

TM Forum Specification

Partnership Management API REST Specification

TMF668
Release 19.0
July 2019

Latest Update: TM Forum Release 19.0	Member Evaluation
Version 4.0.0	IPR Mode: RAND

NOTICE

Copyright © TM Forum 2018. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to TM FORUM, except as needed for the purpose of developing any document or deliverable produced by a TM FORUM Collaboration Project Team (in which case the rules applicable to copyrights, as set forth in the [TM FORUM IPR Policy](#), must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by TM FORUM or its successors or assigns.

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. +1 973 944 5110
TM Forum Web Page: www.tmforum.org

TABLE OF CONTENTS

NOTICE	2
Table of Contents	3
List of Tables	5
Introduction	6
SAMPLE USE CASES	7
Support of polymorphism and extension patterns	15
RESOURCE MODEL	16
Managed Entity and Task Resource Models	16
Partnership Specification resource	16
Partnership resource	19
Notification Resource Models	26
Partnership Specification Create Event	27
Partnership Specification Change Event	28
Partnership Specification Delete Event	28
Partnership Create Event	28
Partnership Change Event	29
Partnership Delete Event	29
API OPERATIONS	31
Operations on Partnership Specification	31
List partnership specifications	32
Retrieve partnership specification	33
Create partnership specification	34
Patch partnership specification	35
Delete partnership specification	36
Operations on Partnership	37
List partnerships	37
Retrieve partnership	38

Create partnership	40
Patch partnership	41
Delete partnership	42
API NOTIFICATIONS	44
Register listener	44
Unregister listener	45
Publish Event to listener	45
Acknowledgements	47
Release History	47
Contributors to Document	47

LIST OF TABLES

N/A

INTRODUCTION

The following document is the specification of the REST API for Partnership Management. It includes the model definition as well as all available operations.

Basically a partnership specification is defined by declaring a list of role specifications that the partners can play. A role specification is typically characterized by an agreement specification. In addition some role specifications imply the necessity to be linked to billing accounts. A Partnership instance in the other hand refers to its partnership specification and declares the list of partners – that is to say – the parties playing a given role in the partnership.

Technically, the partnership management API defines PartnershipSpecification and Partnership as root resources. RoleSpecifications are sub-objects contained within a PartnershipSpecification. A Partnership declares – as contained instances - the *partners* involved in the partnership. A *partner* may be linked to agreements and accounts (such as billing or settlements accounts) in line with partnership specification (which may declare the used agreement specifications). The declared role name being played by the engaged party (“role” attribute in the RelatedParty structure) should match one of the role specifications declared in the partnership specification.

The following APIs are typically used in conjunction with this API:

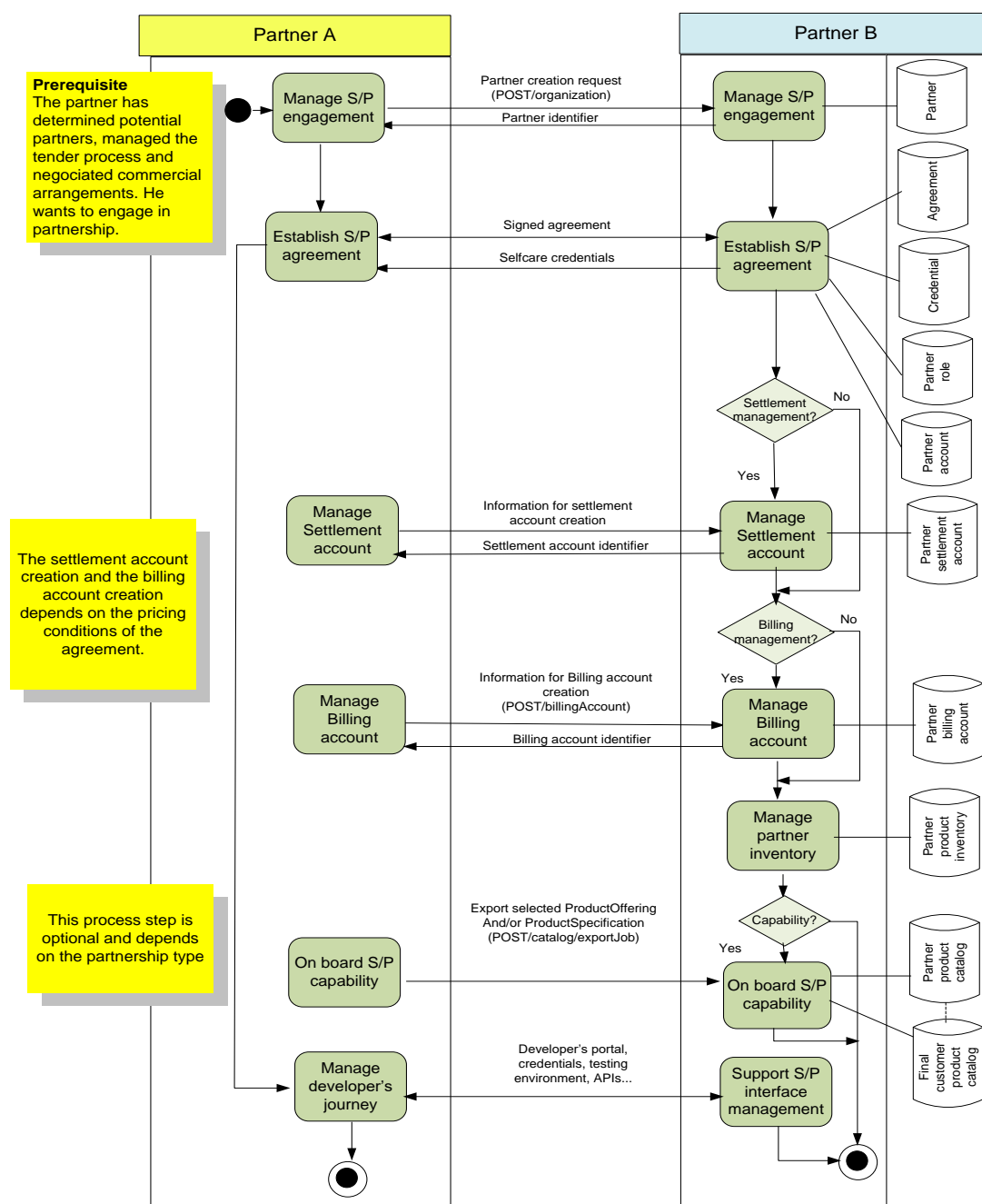
- Party API: defining the parties
- Agreement API: Definition of the terms of an agreement or an agreement specification
- Account API: Definition of billing/settlement/financial accounts that may be attached to the partners
- PartyRole API: Maybe used for creating the Partners entities – representing the coupling of a party with a given role – and then be attached to a given partnership. However Partners can be created directly as sub-objects of a given Partnership.

