TM Forum Specification

Service Problem Management API REST Specification

TMF656
Release 18.5.0
January 2019

Latest Update: TM Forum Release 18.5.0	**Member Evaluation**
Version 3.0.0	**IPR Mode: RAND**

NOTICE

This document and the information contained herein is provided on an "AS IS" basis and TM FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Direct inquiries to the TM Forum office:

4 Century Drive, Suite 100
Parsippany, NJ 07054, USA
Tel No. +1 973 944 5100
Fax No. +1 973 998 7196
TM Forum Web Page: www.tmforum.org

Table of Contents

List of Tables

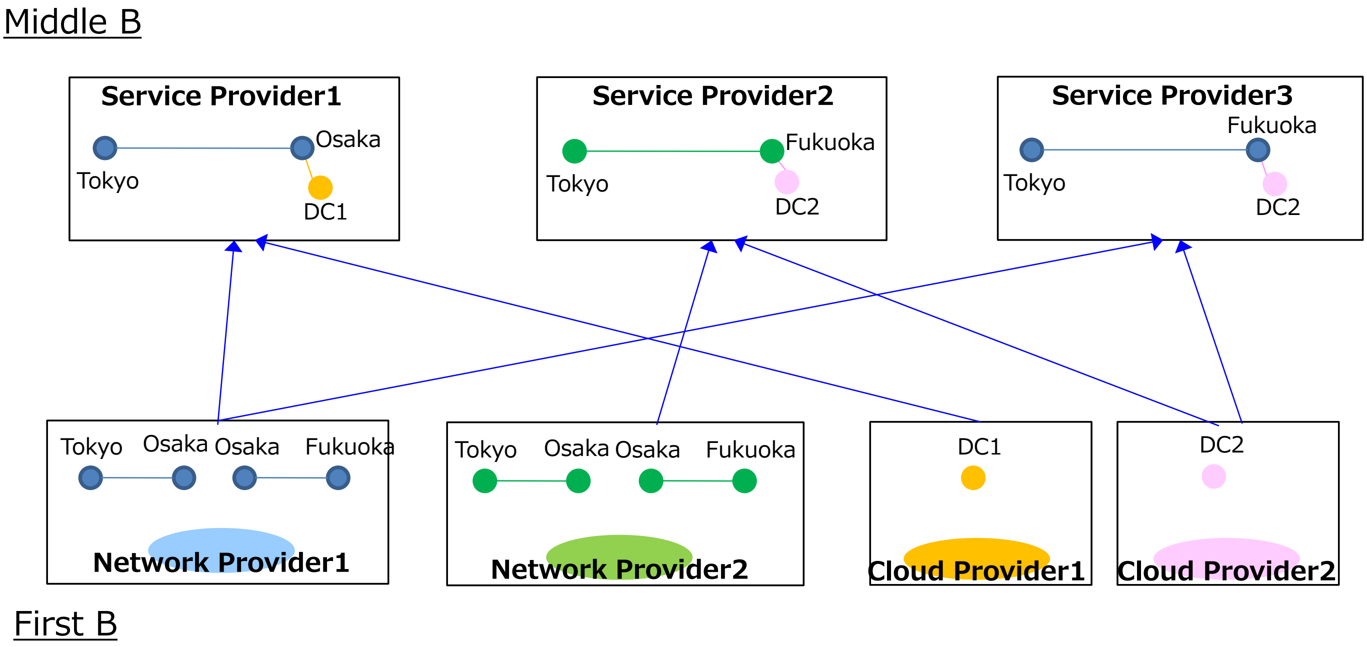
N/A

Introduction

This Service Problem Management API is used for the service providers (Defined as the Middle B) to manage the service problems in their service area. Service problem is generated based on the information declared by Middle B or the event information notified from infrastructure providers (Defined as the First B) who provide the infrastructure of cloud or network. The event information includes alarm information, performance anomaly information, trouble ticket information, SLA violation, maintenance information and prediction information. Middle Bs can refer the service problems and the event information from First Bs and when the service problems occur or its status have been changed, Middle Bs can receive notifications. According to these functions, Middle Bs are able to grasp the service problems quickly and accurately.

SAMPLE USE CASES

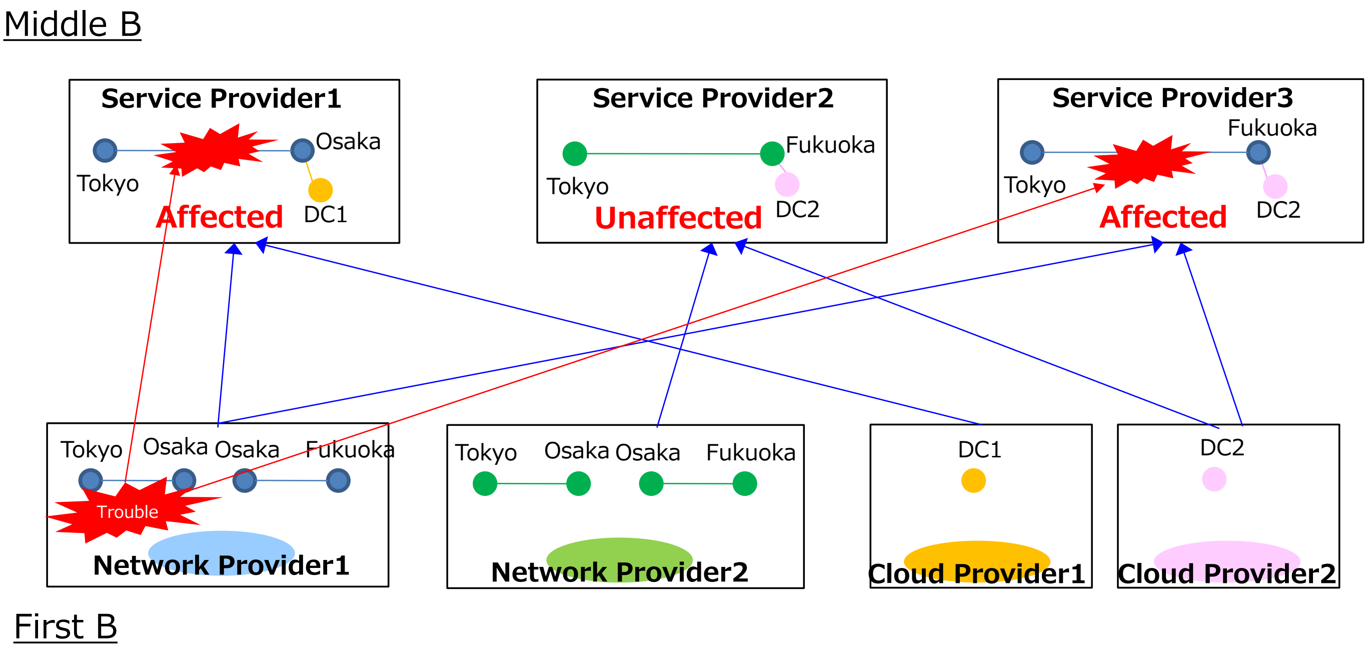
We assume following situation:
There are Network Provider 1 (NP1) and 2 (NP2), which provide network infrastructure, and Cloud Provider 1, 2, which provides cloud infrastructure, as First Bs. Using these infrastructure, Service Provider 1 (SP1), 2 (SP2) and 3 (SP3) are providing their services to their end-users as Middle Bs.



Use case 1

When trouble happened in any resources of NW/Cloud Providers, Service Providers can know their services are affected or not. The specific use case is following:

1. SPM collects configuration information of services provided by service providers in advance using Product Inventory Management API, etc.
2. Each of Middle Bs – Service Provider 1 (SP1), Service Provider 2 (SP2), Service Provider 3 (SP3) – registers the notification destination to SPM SPI.
3. When a fault occurs, SPM receives a trouble ticket from the Network Provider 1 (NP1).
4. SPM creates a Service Problem based on the Trouble Ticket.
5. SPM notifies the Service Problem creation notification to Middle B (SP1, SP3) to notify expected service impact, based on the configuration information collected in advance.
6. When SPM receives a notification that the trouble ticket has changed to "In Progress" state, update the status of the relevant Service Problem. Notify the Service Problem state change notification to Middle B (SP1, SP3).



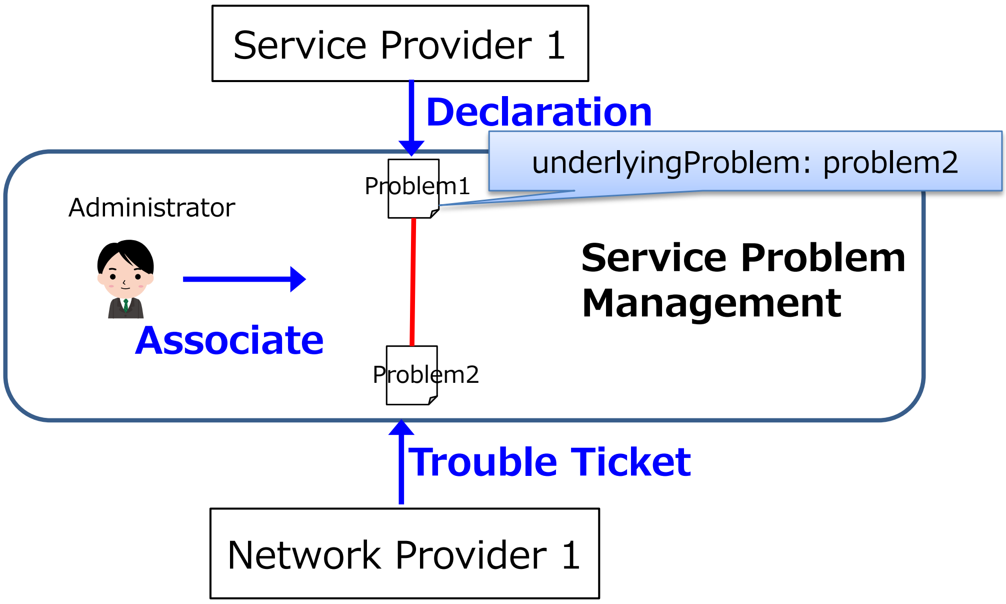
Use case 2

To analyze the past problems, Middle B collects the problem information in the past one year.

