translationBridgeManagement

Service Description

Abstract
This document provides service description for the **translationBridgeManagement** service.

Contents

1 Overview

This document describes the **translationBridgeManagement** service, which enables to find providers whose service operation is accessible by a specified consumer via a translation bridge using the currently available translators. The service also allows to build such bridges, to query or to abort existing translation bridges. An example of this interaction is when a service orchestration system can't find a suitable provider for a specified consumer with a specified service operation and try to determine whether it is possible to use one of the available providers utilizing various translators.

To enable other systems to use, to consume it, this service needs to be offered through the ServiceRegistry.

The **translationBridgeManagement** service contains the following operations:

- *discovery* checks a list of possible providers for a service operation and returns only those that can be consumed by the specified consumer via a translation bridge using the currently available translators;
- *negotiation* allows systems to create a translation bridge for a specified consumer to one selected provider's one particular service operation;
- *abort* aborts existing bridges in bulk;
- *query* lists the translation bridges that match the filtering requirements.

The rest of this document is organized as follows. In Section 2, we describe the abstract message operations provided by the service. In Section 3, we end the document by presenting the data types used by the mentioned operations.

1.1 How This Service Is Meant to Be Used

This service should be used primarily by service orchestration systems.

During service orchestration, if no candidates are found that matches all the requirements and the requester allows translation, an orchestration system should call the *discovery* operation with ”almost good” service instance candidates. ”Almost good” means that these candidates provides the required service operation but the direct communication with the specified consumer is not possible because of missing common interfaces and/or different data models. Operation *discovery* returns the identifiers those candidates that can be accessed by the consumer via a translation bridge alongside with an identifier.

Service orchestration systems then should call the *negotiation* operation with the above-mentioned identifier and one of the candidates and the operation builds the translation bridge (based on the data collected during the *discovery*) and return its access information.

After a successful *negotiation*, the service orchestration should return the translation bridge as result to its consumer.

Support systems can use *query* operation to gather information about translation bridges in the Local Cloud. *abort* operation can be used to dismantle existing translation bridges.

1.2 Important Delimitations

The requester has to identify itself to use any of the operations.

1.3 Access policy

The service is only available for operators, dedicated Core/Support systems and those who have the proper authorization rights to consume it.

2 Service Operations

This section describes the abstract signatures of each operations of the service. In particular, each subsection names an operation, an input type and one or two output types (unsuccessful operations can return different structure), in that order. The input type is named inside parentheses, while the output type is preceded by a colon. If the operation has two output types, they are separated by a slash. Input and output types are only denoted when accepted or returned, respectively, by the operation in question. All abstract data types named in this section are defined in Section 3.

2.1 operation discovery (TranslationDiscoveryMgmtRequest) : TranslationDiscoveryResponse / ErrorResponse

Operation *discovery* checks a list of possible providers for a service operation and returns only those that can be consumed by the specified consumer via a translation bridge using the currently available translators. The request data must meet the following criteria:

- The consumer's name must follow the PascalCase naming convention, can contain maximum of 63 characters of letters (English alphabet) and numbers, and have to start with a letter.
- The target operation is mandatory and must follow the kebab-case naming convention. Operation can contain maximum 63 characters of letters (English alphabet) and numbers, and have to start with a letter.
- Interface template name list is mandatory and can not be empty. Every interface template name must follow the snake_case naming convention, can contain maximum 63 characters of lower-case letters (English alphabet) and numbers, and have to start with a letter.
- Candidates list is mandatory can not be empty. Every candidate has to specify its service instance id, provider's name, service definition name and information about its interfaces:
  - Service instance id must be a valid ServiceInstanceID.
  - Provider's name must follow the PascalCase naming convention, can contain maximum of 63 characters of letters (English alphabet) and numbers, and have to start with a letter.
  - Service definition must follow the camelCase naming convention can contain maximum of 63 characters of letters (English alphabet) and numbers, and have to start with a letter.
  - Interface list can not be empty and its every element must meet the following criteria:
    - Template name is mandatory and must follow the snake_case naming convention, can contain maximum 63 characters of lower-case letters (English alphabet) and numbers, and have to start with a letter.
    - Policy is mandatory and must be a valid SecurityPolicy.
    - Properties must contain a structure that is a valid DataModelMap which contains an entry for the specified operation. This is not necessary for operations without any input or output payload.
    - Properties must contain every interface-dependent information that is necessary for access (for example, address, port, path, topic, etc.).
- Flags are optional, but if they are specified, each flag name must be a valid TranslationDiscoveryFlag value.