SAMPLE USE CASES

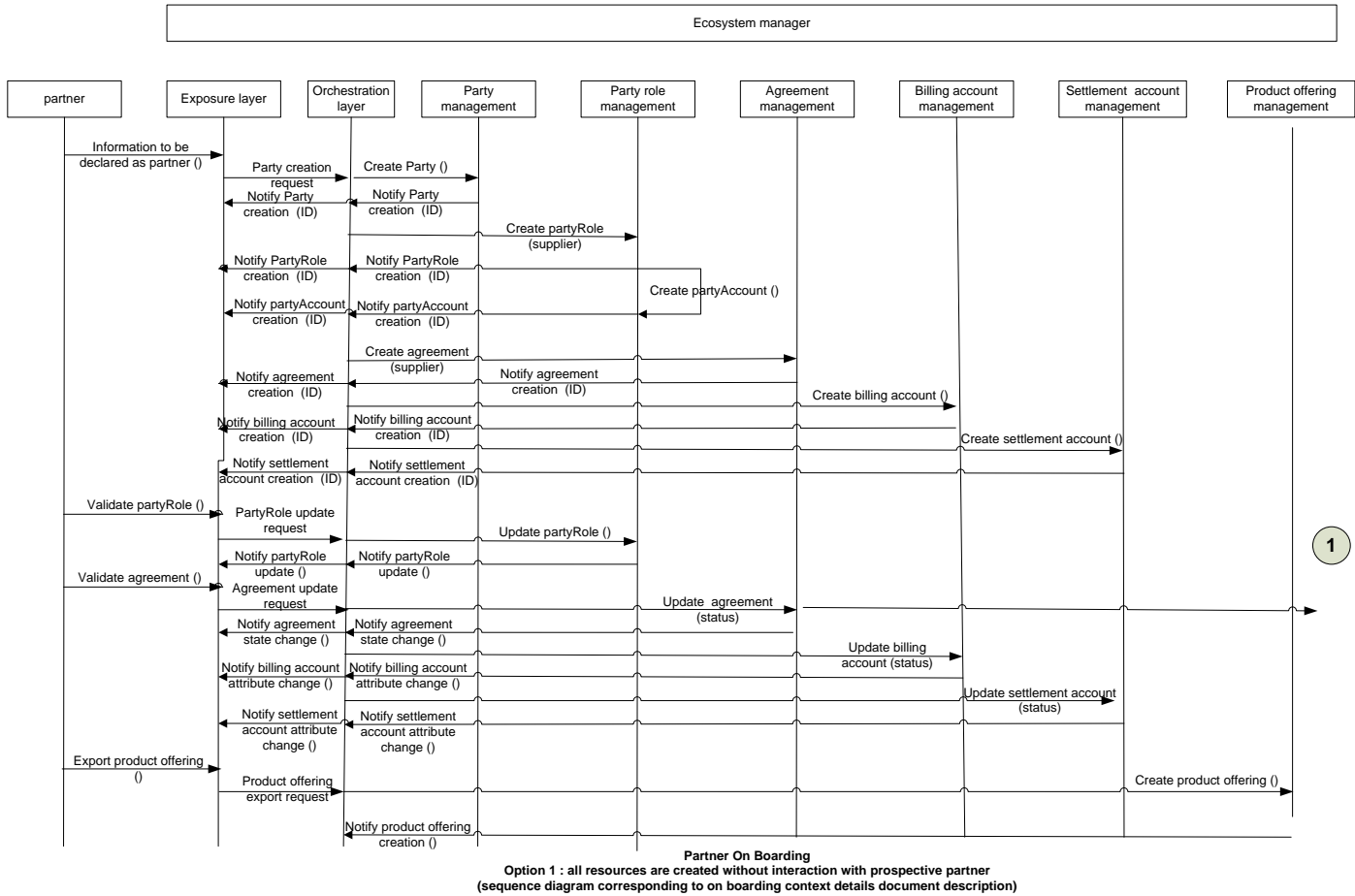
In this section we provide some typical API usage scenarios described in a light-weight fashion. The intent is not to describe all possible contexts of use of the API (The process for building a partnership is out of scope of this api specification).

A Global view

The figure below depicts an on-boarding business process at a very high level of abstraction.



Option 1: All resources are created without interaction with prospective partner

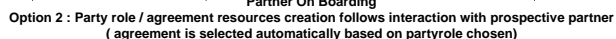


This sequence diagram describes a partner on boarding process where:

- prospective partner is not known by ecosystem manager system
- all resources (party, partyRole, Party account, agreement, billing account and / or settlement account) are created without interaction with prospective partner
- prospective partner validate partyRole and, then, partyRole is validated
- prospective partner validate agreement and, then, agreement, billing account and / or settlement account are updated
- prospective partner validate

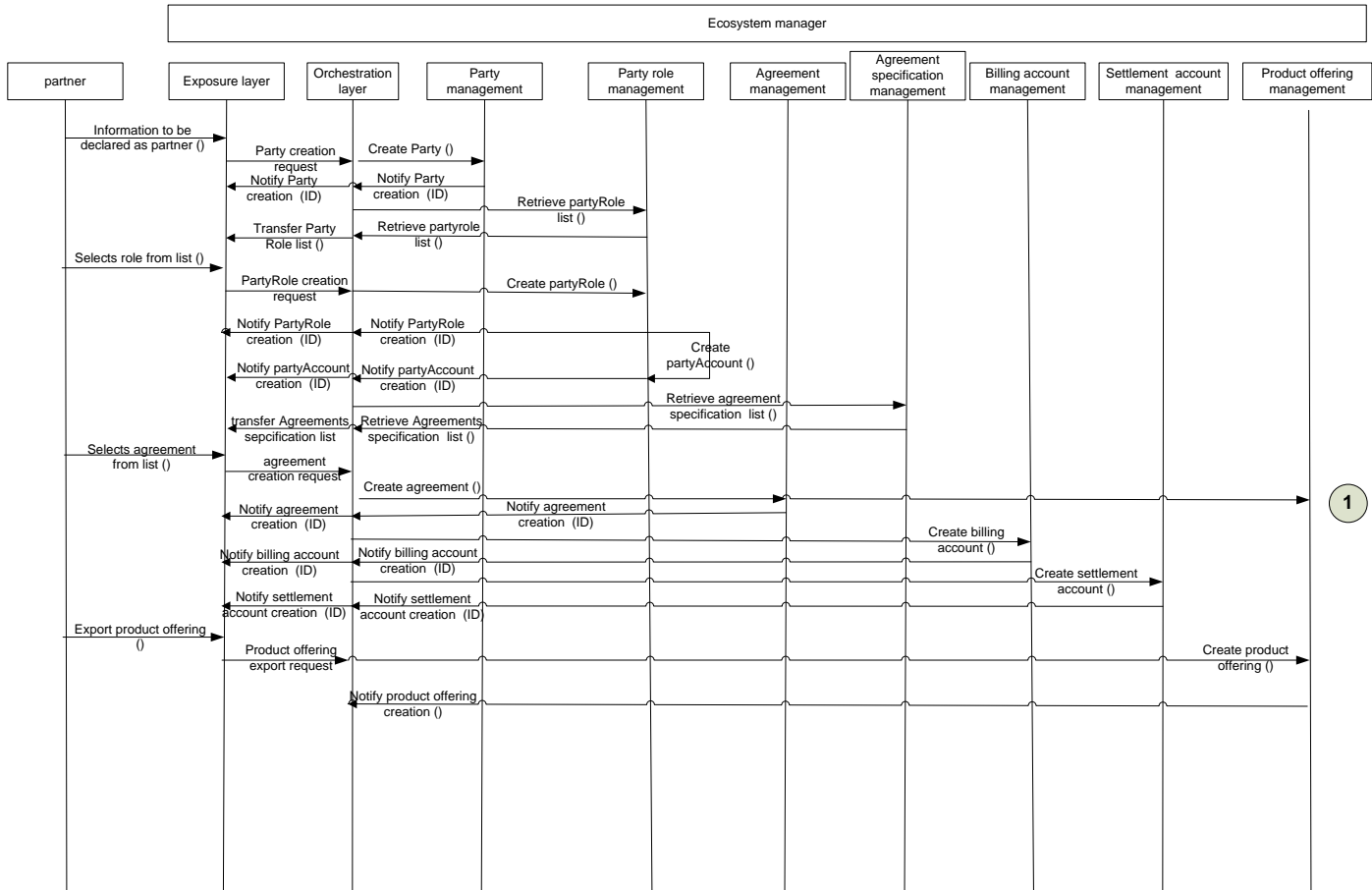
Option 2: Party role / agreement resources creation follows interaction with prospective partner

(Agreement is selected automatically based on partyrole chosen)



- prospective partner is not known by ecosystem manager system
- party resources is created
- list of partyRole is retrieved
- Prospective partner selects a partyRole
- PartyRole, party account , agreement billing account and / or settlement account resources are created
- prospective partner validate agreement and, then, agreement, billing account and / or settlement account are updated

(Agreement is selected by prospective partner)