Use case 3

Service providers can declare a new service problem based on trouble declarations from their end-users. In addition, the SPM administrator can associate the service problem, based on the Middle B declaration, with another problem based on a First B event such as a Trouble Ticket. The specific use case is following:

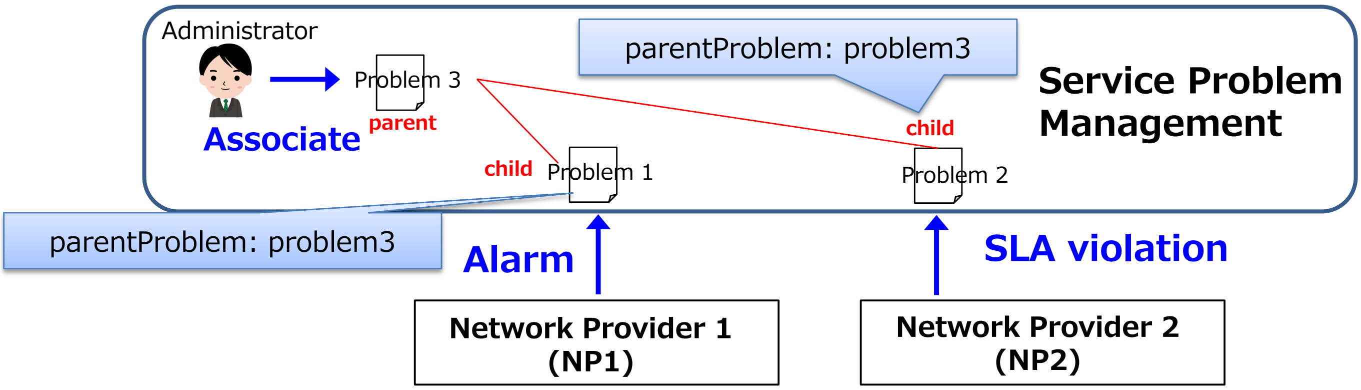
1. Based on the report from the user that there is a problem in the Internet access, Middle B (SP1) gets the current service problem.
2. After SPM collects the current Service Problems, returns that there are no problems related to the service of the Middle B (SP1).
3. In order to request the analysis of this event, Middle B declares a new service problem.
4. Since the SPM administrator found that necessary detailed information was insufficient, SPM administrator requests additional information about the behavior of the Middle B side.
5. SP1 collects the specified additional information and registers it.
6. SPM administrator checks the additional information, and accepts the Service Problem (Problem 1).
7. First B(NP1) registers a detected problem event to the trouble ticket, and notice a new trouble ticket generation to SPM. The SPM creates a Service Problem (Problem 2) based on the trouble ticket.
8. Since the two problems affects the same location, SPM administrator determines that the declared problem (Problem 1) and the new problem based on the new trouble ticket (Problem 2) is associated. SPM administrator associates Problem 1 with 2. In this case, Problem 1 will have Problem2 as the association “underlyingProblem”. Note that Problem 1 can have an “parentProblem” as another association if he would like to group those problems.
9. Since the Problem 1 was changed to add a “underlyingProblem”, Service Problem Change Notification is sent to SP1.



Use case 4

The SPM administrator can associate and group multiple service problems so that service providers can easily recognize what the real problem is. The specific use case is following:

1. SPM receives an alarm from NP1 and creates a Service Problem based on it (Problem 1).
2. SPM receives an SLA violation from NP2 and creates a Service Problem based on it (Problem 2).
3. By analyzing problems, SPM administrator determines that Problem 1 and 2 are the same problem. SPM administrator creates a new Service Problem (Problem 3) in order to group and associate Problem 1, 2 and 3. In this case, Problem 3 is a parent and Problem 1 and 2 are children.



Service Problem Lifecycle

ServiceProblem states:

Following the available status values for a service problem are listed. The status value is in accordance with Trouble Ticket API. The state graphic gives an overview of the allowed status changes

- Submitted
- Rejected

- Acknowledged
- In Progress
  - Held
  - Pending
- Resolved
- Closed
- Cancelled

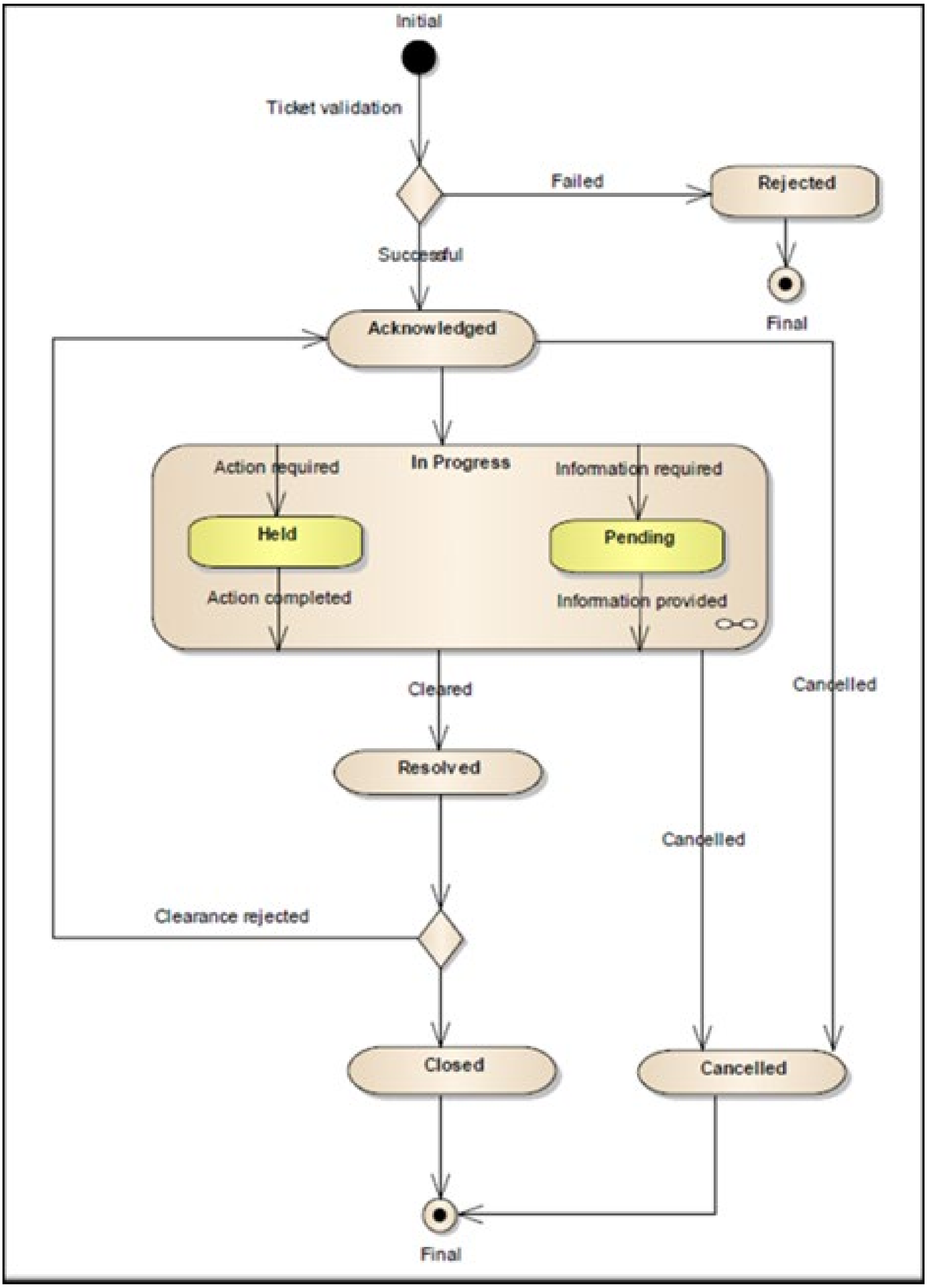


Figure 1 – Life Cycle

State	Description
Submitted	The initial state of a service problem when created by a service problem originator
Acknowledged	The Service Problem was accepted and allocated a unique service problem id by Service problem handler.
In Progress	The service problem was validated by the service problem handler and is being processed.
Resolved	The fault indicated in the service problem was corrected by the service problem handler and acknowledgement is awaited from its originator.
Closed	The service Problem's originator has acknowledged the 'Resolved' state of the service problem, or the timeframe for acknowledgement has passed without response from service problem originator.
Rejected	The service problem was rejected because it: • is not submitted • provides invalid information • fails to meet the Business rules in respect of the product which originator is raising a service problem against • is otherwise defective
In Progress – Pending	Service problem handler is awaiting further confirmation on details of a Fault from originator before it can progress the Fault. An example is where appointment information is required.
In Progress - Held	Service problem handler is confirming further details internally before completing a service problem. An example is where service problem handler for network infrastructure spare parts to progress with the fault rectification.
Cancelled	The service problem was cancelled because it: • Was cancelled by service problem originator

Support of polymorphism and extension patterns

Support of polymorphic collections and types and schema based extension is provided by means of a list of generic meta-attributes that we describe below. Polymorphism in collections occurs when entities inherit from base entities, for instance a RouterProblem and MainSwitchProblem inheriting properties from the ServiceProblem entity.
Generic support of polymorphism and pattern extensions is described in the TMF API Guidelines v3.0 Part 2 document.

The @type attribute provides a way to represent the actual class type of an entity. For example, within a list of ServiceProblem instances some may be instances of RouterProblem where other could be instances of MainSwitchProblem. The @type gives this information. All resources and sub-resources of this API have a @type attributes that can be provided when this is useful.

The @referredType can be used within reference entities (like for instance an ServiceProblemRef object) to explicitly denote the actual entity type of the referred class. Notice that in reference entities the @type, when used, denotes the class type of the reference itself, such as ServiceProblemRef, and not the class type of the referred object. However, since reference classes are rarely sub-classed, @type is generally not useful in reference objects.

The @schemaLocation property can be used in resources to allow specifying user-defined properties of an Entity or to specify the expected *characteristics* of an entity.

The @baseType attribute gives a way to provide explicitly the base of class of a given resource that has been extended.

