serviceOrchestrationStoreManagement

Service Description

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

Contents

1 Overview

This document describes the **serviceOrchestrationStoreManagement** service, which enables systems with administrative rights to manage the peer-to-peer orchestration rules within an Eclipse Arrowhead Local Cloud (LC). These rules associate consumer systems to service instances and can be used during orchestration process in case of simple-store strategy.

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

- *query* lists the stored rules that match the filtering requirements;
- *create* adds new orchestration rules to the LC in bulk;
- *modify-priorities* updates the priority of the given rules in bulk;
- *remove* removes stored orchestration rules in bulk.

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 can be used to define peer-to-peer rules that determine the orchestration in a Local Cloud that uses simple-store strategy. These rules associate a consumer system with a service instance identifier. Each rule has a priority, and higher-priority rules appear earlier in the orchestration results. Priority is the only attribute that can be modified in a rule.

1.2 Important Delimitations

The consumer 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. The **serviceOrchestrationStoreManagement** service is used to perform *query, create, modify-priorities* and *remove* operations on orchestration rules. 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 query (OrchestrationSimpleStoreQueryRequest) : OrchestrationSimpleStoreListResponse / ErrorResponse

Operation query lists the orchestration store records 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

2.2 operation create (OrchestrationSimpleStoreListRequest) : OrchestrationSimpleStoreListResponse / ErrorResponse

This operation requires at least the consumer's identity and the orchestration store entries to create. The result of this operation is a list of rules that were created. It is not allowed to store rules where the consumer system name, service instance identifier and priority all have the same value as in another rule.

2.3 operation modify-priorities (PriorityRequest) : OrchestrationSimpleStoreListResponse / ErrorResponse

This operation requires at least the consumer's identity and the rule identifiers with their new priorities. The result of this operation is a list of all the rules referenced by the identifiers after the update (even if they were not actually modified). When specifying priorities, care must be taken to ensure that the rules remain unique according to consumer system name, service instance identifier and priority.

2.4 operation remove (OrchestrationSimpleStoreRemoveRequest) : OperationStatus / ErrorResponse

Operation *remove* requires the rule identifiers and removes their associated store orchestration records.

3 Information Model

Here, all data objects that can be part of the **serviceOrchestration** 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 3.12, which are used to represent things like hashes and identifiers.

3.1 struct OrchestrationSimpleStoreQueryRequest

Field	Type	Mandatory	Description
pagination	PageRequest	yes	Paging information about the queried store records.
ids	List<StoreEntryID>	no (yes)	Requester is looking for store entries with any of the specified identifiers.
consumerNames	List<SystemName>	no (yes)	Requester is looking for store entries with any of the consumer system names.
serviceDefinitions	List<ServiceName>	no (yes)	Requester is looking for store entries with any of the specified service definition names.
serviceInstanceIds	List<ServiceInstanceID>	no (yes)	Requester is looking for store entries with any of the specified service instance identifiers.
minPriority	Number	no	Requester is looking for store entries with at least the specified priority.
maxPriority	Number	no	Requester is looking for store entries with at most the specified priority.
createdBy	SystemName	no (yes)	Requester is looking for store entries that were created by a system with the specified name.

Note: One of the following fields must be specified: *ids, consumerNames, serviceDefinitions, serviceInstanceIds* or *createdBy*.

3.2 struct PageRequest

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.

3.3 struct OrchestrationSimpleStoreListResponse

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

3.4 struct OrchestrationSimpleStoreResponse

Field	Type	Description
id	StoreEntryID	Unique identifier of the store entry.
consumer	SystemName	Name of the consumer system that the rule applies to.
serviceDefinition	ServiceName	Name of the service definition that the rule applies to.
serviceInstanceId	ServiceInstanceID	Identifier of the service instance that the rule applies to.
priority	Number	Priority of the rule.
createdBy	SystemName	The rule was created by this system.
updatedBy	SystemName	The rule was updated by this system.
createdAt	DateTime	The rule was created at this timestamp.
updatedAt	DateTime	The rule was last updated at this timestamp.

3.5 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.

3.6 struct OrchestrationSimpleStoreListRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
entries	List<OrchestrationSimpleStoreRequest>	yes	A list of store entries to create.

3.7 struct Identity

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

3.8 struct OrchestrationSimpleStoreRequest

Field	Type	Mandatory	Description
consumer	SystemName	yes	The name of the consumer system that the rule applies to.
serviceInstanceId	ServiceInstanceID	yes	The name of the service instance identifier that the rule applies to.
priority	Number	yes	The priority of the rule.

3.9 struct OrchestrationSimpleStorePriorityRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
priorities	PriorityMap	yes	The identifiers of the store entries to update and their associated priority values.

3.10 struct PriorityMap

An Object which maps StoreEntryID identifiers to number values.

3.11 struct OrchestrationSimpleStoreRemoveRequest

Field	Type	Mandatory	Description
ids	List<StoreEntryID>	yes	A list of identifiers of the store entries to remove.

3.12 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
DateTime	Pinpoints a specific moment in time.
ErrorType	Any suitable type chosen by the implementor of service.
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.
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.
StoreEntryID	Unique string identifier.
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	5.2.0		Xxx Yyy

5.2 Quality Assurance

No.	Date	Version	Approved by
1	YYYY-MM-DD	5.2.0	