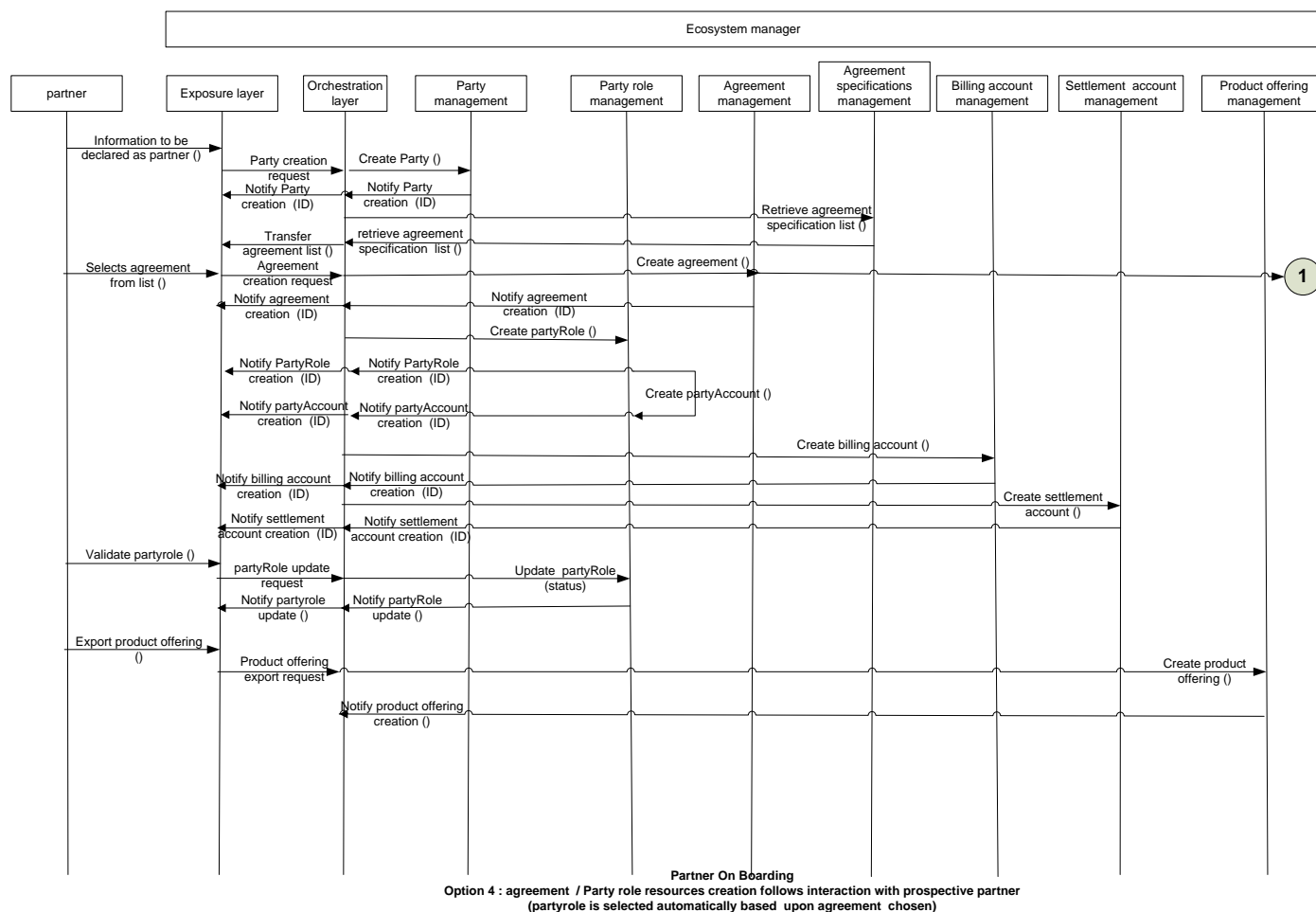
Partner On Boarding
Option 3: Party role / agreement resources creation follows interaction with prospective partner
(agreement is selected by prospective partner)

This sequence diagram describes a partner on boarding process where:

- prospective partner is not known by ecosystem manager system
- party resources is created
- list of partyRole is retrieved
- prospective partner selects a partyRole
- partyRole and party account resources are created
- list of agreement specifications is retrieved
- prospective partner selects an agreement specification
- agreement, billing account and / or settlement account resources are created

Option 4: agreement / party role resources creation follows interaction with prospective partner

(Party role is selected automatically based upon agreement chosen)

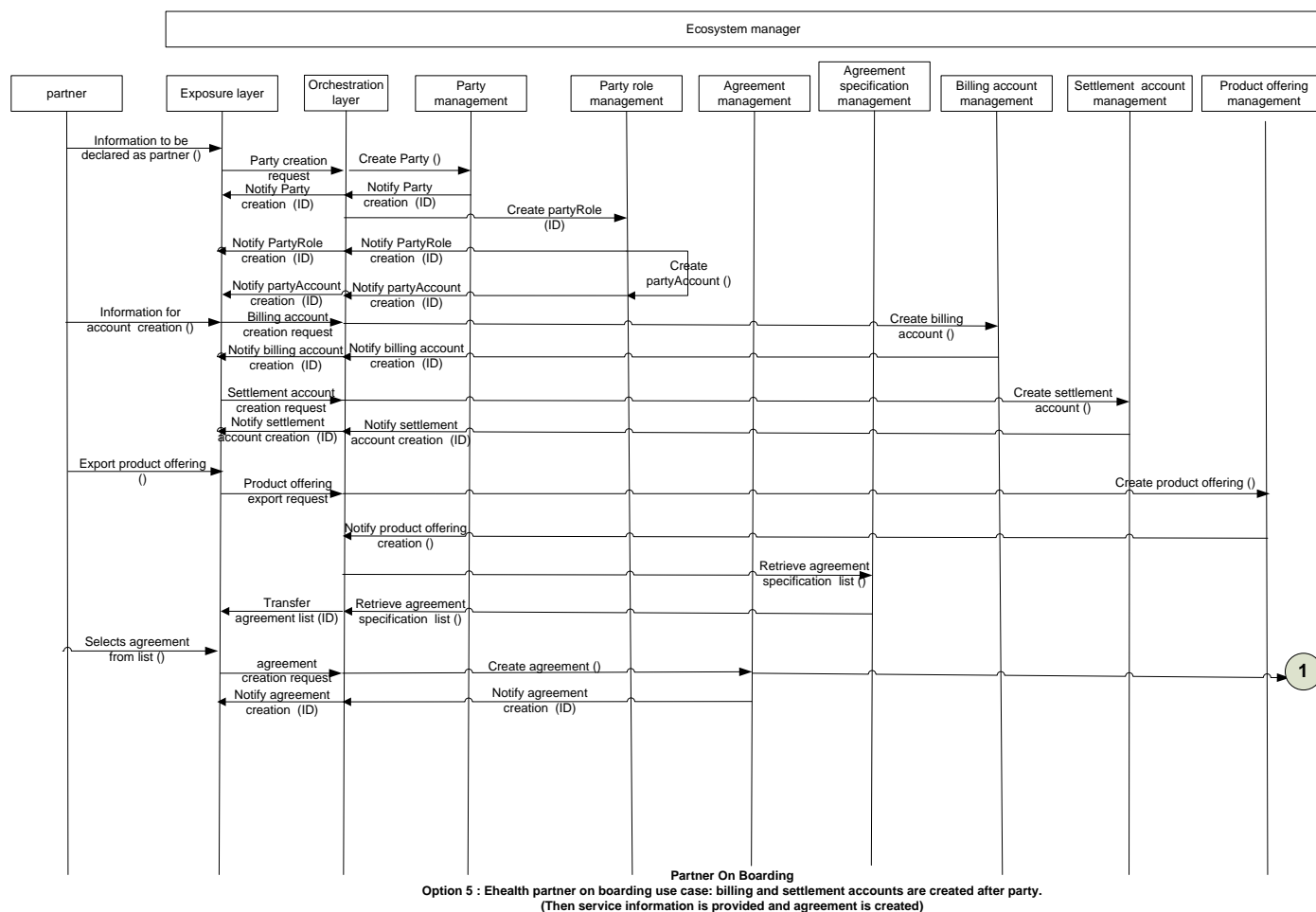


This sequence diagram describes a partner on boarding process where:

- prospective partner is not known by ecosystem manager system
- party resources is created
- list of agreement specifications is retrieved
- prospective partner selects an agreement specification
- agreement, billing account and / or settlement account resources are created
- partyRole and partyAccount are created depending on agreement specification selected
- Prospective partner validates partyRole
- PartyRole is updated

Option 5: E-health partner on boarding use case: billing and settlement accounts are created after party