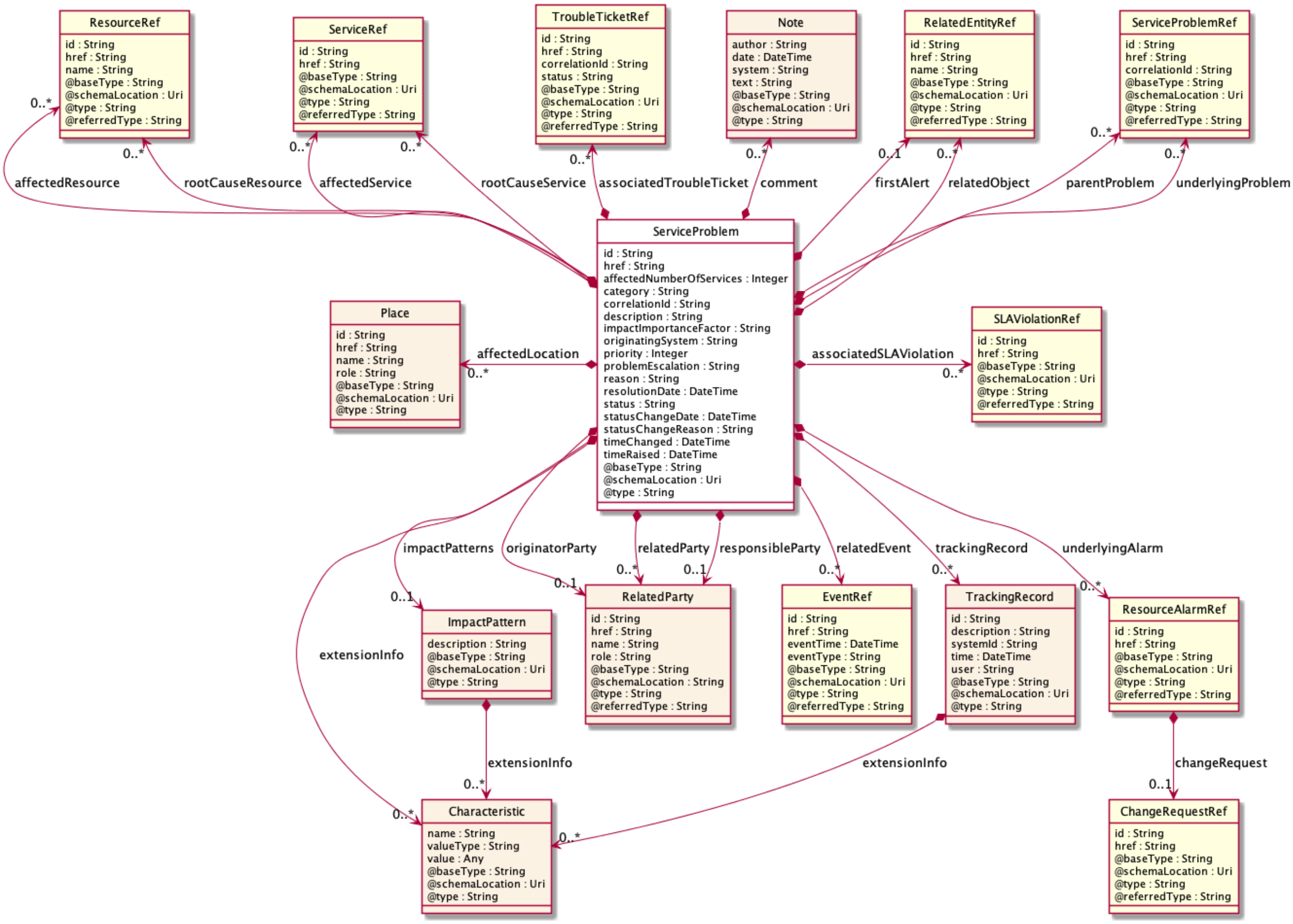
RESOURCE MODEL

Managed Entity and Task Resource Models

Service Problem resource

The problem information for Middle B which is abstracted in the service layer from the issued event information by First B.

Resource model



Field descriptions

ServiceProblem fields

affectedLocation A list of places (Place [*]). A list of the locations affected by the problem. At least one of affectedResource, affectedService or affectedLocation should be present.

affectedNumberOfServices An integer. Number of affected services.

affectedResource A list of resource references (ResourceRef [*]). A list of the resources affected by the problem. At least one of affectedResource, affectedService or affectedLocation should be present.

affectedService A list of service references (ServiceRef [*]). List of affected services. At least one of affectedResource, affectedService or affectedLocation should be present.

associatedSLAViolation A list of s l a violation references (SLAViolationRef [*]). A List of SLA violations associated with this problem.

associatedTroubleTicket A list of trouble ticket references (TroubleTicketRef [*]). A list of trouble tickets associated with this problem.

category A string. Classifier for the problem. Settable. For example, this is used for distinguish the category of problem originator in [role].[category] format. Example: serviceProvider.declarer, supplier.originated, system.originated.

comment A list of notes (Note [*]). A list of comments or notes made on the problem.

correlationId A string. Additional identifier coming from an external system.

description A string. Free form text describing the Service Problem.

extensionInfo A list of characteristics (Characteristic [*]). A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces.

firstAlert A related entity reference (RelatedEntityRef). Indicates what first alerted the system to the problem. It is not the root cause of the Service Problem. Examples: Threshold crossing alert.

href A string. Reference to the Service Problem.

id A string. Identifier of the service problem.

impactImportanceFactor A string. Impact Importance is characterized by an Impact Importance Factor: overall importance of the impact of all the affected services, e.g. 0 (zero impact) to 100 (worst impact). The Impact Importance is a calculated field which is set by the OSS determining the impact.

impactPatterns An impact pattern (ImpactPattern). Define the patterns of impact (optional)- e.g. other service characteristics- Used when defining impact through another pattern than the predefined attributes.

originatingSystem A string. Indicates where the problem was generated.

originatorParty A related party (RelatedParty). Individual or organization that created the problem.

parentProblem
A list of service problem references (ServiceProblemRef [*]). The parent problem to which this problem is attached.

priority
An integer. An indication varying from 1 (highest) to 10 (lowest) of how important it is for the service provider to correct the Service Problem.

problemEscalation
A string. Indicates if this service problem has been escalated or not. Possible values are 0 to 10. A value of zero means no escalation. The meanings of values 1-10 are to be determined by the user of the interface, but they show increasing levels of escalation.

reason
A string. Free text or optionally structured text. It can be Unknown.

relatedEvent
A list of event references (EventRef [*]). List of events associated to this problem.

relatedObject
A list of related entity references (RelatedEntityRef [*]). List of objects associated to this problem.

relatedParty
A list of related parties (RelatedParty [*]). List of parties or party roles playing a role within the service problem.

resolutionDate
A date time (DateTime). Time the problem was resolved.

responsibleParty
A related party (RelatedParty). Individual or organization responsible for handling this problem.

rootCauseResource
A list of resource references (ResourceRef [*]). Resource(s) that are associated to the underlying service problems that are the Root Cause of this one if any (used only if applicable).

rootCauseService
A list of service references (ServiceRef [*]). Service(s) that are associated to the underlying service problems that are the Root Cause of this one if any (used only if applicable).

status
A string. The current status of the service problem. Possible values are Submitted, Rejected, Acknowledged, In Progress [Held, Pending], Resolved, Closed, and Cancelled.

statusChangeDate
A date time (DateTime). Time the problem was last status changed.

statusChangeReason
A string. The reason of state change.

timeChanged
A date time (DateTime). Time the problem was last changed.

timeRaised
A date time (DateTime). Time the problem was raised.

trackingRecord
A list of tracking records (TrackingRecord [*]). List of tracking records that allow the tracking of modifications on the problem.The tracking records should not be

	embedded in the problem to allow retrieving the problem without the tracking records.
underlyingAlarm	A list of resource alarm references (ResourceAlarmRef [*]). A list of alarms underlying this problem.
underlyingProblem	A list of service problem references (ServiceProblemRef [*]). A list of underlying problems. Relevant only if this problem is derived from other problems.

Characteristic sub-resource

Describes a given characteristic of an object or entity through a name/value pair.

name	A string. Name of the characteristic.
value	An any (Any). The value of the characteristic.
valueType	A string. Data type of the value of the characteristic.

ImpactPattern sub-resource

Define the patterns of impact (optional), such as other service characteristics- Used when defining impact through another pattern than the pre-defined attributes.

description	A string. Basic description of the impact pattern.
extensionInfo	A list of characteristics (Characteristic [*]). A generic list of any type of elements. Used for extensions or loose element encapsulation from other namespaces.

Note sub-resource

Extra information about a given entity.

author	A string. Author of the note.
date	A date time (DateTime). Date of the note.
system	A string. Describes the system from which the action related to this note was done.
text	A string. Text of the note.

Place sub-resource

Place reference. Place defines the places where the products are sold or delivered.

href	A string. Unique reference of the place.
id	A string. Unique identifier of the place.
name	A string. A user-friendly name for the place, such as [Paris Store], [London Store], [Main Home].

role	A string. Role of the place, such as: [home delivery], [shop retrieval]).

RelatedParty sub-resource

RelatedParty reference. A related party defines party or party role linked to a specific entity.

@baseType	A string. When sub-classing, this defines the super-class.
@referredType	A string. The actual type of the target instance when needed for disambiguation.
@schemaLocation	A string. A URI to a JSON-Schema file that defines additional attributes and relationships.
@type	A string. When sub-classing, this defines the sub-class entity name.
href	A string. Reference of the related party, could be a party reference or a party role reference.
id	A string. Unique identifier of a related party.
name	A string. Name of the related party.
role	A string. Role of the related party.

TrackingRecord sub-resource

Tracking records allow the tracking of modifications on the problem. The tracking records should not be embedded in the problem to allow retrieving the problem without the tracking records.

description	A string. Describes the action being done, such as: ack, clear.
extensionInfo	A list of characteristics (Characteristic [*]). A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces.
id	A string. Identifier of the TrackingRecord.
systemId	A string. Describes the system Id from which the action was done.
time	A date time (DateTime). Describes the time at which the action was done.
user	A string. Describes the user doing the action.

ChangeRequestRef relationship

Reference to a Change Request.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
href	A string. The reference link to the change request.
id	A string. The identifier of the change request.

EventRef relationship

Events linked with service problem.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
eventTime	A date time (DateTime). Time the event occurred.
eventType	A string. Type of the event.
href	A string. event reference.
id	A string. ID of the event.

RelatedEntityRef relationship

A reference to an entity, where the type of the entity is not known in advance.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
href	A string. The hyperlink to access an entity.
id	A string. The identifier of an entity.
name	A string. The name of an entity.

ResourceAlarmRef relationship

A set of alarm ids identifying the alarms that are underlying this problem.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
changeRequest	A change request reference (ChangeRequestRef). Reference to a Change Request.
href	A string. Reference of the Alarm.
id	A string. Unique identifier of the Alarm.

