translationBridge

Service Description

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

Contents

1 Overview

This document describes the **translationBridge** service, which enables to find providers whose service operation is accessible by the requester via a translation bridge using the currently available translators. The service also allows to build or abort such bridges. An example of this interaction is when a consumer system can't find a suitable provider for 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 **translationBridge** 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 requester via a translation bridge using the currently available translators;
- *negotiation* allows consumers to create a translation bridge to one selected provider's one particular service operation (should be used after a successful discovery but it is possible to do that during the negotiation if no prior discovery is happened);
- *abort* aborts an existing bridge created by the requester.

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 by a consumer system in Local Clouds where simple rule-based service orchestration is available or there is no service orchestration at all.

Consumer systems should call the *discovery* operation with ”almost good” service instance candidates that comes from the Service Registry. ”Almost good” means that these candidates provides the required service operation but the direct communication 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.

Consumer 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.

Please note that it is possible to use the *negotiation* operation without using the *discovery* operation first. In that case consumer can't provide the identifier mentioned before, but needs to specify additional requirements. If no identifier is provided, the *negotiation* automatically executes the *discovery* operation on that one candidate that is specified before trying to build a translation bridge.

After the consumer has finished the work with the translation bridge, it should call the *abort* to eliminate the bridge and free up resources.

1.2 Important Delimitations

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

1.3 Access policy

Available for anyone within the Local Cloud.

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 (TranslationDiscoveryRequest) : TranslationDiscoveryResponse / ErrorResponse

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

- With this operation the requester can only found good candidates to accessing one of their service operation via a translation bridge for itself.
- 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.).

2.2 operation negotiation (TranslationNegotiationRequest) : TranslationNegotiationResponse / ErrorResponse

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

- Target is mandatory.
- If bridge identifier is specified only the service instance identifier of the target is necessary.
- In either case, service instance identifier is mandatory and must be a valid ServiceInstanceID.
- If bridge identifier is missing, a *discovery* operation is automatically called before the negotiation steps where the only candidate is the target. That means target must meet the criteria about candidates mentioned at operation *discovery*.
- The same is true in the case of operation and interface template names parameters.

2.3 operation abort (TranslationAbortRequest) : OperationStatus / ErrorResponse

Operation *abort* aborts an existing bridge. The abort data must meet the following criteria:

- The bridge identifier is mandatory.
- Requester can only abort translation bridges that was created by its request.

3 Information Model

Here, all data objects that can be part of the **translationBridge** 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 3.17, which are used to represent things like hashes and identifiers.

3.1 struct TranslationDiscoveryRequest

Field	Type	Mandatoı	Description
authentication	Identity	yes	The requester of the operation.
candidates	List<TranslationDiscoveryService-InstanceDescriptor>	yes	Possible service instances for consuming via a translation bridge.
operation	ServiceOperationName	yes	The operation that the requester wants to consume.
interfaceTemplateNames	List<InterfaceName>	yes	The name of the interfaces that the consumer can use.
inputDataModelId	DataModelIdentifier	yes (no)	The identifier of the data model that the requester can use as input payload of the specified operation. Must be omitted if there is no input payload.
outputDataModelId	DataModelIdentifier	yes (no)	The identifier of the data model that the requester can use as response payload of the specified operation. Must be omitted if there is no response payload.

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	TranslationDiscoveryServiceDefinition-Descriptor	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 struct TranslationDiscoveryResponse

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

3.9 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.

3.10 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.

3.11 struct DataModelMap

An Object that maps the key *dataModels* to an Object, in which keys are ServiceOperationNames and the values are Objects containing DataModelIdentifiers 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>
    }
}
```

3.12 struct TranslationNegotiationRequest

Field	Type	Mandatoı	Description
authentication	Identity	yes	The requester of the operation.
bridgeId	BridgeIdentifier	no	The unique identifier of the results of a previously executed discovery operation.
target	TranslationNegotiationService-InstanceDescriptor	yes	Information about the target service instance.
operation	ServiceOperationName	no (yes)	The operation that the requester wants to consume. Mandatory if *bridgeId* is not specified.
interfaceTemplateName	InterfaceName	no (yes)	The name of the interface that the consumer can use. Mandatory if *bridgeId* is not specified
inputDataModelId	DataModelIdentifier	no (yes)	The identifier of the data model that the requester can use as input payload of the specified operation. Can be omitted if *bridgeId* is specified and must be omitted if there is no input payload.
outputDataModelId	DataModelIdentifier	no (yes)	The identifier of the data model that the requester can use as response payload of the specified operation. Can be omitted if *bridgeId* is specified and must be omitted if there is no response payload.

3.13 struct TranslationNegotiationServiceInstanceDescriptor

Field	Type	Mandatory	Description
instanceId	ServiceInstanceID	yes	Unique identifier of the service instance.
provider	TranslationDiscoverySystem-Descriptor	no (yes)	Provider system. Mandatory if the *bridgeId* is not specified in the containing model.
serviceDefinition	TranslationDiscoveryService-DefinitionDescriptor	no (yes)	Service definition. Mandatory if the *bridgeId* is not specified in the containing model.
interfaces	List<TranslationDiscoveryService-InstanceInterfaceDescriptor>	no (yes)	Available access interfaces of the service instance. Mandatory if the *bridgeId* is not specified in the containing model.

3.14 struct TranslationNegotiationResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
bridgeId	BridgeIdentifier	Unique identifier of the translation bridge.
bridgeInterface	TranslationNegotiationService-InstanceInterfaceDescriptor	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 TranslationAbortRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
bridgeId	BridgeIdentifier	yes	The unique identifier of the translation bridge.

3.17 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
BridgeIdentifier	A unique character sequence that identifies translation bridges and/or discovery information about potential translation bridges.
DataModelIdentifier	A string identifier that is intended to be both human and machine-readable. Must follow camelCase naming convention.
DateTime	Pinpoints a specific moment in time.
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.

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	