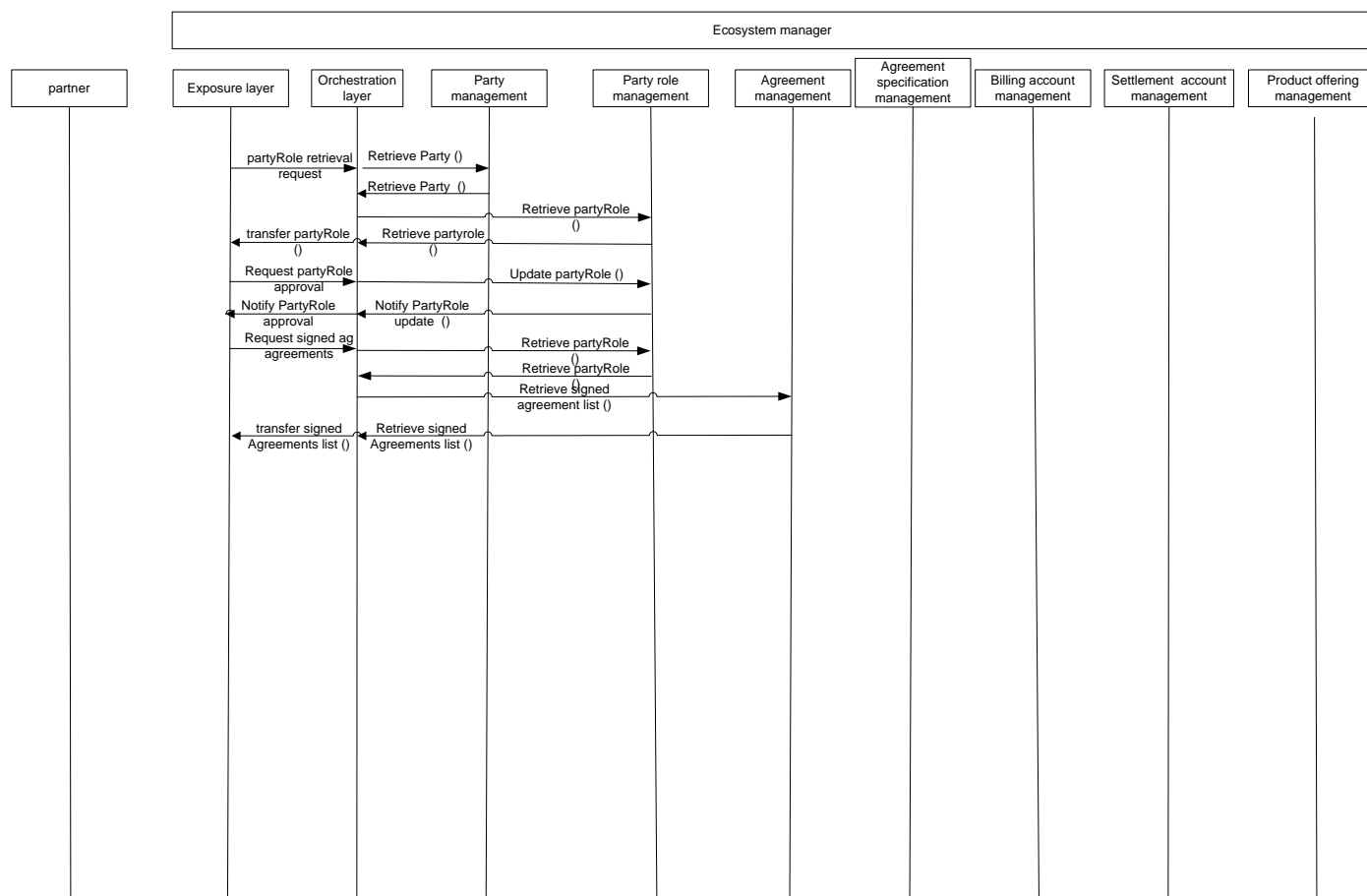
(Then service information is provided and agreement is created)



This sequence diagram describes the eHealth partner on boarding process where:

- prospective partner is not known by ecosystem manager system
- Creation of partner includes creation of party, partyRole and Party account
- Creation of billing account is triggered by an account creation request from partner
- "Service info" provision by partner triggers product offering creation
- Creation of product offering triggers agreement list notification.

Option 6: Party has already selected a party role, party role needs to be approved, agreement signed by party role needs to be retrieved

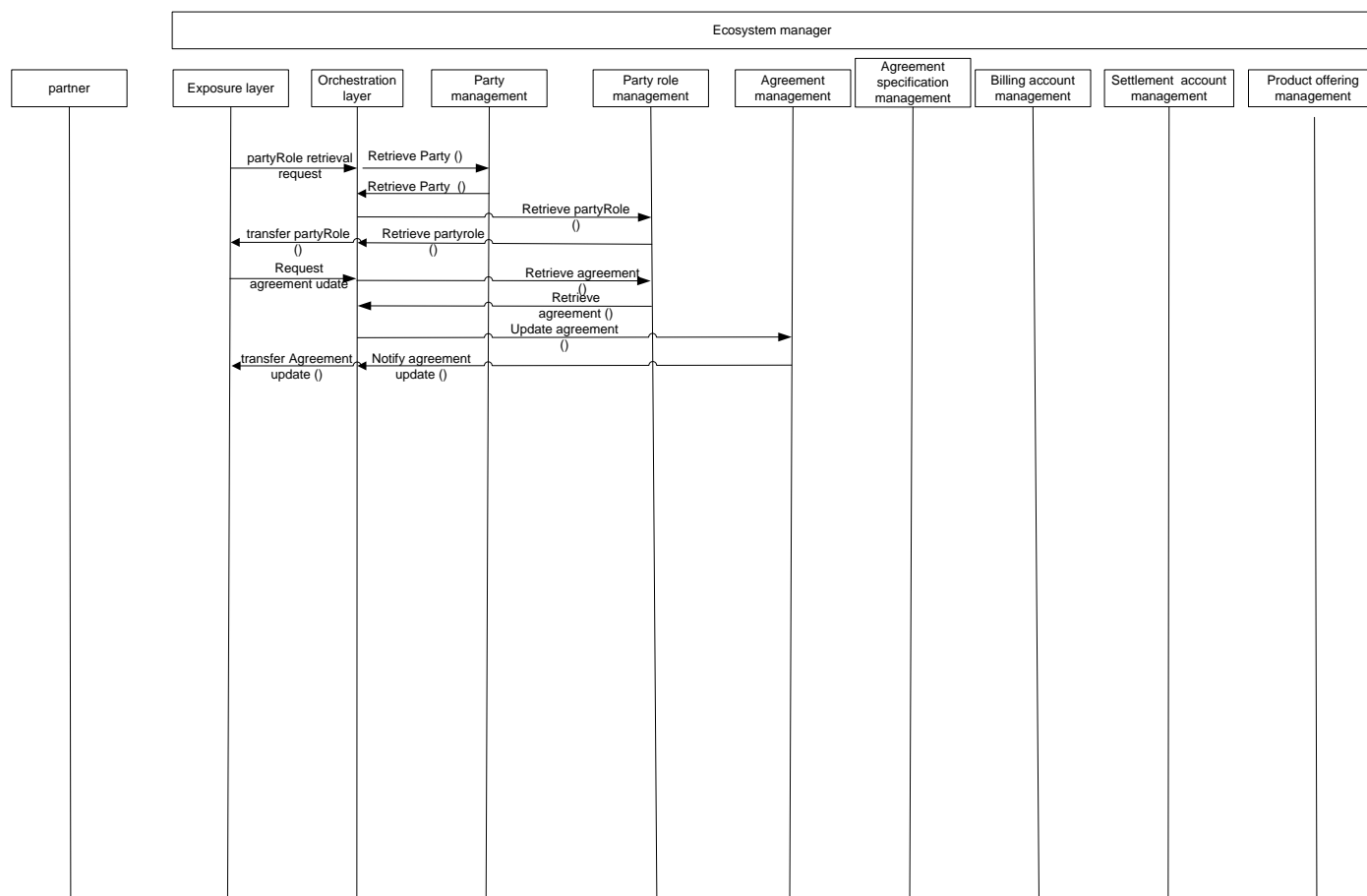


Partner On Boarding
Option 6: party has already selected a partyRole, partyRole needs to be approved, Agreement signed by partyRole needs to be retrieved

This sequence diagram describes a partner on boarding process where:

- prospective partner already exists and needs to be approved
- Party is retrieved (relevance to be confirmed)
- partyRole is retrieved
- PartyRole is approved
- list of agreements signed by partner (partyRole) are retrieved

Option 7: Party role exist, agreement is updated



Partner On Boarding
Option 7: partyRole exist, Agreement is updated

This sequence diagram describes a partner on boarding process where:

- prospective partner already exists and agreement needs to be updated
- Party is retrieved (relevance to be confirmed)
- partyRole is retrieved
- list of agreements related to party role are retrieved
- agreement is updated

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 `BillingAccount` and `SettlementAccount` inheriting properties from the abstract `Account` 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 `Account` instances some may be instances of `BillingAccount` where other could be instances of `SettlementAccount`. 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 `AccountRef` 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 `BillingAccountRef` or `SettlementAccountRef`, 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.