ResourceRef relationship

@referredType	A string. The actual type of the target instance when needed for disambiguation.
href	A string. Reference of the supporting resource.
id	A string. Unique identifier of the supporting resource.
name	A string. Name of the resource supporting the service.

SLAViolationRef relationship

@referredType	A string. The actual type of the target instance when needed for disambiguation.
href	A string.
id	A string.

ServiceProblemRef relationship

@referredType	A string. The actual type of the target instance when needed for disambiguation.
correlationId	A string. Additional identifier coming from an external system.
href	A string. Reference of the Problem.
id	A string. Unique identifier of the Problem.

ServiceRef relationship

Service reference, for when Service is used by other entities.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
href	A string. reference of the service.
id	A string. Id of the service.

TroubleTicketRef relationship

A trouble ticket represents a record used for reporting and managing the resolution of resource problems.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
correlationId	A string. Additional identifier coming from an external system.
href	A string. Reference of the trouble ticket.
id	A string. Unique identifier of the trouble ticket.
status	A string. The current status of the Trouble Ticket.

Json representation sample

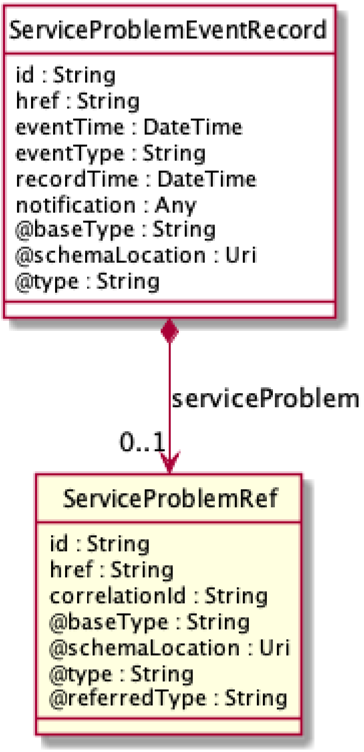
We provide below the json representation of an example of a 'ServiceProblem' resource object

```
{
   "id": "problemxxxx0000",
   "correlationId": "543251",
   "originatingSystem": "System_001",
   "category": "supplier.originated",
   "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/problemxxxx0000",
   "impactImportanceFactor": "0",
   "priority": "1",
   "description": "connection failure between Tokyo and Osaka",
   "problemEscalation": "0",
   "timeRaised": "2017-10-25T12:14:16.361Z",
   "timeChanged": "2017-10-30T12:13:16.361Z",
   "statusChangeDate": "2017-10-29T12:00:00.361Z",
   "statusChangeReason": "problem analysis has been completed in NP1",
   "resolutionDate": "2017-10-29T12:00:00.361Z",
   "status": "resolved",
   "reason": "Failure of resource NP1_Resource_1 in NP1",
   "affectedNumberOfServices": "2",
   "firstAlert": {
     "alertType": "TroubleTicket",
     "id": "NP1_TT_0000000",
     "href": "https://mycsp.com:8080/tmf-api/troubleTicketManagement/v3/troubleTicket/NP1_TT_000000"
   },
   "responsibleParty": {
     "role": "Supplier",
     "id": "NP1",
     "href": "https://mycsp.com:8080/tmf-api/partyManagement/v3/party/NP1",
     "@referredType": "Organization"
   },
   "originatorParty": {
     "role": "Supplier",
     "id": "NP1",
     "href": "https://mycsp.com:8080/tmf-api/partyManagement/v3/party/NP1",
     "@referredType": "Organization"
   },
   "relatedParty": [
     {
       "role": "Supplier",
       "id": "NP1",
       "href": "https://mycsp.com:8080/tmf-api/partyManagement/v3/party/NP1",
       "@referredType": "Organization"
     },
     {
       "role": "Partner",
       "id": "SP1",
       "href": "https://mycsp.com:8080/tmf-api/partyManagement/v3/party/SP1",
       "@referredType": "Organization"
     },
     {
       "role": "Partner",
       "id": "SP3",
       "href": "https://mycsp.com:8080/tmf-api/partyManagement/v3/party/SP3",
       "@referredType": "Organization"
     }
   ],
   "affectedService": [
     {
       "id": "NP1_Tokyo_Osaka",
       "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v3/service/NW1_Tokyo_Osaka"
     },
     {
       "id": "NP1_Tokyo_xxxx",
       "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v3/service/NW1_Tokyo_xxxx"
     }
   ],
   "affectedResource": [
     {
       "id": "NP1_RES_0001",
       "href": "https://mycsp.com:8080/tmf-api/resourceInventoryManagement/v3/resource/NW1_RES_0001"
     }
   ],
   "affectedLocation": [
     {
       "id": "Loc000000",
       "href": "https://mycsp.com:8080/tmf-api/geographicAddressManagement/v3/geographicAddress/Loc000000/",
       "name": "144 Main Street Tokyo 51663556",
       "role": "VPN Endpoint",
       "@type": "GeographicAddress"
     },
     {
       "id": "Loc000001",
       "href": "https://mycsp.com:8080/tmf-api/geographicAddressManagement/v3/geographicAddress/Loc000001/",
       "name": "351 Main Street Osaka 3546365",
       "role": "VPN Endpoint",
       "@type": "GeographicAddress"
     }
   ],
   "associatedTroubleTicket": [
     {
       "id": "NP1_TT_0000000",
       "href": "https://mycsp.com:8080/tmf-api/troubleTicketManagement/v3/TroubleTicketRef/NP1_TT_000000"
     }
   ],
   "underlyingAlarm": [
     {
       "id": "NP1_A_0000000",
       "href": "https://mycsp.com:8080/tmf-api/alarmManagement/v3/resourceAlarm/NP1_A_000000",
       "@referredType": "ResourceAlarm"
     }
   ],
   "associatedSLAViolation": [
     {
       "id": "NP1_SLA_0000000",
       "href": "https://mycsp.com:8080/tmf-api/SLAManagement/v3/SLAViolationRef/NP1_SLA_000000"
     }
   ],
   "relatedEvent": [
     {
       "eventType": "prediction",
       "id": "prediction_0001",
       "href": "https://mycsp.com:8080/tmf-api/eventManagement/v3/event/prediction_0001",
       "eventTime": "2014-12-20T17:00:00Z"
     }
   ],
   "relatedObject": [
     {
       "id": "product0001",
       "href": "https://mycsp.com:8080/tmf- api/productInventoryManagement/v3/product/product0001",
       "@referredType": "Product"
     }
   ],
   "rootCauseService": [
     {
       "id": "NP1_Tokyo_Osaka",
       "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v3/service/NW1_Tokyo_Osaka"
     }
   ],
   "rootCauseResource": [
     {
       "id": "NP1_Resource_1",
       "href": "https://mycsp.com:8080/tmf-api/resourceInventoryManagement/v3/resource/NP1_Resource_1"
     }
   ],
   "parentProblem": [
     {
       "id": "problemxxxx0001",
       "correlationId": "xxxxxxxx",
       "href": "https://mycsp.com:8080/tmf-api/service ProblemManagement/v3/serviceProblem/problemxxxx0001"
     }
   ],
   "underlyingProblem": [
     {
       "id": "problemxxxx0001",
       "correlationId": "xxxxxxxx",
       "href": "https://mycsp.com:8080/tmf-api/serviceProblemMa nagement/v3/serviceProblem/problemxxxx0001"
     }
   ],
   "trackingRecord": [
     {
       "description": "yyy cleared the problem",
       "systemId": "xxxx",
       "time": "2014-12-20T17:00:00Z",
       "user": "NPUSER1",
       "id": "TR_xxxx",
       "extensionInfo": [
         {
           "name": "vendorComment",
           "valueType": "string",
           "value": "Watch out for the dog"
         }
       ]
     }
   ],
   "comment": [
     {
       "author": {
         "id": "SPM_handler_01",
         "href": "https://mycsp.com:8080/tmf-api/partyManagement/v3/party/SPM_handler_01",
         "@referredType": "Individual"
       },
       "date": "2014-12-20T17:00:00Z",
       "systemId": "System_002",
       "text": "receive trouble ticket from NP1, and create this Service Problem"
     },
     {
       "author": {
         "id": "NP1_handler_11",
         "href": "https://mycsp.com:8080/tmf-api/partyManagement/v3/party/NP1_handler_11",
         "@referredType": "Individual"
       },
       "date": "2014-12-20T17:00:00Z",
       "systemId": "System_002",
       "text": "status changed to Progress-Held"
     }
   ],
   "impactPatterns": {
     "description": "Many services are at risk in this problem",
     "extensionInfo": [
       {
         "name": "ImpactProbability",
         "valueType": "boolean",
         "value": true
       }
     ]
   },
   "extensionInfo": [
     {
       "name": "EstimatedCost",
       "valueType": "integer",
       "value": 20
     }
   ]
 }
```

Service Problem Event Record resource

A record of an event (related to a service problemn) received from another system.

Resource model



Field descriptions

<u>*ServiceProblemEventRecord* fields</u>

eventTime	A date time (DateTime). Time at which the event occurred.
eventType	A string. Type of the recorded event.
href	A string. reference to this resource.
id	A string. Identifier of the service problem event record.
notification	An any (Any). A notification from the possible notifications for Service Problem (such as creation, status change, information required, change).
recordTime	A date time (DateTime). Time at which the record was created.
serviceProblem	A service problem reference (ServiceProblemRef). The service problem to which this record applies.

<u>*ServiceProblemRef* relationship</u>

@referredType	A string. The actual type of the target instance when needed for disambiguation.

correlationId A string. Additional identifier coming from an external system.

href A string. Reference of the Problem.

id A string. Unique identifier of the Problem.

