



TM Forum Specification

Partnership Type Management API REST Specification

TMF668
Release 17.0.1
November 2017

Latest Update: TM Forum Release 17	TM Forum Approved
Version 1.0.1	IPR Mode: RAND

NOTICE

Copyright © TM Forum 2017. 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.

TM FORUM invites any TM FORUM Member or any other party that believes it has patent claims that would necessarily be infringed by implementations of this TM Forum Standards Final Deliverable, to notify the TM FORUM Team Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this deliverable.

The TM FORUM invites any party to contact the TM FORUM Team Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this TM FORUM Standards Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this TM FORUM Standards Final Deliverable. TM FORUM may include such claims on its website, but disclaims any obligation to do so.

TM FORUM takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this TM FORUM Standards Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on TM FORUM's procedures with respect to rights in any document or deliverable produced by a TM FORUM Collaboration Project Team can be found on the TM FORUM website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this TM FORUM Standards Final Deliverable, can be obtained from the TM FORUM Team Administrator. TM FORUM makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

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

TABLE OF CONTENTS

NOTICE	2
Table of Contents.....	4
List of Tables	5
Introduction	6
SAMPLE USE CASES	7
RESOURCE MODEL	19
Managed Entity and Task Resource Models	19
Partnership Type resource	19
Notification Resource Models	21
Partnership Type Creation Notification	22
Partnership Type Remove Notification.....	22
API OPERATIONS.....	24
Operations on Partnership Type.....	24
List partnership types	24
Retrieve partnership type.....	26
Create partnership type	27
Patch partnership type.....	28
Delete partnership type.....	30
API NOTIFICATIONS.....	31
Register listener	31
Unregister listener	31
Publish Event to listener	32
Acknowledgments.....	34
Release History	34
Contributors to Document.....	34

LIST OF TABLES

N/A

INTRODUCTION

The Partnership Type API provides standardized mechanisms for creating partnership types. It is one of the APIs involved in an onboarding process.

The following resources are managed by this API:

- **PartnershipType:** Identifies a type of a partnership between parties, including the list of roles types that are permitted (i.e Buyer, Seller, Developer). Role types may refer to agreement specifications to be signed by parties playing the role.

The API allows the retrieval, creation, update and deletion of partnership type and its owned sub-resources.

API Dependencies

This API has strong dependencies with the following management APIs:

- Party Management API: used to query, create, update or delete information on *individuals* or *organizations* that will be onboarded.
- Agreement Management API: used to query, create, update or delete agreements and agreement specifications. These agreements need to be created and updated when signed by the involved parties.
- Account Management API: used to retrieve, create, update or delete different kinds of accounts that may be needed in the context of the onboarding process, such as billing or settlement accounts and financial accounts.
- Party Role Management API: used to retrieve the party roles defined for a partnership type. A party role in this context represents the fact that a given *party* will play a given identified role type in the partnership being defined.

Other indirect API dependencies when using this API are:

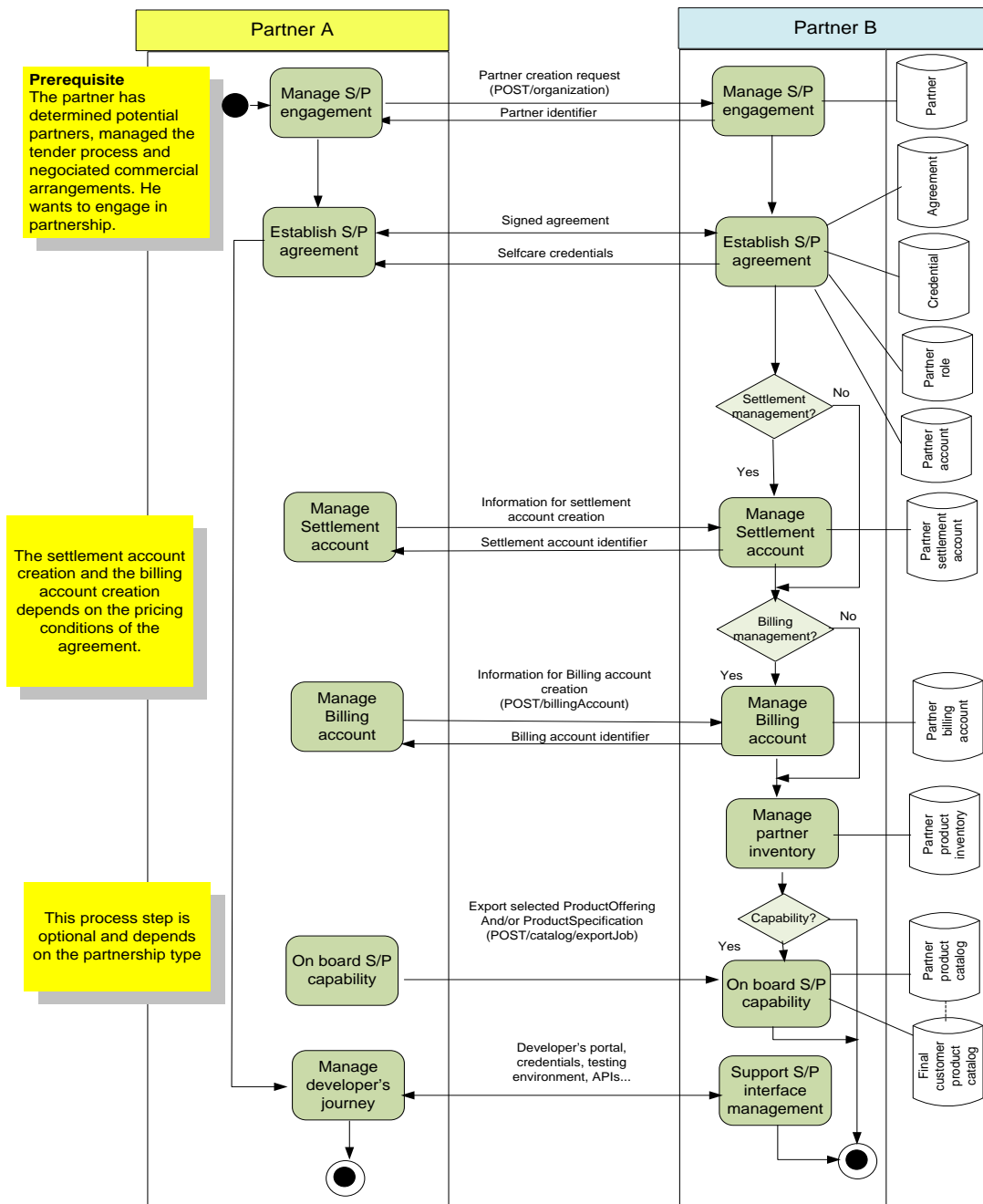
- Product Catalog Management API: used to connect agreements to product offerings
- Product Inventory Management API: retrieval of products related to product offerings.
- Product Ordering management API: establishing order on available products.