2.2 operation negotiation (TranslationNegotiationMgmtRequest) : TranslationNegotiationResponse / ErrorResponse

Operation *negotiation* allows to create a translation bridge for a specified consumer to one selected provider's one particular service operation. The request data must meet the following criteria:

- Bridge identifier is mandatory and must be a valid TranslationBridgeID.
- Target instance identifier is mandatory and must be a valid ServiceInstanceID.

2.3 operation abort (TranslationAbortMgmtRequest) : TranslationAbortMgmtResponse / ErrorResponse

Operation *abort* aborts a list of existing bridges. The abort data must meet the following criteria:

- At least one bridge identifier is mandatory.
- All specified identifier must be a valid TranslationBridgeID.

2.4 operation query (TranslationQueryRequest) : TranslationQueryListResponse / ErrorResponse

Operation *query* lists the translation bridges that match the filtering requirements. The query data must meet the following criteria:

- The operation returns results in pages. There are default page data settings, but the requester can provide a custom specification.
- If page number is specified, the page size must be specified as well and vice versa.
- In some Local Clouds there is a maximum page size.
- If a filter expects a list, there is an OR relation between the elements of the filter.
- There is an AND relation between different kind of filters.

3 Information Model

Here, all data objects that can be part of the **translationBridgeManagement** service are listed and must be respected by the hosting system. Note that each subsection, which describes one type of object, begins with the *struct* keyword, which is used to denote a collection of named fields, each with its own data type. As a complement to the explicitly defined types in this section, there is also a list of implicit primitive types in Section 3.24, which are used to represent things like hashes and identifiers.

3.1 struct TranslationDiscoveryMgmtRequest

Field	Type	Mandator	Description
authentication	Identity	yes	The requester of the operation.
candidates	List<TranslationDiscoveryServiceInstanceDescriptor>	yes	Possible service instances for consuming via a translation bridge.
consumer	SystemName	yes	The system that needs a translation bridge.
operation	ServiceOperationName	yes	The operation that the consumer wants to consume.
interfaceTemplateNames	List<InterfaceName>	yes	The name of the interfaces that the consumer can use.
inputDataModelId	DataModelID	yes (no)	The identifier of the data model that the consumer can use as input payload of the specified operation. Must be omitted if there is no input payload.
outputDataModelId	DataModelID	yes (no)	The identifier of the data model that the consumer can use as response payload of the specified operation. Must be omitted if there is no response payload.
flags	TranslationDiscoveryFlagMap	no	Flags to fine tune the discovery process.

3.2 struct Identity

An Object which describes the identity of a system. It also contains whether the identified system has higher level administrative rights.

3.3 struct TranslationDiscoveryServiceInstanceDescriptor

Field	Type	Mandatory	Description
instanceId	ServiceInstanceID	yes	Unique identifier of the service instance.
provider	TranslationDiscoverySystemDescriptor	yes	Provider system.
serviceDefinition	TranslationDiscoveryServiceDefinitionDescriptor	yes	Service definition.
interfaces	List<TranslationDiscoveryServiceInstanceInterfaceDescriptor>	yes	Available access interfaces of the service instance.

3.4 struct TranslationDiscoverySystemDescriptor

Field	Type	Mandatory	Description
name	SystemName	yes	Unique name of the system.

3.5 struct TranslationDiscoveryServiceDefinitionDescriptor

Field	Type	Mandatory	Description
name	ServiceName	yes	Unique name of the service definition.

3.6 struct TranslationDiscoveryServiceInstanceInterfaceDescriptor

Field	Type	Mandatory	Description
templateName	InterfaceName	yes	The name of the interface template that describes the interface structure.
policy	SecurityPolicy	yes	The security of the interface.
properties	Metadata	yes	Interface template-specific data.

3.7 struct Metadata

An Object which maps String keys to primitive, Object or list values.

3.8 sruct TranslationDiscoveryFlagMap

An Object which maps TranslationDiscoveryFlag keys to Boolean values.

3.9 struct TranslationDiscoveryResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
bridgeId	TranslationBridgeID	Unique identifier related to discovery results.
candidates	List<TranslationBridgeCandidate>	Information about service instances that can be accessed via a translation bridge.

3.10 struct TranslationBridgeCandidate

Field	Type	Description
serviceInstanceId	ServiceInstanceID	Unique identifier of the service instance.
interfaceTemplateName	InterfaceName	Name of the interface that can be used to access the specified operation of the service instance.

