interfaceBridgeManagement

Service Description

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

Contents

1 Overview

This document describes the **interfaceBridgeManagement** service, which provides functionality to translate between interface templates independently and to leverage other providers for data model translation, in order to create a translation bridge between incompatible consumers and providers. To enable other systems to consume this service, it needs to be offered through the ServiceRegistry and the handled interface template names have to be included as part of the service metadata.

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

- *check-targets* removes the incompatible interfaces from target providers (or the target provider if all of its interfaces are incompatible);
- *initialize-bridge* prepares a translation bridge and returns its service interface details;
- *abort-bridge* destroys a translation bridge;

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 is primarily provided to the TranslationManager Support System to enable the organization of a translation bridge. The TranslationManager makes the target provider candidates for a specific service filtered using the *check-targets* operation that cleans the candidate provider list from untranslatable interfaces (therefore from incompatible providers as well).

When the TranslationManager calculates the possible translation bridges and one is chosen to be initialized, it calls the *initilaize-bridge* operation. The init request describes the entire translation bridge: the target service interface to connect; the input data model translation service (if necessary) to use; the result data model translation service (if necessary) to use and the authorization tokens (if required). Once the translation bridge is ready to use, a dynamically created access interface is returned.

The *abort-bridge* operation can be used to destroy existing bridges, however inactive translation bridges should be dismantled automatically.

1.2 Important Delimitations

Interface translation is only possible between interface templates and in the directions that are defined in the service metadata; and if proper data model translator systems are available when required.

1.3 Access policy

Available for systems with management role, primarily for the TranslationManager Support System.

2 Service Operations

This section describes the abstract signatures of each operations of the service. The **interfaceBridgeManagement** service is used to initialize and abort interface translation bridges. 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 check-targets (TranslationCheckTargetsRequest) : TranslationCheckTargetsResponse / ErrorResponse

Operation *check-targets* validates a target provider list by removing untranslatable service interfaces. An interface is considered translatable only if all required properties are present. Any provider left without interfaces is removed from the list.

2.2 operation initialize-bridge (TranslationBridgeInitializationRequest) : AccessInterfaceResponse / ErrorResponse

Operation *initialize-bridge* sets up a translation bridge based on the given interfaces, additional data model translation services (if necessary) and returns a dynamically generated access interface. Accessing this interface requires the client to supply the translation bridge ID as an access token.

2.3 operation abort-translation (TranslationBridgeAbortRequest) : OperationStatus / ErrorResponse

Operation *abort-translation* terminates an existing translation bridge without checking whether it is actively used or not.

3 Information Model

Here, all data objects that can be part of the **interfaceBridgeManagement** 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 3.11, which are used to represent things like hashes and identifiers.

3.1 struct TranslationCheckTargetsRequest

Field	Type	Mandatory	Description
accessToken	String	no	The access token that might be required.
targetOperation	ServiceOperationName	yes	The target service-operation.
targets	List<TranslationTargetDTO>	yes	List of candidate service providers to filter based on translatable interfaces.

3.2 struct TranslationTargetDTO

Field	Type	Mandatory	Description
instanceId	ServiceInstanceID	yes	The service instance identifier.
interfaces	List<ServiceInterfaceDescriptor>	yes	List of available interfaces.

3.3 struct ServiceInterfaceDescriptor

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

3.4 struct Map

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

3.5 struct TranslationCheckTargetsResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
targets	List<TranslationTargetDTO>	List of candidate service providers with translatable interfaces.

3.6 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.

3.7 struct TranslationBridgeInitializationRequest

Field	Type	Mandatory	Description
accessToken	String	no	The access token that might be required.
bridgeId	TranslationBridgeID	yes	The given identifier of the intended translation bridge.
operation	ServiceOperationName	yes	The target service-operation.
inputInterface	InterfaceName	yes	The name of the interface template to be translated.
targetInterface	InterfaceName	yes	The name of the interface template of the target service-operation.
targetInterfaceProperties	Map	yes	Interface specific data of the target service-operation.
inputDataModelRequirement	DataModelID	no (yes)	The data model identifier of what the consumer can provide as the target service-operation input. It is mandatory if the target operation has input payload and input needs to be translated.
inputDataModelTranslator	DataModelTranslationDescriptor	no	The data model translation provider for translating the input of the target service-operation.
resultDataModelRequirement	DataModelID	no (yes)	The data model identifier of what the consumer can handle as the target service-operation result. It is mandatory if the target operation has result payload and result needs to be translated.
resultDataModelTranslator	DataModelTranslationDescriptor	no	The data model translation provider for translating the output of the target service-operation.
authorizationToken	String	no	Authorization token for the target service-operation if required.
interfaceTranslatorSettings	Map	no	Any configuration data that might be required by the interface translation process.

3.8 struct DataModelTranslationDescriptor

Field	Type	Mandatory	Description
fromModelId	DataModelID	yes	The identifier of the data model to be translated.
toModelId	DataModelID	yes	The identifier of the required result data model.
interfaceProperties	Map	yes	Interface specific data of the model translation provider.
configurationSettings	Map	no	Any configuration data that might be required by the model translation provider.

3.9 struct AccessInterfaceResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
templateName	InterfaceName	The name of the interface template.
protocol	Protocol	The communication protocol of the interface.
policy	SecurityPolicy	The security level of the interface.
properties	Map	Interface template-specific data.

3.10 struct TranslationBridgeAbortRequest

Field	Type	Mandatory	Description
accessToken	String	no	The access token that might be required.
bridgeID	TranslationBridgeID	yes	The identifier of the translation bridge to be terminated.

3.11 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
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.
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.
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.
TaskID	A unique string identifier that belongs to a translation task.
TranslationBridgeID	A unique character sequence that is associated with an existing translation bridge.

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	