SAMPLE USE CASES

In this section we provide some typical API usage scenarios concerning partnerships described in a light-weight fashion. The intent is not to describe all possible contexts of use of the API. Much more details on these use cases can be found in “Open Digital Business Scenarios and Use Cases” document.

A Global view

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



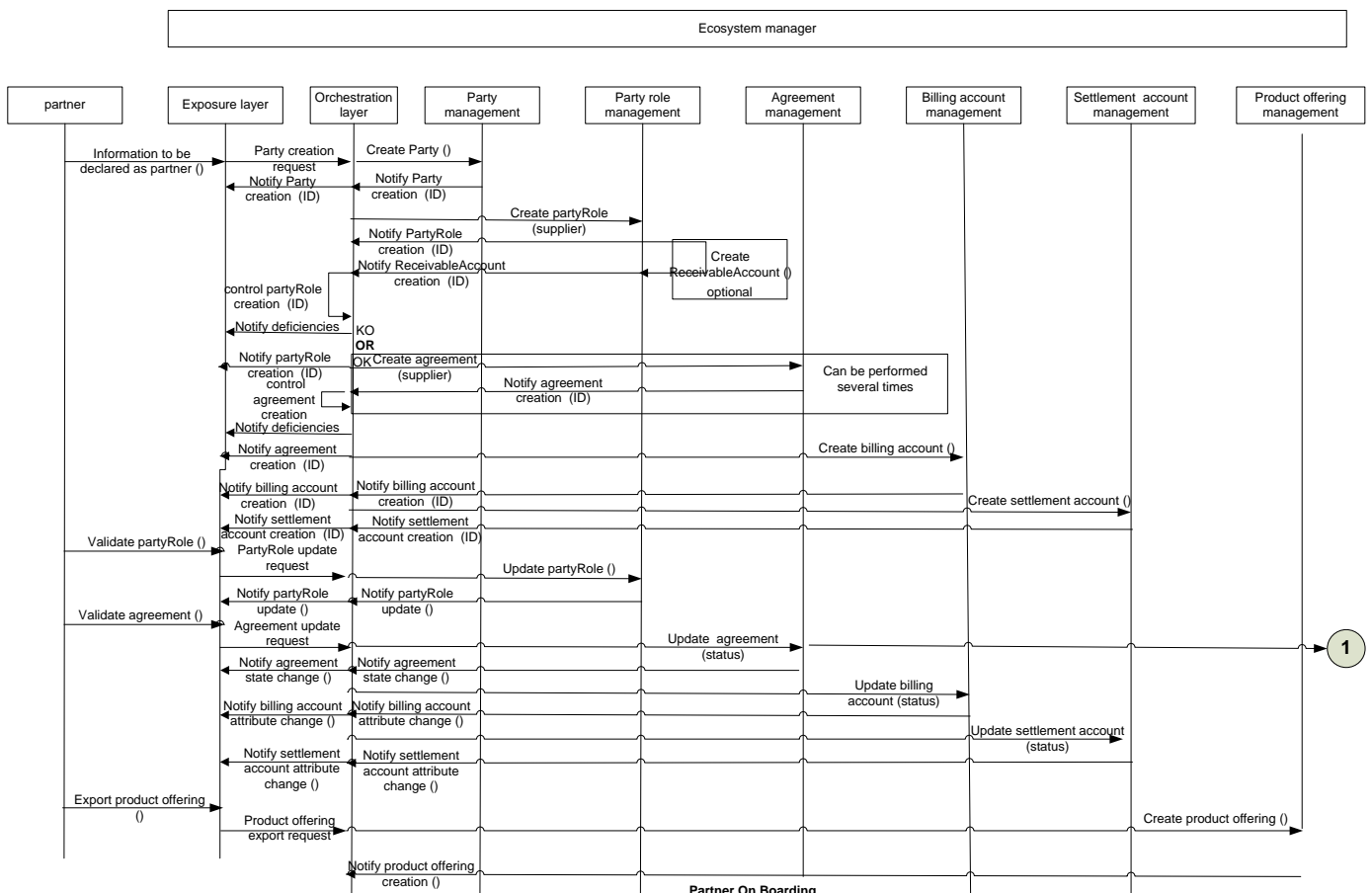
Usage options for the API

In the following we provide a list of sequence diagrams illustrating different usages of the API. A partner starts the interaction by accessing an “exposure layer”, which in turn communicates with an “orchestration layer” which make calls to the On Boarding API.

Note: In these diagrams we assume that the “orchestration layer” performs direct calls to the APIs and implements as well the “hub” client listener interfaces to receive the notifications from the API. In contrast the way how the “orchestration layer” communicates with the “exposure layer” could be done in any way and does not assume for the “exposure layer” to support the API notification listener interfaces.

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

(Sequence diagram corresponding to on boarding context details document description)



Option 1 : all resources are created without interaction with prospective partner (sequence diagram corresponding to on boarding context details document description)

This sequence diagram describes a partner on boarding process where:

- Prospective partner is not known by ecosystem manager system
- All resources (party, partyRole, agreement, billing account and / or settlement account) are created without interaction with prospective partner
- Receivable account is created optionally
- Business logic:

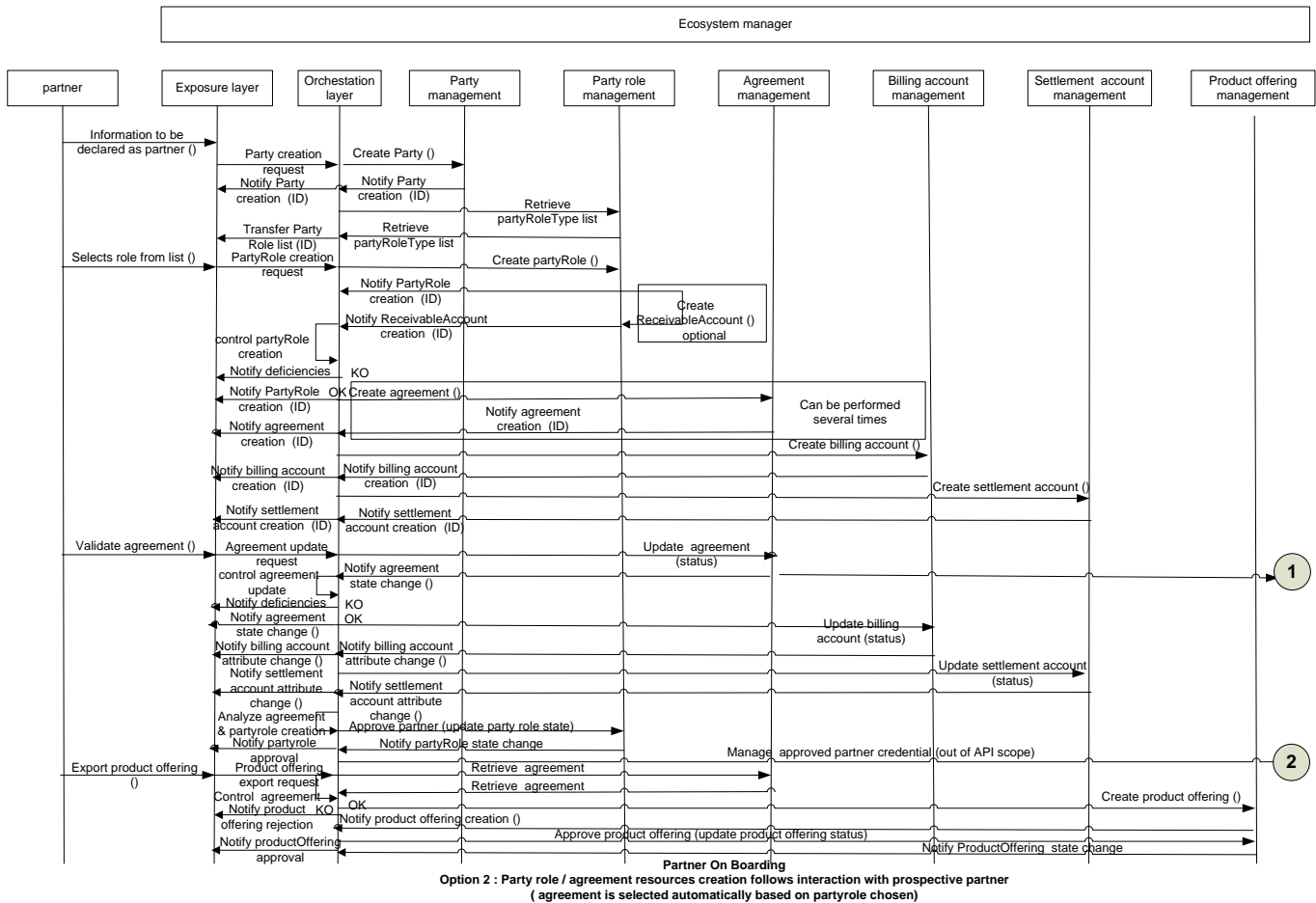
- Controls party role creation and notifies exposure layer of partyRole creation or deficiencies if there are errors or inconsistencies
- Controls agreement creation and notifies exposure layer of agreement creation or deficiencies if there is errors or inconsistencies
- Agreement creation can be performed several times
- Prospective partner validate partyRole and, then, partyRole is updated
- Prospective partner validate agreement and, then, agreement, billing account and / or settlement account are updated
- Business logic manage partner credentials (out of API scope)
- Partner creates a product offering if allowed

Note:

- For the sake of readability, following actions are not presented on the sequence diagram
 - Subsequent actions in case of deficiencies when controlling partyRole or Agreement creation,
 - Partner approval when it depends on partyRole and agreement creation
 - Verification if partner productOffering creation request is covered by an agreement
 - Product offering approval or rejection by ecosystem manager

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

(Agreement is selected automatically based on partyrole 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 partyRoleType is retrieved
- Prospective partner selects a partyRoleType
- PartyRole is created and receivable account can be optionally created
- Business logic controls party role creation and notifies exposure layer of partyRole creation or deficiencies if there is errors or inconsistencies
- Agreement, billing account and / or settlement account resources are created
- Agreement creation can be performed several times
- Prospective partner validate agreement and, then, agreement, billing account and / or settlement account are updated
- Business logic controls agreement creation and notifies exposure layer of agreement creation or deficiencies if there is errors or inconsistencies
- Partner is approved if its approval depends on partyRole and agreement creation
- Business logic manage partner credentials (out of API scope)

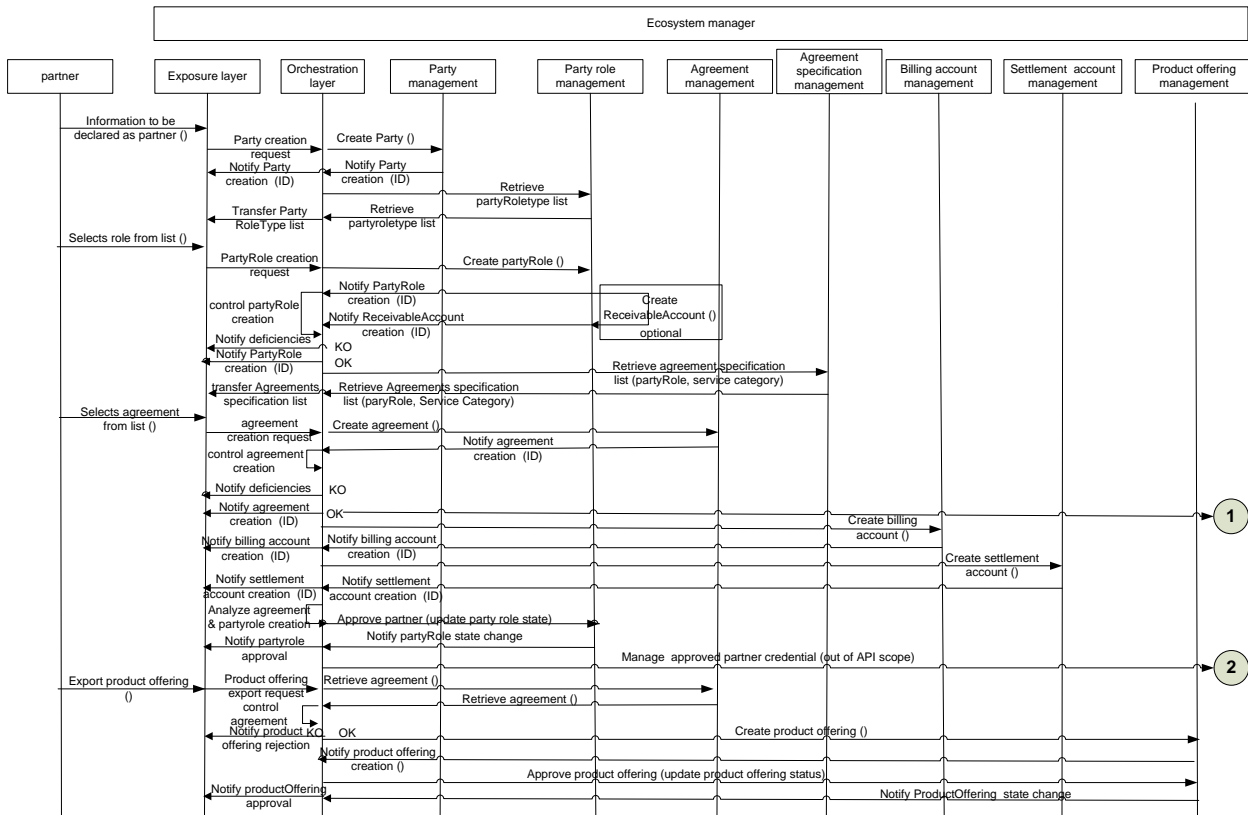
- Business logic checks if partner productOffering creation request is covered by an agreement
- Partner creates a product offering if allowed
- Ecosystem manager approves or rejects product offering creation