Json representation sample

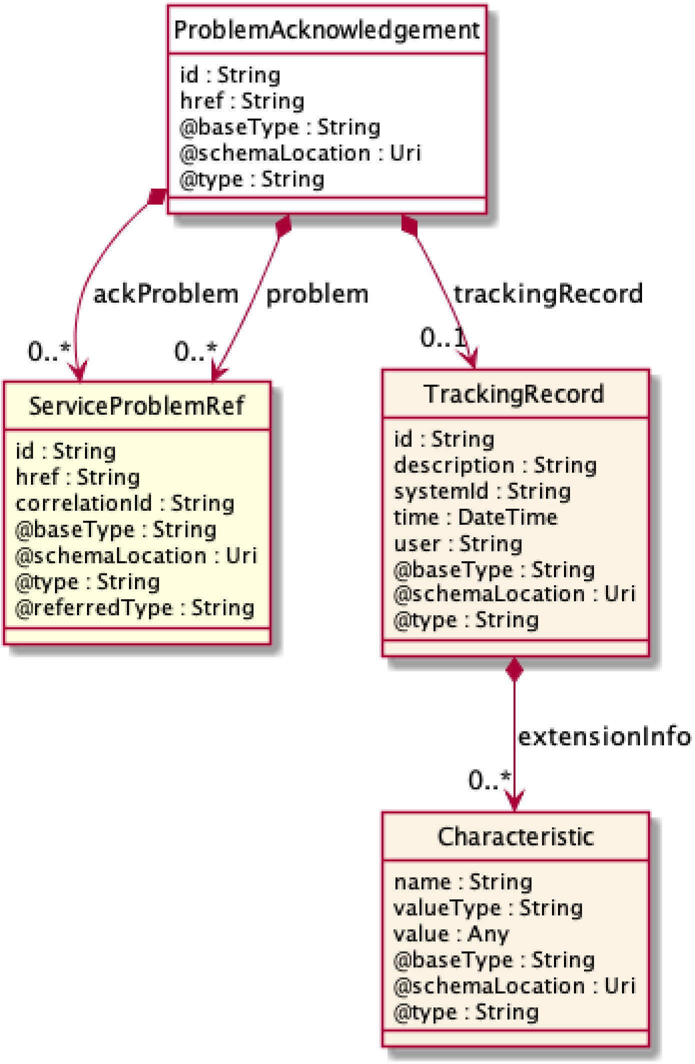
We provide below the json representation of an example of a 'ServiceProblemEventRecord' resource object

```
{
   "id": "42",
   "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblemEventRecord/42",
   "recordTime": "2016-08-08T10:45:30.0Z",
   "eventType": "ServiceProblemCreationNotification",
   "eventTime": "2016-08-08T10:45:25.0Z",
   "serviceProblemId": "SP001",
   "notification": {
      "eventType": "ServiceProblemCreationNotification",
      "eventTime": "2016-08-08T10:45:25.0Z",
      "eventId": "92775",
      "event": {}
   },
   "serviceProblem": {
      "id": "problemxxxx0000",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/problemxxxx0000"
   }
}
```

Problem Acknowledgement resource

Task resource that requests acknowledgement of problems by the problem handler.

Resource model



Field descriptions

ProblemAcknowledgement fields

ackProblem A list of service problem references (ServiceProblemRef [*]). The problems that were acknowledged, populated in the output to this task.

href A string. Reference to this task resource.

id A string. Unique identifier of this task resource.

problem A list of service problem references (ServiceProblemRef [*]). The problems to be acknowledged, relevant in the input to this task.

trackingRecord A tracking record (TrackingRecord). A record of the action taken in this acknowledgement.

Characteristic sub-resource

Describes a given characteristic of an object or entity through a name/value pair.

name	A string. Name of the characteristic.
value	An any (Any). The value of the characteristic.
valueType	A string. Data type of the value of the characteristic.

TrackingRecord sub-resource

Tracking records allow the tracking of modifications on the problem. The tracking records should not be embedded in the problem to allow retrieving the problem without the tracking records.

description	A string. Describes the action being done, such as: ack, clear.
extensionInfo	A list of characteristics (Characteristic [*]). A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces.
id	A string. Identifier of the TrackingRecord.
systemId	A string. Describes the system Id from which the action was done.
time	A date time (DateTime). Describes the time at which the action was done.
user	A string. Describes the user doing the action.

ServiceProblemRef relationship

@referredType	A string. The actual type of the target instance when needed for disambiguation.
correlationId	A string. Additional identifier coming from an external system.
href	A string. Reference of the Problem.
id	A string. Unique identifier of the Problem.

Json representation sample

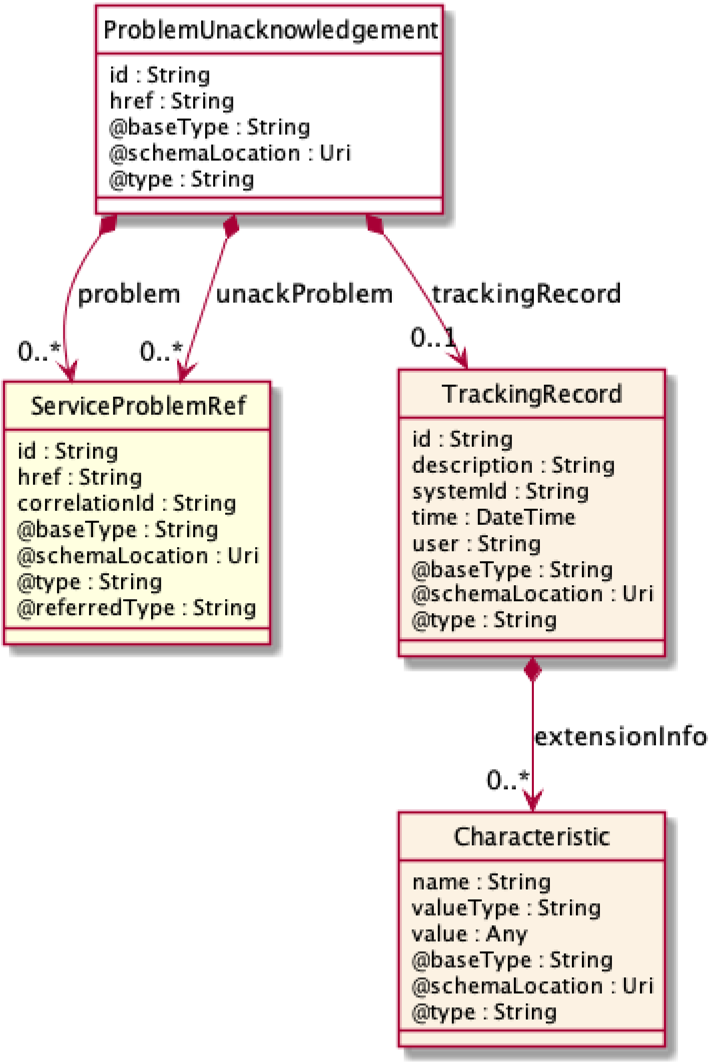
We provide below the json representation of an example of a 'ProblemAcknowledgement' resource object

```
{
  "problem": [
    {
      "id": "41",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/41"
    },
    {
      "id": "42",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/42"
    },
    {
            "id": "43",
            "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
        }
    ],
    "trackingRecord": {
        "description": "yyy ack the problem",
        "systemId": "xxxx",
        "time": "2014-12-20T17:00:00Z",
        "user": "NP1"
    }
}
```

Problem Unacknowledgement resource

Task resource that requests unacknowledgement of problems, rolling back the status of the problems from Acknowledged to Submitted.

Resource model



Field descriptions

ProblemUnacknowledgement fields

href A string. Reference to this task resource.

id A string. Unique identifier of this task resource.

problem A list of service problem references (ServiceProblemRef [*]). The problems to be unacknowledged, relevant in the input to this task.

trackingRecord A tracking record (TrackingRecord). A record of the action taken in this acknowledgement.

unackProblem A list of service problem references (ServiceProblemRef [*]). The problems that were unacknowledged, populated in the output to this task.

Characteristic sub-resource

Describes a given characteristic of an object or entity through a name/value pair.

name A string. Name of the characteristic.

value An any (Any). The value of the characteristic.

valueType A string. Data type of the value of the characteristic.

TrackingRecord sub-resource

Tracking records allow the tracking of modifications on the problem. The tracking records should not be embedded in the problem to allow retrieving the problem without the tracking records.

description A string. Describes the action being done, such as: ack, clear.

extensionInfo A list of characteristics (Characteristic [*]). A generic list of any type of elements. Used for vendor Extensions or loose element encapsulation from other namespaces.

id A string. Identifier of the TrackingRecord.

systemId A string. Describes the system Id from which the action was done.

time A date time (DateTime). Describes the time at which the action was done.

user A string. Describes the user doing the action.

ServiceProblemRef relationship

@referredType A string. The actual type of the target instance when needed for disambiguation.

correlationId A string. Additional identifier coming from an external system.

href A string. Reference of the Problem.

id A string. Unique identifier of the Problem.

Json representation sample

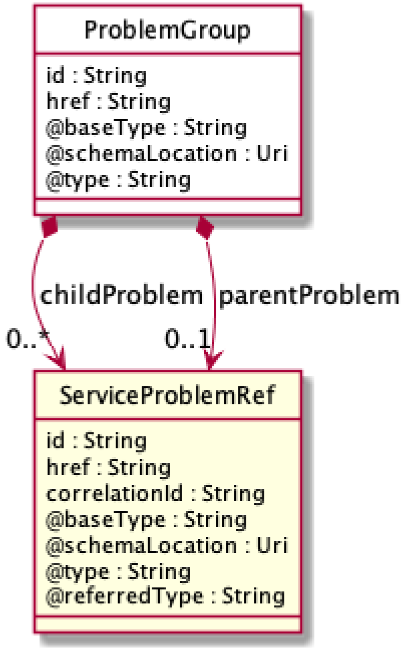
We provide below the json representation of an example of a 'ProblemUnacknowledgement' resource object

```
{
   "problem": [
      {
         "id": "41",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemMan agement/v3/serviceProblem/41"
      },
      {
         "id": "42",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/42"
      },
      {
         "id": "43",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
      }
   ],
   "trackingRecord": {
      "description": "yyy unack the problem",
      "systemId": "xxxx",
      "time": "2014-12-20T17:00:00Z",
      "user": "NP1"
   }
}
```

Problem Group resource

Task resource that requests Service Problems to be grouped together into a parent and set of children.

Resource model



Field descriptions

ProblemGroup fields

childProblem	A list of service problem references (ServiceProblemRef [*]). List of problems to be grouped under a parent problem.
href	A string. Reference to this task resource.
id	A string. Unique identifier of this task resource.
parentProblem	A service problem reference (ServiceProblemRef). The parent problem to which the problems are to be grouped.

ServiceProblemRef relationship

@referredType	A string. The actual type of the target instance when needed for disambiguation.
correlationId	A string. Additional identifier coming from an external system.
href	A string. Reference of the Problem.
id	A string. Unique identifier of the Problem.