3.11 struct ErrorResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
errorMessage	String	Description of the error.
errorCode	Number	Numerical code of the error.
type	ErrorType	Type of the error.
origin	String	Origin of the error.

3.12 struct DataModelMap

An Object that maps the key *dataModels* to an Object, in which keys are ServiceOperationNames and the values are Objects containing DataModelIDs using keys *input* and *output*. If an operation has input payload then the identifier of the input data model is assigned to the key *input*. If the operation has response payload then the identifier of the response data model is assigned to the key *output*. If an operation has no input and/or response payload, the appropriate key-value pair must be omitted. If an operation has no payload at all, then that operation can be omitted from the structure.

Should looks like this:

```
dataModels: {
    <operation name>: {
        input: <input data model id>
        output: <output data model id>
    }
}
```

3.13 struct TranslationNegotiationMgmtRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
bridgeId	TranslationBridgeID	yes	The unique identifier of the results of a previously executed discovery operation.
targetInstanceId	ServiceInstanceID	yes	Identifier of the target service instance.

3.14 struct TranslationNegotiationResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
bridgeId	TranslationBridgeID	Unique identifier of the translation bridge.
bridgeInterface	TranslationNegotiationServiceInstanceInterfaceDescriptor	Access information for the translation bridge.
tokenExpiresAt	DateTime	Token expiry date. Only filled if target provider's operation needs an expirable token.
tokenUsageLimit	Number	A number about how many times a token can be used. Only filled if target provider's operation needs a usage limited token.

3.15 struct TranslationNegotiationServiceInstanceInterfaceDescriptor

Field	Type	Description
templateName	InterfaceName	The name of the interface template that describes the interface structure.
protocol	Protocol	The communication protocol of the interface.
policy	SecurityPolicy	The security of the interface.
properties	Metadata	Interface template-specific data.

3.16 struct TranslationAbortMgmtRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
ids	List<TranslationBridgeID>	yes	Unique identifiers of the translation bridges to delete.

3.17 struct TranslationAbortMgmtResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
results	TranslationAbortResults	Results of the abort jobs.

3.18 struct TranslationAbortResults

An Object which maps TranslationBridgeID keys to Boolean values. The value is true if and only if the appropriate translation bridge in not working after the operation.

3.19 struct TranslationQueryRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
pageNumber	Number	no (yes)	The number of the requested page. It is mandatory, if page size is specified.
pageSize	Number	no (yes)	The number of entries on the requested page. It is mandatory, if page number is specified.
pageSortField	String	no	The identifier of the field which must be used to sort the entries.
pageDirection	Direction	no	The direction of the sorting.
bridgeIds	List<TranslationBridgeID>	no	Requester is looking for translation bridges with any of the specified identifiers.
creators	List<SystemName>	no	Requester is looking for translation bridges that are created by any of the specified systems.
statuses	List<TranslationBridgeStatus>	no	Requester is looking for translation bridges with any of the specified statuses.
consumers	List<SystemName>	no	Requester is looking for translation bridges that are created for any of the specified systems.

providers	List<SystemName>	no	Requester is looking for translation bridges where the target system is in the specified list.
serviceDefinitions	List<ServiceName>	no	Requester is looking for translation bridges with any of the specified names as target service definition.
interfaceTranslators	List<SystemName>	no	Requester is looking for translation bridges where the used interface translator is in the specified list.
dataModelTranslators	List<SystemName>	no	Requester is looking for translation bridges where the input or result data model translator is in the specified list.
creationFrom	DateTime	no	Requester is looking for translation bridges that was created after the specified moment.
creationTo	DateTime	no	Requester is looking for translation bridges that was created before the specified moment.
alivesFrom	DateTime	no	Requester is looking for translation bridges that was used after the specified moment.
alivesTo	DateTime	no	Requester is looking for translation bridges that was used before the specified moment.
minUsage	Number	no	Requester is looking for translation bridges was used *minUsage* times at least.
maxUsage	Number	no	Requester is looking for translation bridges was used *maxUsage* times at the most.

3.20 struct TranslationQueryListResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
entries	List<TranslationQueryResponse>	A page of translation bridges.
count	Number	Total number of translation bridges that match the filters.