Note:

- For the sake of readability, Partner subsequent action in case of deficiencies when controlling partyRole or Agreement creation are not presented on the sequence diagram

Option 3: party role / agreement resources creation follows interaction with prospective partner

(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 partyRoleType is retrieved
- Prospective partner selects a partyRoleType
- PartyRole is created and receivable account can be optionally created
- Business logic controls party role creation and notifies exposure layer of partyRole creation or deficiencies if there is errors or inconsistencies
- List of agreement specifications is retrieved based on partyRoleType and/or service category
- Prospective partner selects an agreement specification
- Agreement is created
- Business logic controls agreement creation and notifies exposure layer of agreement creation or deficiencies if there is errors or inconsistencies
- Agreement creation can be performed several times
- Billing account and / or settlement account resources are created
- Partner is approved if its approval depends on partyRole and agreement creation
- Business logic manage partner credentials (out of API scope)

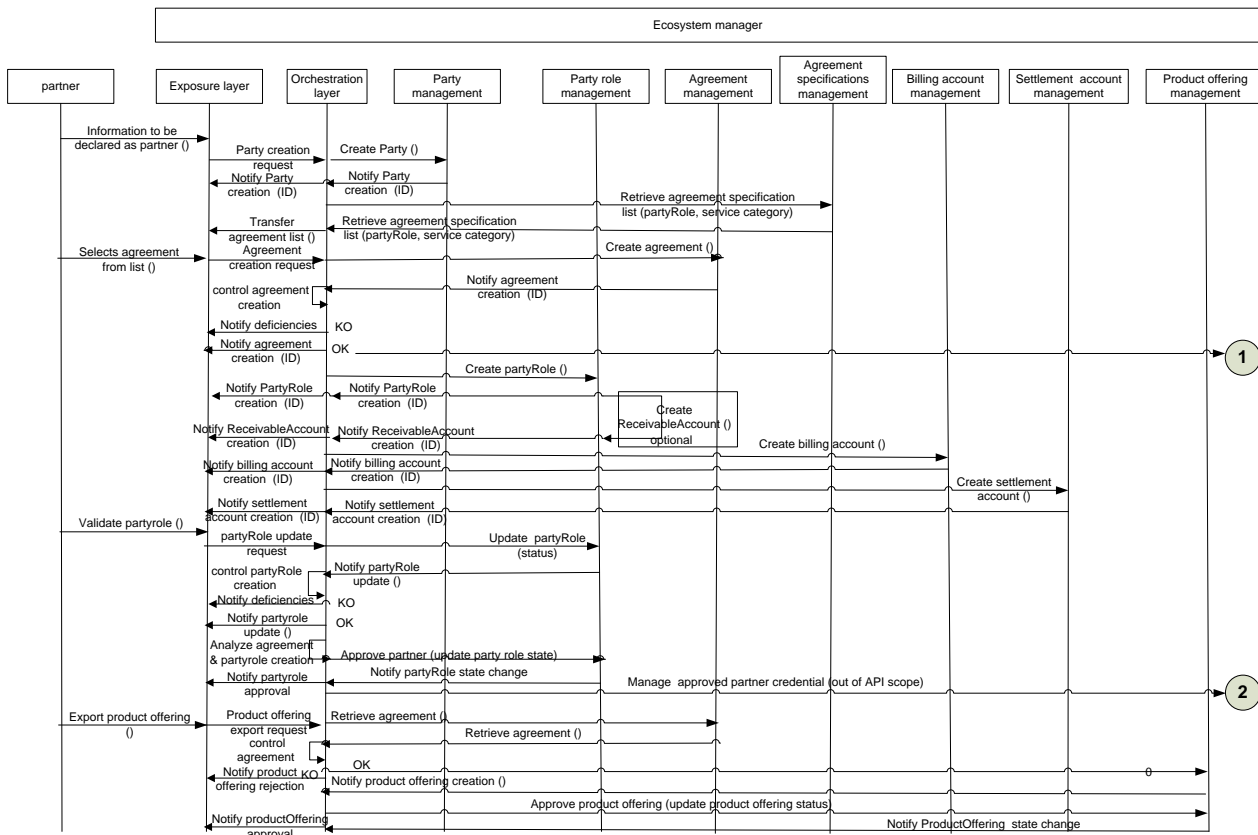
- Business logic checks if partner productOffering creation request is covered by an agreement
- Partner creates a product offering if allowed
- Ecosystem manager approves or rejects product offering creation

Note:

- For the sake of readability, Partner subsequent action in case of deficiencies when controlling partyRole or Agreement creation are not presented on the sequence diagram

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

(Party role is selected automatically based upon agreement chosen)



Partner On Boarding
Option 4 : agreement / Party role resources creation follows interaction with prospective partner
(partyrole 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 based on partyRoleType and/or service category
- Prospective partner selects an agreement specification
- Agreement is created
- Agreement creation can be performed several times
- Business logic controls agreement creation and notifies exposure layer of agreement creation or deficiencies if there is errors or inconsistencies
- Billing account and / or settlement account resources are created