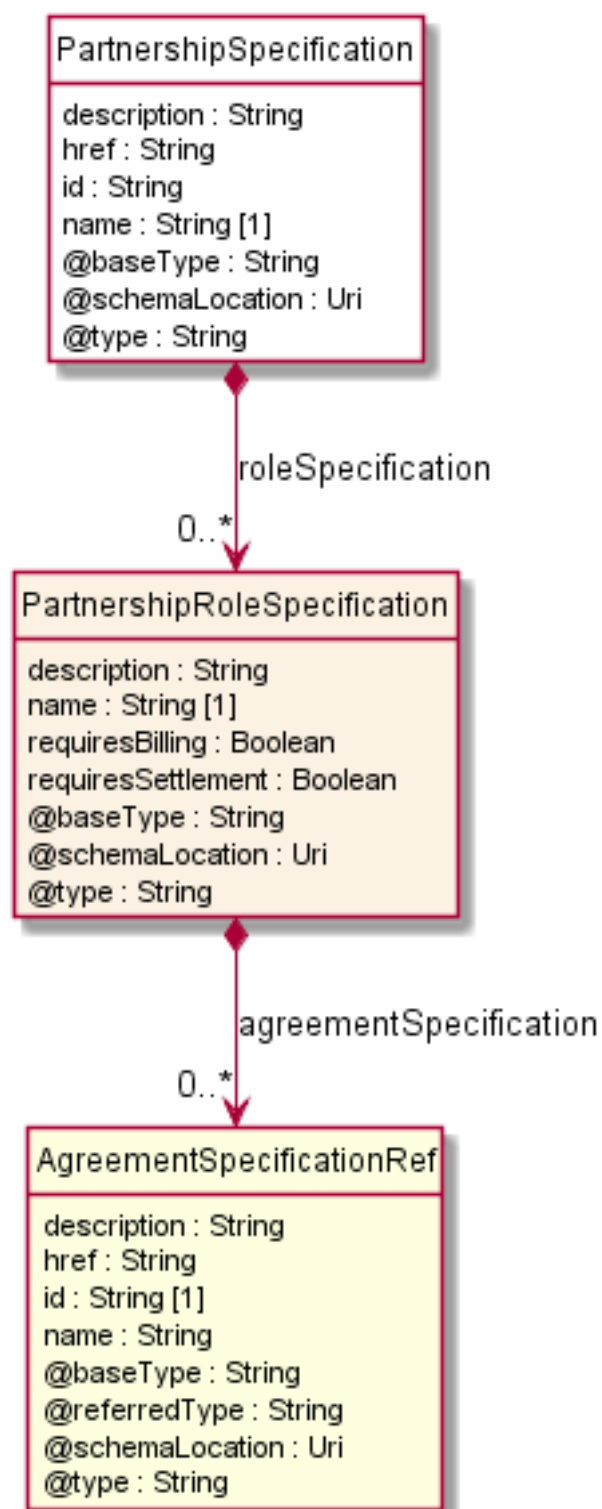
RESOURCE MODEL

Managed Entity and Task Resource Models

PARTNERSHIP SPECIFICATION RESOURCE

A partnership specification contains all the information for the setup of a partnership of a given kind. This includes the list of identified role types for the partnership with the corresponding agreement specifications.

Resource model



Field descriptions

PartnershipSpecification fields

description A string. An explanatory text regarding this partnership specification.

href	A string. The reference url for this partnership specification.
id	A string. The identifier of the partnership specification.
name	A string. An identifying name for the partnership specification.
roleSpecification	A list of partnership role specifications (PartnershipRoleSpecification [*]). The list of roles specifications a engaged party will be able to play for this kind of partnership.

PartnershipRoleSpecification sub-resource

A PartnershipRoleSpecification represents a specification of a role defined in the context of a given type of partnership (PartnershipSpecification), such as Buyer, Seller.

description	A string. An explanatory text documenting the role specification.
name	A string. The name of the role specification. This name uniquely identifies the role specification within the partnership.
requiresBilling	A boolean. Indicates whether billing operations will be associated to parties playing the role.
requiresSettlement	A boolean. Indicates whether settlement operations will be associated to parties playing the role.
agreementSpecification	A list of agreement specification references (AgreementSpecificationRef [*]). An AgreementSpecification represents a template of an agreement that can be used when establishing partnerships.

AgreementSpecificationRef relationship

AgreementSpecification reference. An AgreementSpecification represents a template of an agreement that can be used when establishing partnerships.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
href	A string. Reference of the related entity.
id	A string. Unique identifier of a related entity.
name	A string. Name of the agreement specification.
description	A string. A narrative that explains in detail what the agreement specification is about.

Json representation sample

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

```
{
  "@type": "PartnershipSpecification",
```

```

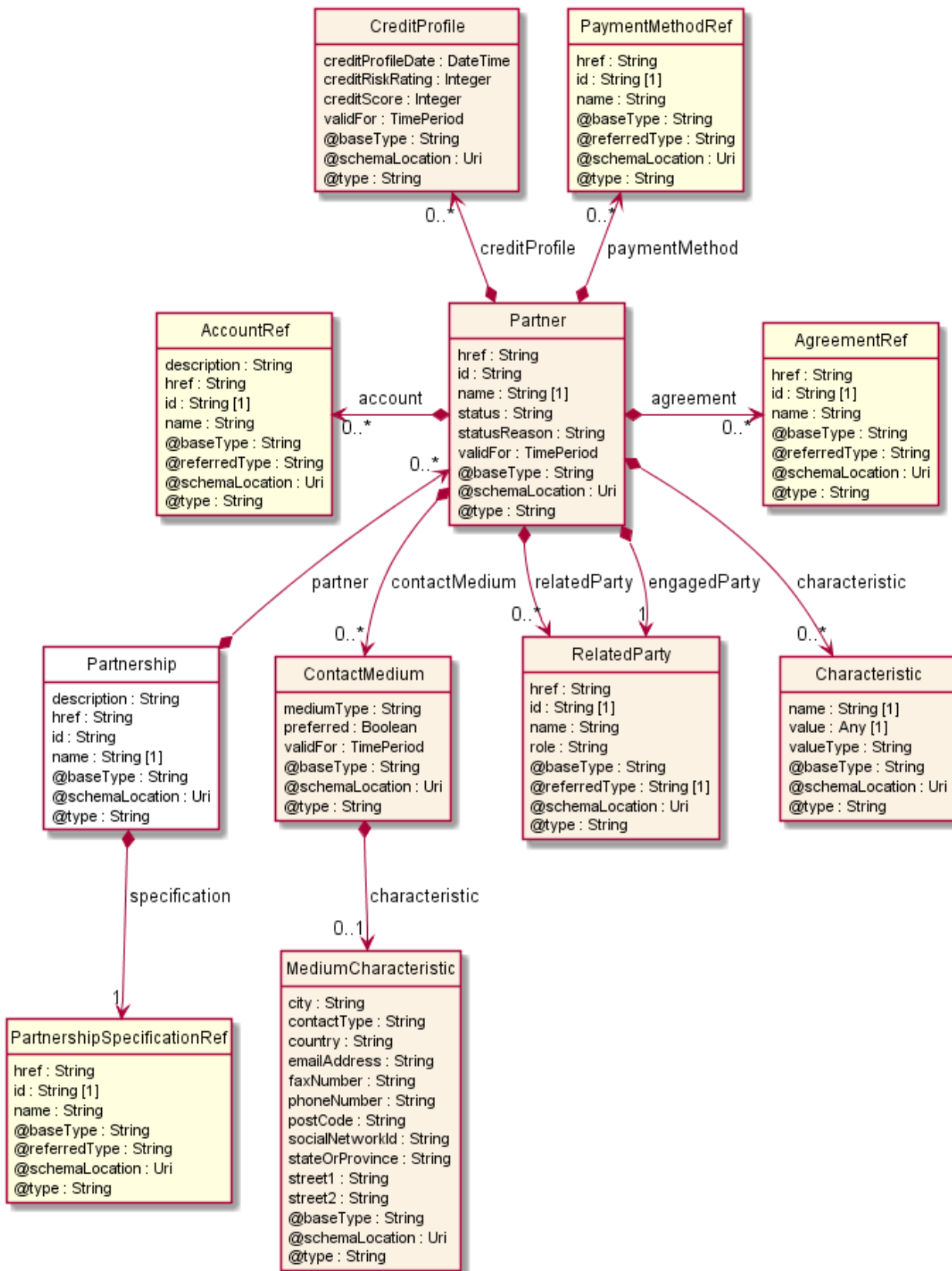
"description": "This partnership specification gives the template for defining a partnership with a content provider",
"href": "https://host:port/tmf-api/partnershipManagement/v4/partnershipSpecification/9047",
"id": "9047",
"name": "Dream Partnership template",
"roleSpecification": [
  [
    {
      "@type": "RoleSpecification",
      "name": "ContentProvider",
      "agreementSpecification": [
        {
          "name": "ContentLicenseAgreement",
          "id": "33"
        }
      ]
    }
  ],
  {
    "@type": "RoleSpecification",
    "name": "CloudProvider"
  },
  {
    "@type": "RoleSpecification",
    "name": "Developer",
    "agreementSpecification": [
      {
        "name": "ProfitShareAgreement",
        "id": "32"
      }
    ]
  },
  {
    "@type": "RoleSpecification",
    "name": "Tester"
  }
]
]
}

