand parameters for the Placement optimization.



Field Name	Tag Name	MOC	Format	Val	Notes
Demand Info {	demandInfo	M	Container		Demands parameters for Placement optimization. Start of information structure.
Resource Instance Type	resourceInstanceType	M	Enum	{“vVIGaaS”, “vHNPortalaas”, “vHNGWaaS”}	The type of a resource.
Resource Instance Primary Secondary	resourceInstancePriSec	M	Enum	{“PRIMARY”, “SEC- ONDARY”}	The category of a resource instance.
Pair Number	pairNumber	M	Enum	{“1”, “2”}	Identity of a pair of resources.
Preferred Candidate Info{ []	preferredCandidateInfo	O	List of Contain- ers		A list of preferred candidates.
}					End of information structure.

### 10.1.11 Preferred Candidate Information Structure

This structure captures the information on preselected candidates for the Placement Optimization. This structure is not in scope for 1702.

Field Name	Tag Name	MOC	Format	Val	Notes
Preferred Candidate Info {	preferredCandidateInfo	M	Container		Preferred candidates information. Start of information structure.
Inventory Type	inventoryType	M	Enum	{“vnfInstance”, “aicRegion”}	The type of inventory
Candidate ID	candidateId	M	String		ID of a candidate for the selected inventory type.
}					End of information structure.

### 10.1.12 Solution Information Structure

This is the content posted to the callback URL in asynchronous request scenario.

Field Name	Tag Name	MOC	Format	Val	Notes
Solution Info {	solution	M	Container		Solutions for a request. Start of information structure.
Composite Solution{[]	placement	O	List of Container		Solutions for placement optimization.
License Info{ }	licenseInfo	M	Container		License information of resources. End of information structure.

### 10.1.13 License Information Structure

The following structure captures the license information of resources.

Field Name	Tag Name	MOC	Format	Val	Notes
License Info	licenseInfo	M	Container		License information of resources.
{					Start of information structure.
Feature Group ID	featureGroupId	O	String		A&AT feature group ID of a resource.
}					End of information structure.

### 10.1.14 Composite Solution Information Structure

This structure captures a solution for each demand in relation with each other (referred as a composite solution).

Field Name	Tag Name	MOC	Format	Val	Notes
Composite Solution	compositeSolution	M	Container		A composite solution for given demands.
{					Start of information structure.
Demand Solution Info{ }[]		M	List of Container		Solution Set for given demands.
}					End of information structure.

### 10.1.15 Demand Solution Information Structure

This structure captures solutions on the requested demand parameters.

Field Name	Tag Name	MOC	Format	Val	Notes
Demand Solution Info	demandSolutionInfo	M	Container		Solution for given demands.  Start of information structure.
{					
Solution Id	solutionId	M	Integer		Id of a solution.
Resource Instance Type	resourceInstanceType	M	Enum	{“vVIGaaS”, “vHNPortalaaS”, “vHNGWaaS”}	The type of a resource.
Resource Instance Primary Secondary	resourceInstancePriSec	M	Enum	{“PRIMARY”, “SEC-ONDARY”}	The category of a resource instance.
Pair Number	pairNumber	M	Enum	{“1”, “2”}	Identity of a pair of resources.
Inventory Type	inventoryType	M	Enum	{“vnfInstance”, “aicRegion”}	The type of inventory
Service Instance ID	serviceInstanceId	C	String		VNF instance Id if an inventory type is vnfInstance.
VNF Host Name	vnfHostName	C	String		VNF host name if an inventory type is vnfInstance.
cloudRegionID	cloudRegionId	C	String		Cloud region Id if an inventory type is aicRegion.
Cloud Owner	cloudOwner	C	Enum	{“att_aic”}	Cloud owner of a candidate for the selected inventory type.
}					End of information structure.

## 11 Error Handling

Note that, in some scenarios, a consumer may receive error responses that originate from the network or SNIRO software layers. The content of these responses will conform to the exception schema specified here.

SNIRO sends exceptions to a consumer in response to invocation errors. An exception is an HTTP response. Two types of exceptions are defined: optimization service exceptions and optimization policy exceptions. The content is defined to have these fields:

Field Name	Data Types	Required?	Description
messageId	xs:string	Yes	Unique message identifier of the format ABCnnnn where ABC is either SVC for Service Exceptions or POL for Policy Exception. Exception numbers may be in the range of 0001 to 9999, where 0001 to 2999 are defined by OMA (see OMAs Common definitions for RESTful Network APIs for details) and 3000-9999 are available and undefined
text	xs:string	Yes	Message text, with replacement variables marked with %n, where n is an index into the list of <code>variables<sub>i</sub></code> elements, starting at 1
variable	xs:string	No	List of zero or more strings that represent the contents of the variables used by the message text.

When an optimization service is not able to process a request, and retrying the request

with the same information will also result in a failure, and if the issue is related to an optimization service issue, then the service will issue a fault using the optimization service exception fault message. Examples of optimization service exceptions include invalid input, lack of availability of a required resource or a processing error.

## 11.1 Optimization Service Exception

An optimization service exception uses the letters 'SVC' at the beginning of the message identifier. SVC service exceptions used by the Generic Event Listener API are defined below.

