service-discovery

Service Description

Abstract
This document provides service description for the **service-discovery** service.

Contents

1 Overview **4**

1.1 How This Service Is Meant to Be Used . . . 4
1.2 Important Delimitations . . . 4
1.3 Access policy . . . 4

2 Service Operations **5**

2.1 operation register . . . 5
2.2 operation lookup . . . 5
2.3 operation revoke . . . 5

3 Information Model **7**

3.1 struct ServiceRegistrationRequest . . . 7
3.2 struct Identity . . . 7
3.3 struct Metadata . . . 7
3.4 struct ServiceInterfaceRequest . . . 7
3.5 struct ServiceRegistrationResponse . . . 8
3.6 struct SystemDescriptor . . . 8
3.7 struct AddressDescriptor . . . 8
3.8 struct DeviceDescriptor . . . 9
3.9 struct ServiceDefinitionDescriptor . . . 9
3.10 struct ServiceInterfaceDescriptor . . . 9
3.11 struct ErrorResponse . . . 9
3.12 struct ServiceLookupRequest . . . 10
3.13 struct MetadataRequirements . . . 10
3.14 struct ServiceLookupResponse . . . 11
3.15 struct ServiceLookupResult . . . 11
3.16 struct ServiceRevokeRequest . . . 12
3.17 Primitives . . . 12

4 References **13**

5 Revision History **14**

5.1 Amendments 14

5.2 Quality Assurance 14

1 Overview

This document describes the **service-discovery** service, which enables both application and core/support systems to register and revoke their service instances to/from Local Cloud. It also enables to lookup for service instances. Service and service instance representation is mandatory for the base functionalities of a Local Cloud therefore it is an integral part of the implementation of the requirements in Service Registry Core System. An example of this interaction when a provider registers its service instances to offer them to other systems in the Local Cloud. To enable other systems to use, to consume it, this service needs to be offered through the Service Registry.

The **service-discovery** service contains the following operations:

- *register* adds new service instance to the Local Cloud;
- *revoke* removes a service instance from the Local Cloud;
- *lookup* lists the service instances that matches 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

A provider system can use the *register* operation of **service-discovery** service to register its service instances. When a system is shutting down or stop offering services, it should remove its service instances from the Local Cloud by using *revoke* operation.

A consumer system can use the *lookup* operation to find the appropriate service instance which it can consume later.

1.2 Important Delimitations

As a general rule, the requester has to identify itself to use any of the operations.

However, there are some special cases when looking for a service instance can be requested anonymously. For example, when somebody looks for a service instance that it has to consume to get an identity (some kind of login).

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. The **service-discovery** service is used to *register*, *lookup* and *revoke* service instances. 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 register (ServiceRegistrationRequest) : ServiceRegistrationResponse / ErrorResponse

The registration data must meet the following criteria:

- The requester system has to be present in the Local Cloud.
- Service definition names are case insensitive and have to be unique within the Local Cloud.
- Service definition names can contain maximum 63 character of letters (english alphabet), numbers and dash (-), and have to start with a letter (also cannot end with dash).
- If an expiration date is specified, it cannot points to a past date.
- Keys in the metadata structure can not contain dot (.) character.
- At least one interface must be specified with its template name, the used policy and all the necessary template-specific properties.
- If the specified interface template is not present in the Local Cloud, a protocol is also need to be defined. In some Local Clouds using previously unknown interface templates during the service instance registration may be forbidden and thus rejected.
- Service instances can register multiple times, the appropriate old service instances are discarded.

2.2 operation lookup (ServiceLookupRequest) : ServiceLookupResponse / ErrorResponse

The lookup data must meet the following criteria:

- 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.
- To use this operation, an application system must specify at least one service instance id OR one provider name OR one service definition.
- In some Local Clouds, operation *lookup* can be restricted, which means only the ”publicly” available service instances are returned. To gain access to a non-public service instance, an application system must use the **orchestration** service.

2.3 operation revoke (ServiceRevokeRequest) : OperationStatus / ErrorResponse

The input operation data must meet the following criteria:

- With this operation a system can revoke their own service instances.

3 Information Model

Here, all data objects that can be part of the **service-discovery** 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 ServiceRegistrationRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
serviceDefinitionName	Name	yes	The service definition of the instance.
version	Version	no	Version of the service instance.
expiresAt	DateTime	no	The moment of the future from which the service instance will not available.
metadata	Metadata	no	Additional information about the service instance.
interfaces	List<ServiceInterfaceRequest>	yes	Available access interfaces of the service instance.

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 Metadata

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

3.4 struct ServiceInterfaceRequest

Field	Type	Mandatory	Description
templateName	InterfaceTemplate	yes	The name of the interface template that describes the interface structure.
protocol	Protocol	no (yes)	The communication protocol of the interface. Only mandatory if the interface template is not previously known in the Local Cloud.
policy	SecurityPolicy	yes	The security of the interface.
properties	Metadata	yes	Interface template-specific data.

3.5 struct ServiceRegistrationResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
instanceId	Name	Unique identifier of the registered service instance.
provider	SystemDescriptor	Information about the service instance provider system.
serviceDefinition	ServiceDefinitionDescriptor	Information about the service definition.
version	Version	Version of the service instance.
expiresAt	DateTime	The moment of the future from which the service instance will not be available.
metadata	Metadata	Additional information about the registered service instance.
interfaces	List<ServiceInterfaceDescriptor>	Available access interfaces of the service instance.
createdAt	DateTime	Service instance was registered at this timestamp.
updatedAt	DateTime	Service instance was modified at this timestamp.