```

PARTNERSHIP RESOURCE

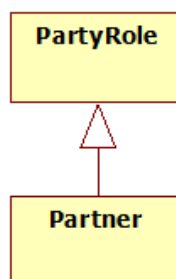
A partnership represents a formalized collaboration between various parties, each party playing a role in the partnership. The parties playing a role in the partnership represented by the Partner entity. The structure of a partnership is specified by a PartnershipSpecification which contain all the information for the setup of the partnership, including the list of identified roles and the agreement that should apply.

Resource model



Note on Partner inheritance: The Partner entity is conceptually a sub-class of the generic PartyRole entity that is defined within within the PartyRole API, as depicted by the diagram below. The structure is

the same except that attribute descriptions in Partner entity are overridden to match the vocabulary that is accurate for partnerships.



Field descriptions

Partnership fields

description	A string. An explanatory text regarding this partnership.
href	A string. The reference url for this partnership.
id	A string. The identifier of the partnership.
name	A string. An identifying name for the partnership.
partner	A list of partners (Partner [*]). The list of partners of the partnership, where a partner represents a party playing a given role. Hence a partner structure includes primarily a reference to the engaged party and a role name matching necessarily the name of one of the role specifications defined in the specification of the partnership.
specification	A partnership specification reference (PartnershipSpecificationRef). The specification of this partnership.

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.

ContactMedium sub-resource

Indicates the contact medium that could be used to contact the party.

characteristic	A medium characteristic (MediumCharacteristic). Any additional characteristic(s) of this contact medium.
----------------	--

mediumType	A string. Type of the contact medium, such as: email address, telephone number, postal address.
preferred	A boolean. If true, indicates that is the preferred contact medium.
validFor	A time period. The time period that the contact medium is valid for.

CreditProfile sub-resource

Credit profile for the party (containing credit scoring, ...). By default only the current credit profile is retrieved. It can be used as a list to give the party credit profiles history, the first one in the list will be the current one.

creditProfileDate	A date time (DateTime). The date the profile was established.
creditRiskRating	An integer. This is an integer whose value is used to rate the risk.
creditScore	An integer. A measure of a person or organizations creditworthiness calculated on the basis of a combination of factors such as their income and credit history.
validFor	A time period. The period for which the profile is valid.

MediumCharacteristic sub-resource

Describes the contact medium characteristics that could be used to contact a party (an individual or an organization).

city	A string. The city.
contactType	A string. The type of contact, for example: phone number such as mobile, fixed home, fixed office. postal address such as shipping instalation....
country	A string. The country.
emailAddress	A string. Full email address in standard format.
faxNumber	A string. The fax number of the contact.
phoneNumber	A string. The primary phone number of the contact.
postCode	A string. Postcode.
socialNetworkId	A string. Identifier as a member of a social network.
stateOrProvince	A string. State or province.
street1	A string. Describes the street.
street2	A string. Complementary street description.

Partner sub-resource

A party playing a given role in the partnership.

account	A list of account references (AccountRef [*]). A account may be a party account or a financial account.
agreement	A list of agreement references (AgreementRef [*]). An agreement represents a contract or arrangement, either written or verbal and sometimes enforceable by law, such as a service level agreement or a customer price agreement. An agreement involves a number of other business entities, such as products, services, and resources and/or their specifications.
characteristic	A list of characteristics (Characteristic [*]). Describes the characteristic of a partner.
contactMedium	A list of contact mediums (ContactMedium [*]). Indicates the contact medium that could be used to contact the party.
creditProfile	A list of credit profiles (CreditProfile [*]). Credit profile for the party (containing credit scoring, ...). By default only the current credit profile is retrieved. It can be used as a list to give the party credit profiles history, the first one in the list will be the current one.
engagedParty	A related party (RelatedParty). The reference to the party engaged in this partnership.
href	A string. Url used to reference the partner.
id	A string. Unique identifier for Partners.
name	A string. A word, term, or phrase by which the Partner is known and distinguished from other Partners.
paymentMethod	A list of payment method references (PaymentMethodRef [*]). A payment method defines a specific mean of payment (e.g direct debit).
relatedParty	A list of related parties (RelatedParty [*]). Related Entity reference. A related party defines party or party role linked to a specific entity.
status	A string. Used to track the lifecycle status of the partner.
statusReason	A string. A string providing an explanation on the value of the status lifecycle. For instance if the status is Rejected, statusReason will provide the reason for rejection.
validFor	A time period. The time period that the Partner is valid for.

RelatedParty sub-resource

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

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

href	A string. Reference of the related entity.
id	A string. Unique identifier of a related entity.
name	A string. Name of the related entity.
role	A string. Role played by the related party.

AccountRef relationship

Account reference. A account may be a party account or a financial account.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
description	A string. Detailed description of the account.
href	A string. Reference of the account.
id	A string. Unique identifier of the account.
name	A string. Name of the account.

AgreementRef relationship

Agreement reference. An agreement represents a contract or arrangement, either written or verbal and sometimes enforceable by law, such as a service level agreement or a customer price agreement. An agreement involves a number of other business entities, such as products, services, and resources and/or their specifications.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
href	A string. Reference of the related entity.
id	A string. Unique identifier of a related entity.
name	A string. Name of the agreement.

PartnershipSpecificationRef relationship

PartnershipSpecification reference. A partnership specification contains all the information for the setup of a partnership of a given kind. This includes the list of identified role types for the partnership with the corresponding agreement specifications.

@referredType	A string. The actual type of the target instance when needed for disambiguation.
href	A string. Reference url for the partnership specification.
id	A string. Identifier of the partnership specification.
name	A string. Name of the partnership specification.

PaymentMethodRef relationship

PaymentMethod reference. A payment method defines a specific mean of payment (e.g direct debit).