Json representation sample

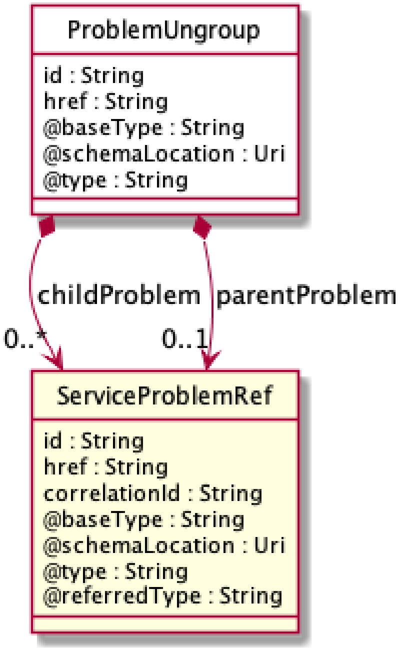
We provide below the json representation of an example of a 'ProblemGroup' resource object

```
{
   "childProblem": [
      {
         "id": "41",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/41"
      },
      {
         "id": "42",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/42"
      },
      {
         "id": "43",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
      }
   ],
   "parentProblem": {
      "id": "43",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
   }
}
```

Problem Ungroup resource

Task resource that requests Service Problems to be ungrouped from a parent.

Resource model



Field descriptions

ProblemUngroup fields

childProblem A list of service problem references (ServiceProblemRef [*]). List of problems to be ungrouped from a parent problem.

href A string. Reference to this task resource.

id A string. Unique identifier of this task resource.

parentProblem A service problem reference (ServiceProblemRef). The parent problem from which the problems are to be ungrouped.

ServiceProblemRef relationship

@referredType A string. The actual type of the target instance when needed for disambiguation.

correlationId A string. Additional identifier coming from an external system.

href A string. Reference of the Problem.

id A string. Unique identifier of the Problem.

Json representation sample

We provide below the json representation of an example of a 'ProblemUngroup' resource object

```
{
  "childProblem": [
    {
      "id": "41",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/41"
    },
    {
      "id": "42",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/42"
    },
    {
      "id": "43",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
    }
  ],
  "parentProblem": {
    "id": "43",
    "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
  }
}
```

Notification Resource Models

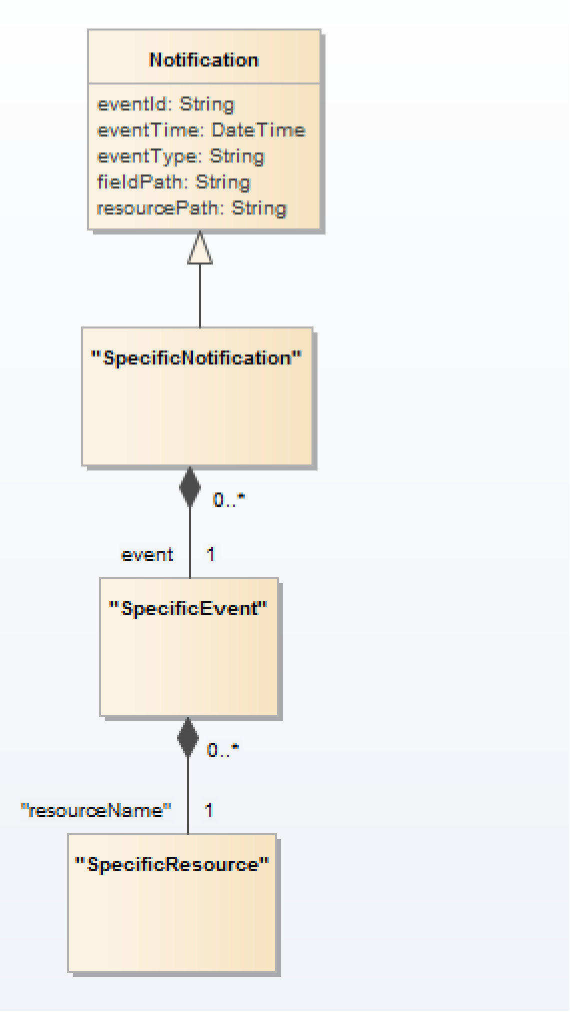
4 notifications are defined for this API

Notifications related to ServiceProblem:

- ServiceProblemCreateNotification
- ServiceProblemStateChangeNotification
- ServiceProblemAttributeValueChangeNotification
- ServiceProblemInformationRequiredNotification

The notification structure for all notifications in this API follow the pattern depicted by the figure below.
A notification resource (depicted by "SpecificNotification" placeholder) is a sub class of a generic Notification structure containing an id of the event occurrence (eventId), an event timestamp (eventTime), and the name of the notification resource (eventType).

This notification structure owns an event structure ("SpecificEvent" placeholder) linked to the resource concerned by the notification using the resource name as access field ("resourceName" placeholder).



Service Problem Create Notification

Notification ServiceProblemCreateNotification case for resource ServiceProblem

Json representation sample

We provide below the json representation of an example of a 'ServiceProblemCreateNotification' notification object

```
{
   "eventId":"00001",
   "eventTime":"2015-11-16T16:42:25-04:00",
   "eventType":"ServiceProblemCreateNotification",
    "event": {
      "serviceProblem" :
         {-- SEE ServiceProblem RESOURCE SAMPLE --}
   }
}
```

Service Problem State Change Notification

Notification ServiceProblemStateChangeNotification case for resource ServiceProblem

Json representation sample

We provide below the json representation of an example of a 'ServiceProblemStateChangeNotification' notification object

```
{
   "eventId":"00001",
   "eventTime":"2015-11-16T16:42:25-04:00",
   "eventType":"ServiceProblemStateChangeNotification",
    "event": {
      "serviceProblem" :
         {-- SEE ServiceProblem RESOURCE SAMPLE --}
   }
}
```

Service Problem Attribute Value Change Notification

Notification ServiceProblemAttributeValueChangeNotification case for resource ServiceProblem

Json representation sample

We provide below the json representation of an example of a 'ServiceProblemAttributeValueChangeNotification' notification object

```
{
   "eventId":"00001",
   "eventTime":"2015-11-16T16:42:25-04:00",
   "eventType":"ServiceProblemAttributeValueChangeNotification",
    "event": {
      "serviceProblem" :
         {-- SEE ServiceProblem RESOURCE SAMPLE --}
   }
}
```

Service Problem Information Required Notification

Notification ServiceProblemInformationRequiredNotification case for resource ServiceProblem

Json representation sample

We provide below the json representation of an example of a 'ServiceProblemInformationRequiredNotification' notification object

```
{
    "eventId":"00001",
    "eventTime":"2015-11-16T16:42:25-04:00",
    "eventType":"ServiceProblemInformationRequiredNotification",
     "event": {
        "serviceProblem" :
            {-- SEE ServiceProblem RESOURCE SAMPLE --}
    }
}
```

API OPERATIONS

Remember the following Uniform Contract:

Operation on Entities	Uniform API Operation	Description
Query Entities	GET Resource	GET must be used to retrieve a representation of a resource.
Create Entity	POST Resource	POST must be used to create a new resource
Partial Update of an Entity	PATCH Resource	PATCH must be used to partially update a resource
Complete Update of an Entity	PUT Resource	PUT must be used to completely update a resource identified by its resource URI
Remove an Entity	DELETE Resource	DELETE must be used to remove a resource
Execute an Action on an Entity	POST on TASK Resource	POST must be used to execute Task Resources
Other Request Methods	POST on TASK Resource	GET and POST must not be used to tunnel other request methods.

Filtering and attribute selection rules are described in the TMF REST Design Guidelines Part 1 Document.

Notifications are also described in a subsequent section.

Operations on Service Problem

List service problems

GET /serviceProblem?fields=...&{filtering}

Description

This operation list service problem entities.
Attribute selection is enabled for all first level attributes.
Filtering may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving multiple service problems, in this example closed items first reported from a specific service

Request

```
GET
{apiRoot}/serviceProblem?fields=id,href,statusChangeDate,category&status=closed&priority=1&firstAlert.id=NP1_TT_0000000
Accept: application/json
```

Response

```
200

[
    {
        "id": "5351",
        "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/5351",
        "statusChangeDate": "2017-10-29T12:00:00.361Z",
        "category": "supplier.originated"
    },
    {
        "id": "5352",
        "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/5352",
        "name": "vDPI serial 1355445",
        "category": "serviceProvider.originated"
    }
]
```

Retrieve service problem

GET /serviceProblem/{id}?fields=...&{filtering}

Description

This operation retrieves a service problem entity.
Attribute selection is enabled for all first level attributes.
Filtering on sub-resources may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving a specific service problem.

Request

```
GET {apiRoot}/serviceProblem/problemxxxx0000
Accept: application/json
```