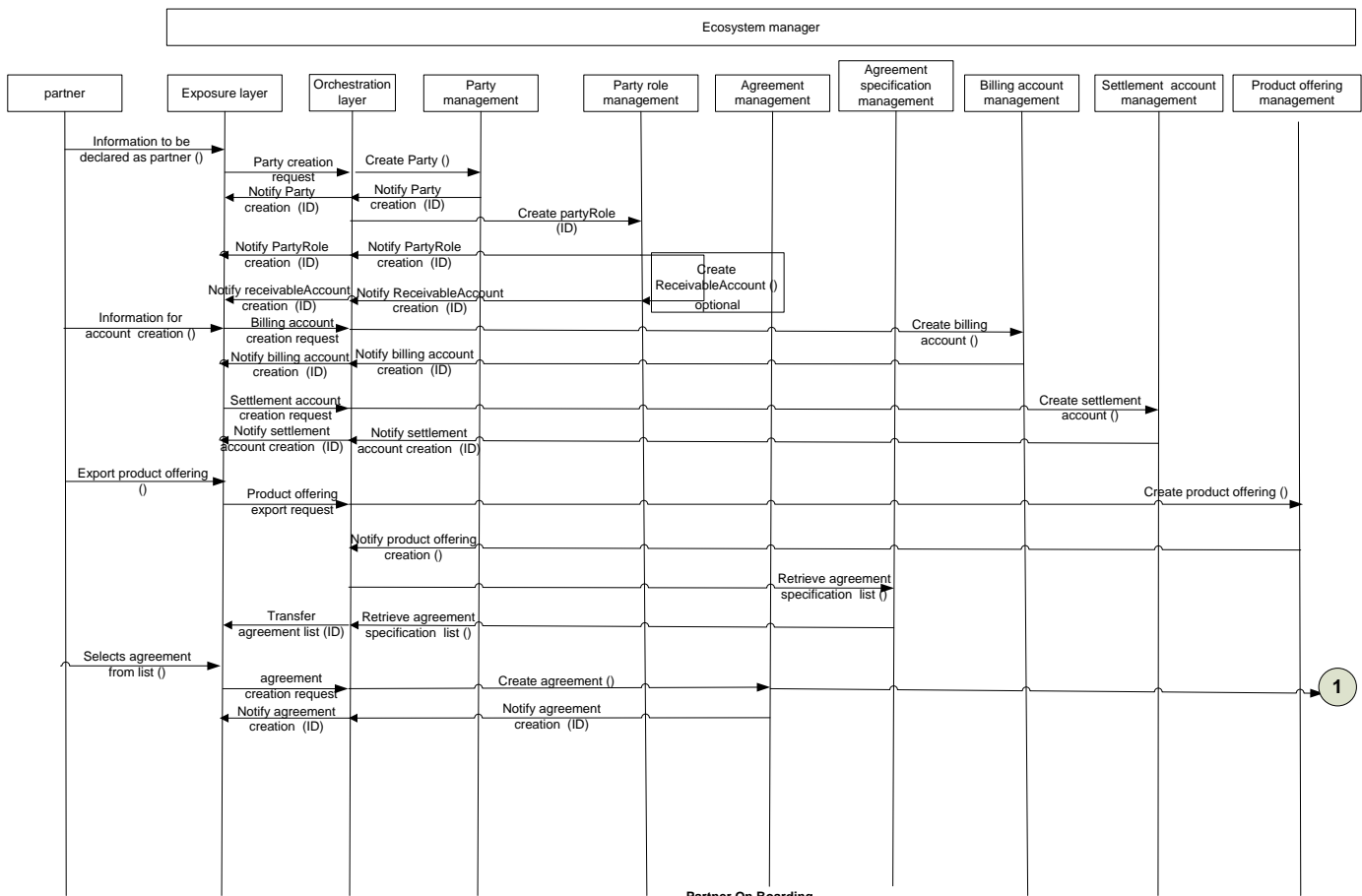
- PartyRole is created depending on agreement specification selected and receivableAccount is optionally created
- prospective partner validates partyRole and PartyRole is updated
- Business logic controls party role creation and notifies exposure layer of partyRole creation or deficiencies if there is errors or inconsistencies
- Partner is approved if its approval depends on partyRole and agreement creation
- Business logic manage partner credentials (out of API scope)
- Business logic checks if partner productOffering creation request is covered by an agreement
- Partner creates a product offering if allowed
- Ecosystem manager approves or rejects product offering creation

Note:

- For the sake of readability, Partner subsequent action in case of deficiencies when controlling partyRole or Agreement creation are not presented on the sequence diagram

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

(Then service information is provided and agreement is created)



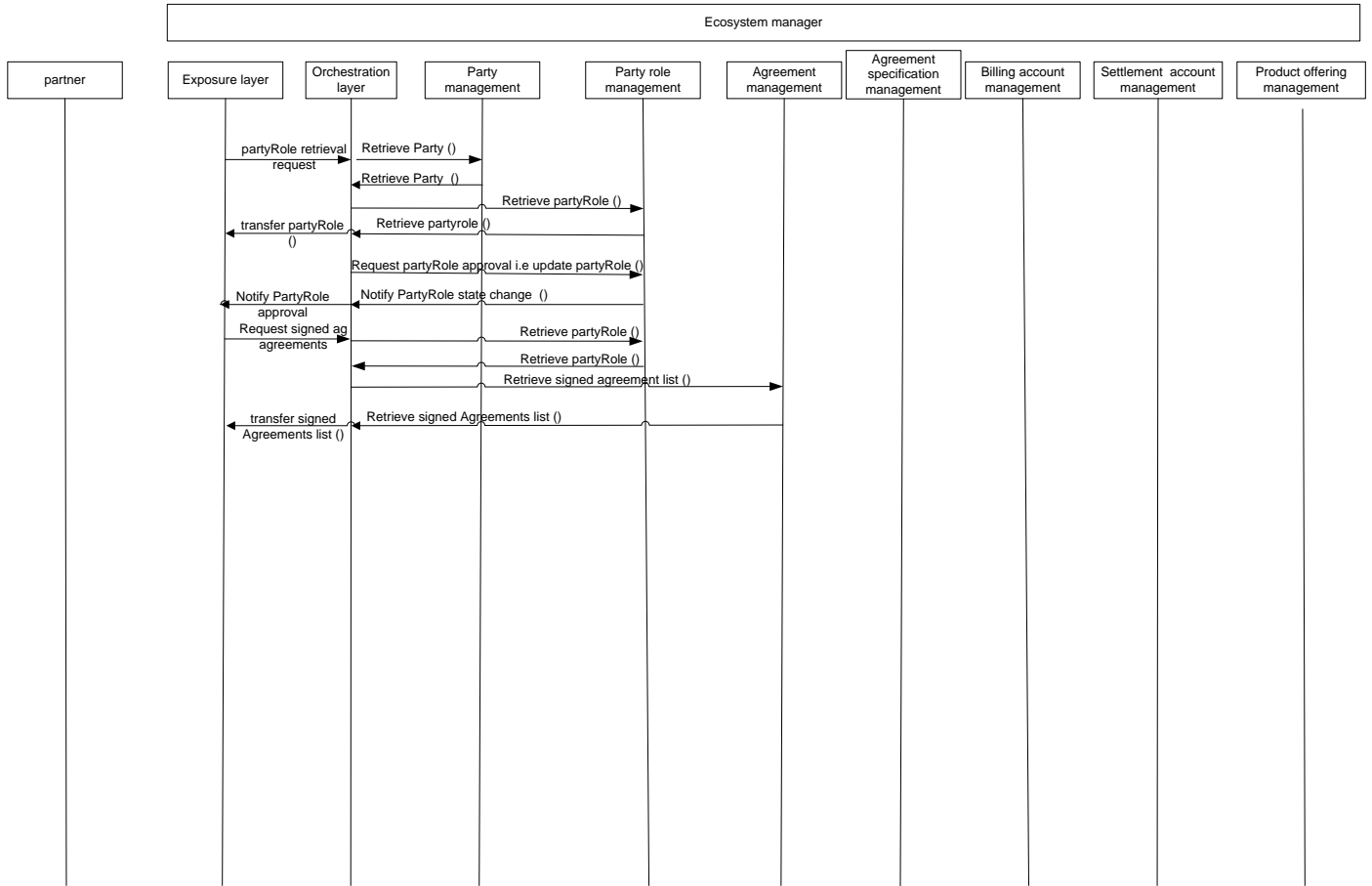
Partner On Boarding
Option 5 : Ehealth partner on boarding use case: billing and settlement accounts are created after party.
(Then service information is provided and agreement is created)