Message Id	Comment	Text	Variable	HTTP Code
SVC0001	General Service Error	Custom Error Message	None	400
SVC0002	Bad Parameter	Invalid input value for message part %1	%1: Message Part	400
SVC1000	No Server Resources	No server resources available to process a request	None	500
SVC2000	More elaborate version of SVC0001	The following service error occurred:%1. Error code is %2.	%1 is human readable description and %2 is an error code	400

When an optimization service is not able to complete because the request fails to meet a policy criteria, then the service will issue a fault using the optimization policy exception fault message. To clarify how an optimization policy exception differs from an optimization service exception, consider that all the input to an operation may be valid as meeting the required input for the operation (thus no optimization service exception), but using that input in the execution of the optimization service may result in conditions that require the service not to complete. Examples of optimization policy exceptions include privacy violations, requests not permitted under a governing service agreement, or input content not acceptable to the service provider.

## 11.2 Optimization Policy Exception

An Optimization Policy Exception uses the letters 'POL' at the beginning of the message identifier. POL policy exceptions used by the Generic Event Listener API are defined below.

Message Id	Comment	Text	Variable	HTTP Code
POL0001	General policy error	A policy error occurred	None	401
POL1009	User not provisioned for service	User has not been provisioned for service	None	401
POL1010	User suspended from service	User has been suspended for service	None	401
POL2000	More elaborate version of POL0001	The following policy error occurred : %1. Error code is %2.	%1 is the human readable description of the error and %2 is the error code	401
POL9003	Message size exceeds limit	Message content size exceeds the allowable limit	None	400

## 12 Error Codes

Code	Reason Phrase	Description
201	Created	The optimization solution is successfully created.
202	Accepted	The request has been accepted for processing, but the processing has not been completed. The request might or might not eventually be acted upon, as it might be disallowed when processing actually takes place.
400	Bad Request	Many possible reasons not specified by the other codes (e.g., missing required parameters or incorrect format). The response body may include a further exception code and text. HTTP 400 errors may be mapped to SVC0001 (general service error), SVC0002 (bad parameter), SVC2000 (general service error with details) or PO9003 (message content size exceeds the allowable limit).
401	Unauthorized	Authentication failed or was not provided. HTTP 401 errors may be mapped to POL0001 (general policy error) or POL2000 (general policy error with details).
404	Not Found	Not Found The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent.
405	Method Not Allowed	A request was made of a resource using a request method not supported by that resource (e.g., using PUT on a REST resource that only supports POST).
409	Locked	The request could not be completed due to a conflict with the current state of the resource.
500	Internal Server Error	The server encountered an internal error or timed out; please retry (general catch-all server-side error).HTTP 500 errors may be mapped to SVC1000 (no server resources).
503	Service Unavailable	The server is currently unable to handle the request due to a temporary overloading or maintenance of the server. The implication is that this is a temporary condition which will be alleviated after some delay.

## 13 Sample Policy Error Message

```

HTTPS/1.1 400 Bad Request
content-type: application/json
content-length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
    
```

```
{
  "requestError": {
    "policyException": {
      "messageId": " POL9003",
      "text": "Message content size exceeds the allowable limit",
    }
  }
}
```

## 14 Sample Optimization Service Error Message

```
HTTPS/1.1 400 Bad Request
content-type: application/json
content-length: 12345
Date: Thu, 04 Jun 2009 02:51:59 GMT
```

```
{
  "requestError": {
    "serviceException": {
      "messageId": "SVC2000",
      "text": "Missing Parameter: %1. Error code is %2",
      "variables": ["severity", 400]
    }
  }
}
```

## 15 Authentication

### 15.1 Authorization

Clients must identify themselves to the API server. Client credentials are passed using HTTP Basic Authentication. Credentials must not be passed on the query string. Credentials must be sent in an Authorization header as follows:

- The username and password are formed into one string as “username:password”.
- The resulting string is Base64 encoded to produce the encoded credential.
- The encoded credential is communicated Authorization HTTP header.

For example,

Header	Value
Authorization	Basic am01Nxe45NJC52dgn09sNJDF02seg92452

## 15.2 Encryption

Because client credentials are merely encoded but not encrypted, HTTPS (rather than HTTP) should be used. HTTPS will encrypt and protect message contents. Data is not transmitted over the internet.

## 15.3 Firewall

N/A

## 16 Disaster Recovery

Standard SNIRO disaster recovery.

## 17 Testing Consideration

TBD

## 18 Related Documents

TBD

## 19 Acceptance and Approval

### 19.1 Overview

The Approvers of this work product agree that this document is acceptable and complete to the best of their knowledge and will be used by the project team as an official deliverable for the project. It is further agreed that this document can now be baselined and any changes to these sections from this point forward must follow the Managing Change in the IT UP.

Embed evidence of approval in the review table below, or use the PRISM Approval Functionality in the Project Workflow Module Workflow Template View.

## 19.2 Approvers

ATT UIDName	Role	GroupApplication	Approved Version, Date, Evidence
NA	NA	NA	See PRISM for evidence of approval.