Response

```
200

{
   "id": "problemxxxx0000",
   "correlationId": "543251",
   "originatingSystem": "System_001",
   "category": "supplier.originated",
   "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/problemxxxx0000",
   "impactImportanceFactor": "0",
   "priority": "1",
   "description": "connection failure between Tokyo and Osaka",
   "problemEscalation": "0",
   "timeRaised": "2017-10-25T12:14:16.361Z",
   "timeChanged": "2017-10-30T12:13:16.361Z",
   "statusChangeDate": "2017-10-29T12:00:00.361Z",
   "statusChangeReason": "problem analysis has been completed in NP1",
   "resolutionDate": "2017-10-29T12:00:00.361Z",
   "status": "resolved",
   "reason": "Failure of resource NP1_Resource_1 in NP1",
   "affectedNumberOfServices": "2",
   "firstAlert": {
      "alertType": "TroubleTicket",
      "id": "NP1_TT_0000000",
      "href": "https://mycsp.com:8080/tmf-api/troubleTicketManagement/v3/troubleTicket/NP1_TT_000000"
   },
   "responsibleParty": {
      "role": "Supplier",
      "id": "NP1",
      "href": "https://mycsp.com:8080/tmf-api/partyManagement/v3/party/NP1",
      "@referredType": "Organization"
   },
   "originatorParty": {
      "role": "Supplier",
      "id": "NP1",
      "href": "https://mycsp.com:8080/tmf-api/partyManagement/v3/party/NP1",
      "@referredType": "Organization"
   },
   "relatedParty": [
      {
        "role": "Supplier",
        "id": "NP1",
        "href": "https://mycsp.com:8080/tmf-api/partyManagement/v3/party/NP1",
        "@referredType": "Organization"
      },
      {
        "role": "Partner",
        "id": "SP1",
        "href": "https://mycsp.com:8080/tmf-api/partyManagement/v3/party/SP1",
        "@referredType": "Organization"
      },
      {
        "role": "Partner",
        "id": "SP3",
        "href": "https://mycsp.com:8080/tmf-api/partyManagement/v3/party/SP3",
        "@referredType": "Organization"
      }
   ],
   "affectedService": [
      {
        "id": "NP1_Tokyo_Osaka",
        "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v3/service/NW1_Tokyo_Osaka"
      },
      {
        "id": "NP1_Tokyo_xxxx",
        "href": "https://mycsp.com:8080/tmf- api/serviceInventoryManagement/v3/service/NW1_Tokyo_xxxx"
      }
   ],
   "affectedResource": [
      {
        "id": "NP1_RES_0001",
        "href": "https://mycsp.com:8080/tmf-api/resourceInventoryManagement/v3/resource/NW1_RES_0001"
      }
   ],
   "affectedLocation": [
      {
        "id": "Loc000000",
        "href": "https://mycsp.com:8080/tmf-api/geographicAddressManagement/v3/geographicAddress/Loc000000/",
        "name": "144 Main Street Tokyo 51663556",
        "role": "VPN Endpoint",
        "@type": "GeographicAddress"
      },
      {
        "id": "Loc000001",
        "href": "https://mycsp.com:8080/tmf-api/geographicAddressManagement/v3/geographicAddress/Loc000001/",
        "name": "351 Main Street Osaka 3546365",
        "role": "VPN Endpoint",
        "@type": "GeographicAddress"
      }
    ],
    "associatedTroubleTicket": [
      {
        "id": "NP1_TT_0000000",
        "href": "https://mycsp.com:8080/tmf-api/troubleTicketManagement/v3/TroubleTicketRef/NP1_TT_000000"
      }
    ],
    "underlyingAlarm": [
      {
        "id": "NP1_A_0000000",
        "href": "https://mycsp.com:8080/tmf-api/alarmManagement/v3/resourceAlarm/NP1_A_000000",
        "@referredType": "ResourceAlarm"
      }
    ],
    "associatedSLAViolation": [
      {
        "id": "NP1_SLA_0000000",
        "href": "https://mycsp.com:8080/tmf-api/SLAManagement/v3/SLAViolationRef/NP1_SLA_000000"
      }
    ],
    "relatedEvent": [
      {
        "eventType": "prediction",
        "id": "prediction_0001",
        "href": "https://mycsp.com:8080/tmf-api/eventManagement/v3/event/prediction_0001",
        "eventTime": "2014-12-20T17:00:00Z"
      }
    ],
    "relatedObject": [
      {
        "id": "product0001",
        "href": "https://mycsp.com:8080/tmf-api/productInventoryManagement/v3/product/product0001",
        "@referredType": "Product"
      }
    ],
    "rootCauseService": [
      {
        "id": "NP1_Tokyo_Osaka",
        "href": "https://mycsp.com:8080/tmf- api/serviceInventoryManagement/v3/service/NW1_Tokyo_Osaka"
      }
    ],
    "rootCauseResource": [
      {
        "id": "NP1_Resource_1",
        "href": "https://mycsp.com:8080/tmf-api/resourceInventoryManagement/v3/resource/NP1_Resource_1"
      }
    ],
    "parentProblem": [
      {
        "id": "problemxxxx0001",
        "correlationId": "xxxxxxxx",
        "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/problemxxxx0001"
      }
    ],
    "underlyingProblem": [
      {
        "id": "problemxxxx0001",
        "correlationId": "xxxxxxxx",
        "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/problemxxxx0001"
      }
    ],
    "trackingRecord": [
      {
        "description": "yyy cleared the problem",
        "systemId": "xxxx",
        "time": "2014-12-20T17:00:00Z",
        "user": "NPUSER1",
        "id": "TR_xxxx",
        "extensionInfo": [
          {
            "name": "vendorComment",
            "valueType": "string",
            "value": "Watch out for the dog"
          }
        ]
      }
    ],
    "comment": [
      {
        "author": {
          "id": "SPM_handler_01",
          "href": "https://mycsp.com:8080/tmf-api/partyManagement/v3/party/SPM_handler_01",
          "@referredType": "Individual"
        },
        "date": "2014-12-20T17:00:00Z",
        "systemId": "System_002",
        "text": "receive trouble ticket from NP1, and create this Service Problem"
      },
      {
        "author": {
          "id": "NP1_handler_11",
          "href": "https://mycsp.com:8080/tmf- api/partyManagement/v3/party/NP1_handler_11",
          "@referredType": "Individual"
        },
        "date": "2014-12-20T17:00:00Z",
        "systemId": "System_002",
        "text": "status changed to Progress-Held"
      }
    ],
    "impactPatterns": {
      "description": "Many services are at risk in this problem",
      "extensionInfo": [
        {
          "name": "ImpactProbability",
          "value": {
            "@type": "boolean",
            "value": true
          }
        }
```

```
      ]
   },
   "extensionInfo": [
      {
         "name": "EstimatedCost",
         "value": {
            "@type": "integer",
            "value": "20"
         }
      }
   ]
}
```

Create service problem

POST /serviceProblem

Description

This operation creates a service problem entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a ServiceProblem, including any possible rule conditions and applicable default values. Notice that it is up to an implementer to add additional mandatory attributes.

Mandatory Attributes	Rule
category	
priority	
description	
reason	
originatorParty	

Non Mandatory Attributes	Default Value	Rule
affectedLocation		
affectedNumberOfServices	0	
affectedResource		
affectedService		
associatedSLAViolation		
associatedTroubleTicket		
comment		
correlationId		
extensionInfo		
firstAlert		
impactImportanceFactor		
impactPatterns		
originatingSystem		
parentProblem		

Non Mandatory Attributes	Default Value	Rule
problemEscalation		
relatedEvent		
relatedObject		
relatedParty		
resolutionDate		
responsibleParty		
rootCauseResource		
rootCauseService		
status		
statusChangeDate		
statusChangeReason		
timeChanged		
timeRaised		
trackingRecord		
underlyingAlarm		
underlyingProblem		

Additional Rules

The following table provides additional rules indicating mandatory fields in sub-resources or relationships when creating a ServiceProblem resource.

Context		Mandatory Sub-Attributes
relatedParty		id, href

Usage Samples

Here's an example of a request for creating a specific service problem. In this example the request only passes mandatory attributes.

Request

```
POST {apiRoot}/serviceProblem
Content-Type: application/json

{
  "category": "serviceProvider.declared",
  "priority": "1",
  "description": "Internet connection error",
  "reason": "unknown",     "originatorParty": {
    "role": " ServiceProvider ",
    "id": " SP_00001",
    "href": "https://mycsp.com:8080/tmf-api/partyManagement/v3/party/SP_00001",
    "@referredType": "Organization"
  },
  "affectedService": [
    {
            "id": "SP00001_Service_001",
            "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v3/service/SP00001_Service_001"
        }
    ]
}
```

Response

```
201

{
    "id": "sp_001",
    "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/sp_001",
    "category": "serviceProvider.declared",
    "priority": "1",
    "description": "Internet connection error",
    "reason": "unknown",
    "originatorParty": {
        "role": " ServiceProvider ",
        "id": "SP_00001",
        "href": "https://mycsp.com:8080/tmf- api/partyManagement/v3/party/SP_00001",
        "@referredType": " Organization "
    },
    "affectedService": [
        {
            "id": "SP00001_Service_001",
            "href": "https://mycsp.com:8080/tmf- api/serviceInventoryManagement/v3/service/SP00001_Service_001"
        }
    ]
}
```

Patch service problem

PATCH /serviceProblem/{id}

Description

This operation allows partial updates of a service problem entity. Support of json/merge (https://tools.ietf.org/html/rfc7386) is mandatory, support of json/patch (http://tools.ietf.org/html/rfc5789) is optional.

Note: If the update operation yields to the creation of sub-resources or relationships, the same rules concerning mandatory sub-resource attributes and default value settings in the POST operation applies to the PATCH operation. Hence these tables are not repeated here.

Patchable and Non Patchable Attributes

The tables below provide the list of patchable and non patchable attributes, including constraint rules on their usage.

Patchable Attributes	Rule
affectedLocation	
affectedNumberOfServices	
affectedResource	
affectedService	
associatedSLAViolation	
associatedTroubleTicket	
category	
comment	
description	
extensionInfo	
impactImportanceFactor	
impactPatterns	
originatorParty	
parentProblem	
priority	
problemEscalation	
reason	
relatedEvent	
relatedObject	
relatedParty	
resolutionDate	
responsibleParty	
rootCauseResource	
rootCauseService	
status	
statusChangeDate	
statusChangeReason	
timeChanged	
underlyingAlarm	
underlyingProblem	

Non Patchable Attributes	Rule
correlationId	
firstAlert	
href	
id	
originatingSystem	
timeRaised	
trackingRecord	

Usage Samples

Here's an example of a request for patching a service problem. In this example, a new description is set.

Request

```
PATCH {apiRoot}/serviceProblem/problemxxxx0000
Content-Type: application/merge-patch+json

{
   "description": "connection failure between Tokyo and Osaka at 5:00"
}
```

Response

```
200

{
   "id": "sp_001",
   "href": "https://mycsp.com:8080/tmf- api/serviceProblemManagement/v3/serviceProblem/sp_001",
   "category": "serviceProvider.declared",
   "priority": "1",
   "description": "connection failure between Tokyo and Osaka at 5:00",
   "reason": "unknown",
   "originatorParty": {
      "role": "Service Provider",
      "id": "SP_00001",
      "href": "https://mycsp.com:8080/tmf-api/partyManagement/v3/party/SP_00001",
      "@referredType": "Supplier"
   },
   "affectedService": [
      {
         "id": "SP00001_Service_001",
         "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v3/service/SP00001_Service_001"
      }
   ]
}
```

Here's an example of a request for patching a service problem. In this example, an affected service is added.

Request

```
PATCH {apiRoot}/serviceProblem/problemxxxx0000
Content-Type: application//json-patch+json

{
   "op": "add",
   "path": "/affectedService",
   "value": {
      "id": "44",
      "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v3/service/service/44"
    }
}
```

Response

```
200

{
    "id": "sp_001",
    "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/sp_001",
    "category": "serviceProvider.declared",
    "priority": "1",
    "description": "connection failure between Tokyo and Osaka at 5:00",
    "reason": "unknown",
    "originatorParty": {
        "role": "Service Provider",
        "id": "SP_00001",
        "href": "https://mycsp.com:8080/tmf-api/partyManagement/v3/party/SP_00001",
        "@referredType": "Supplier"
    },
    "affectedService": [
        {
            "id": "SP00001_Service_001",
            "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v3/service/SP00001_Service_001"
        },
        {
            "id": "44",
            "href": "https://mycsp.com:8080/tmf-api/serviceInventoryManagement/v3/service/44"
        }
    ]
}
```

Delete service problem

DELETE /serviceProblem/{id}

Description

This operation deletes a service problem entity.