3.21 struct TranslationQueryResponse

Field	Type	Description
bridgeId	TranslationBridgeID	Unique identifier of the translation bridge.
status	TranslationBridgeStatus	Status of the translation bridge.
usageReportCount	Number	Information about how many times the translation bridge was used.
alivesAt	DateTime	The moment when the translation was used last time.
message	String	Error message if the translation bridge is in error state.
consumer	SystemName	The translation bridge is constructed for this consumer.
provider	SystemName	The translation bridge is constructed to use this system.
serviceDefinition	ServiceName	The translation bridge is constructed to use this service.
operation	ServiceOperationName	The translation bridge is constructed to use this operation.
interfaceTranslator	SystemName	The translation bridge is using this system as interface translator.
interfaceTranslatorData	TranslationInterfaceTranslationDataDescriptor	Additional information about the interface translator.
inputDataModelTranslator	SystemName	The translation bridge is using this system as input data model translator.
inputDataModelTranslatorData	TranslationDataModelTranslationDataDescriptor	Additional information about the data model translator for input translation.
outputDataModelTranslator	SystemName	The translation bridge is using this system as output data model translator.
outputDataModelTranslatorData	TranslationDataModelTranslationDataDescriptor	Additional information about the data model translator for output translation.
createdBy	SystemName	The system that requested the translation bridge construction.
createdAt	DateTime	Translation bridge was registered at this timestamp.
updatedAt	DateTime	Translation bridge was modified at this timestamp.

3.22 struct TranslationInterfaceTranslationDataDescriptor

Field	Type	Description
fromInterfaceTemplate	InterfaceName	Translator accepts requests using this interface.
toInterfaceTemplate	InterfaceName	Translator sends requests to target provider using this interface.
token	String	An authorization token for the target operation (if required).
interfaceProperties	Metadata	Access interface data of the interface translator.
configurationSettings	Metadata	Additional configuration of the interface translator.

3.23 struct TranslationDataModelTranslationDataDescriptor

Field	Type	Description
fromModelId	DataModelID	Translator needs to translate from this data model.
toModelId	DataModelID	Translator needs to translate to this data model.
interfaceProperties	Metadata	Access interface data of the data model translator.
configurationSettings	Metadata	Additional configuration of the data model translator.

3.24 Primitives

Types and structures mentioned throughout this document that are assumed to be available to implementations of this service. The concrete interpretations of each of these types and structures must be provided by any IDD document claiming to implement this service.

Type	Description
Boolean	One out of *true* or *false*.
DataModelID	A string identifier that is intended to be both human and machine-readable. It also defines the specific format and the associated semantics. Must follow camelCase naming convention.
DateTime	Pinpoints a specific moment in time.
Direction	The direction of a sorting operation. Possible values are the representation of ascending or descending order.
ErrorType	Any suitable type chosen by the implementor of service.
InterfaceName	A string identifier of an interface descriptor. Must follow snake_case naming convention.
List<A>	An *array* of a known number of items, each having type A.
Number	Decimal number.
Object	Set of primitives and possible further objects.

OperationStatus	Logical, textual or numerical value that indicates whether an operation is a success or a failure. Multiple values can be used for success and error cases to give additional information about the nature of the result.
Protocol	A string representation of a communication protocol.
SecurityPolicy	Any suitable security policy chosen by the implementor of service.
ServiceInstanceID	A composite string identifier that is intended to be both human and machine-readable. It consists of the instance's provider name, service definition and version, each separated by a special delimiter character. Each part must follow its related naming convention.
ServiceName	A string identifier that is intended to be both human and machine-readable. Must follow camelCase naming convention.
ServiceOperationName	A string identifier that is intended to be both human and machine-readable. Must follow kebab-case naming convention.
String	A chain of characters.
SystemName	A string identifier that is intended to be both human and machine-readable. Must follow PascalCase naming convention.
TranslationBridgeID	A unique character sequence that identifies translation bridges and/or discovery information about potential translation bridges.
TranslationBridgeStatus	Any suitable status chosen by the implementor of service.
TranslationDiscoveryFlag	Specific string values to fine tune the translation discovery process. Possible values are: *CONSUMER_BLACKLIST_CHECK*, *CANDIDATES_BLACKLIST_CHECK*, *TRANSLATORS_BLACKLIST_CHECK*, *CANDIDATES_AUTH_CHECK*, *TRANSLATORS_AUTH_CHECK*. The first three (when true) uses the Blacklist Support system and the other two (when true) uses the ConsumerAuthorization Core system to check whether the translation bridge is possible or not.

4 References

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	YYYY-MM-DD	5.1.0		Xxx Yyy

5.2 Quality Assurance

No.	Date	Version	Approved by
1	YYYY-MM-DD	5.1.0	