deviceDiscovery

Service Description

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

Contents

1 Overview

This document describes the **deviceDiscovery** service, which enables both application and Core/Support systems to lookup, register and revoke devices on which the Local Cloud's systems are running. Device representation is not necessary for the base functionalities of a Local Cloud but in certain use cases (e.g. enabling onboarding) is needed therefore it is an integral part of the implementation of the requirements in ServiceRegistry Core System. An example of this interaction is the Onboarding support system that has the capability to dynamically add new systems (and devices on which those systems are running) to the Local Cloud. To enable other systems to use, to consume it, this service needs to be offered through the ServiceRegistry.

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

- *register* adds new device to the Local Cloud;
- *revoke* removes a device from the Local Cloud;
- *lookup* lists the devices 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

In some use cases, the topology of the Local Cloud can be important. In these situations, an application system can use the *register* operation of **deviceDiscovery** service to register the device on which the system is running. Before that, the application system should use the *lookup* operation to check whether the device is already registered or not. When a device is removed from the Local Cloud, the *revoke* operation should be called.

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. The **deviceDiscovery** service is used to *register*, *lookup* and *revoke* devices. 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 (DeviceRegistrationRequest) : DeviceRegistrationResponse / ErrorResponse

Operation *register* adds new device to the Local Cloud. The registration data must meet the following criteria:

- Device names are case sensitive, must follow the UPPER_SNAKE_CASE naming convention and have to be unique within the Local Cloud.
- Device names can contain maximum 63 character of uppercase letters (English alphabet), numbers and underscore (_), and have to start with a letter (also cannot end with underscore).
- Device has to have at least one address. It is recommended to use MAC address and/or IP address.
- Keys in the metadata structure can not contain dot (.) character.
- Devices can register multiple times, but only if they are identical (same addresses, same metadata).

2.2 operation lookup (DeviceLookupRequest) : DeviceLookupResponse / ErrorResponse

Operation *lookup* lists the devices that match the filtering requirements. 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.

2.3 operation revoke (DeviceRevokeRequest) : OperationStatus / ErrorResponse

Operation *revoke* removes a device from the Local Cloud. The input operation data must meet the following criteria:

- Devices that have system assignment can not be revoked.

3 Information Model

Here, all data objects that can be part of the **deviceDiscovery** 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 DeviceRegistrationRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
name	DeviceName	yes	Unique identifier of the device.
metadata	Metadata	no	Additional information about the device.
addresses	List<Address>	yes	Different kind of addresses of the device.

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 DeviceRegistrationResponse

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

3.5 struct AddressDescriptor

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

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 DeviceLookupRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
deviceNames	List<DeviceName>	no	Requester is looking for devices with any of the specified names.
addresses	List<Address>	no	Requester is looking for devices with any of the specified addresses.
addressType	AddressType	no	Requester is looking for devices with the specified type of address.
metadataRequirementsList	List<MetadataRequirements>	no	Requester is looking for devices that are matching any of the specified metadata requirements.

3.8 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 is 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.

3.9 struct DeviceLookupResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
entries	List<DeviceLookupResult>	List of device results.
count	Number	Number of returned devices.

3.10 struct DeviceLookupResult

Field	Type	Description
name	DeviceName	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.

3.11 struct DeviceRevokeRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
name	DeviceName	yes	Unique identifier of the device.

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
Address	A string representation of the address.
AddressType	Any suitable type chosen by the implementor of service.
DateTime	Pinpoints a specific moment in time.
DeviceName	A string identifier that is intended to be both human and machine-readable. Must follow the UPPER_SNAKE_CASE naming convention.
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.
String	A chain of characters.

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	