Usage Samples

Here's an example of a request for deleting a service problem.

Request

```
DELETE {apiRoot}/serviceProblem/problemxxxx0000
```

Response

```
204
```

Operations on Service Problem Event Record

List service problem event records

GET /serviceProblemEventRecord?fields=...&{filtering}

Description

This operation list service problem event record entities.
Attribute selection is enabled for all first level attributes.
Filtering may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving multiple event records for a service problems

Request

```
GET {apiRoot}/serviceProblemEventRecord?serviceProblem.id=problemxxxx0000
Accept: application/json
```

Response

```
200

[
  {
    "id": "42",
    "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblemEventRecord/42",
    "recordTime": "2016-08-08T10:45:30.0Z",
    "eventType": "ServiceProblemCreationNotification",
    "eventTime": "2016-08-08T10:45:25.0Z",
    "serviceProblemId": "SP001",
    "notification": {
      "eventType": "ServiceProblemCreationNotification",
      "eventTime": "2016-08-08T10:45:25.0Z",
      "eventId": "92775",
      "event": {}
    },
    "serviceProblem": {
      "id": "problemxxxx0000",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/problemxxxx0000"
    }
  }
]
```

Retrieve service problem event record

GET /serviceProblemEventRecord/{id}?fields=...&{filtering}

Description

This operation retrieves a service problem event record entity.
Attribute selection is enabled for all first level attributes.
Filtering on sub-resources may be available depending on the compliance level supported by an implementation.

Usage Samples

Here's an example of a request for retrieving a specific service problem event record.

Request

```
GET {apiRoot}/serviceProblemEventRecord/42
Accept: application/json
```

Response

```
200

{
  "id": "42",
  "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblemEventRecord/42",
  "recordTime": "2016-08-08T10:45:30.0Z",
  "eventType": "ServiceProblemCreationNotification",
  "eventTime": "2016-08-08T10:45:25.0Z",
  "serviceProblemId": "SP001",
  "notification": {
    "eventType": "ServiceProblemCreationNotification",
    "eventTime": "2016-08-08T10:45:25.0Z",
    "eventId": "92775",
    "event": {}
  },
  "serviceProblem": {
    "id": "problemxxxx0000",
        "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/problemxxxx0000"
    }
}
```

Operations on Problem Acknowledgement

Create problem acknowledgement

POST /problemAcknowledgement

Description

This operation creates a problem acknowledgement entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a ProblemAcknowledgement, including any possible rule conditions and applicable default values. Notice that it is up to an implementer to add additional mandatory attributes.

Mandatory Attributes	Rule
problem	

Non Mandatory Attributes	Default Value	Rule
ackProblem		
trackingRecord		

Usage Samples

Here's an example of a request for acknowledging service problem. In this example the request is handled synchronously so the task resource is not persisted.

Request

```
POST {apiRoot}/problemAcknowledgement
Content-Type: application/json

{
    "problem": [
        {
            "id": "41",
            "href": "https://mycsp.com:8080/tmf-api/servic eProblemManagement/v3/serviceProblem/41"
        },
        {
            "id": "42",
            "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/42"
        },
        {
```

```
      "id": "43",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
    }
  ],
  "trackingRecord": {
    "description": "yyy ack the problem",
    "systemId": "xxxx",
    "time": "2014-12-20T17:00:00Z",
    "user": "NP1"
  }
}
```

Response

```
201

{
  "problem": [
    {
      "id": "41",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/41"
    },
    {
      "id": "42",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/42"
    },
    {
      "id": "43",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
    }
  ],
  "trackingRecord": {
    "description": "yyy ack the problem",
    "systemId": "xxxx",
    "time": "2014-12-20T17:00:00Z",
    "user": "NP1"
  }
}
```

Operations on Problem Unacknowledgement

Create problem unacknowledgement

`POST /problemUnacknowledgement`

Description

This operation creates a problem unacknowledgement entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a ProblemUnacknowledgement, including any possible rule conditions and applicable default values. Notice that it is up to an implementer to add additional mandatory attributes.

Mandatory Attributes	Rule
problem	

Non Mandatory Attributes	Default Value	Rule
trackingRecord		
unackProblem		

Usage Samples

Here's an example of a request for reversing the acknowledgment of a service problem. In this example the request is handled synchronously so the task resource is not persisted.

Request

```
POST {apiRoot}/problemUnacknowledgement
Content-Type: application/json

{
   "problem": [
      {
         "id": "41",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/41"
      },
      {
         "id": "42",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/42"
      },
      {
         "id": "43",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
      }
   ],
   "trackingRecord": {
      "description": "yyy unack the problem",
      "systemId": "xxxx",
      "time": "2014-12-20T17:00:00Z",
      "user": "NP1"
   }
}
```

Response

```
201

{
   "problem": [
      {
         "id": "41",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/41"
      },
      {
         "id": "42",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/42"
      },
      {
         "id": "43",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
      }
   ],
   "trackingRecord": {
      "description": "yyy unack the problem",
      "systemId": "xxxx",
      "time": "2014-12-20T17:00:00Z",
      "user": "NP1"
   }
}
```

Operations on Problem Group

Create problem group

POST /problemGroup

Description

This operation creates a problem group entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a ProblemGroup, including any possible rule conditions and applicable default values. Notice that it is up to an implementer to add additional mandatory attributes.

Mandatory Attributes	Rule
parentProblem	
childProblem	

Non Mandatory Attributes	Default Value	Rule

Usage Samples

Here's an example of a request for requesting service problems to be grouped together with a parent. In this example the request is handled synchronously so the task resource is not persisted.

Request

```
POST {apiRoot}/problemGroup
Content-Type: application/json

{
   "childProblem": [
      {
         "id": "41",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/41"
      },
      {
         "id": "42",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/42"
      },
      {
         "id": "43",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
      }
   ],
   "parentProblem": {
      "id": "43",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
   }
}
```

Response

```
201

{
   "childProblem": [
      {
         "id": "41",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/41"
      },
      {
         "id": "42",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/42"
      },
      {
         "id": "43",
         "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
      }
   ],
   "parentProblem": {
    "id": "43",
    "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
  }
}
```

Operations on Problem Ungroup

Create problem ungroup

POST /problemUngroup

Description

This operation creates a problem ungroup entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a ProblemUngroup, including any possible rule conditions and applicable default values. Notice that it is up to an implementer to add additional mandatory attributes.

Mandatory Attributes	Rule
parentProblem	
childProblem	

Non Mandatory Attributes	Default Value	Rule

Usage Samples

Here's an example of a request for requesting service problems to be ungrouped from a parent. In this example the request is handled synchronously so the task resource is not persisted.

Request

```
POST {apiRoot}/problemUngroup
Content-Type: application/json

{
  "childProblem": [
    {
      "id": "41",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/41"
    },
    {
      "id": "42",
      "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/42"
    },
    {
            "id": "43",
            "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
        }
    ],
    "parentProblem": {
        "id": "43",
        "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
    }
}
```

Response

```
201

{
    "childProblem": [
        {
            "id": "41",
            "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/41"
        },
        {
            "id": "42",
            "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/42"
        },
        {
            "id": "43",
            "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
        }
    ],
    "parentProblem": {
        "id": "43",
        "href": "https://mycsp.com:8080/tmf-api/serviceProblemManagement/v3/serviceProblem/43"
    }
}
```

API NOTIFICATIONS

For every single of operation on the entities use the following templates and provide sample REST notification POST calls.

It is assumed that the Pub/Sub uses the Register and UnRegister mechanisms described in the REST Guidelines part 1. Refer to the guidelines for more details.

Register listener

POST /hub

Description

Sets the communication endpoint address the service instance must use to deliver information about its health state, execution state, failures and metrics. Subsequent POST calls will be rejected by the service if it does not support multiple listeners. In this case DELETE /api/hub/{id} must be called before an endpoint can be created again.

Behavior

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 409 if request is not successful.

Usage Samples

Here's an example of a request for registering a listener.

Request
POST /api/hub Accept: application/json {"callback": "http://in.listener.com"}
Response
201 Content-Type: application/json Location: /api/hub/42 {"id":"42","callback":"http://in.listener.com","query":null}

Unregister listener

DELETE /hub/{id}

Description

Clears the communication endpoint address that was set by creating the Hub.

Behavior

Returns HTTP/1.1 status code 204 if the request was successful.

Returns HTTP/1.1 status code 404 if the resource is not found.

Usage Samples

Here's an example of a request for un-registering a listener.

Request
DELETE /api/hub/42 Accept: application/json
Response
204

Publish Event to listener

POST /client/listener

Description

Clears the communication endpoint address that was set by creating the Hub.

Provides to a registered listener the description of the event that was raised. The /client/listener url is the callback url passed when registering the listener.

Behavior

Returns HTTP/1.1 status code 201 if the service is able to set the configuration.

Usage Samples

Here's an example of a notification received by the listener. In this example “EVENT TYPE” should be replaced by one of the notification types supported by this API (see Notification resources Models section) and EVENT BODY refers to the data structure of the given notification type.

Request

```
POST /client/listener
Accept: application/json

{
   "event": {
              EVENT BODY
         },
   "eventType": "EVENT_TYPE"
}
```

Response

```
201
```

For detailed examples on the general TM Forum notification mechanism, see the TMF REST Design Guidelines Part 1 document.

Acknowledgements

Document History

Release History

Release Number	Date	Release led by:	Description
Release 1.0	20-Jun-2018	Pierre Gauthier TM Forum pgauthier@tmforum.org	First Release of the Document.
Release 3.0.0	16-Jan-2019	Jonathan Goldberg Amdocs Jonathan.Goldberg@amdocs.com	Schema alignment for NaaS APIs

Version History

Release Number	Date	Release led by:	Description
Release 18.5.0	16 Jan 2019	Jonathan Goldberg Amdocs Jonathan.Goldberg@amdocs.com	Schema alignment for NaaS APIs

Contributors to Document

Pierre Gauthier	TM Forum
Kiyotaka Mizuno	NTT
Takayuki Nakamura	NTT
Jonathan Goldberg	Amdocs