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

Json representation sample

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

```
{
  "@type": "Partnership",
  "description": "This partnership ...",
  "href": "https://host:port/tmf-api/partnershipManagement/v4/partnership/54",
  "id": "54",
  "name": "Dream Partnership",
  "specification": {
    "href": "https://host:port/tmf-api/partnershipManagement/v4/partnershipSpecification/9047",
    "name": "Dream Partnership template"
  },
  "partner": [
    {
      "engagedParty": {
        "@referredType": "Organization",
        "href": "https://host:port/tmf-api/partyManagement/v4/organization/12",
        "name": "SuperdataCompany",
        "role": "ContentProvider"
      },
      "account": [
        {
          "@referredType": "BillingAccount",
          "href": "https://host:port/tmf-api/accountManagement/v4/account/10",
          "name": "Superdata Bahamas Account"
        }
      ]
    },
    {
      "engagedParty": {
        "@referredType": "Organization",
        "href": "https://host:port/tmf-api/partyManagement/v4/organization/13",
        "name": "IslandGovernment",
        "role": "ContentProvider"
      }
    },
    {
      "engagedParty": {
        "@referredType": "Organization",
        "href": "https://host:port/tmf-api/partyManagement/v4/organization/14",

```

```
    "name": "BestCloudCompany",
    "role": "CloudProvider"
  },
  {
    "engagedParty": {
      "@type": "Individual",
      "href": "https://host:port/tmf-api/partyManagement/v4/individual/66",
      "name": "Doctor Hack",
      "role": "Developer"
    }
  },
  {
    "engagedParty": {
      "@type": "Individual",
      "href": "https://host:port/tmf-api/partyManagement/v4/individual/62",
      "name": "Tester"
    }
  }
]
```

Notification Resource Models

6 notifications are defined for this API

Notifications related to PartnershipSpecification:

- PartnershipSpecificationCreateEvent
- PartnershipSpecificationChangeEvent
- PartnershipSpecificationDeleteEvent

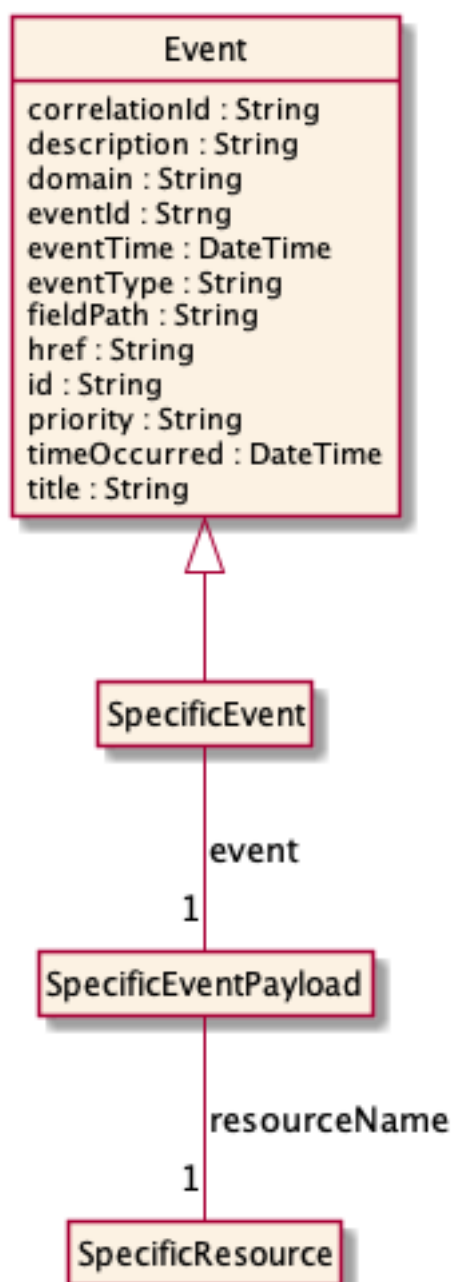
Notifications related to Partnership:

- PartnershipCreateEvent
- PartnershipChangeEvent
- PartnershipDeleteEvent

The notification structure for all notifications in this API follow the pattern depicted by the figure below.

A notification event resource (depicted by "SpecificEvent" placeholder) is a sub class of a generic Event structure containing at least an id of the event occurrence (eventId), an event timestamp (eventTime), and the name of the resource (eventType).

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



PARTNERSHIP SPECIFICATION CREATE EVENT

Notification `PartnershipSpecificationCreateEvent` case for resource `PartnershipSpecification`

Json representation sample

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

```
{
  "eventId": "00001",
  "eventTime": "2015-11-16T16:42:25-04:00",
  "eventType": "PartnershipSpecificationCreateEvent",

```

```
"event": {
  "partnershipSpecification" :
    {-- SEE PartnershipSpecification RESOURCE SAMPLE --}
}
```

PARTNERSHIP SPECIFICATION CHANGE EVENT

Notification PartnershipSpecificationChangeEvent case for resource PartnershipSpecification

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"PartnershipSpecificationChangeEvent",
  "event": {
    "partnershipSpecification" :
      {-- SEE PartnershipSpecification RESOURCE SAMPLE --}
  }
}
```

PARTNERSHIP SPECIFICATION DELETE EVENT

Notification PartnershipSpecificationDeleteEvent case for resource PartnershipSpecification

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"PartnershipSpecificationDeleteEvent",
  "event": {
    "partnershipSpecification" :
      {-- SEE PartnershipSpecification RESOURCE SAMPLE --}
  }
}
```

PARTNERSHIP CREATE EVENT

Notification PartnershipCreateEvent case for resource Partnership

Json representation sample

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

```
{
  "eventId": "00001",
  "eventTime": "2015-11-16T16:42:25-04:00",
  "eventType": "PartnershipCreateEvent",
  "event": {
    "partnership": {
      "-- SEE Partnership RESOURCE SAMPLE --"
    }
  }
}
```

PARTNERSHIP CHANGE EVENT

Notification PartnershipChangeEvent case for resource Partnership

Json representation sample

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

```
{
  "eventId": "00001",
  "eventTime": "2015-11-16T16:42:25-04:00",
  "eventType": "PartnershipChangeEvent",
  "event": {
    "partnership": {
      "-- SEE Partnership RESOURCE SAMPLE --"
    }
  }
}
```

PARTNERSHIP DELETE EVENT

Notification PartnershipDeleteEvent case for resource Partnership

Json representation sample

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

```
{
  "eventId": "00001",
  "eventTime": "2015-11-16T16:42:25-04:00",
  "eventType": "PartnershipDeleteEvent",
  "event": {
    "partnership": {
      "-- SEE Partnership 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.

Notifications are also described in a subsequent section.

OPERATIONS ON PARTNERSHIP SPECIFICATION

LIST PARTNERSHIP SPECIFICATIONS

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

Description

This operation list partnership specification 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 PartnershipSpecification resources.

Request
GET /tmf-api/partnershipManagement/v4/partnershipSpecification Accept: application/json
Response
200 <pre>[{ "@type": "PartnershipSpecification", "description": "This partnership specification gives the template for defining a partnership with a content provider", "href": "https://host:port/tmf-api/partnershipManagement/v4/partnershipSpecification/9047", "id": "9047", "name": "Dream Partnership template", "roleSpecification": [[{ "@type": "PartnershipRoleSpecification", "name": "ContentProvider", "agreementSpecification": [{ "name": "ContentLicenseAgreement", "id": "33" }] }], { "@type": "PartnershipRoleSpecification", "name": "CloudProvider" }, { "@type": "PartnershipRoleSpecification", "name": "Developer", </pre>


```
[
  {
    "agreementSpecification": [
      {
        "name": "ProfitShareAgreement",
        "id": "32"
      }
    ]
  },
  {
    "@type": "PartnershipRoleSpecification",
    "name": "Tester"
  }
]
```

RETRIEVE PARTNERSHIP SPECIFICATION

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

Description

This operation retrieves a partnership specification 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 PartnershipSpecification resource

Request
GET /tmf-api/partnershipManagement/v4/partnershipSpecification/1140 Accept: application/json
Response
200 { "@type": "PartnershipSpecification", "description": "This partnership specification gives the template for defining a partnership with a content provider", "href": "https://host:port/tmf-api/partnershipManagement/v4/partnershipSpecification/9047", "id": "9047", "name": "Dream Partnership template", "roleSpecification": [[]]