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, optionally, ReceivableAccount
- 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 partyRole, partyRole needs to be approved, Agreement signed by partyRole needs to be retrieved

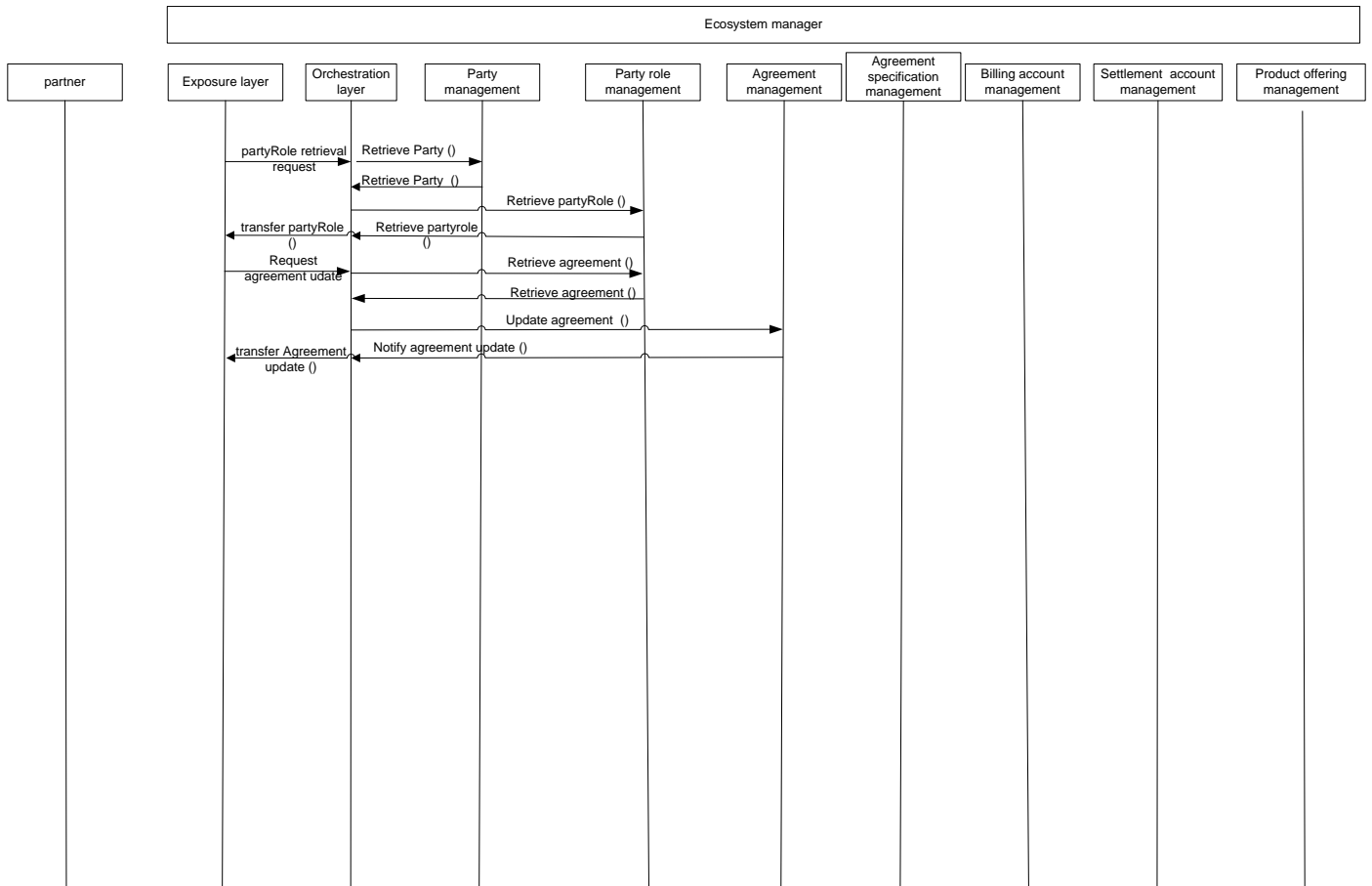


Partner On Boarding
 Option 6: party has already selected a partyRole, partyRole needs to be approved, Agreement signed by partyRole needs to be retrieved

This sequence diagram describes a partner on boarding process where:

- prospective partner already exists and needs to be approved
- If partyRole identifier is not known
 - o Party is retrieved (relevance to be confirmed)
 - o partyRole is retrieved
- If partyrole identifier is known
 - o partyRole is retrieved
- Ecosystem manager request partyRole approval
 - o partyRole status is updated
 - o partyRole state change is notified
- list of agreements signed by partner (partyRole) are retrieved