3.6 struct SystemDescriptor

Field	Type	Description
name	Name	Unique identifier of the system.
metadata	Metadata	Additional information about the system.
version	Version	Version of the system.
addresses	List<AddressDescriptor>	Different kind of addresses of the system.
device	DeviceDescriptor	Information about the device on which the system is running.
createdAt	DateTime	System was registered at this timestamp.
updatedAt	DateTime	System was modified at this timestamp.

3.7 struct AddressDescriptor

Field	Type	Description
type	AddressType	Type of the address.
address	Address	Address.

3.8 struct DeviceDescriptor

Field	Type	Description
name	Name	Unique identifier of the device.
metadata	Metadata	Additional information about the device.
addresses	List<AddressDescriptor>	Different kind of addresses of the device.
createdAt	DateTime	Device was registered at this timestamp.
updatedAt	DateTime	Device was modified at this timestamp.

3.9 struct ServiceDefinitionDescriptor

Field	Type	Description
name	Name	Unique identifier of the service definition.
createdAt	DateTime	Service definition was registered at this timestamp.
updatedAt	DateTime	Service definition was modified at this timestamp.

3.10 struct ServiceInterfaceDescriptor

Field	Type	Description
templateName	InterfaceTemplate	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.

3.11 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 ServiceLookupRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
verbose	Boolean	no	If true detailed system and device information also returns (only if the provider supports it).
instanceIds	List<Name>	no (yes)	Requester is looking for service instances with any of the specified names. Mandatory if no providerNames nor serviceDefinitionNames are specified.
providerNames	List<Name>	no (yes)	Requester is looking for service instances that are provided by any of the specified systems. Mandatory if no serviceInstanceIds nor serviceDefinitionNames are specified.
serviceDefinitionNames	List<Name>	no (yes)	Requester is looking for service instances with any of the specified service definition names. Mandatory if no serviceInstanceIds nor providerNames are specified.
versions	List<Version>	no	Requester is looking for service instances with any of the specified versions.
alivesAt	DateTime	no	Request is looking for service instances that will be available at the specified moment of the future.
metadataRequirementsList	List<MetadataRequirements>	no	Requester is looking for service instances that are matching any of the specified metadata requirements.
interfaceTemplateNames	List<InterfaceTemplate>	no	Requester is looking for service instances with any of the specified interface template names.
interfacePropertyRequirementsList	List<MetadataRequirements>	no	Requester is looking for service instances with interfaces that are matching any of the specified properties requirements.

policies	List<SecurityPolicy>	no	Requester is looking for service instances with any of the specified security policies.

3.13 struct MetadataRequirements

A special Object which maps String keys to Object, primitive or list values, where

- Keys can be paths (or multi-level keys) which access a specific value in a Metadata structure, where parts of the path are delimited with dot character (e.g. in case of ”key.subkey” path we are looking for the key named ”key” in the metadata, which associated with an embedded object and in this object we are looking for the key named ”subkey”).
- Values are special Objects with two fields: an operation (e.g. less than) and an actual value (e.g. a number). A metadata is matching a requirement if the specified operation returns true using the metadata value referenced by a key path as first and the actual value as second operands.
- Alternatively, values can be ordinary primitives, lists or Objects. In this case the operation is equals by default.

3.14 struct ServiceLookupResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
entries	List<ServiceLookupResult>	List of service instance results.
count	Number	Number of returned service instances.

3.15 struct ServiceLookupResult

Field	Type	Description
instanceId	Name	Unique identifier of the service instance.
provider	SystemDescriptor	Information about the service instance provider system.
serviceDefinition	ServiceDefinitionDescriptor	Information about the service definition.
version	Version	Version of the service instance.
expiresAt	DateTime	The moment of the future from which the service instance will not available.
metadata	Metadata	Additional information about the service instance.
interfaces	List<ServiceInterfaceDescriptor>	Available access interfaces of the service instance.
createdAt	DateTime	Service instance was registered at this timestamp.
updatedAt	DateTime	Service instance was modified at this timestamp.

3.16 struct ServiceRevokeRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
instanceId	Name	yes	Unique identifier of the service instance.

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
Address	A string representation of the address.
AddressType	Any suitable type chosen by the implementor or service.
Boolean	One out of `true` or `false`.
DateTime	Pinpoints a specific moment in time.
ErrorType	Any suitable type chosen by the implementor of service.
InterfaceTemplate	A string identifier of an interface descriptor.
List<A>	An *array* of a known number of items, each having type A.
Name	A string identifier that is intended to be both human and machine-readable.
Number	Decimal number.
Object	Set of primitives and possible further objects.
OperationStatus	Logical, textual or number value that indicates whether an operation is a success or a failure. It can be used multiple values 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.
String	A chain of characters.
Version	Specifies a service instance version. Version must follows the Semantic Versioning.

4 References

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	YYYY-MM-DD	5.0.0		Xxx Yyy

5.2 Quality Assurance

No.	Date	Version	Approved by
1	YYYY-MM-DD	5.0.0	