```

{
  "@type": "PartnershipRoleSpecification",
  "name": "ContentProvider",
  "agreementSpecification": [
    {
      "name": "ContentLicenseAgreement",
      "id": "33"
    }
  ]
},
{
  "@type": "PartnershipRoleSpecification",
  "name": "CloudProvider"
},
{
  "@type": "PartnershipRoleSpecification",
  "name": "Developer",
  "agreementSpecification": [
    {
      "name": "ProfitShareAgreement",
      "id": "32"
    }
  ]
},
{
  "@type": "PartnershipRoleSpecification",
  "name": "Tester"
}
]
}

```

CREATE PARTNERSHIP SPECIFICATION

POST /partnershipSpecification

Description

This operation creates a partnership specification entity.

Mandatory and Non Mandatory Attributes

The following tables provide the list of mandatory and non mandatory attributes when creating a PartnershipSpecification, 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
name	

Non Mandatory Attributes	Rule
description	
roleSpecification	

Usage Samples

Here's an example of a request for creating a PartnershipSpecification resource.

Request
POST /tmf-api/partnershipManagement/v4/partnershipSpecification Content-Type: application/json <pre>{ "name": "Dream Partnership template" }</pre>
Response
201 <pre>{ "@type": "PartnershipSpecification", "href": "https://host:port/tmf-api/partnershipManagement/v2/partnershipSpecification/9047", "id": "9047", "name": "Dream Partnership template" }</pre>

PATCH PARTNERSHIP SPECIFICATION

PATCH /partnershipSpecification/{id}

Description

This operation allows partial updates of a partnership specification 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
description	
name	
roleSpecification	

Non Patchable Attributes	Rule
href	
id	

Usage Samples

Here's an example of a request for patching a PartnershipSpecification resource.
Changing the name (using json-merge).

Request
PATCH /tmf-api/partnershipManagement/v4/partnershipSpecification/42 Content-Type: application/merge-patch+json { "name": "new name" }
Response
200 { Similar JSON as in GET response with name changed }

DELETE PARTNERSHIP SPECIFICATION

DELETE /partnershipSpecification/{id}

Description

This operation deletes a partnership specification entity.

Usage Samples

This operation deletes a PartnershipSpecification resource.

Request
DELETE /tmf-api/partnershipManagement/v4/partnershipSpecification/42

Response

204

OPERATIONS ON PARTNERSHIP**LIST PARTNERSHIPS****GET** /partnership?fields=...&{filtering}**Description**

This operation list partnership 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 listing Partnership resources.

Request

GET /tmf-api/partnershipManagement/v4/partnership
Accept: application/json

Response

200

```
[
  {
    "@type": "Partnership",
    "description": "This partnership ...",
    "href": "https://host:port/tmf-api/partnershipManagement/v4/partnership/54",
    "id": "54",
    "name": "Dream Partnership",
    "specification": {
      "href": "https://host:port/tmf-api/partnershipManagement/v4/partnershipSpecification/9047",
      "name": "Dream Partnership template"
    },
    "partner": [
      {
        "engagedParty": {
          "@referredType": "Organization",
          "href": "https://host:port/tmf-api/partyManagement/v4/organization/12",
```

```
    "name": "SuperdataCompany",
    "role": "ContentProvider"
  },
  "account": [
    {
      "@referredType": "BillingAccount",
      "href": "https://host:port/tmf-api/accountManagement/v4/account/10",
      "name": "Superdata Bahamas Account"
    }
  ]
},
{
  "engagedParty": {
    "@referredType": "Organization",
    "href": "https://host:port/tmf-api/partyManagement/v4/organization/13",
    "name": "IslandGovernment",
    "role": "ContentProvider"
  }
},
{
  "engagedParty": {
    "@referredType": "Organization",
    "href": "https://host:port/tmf-api/partyManagement/v4/organization/14",
    "name": "BestCloudCompany",
    "role": "CloudProvider"
  }
},
{
  "engagedParty": {
    "@referredType": "Individual",
    "href": "https://host:port/tmf-api/partyManagement/v4/individual/66",
    "name": "Doctor Hack",
    "role": "Developer"
  }
},
{
  "engagedParty": {
    "@referredType": "Individual",
    "href": "https://host:port/tmf-api/partyManagement/v4/individual/62",
    "name": "Tester"
  }
}
]
}
```

RETRIEVE PARTNERSHIP

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

Description

This operation retrieves a partnership 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 Partnership resource

Request
<pre>GET /tmf-api/partnershipManagement/v4/partnership/9047 Accept: application/json</pre>
Response
<pre>200 { "@type": "Partnership", "description": "This partnership ...", "href": "https://host:port/tmf-api/partnershipManagement/v4/partnership/54", "id": "54", "name": "Dream Partnership", "specification": { "href": "https://host:port/tmf-api/partnershipManagement/v4/partnershipSpecification/9047", "name": "Dream Partnership template" }, "partner": [{ "engagedParty": { "@referredType": "Organization", "href": "https://host:port/tmf-api/partyManagement/v4/organization/12", "name": "SuperdataCompany", "role": "ContentProvider" }, "account": [{ "@referredType": "BillingAccount", "href": "https://host:port/tmf-api/accountManagement/v4/account/10", "name": "Superdata Bahamas Account" }] }, { "engagedParty": { "@referredType": "Organization", "href": "https://host:port/tmf-api/partyManagement/v4/organization/13", "name": "IslandGovernment", "role": "ContentProvider" } }] }</pre>

```

    }
  },
  {
    "engagedParty": {
      "@referredType": "Organization",
      "href": "https://host:port/tmf-api/partyManagement/v4/organization/14",
      "name": "BestCloudCompany",
      "role": "CloudProvider"
    }
  },
  {
    "engagedParty": {
      "@referredType": "Individual",
      "href": "https://host:port/tmf-api/partyManagement/v4/individual/66",
      "name": "Doctor Hack",
      "role": "Developer"
    }
  },
  {
    "engagedParty": {
      "@referredType": "Individual",
      "href": "https://host:port/tmf-api/partyManagement/v4/individual/62",
      "name": "Tester"
    }
  }
]
}

```

CREATE PARTNERSHIP

POST /partnership

Description

This operation creates a partnership entity.

Mandatory and Non Mandatory Attributes

The following tables provide the list of mandatory and non mandatory attributes when creating a Partnership, 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
name	
specification	

Non Mandatory Attributes	Rule
description	
partner	

Usage Samples

Here's an example of a request for creating a Partnership resource. The id is forced in the request.

Request
POST /tmf-api/partnershipManagement/v4/partnership Content-Type: application/json <pre>{ "name": "Dream Partnership" }</pre>
Response
201 <pre>{ "@type": "Partnership", "href": "https://host:port/tmf-api/partnershipManagement/v2/partnership/54", "id": "54", "name": "Dream Partnership" }</pre>

PATCH PARTNERSHIP

PATCH /partnership/{id}

Description

This operation allows partial updates of a partnership 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
description	
name	
partner	
specification	

Non Patchable Attributes	Rule
href	
id	

Usage Samples

Here's an example of a request for patching a Partnership resource.
Changing requires billing to true (using json-merge).

Request
PATCH /tmf-api/partnershipManagement/v4/partnership/9047 Content-Type: application/merge-patch+json { "name": "new name" }
Response
200 { Similar JSON as in GET response with name changed }

DELETE PARTNERSHIP

DELETE /partnership/{id}

Description

This operation deletes a partnership entity.

Usage Samples

This operation deletes a Partnership resource.

Request
DELETE /tmf-api/partnershipManagement/v4/partnership/9047
Response

204

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 reproduced below.

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
<pre>POST /client/listener Accept: application/json { "event": { EVENT BODY }, "eventType": "EVENT_TYPE" }</pre>
Response
201

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

ACKNOWLEDGEMENTS

RELEASE HISTORY

Release Number	Date	Release led by:	Description
Release 1.0	15/15/2016	Pierre Gauthier TM Forum pgauthier@tmforum.org Mariano Belaunde Orange Labs	First Release of the Document.
Release 2.0	11/06/2018	Mariano Belaunde Orange Labs	Alignment with Guidelines 3.0
Release 4.0	19/07/2019	Mariano Belaunde Orange Labs	Refactoring and Regeneration after schematization

CONTRIBUTORS TO DOCUMENT

Mariano Belaunde	Orange
Jonathan Goldberg	Amdocs
Jacob Avraham	Amdocs
Pierre Gauthier	TM Forum