Option 7: partyRole exist, Agreement is updated

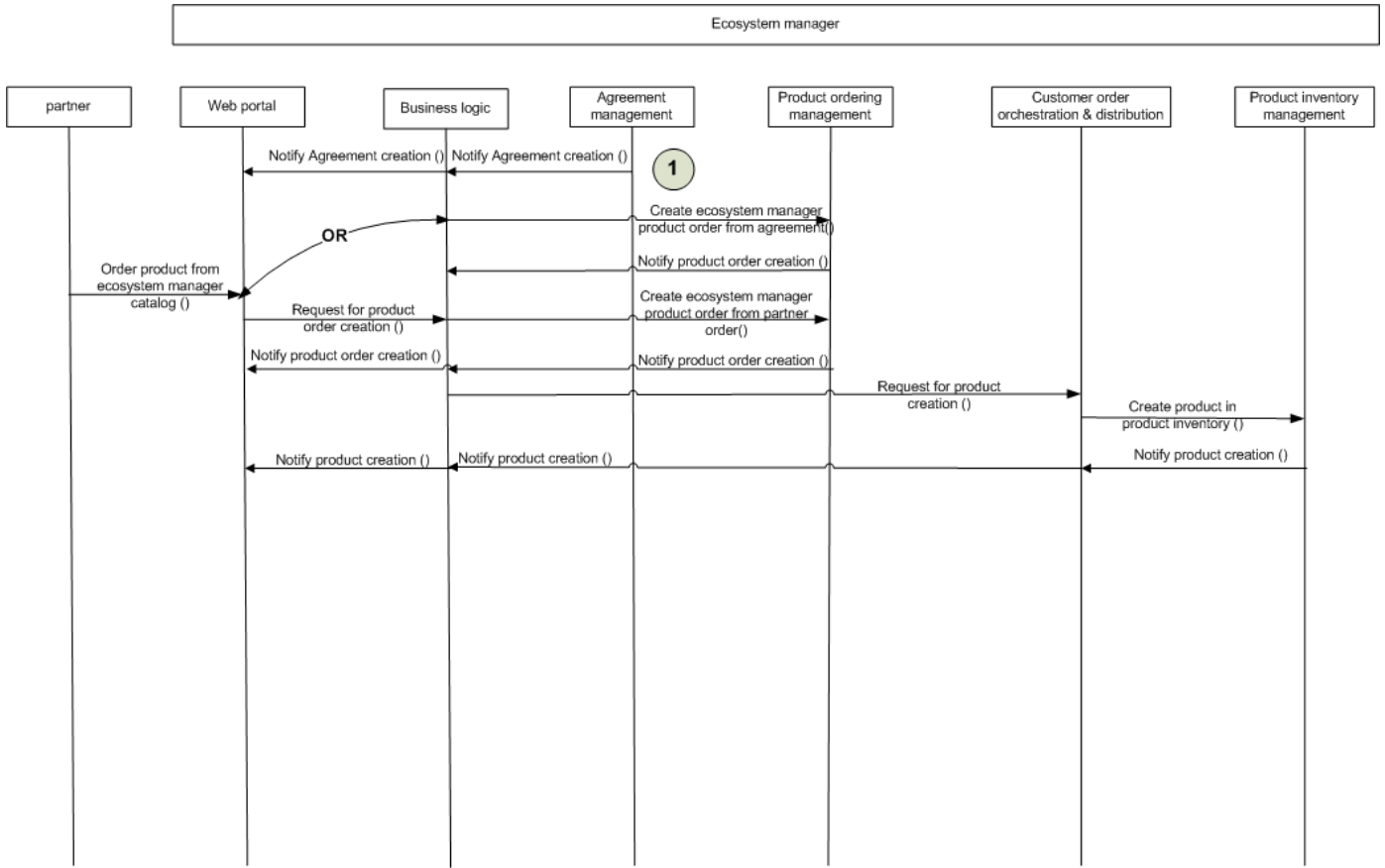


Partner On Boarding
Option 7: partyRole exist, Agreement is updated

This sequence diagram describes a partner on boarding process where:

- prospective partner already exists and agreement needs to be updated
- If partyRole identifier is not known
 - o Party is retrieved (relevance to be confirmed)
 - o partyRole is retrieved
- If partyrole identifier is known
 - o partyRole is retrieved
- list of agreements related to party role are retrieved
- agreement is updated.

Ordering of an ecosystem manager product by partner



Partner On Boarding
Ordering of an ecosystem manager product by partner

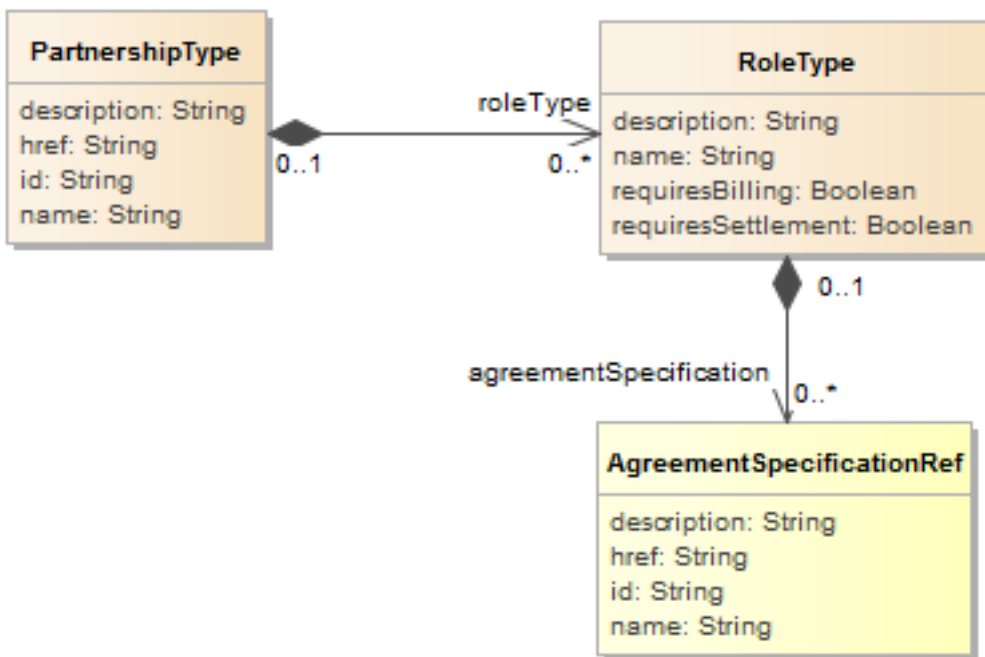
RESOURCE MODEL

Managed Entity and Task Resource Models

PARTNERSHIP TYPE RESOURCE

A partnership type 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

PartnershipType fields

description	A string. An explanatory text regarding this partnership type.
href	A string. The reference url for this partnership type.
id	A string. The identifier of the partnership type.
name	A string. An identifying name for the partnership type.
roleType	A list of role types (RoleType [*]). A RoleType represents the type of a PartyRole, defined in the context of a given type of partnership, such as Buyer, Seller.

RoleType sub-resource

A RoleType represents the type of a PartyRole, defined in the context of a given type of partnership, such as Buyer, Seller.

description	A string. An explanatory text documenting the role type.
name	A string. The name of the role type.
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.

description	A string. A narrative that explains in detail what the agreement specification is about.
href	A string. Reference URL of the agreement specification.
id	A string. Unique identifier of the agreement specification.
name	A string. Name of the agreement specification.

Json representation sample

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

```
{
  "description": "This partnership type ...",
  "href": "https://host:port/onboardingManagement/partnershipType/9364",
  "id": "9364",
  "name": "Dream Partnership",
  "roleType": [
    [
      {
        "name": "ContentProvider",
        "agreementSpecification": [
          {
            "name": "ContentLicenseAgreement",
            "id": "33"
          }
        ]
      }
    ],
    {
      "name": "CloudProvider"
    }
  ]
}
```

```
{
  "name": "Developer",
  "agreementSpecification": [
    {
      "name": "ProfitShareAgreement",
      "id": "32"
    }
  ]
},
{
  "name": "Tester"
}
]
```

Notification Resource Models

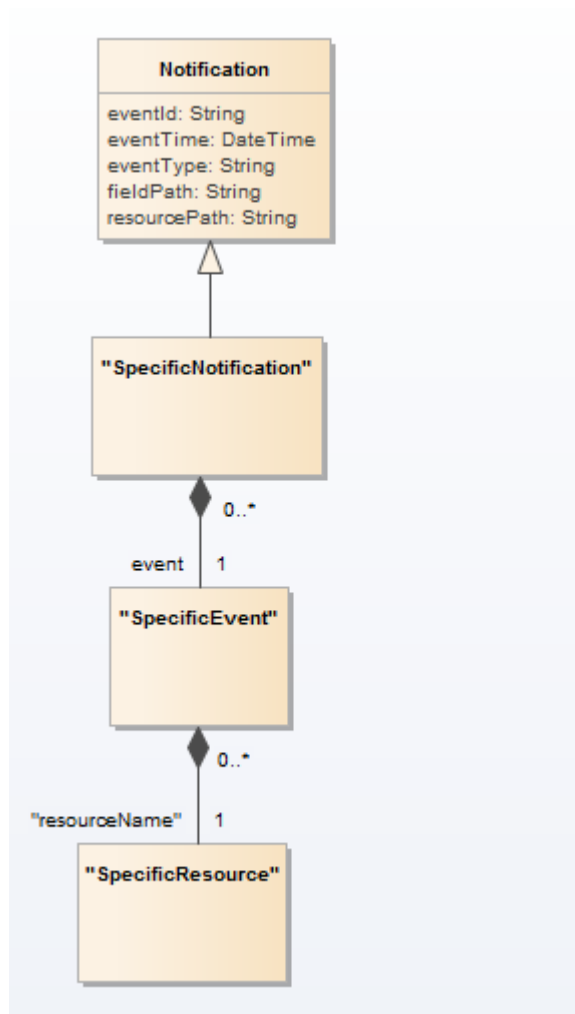
2 notifications are defined for this API

Notifications related to PartnershipType:

- PartnershipTypeCreationNotification
- PartnershipTypeRemoveNotification

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).



PARTNERSHIP TYPE CREATION NOTIFICATION

Notification sent when a new PartnershipType resource is created.

Json representation sample

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

```

{
  "eventId": "00001",
  "eventTime": "2015-11-16T16:42:25-04:00",
  "eventType": "PartnershipTypeCreationNotification",
  "event": {
    "partnershipType":
      {-- SEE PartnershipType RESOURCE SAMPLE --}
  }
}

```

PARTNERSHIP TYPE REMOVE NOTIFICATION

Notification sent when removing a PartnershipType resource.

Json representation sample

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

```
{
  "eventId":"00001",
  "eventTime":"2015-11-16T16:42:25-04:00",
  "eventType":"PartnershipTypeRemoveNotification",
  "event": {
    "partnershipType" :
      {-- SEE PartnershipType 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 TYPE

LIST PARTNERSHIP TYPES

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

Description

This operation list partnership type 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 PartnershipType resources.

Request
GET /onboardingManagement/partnershipType Accept: application/json
Response
<pre> 200 [{ "description": "This partnership type ...", "href": "https://host:port/onboardingManagement/partnershipType/9364", "id": "9364", "name": "Dream Partnership", "roleType": [[{ "name": "ContentProvider", "agreementSpecification": [{ "name": "ContentLicenseAgreement", "id": "33" }] }], { "name": "CloudProvider" }, { "name": "Developer", "agreementSpecification": [{ "name": "ProfitShareAgreement", "id": "32" }] }], { "name": "Tester" } } </pre>

```

    }
  ]
}
]

```

RETRIEVE PARTNERSHIP TYPE

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

Description

This operation retrieves a partnership type 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 PartnershipType resource.

Request
GET /onboardingManagement/partnershipType/9364 Accept: application/json
Response
200 <pre> { "description": "This partnership type ...", "href": "https://host:port/onboardingManagement/partnershipType/9364", "id": "9364", "name": "Dream Partnership", "roleType": [[{ "name": "ContentProvider", "agreementSpecification": [{ "name": "ContentLicenseAgreement", "id": "33" }] }]], }, { </pre>

```

    "name": "CloudProvider"
  },
  {
    "name": "Developer",
    "agreementSpecification": [
      {
        "name": "ProfitShareAgreement",
        "id": "32"
      }
    ]
  },
  {
    "name": "Tester"
  }
]
}

```

CREATE PARTNERSHIP TYPE

POST /partnershipType

Note: this operation is available only to ADMIN API users

Description

This operation creates a partnership type entity.

Mandatory and Non Mandatory Attributes

The following tables provides the list of mandatory and non mandatory attributes when creating a PartnershipType, 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	Default Value	Rule
description		
roleType		

Additional Rules

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

Context	Mandatory Sub-Attributes
roleType	name

Usage Samples

Here's an example of a request for creating a PartnershipType resource. In this example the request only passes mandatory attributes.

Request
POST /onboardingManagement/partnershipType Content-Type: application/json <pre>{ "name": "Dream Partnership" }</pre>
Response
201 <pre>{ "href": "https://host:port/onboardingManagement/partnershipType/9364", "id": "9364", "name": "Dream Partnership" }</pre>

PATCH PARTNERSHIP TYPE

PATCH /partnershipType/{id}

Note: this operation is available only to ADMIN API users

Description

This operation allows partial updates of a partnership type 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
name	
description	

roleType	
----------	--

Non Patchable Attributes	Rule
href	
id	

Usage Samples

Here's an example of a request for patching a PartnershipType resource.

Request
PATCH /onboardingManagement/partnershipType/9364 Content-Type: application/merge-patch+json <pre>{ "name": "new name" }</pre>
Response
201 <pre>{ "description": "This partnership type ...", "href": "https://host:port/onboardingManagement/partnershipType/9364", "id": "9364", "name": "new name", "roleType": [[{ "name": "ContentProvider", "agreementSpecification": [{ "name": "ContentLicenseAgreement", "id": "33" }] }]], { "name": "CloudProvider" }, { "name": "Developer", "agreementSpecification": [{ "name": "ProfitShareAgreement", "id": "32" }] }]</pre>

```
    },  
    {  
      "name": "Tester"  
    }  
  ]  
}
```

DELETE PARTNERSHIP TYPE

DELETE /partnershipType/{id}

Note: this operation is available only to ADMIN API users

Description

This operation deletes a partnership type entity.

Usage Samples

Here's an example of a request for deleting a PartnershipType resource.

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

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.

ACKNOWLEDGMENTS

RELEASE HISTORY

Release Number	Date	Release led by:	Description
Release 1.0	15/04/2016	Pierre Gauthier TM Forum pgauthier@tmforum.org Mariano Belaunde Orange mariano.belaunde@orange.com	First Release of the Document. Generated from the API Data Model.
Release 17.0.1 Version 1.0.1	21/11/2017	Adrienne Walcott	Updated to reflect TM Forum Approved Status

CONTRIBUTORS TO DOCUMENT

Veronique Mauneau	Orange
Jean-Luc Tymen	Orange
Mariano Belaunde	Orange
Elaine Hafer	Ericsson
August-Wilhelm Jagau	Ericsson
Liuyiling (Sammy)	Huawei
Sunruinan	Huawei
Jiang Yisong	Huawei
George Glass	BT
Pierre Gauthier	TM Forum
Andreas Polz	Infonova

Takayuki Nakamura	NTT
-------------------	-----