

systemDiscovery

Service Description

Abstract

This document provides service description for the **systemDiscovery** service.

Contents

1 Overview	3
1.1 How This Service Is Meant to Be Used	3
1.2 Important Delimitations	3
1.3 Access policy	3
2 Service Operations	4
2.1 operation register	4
2.2 operation lookup	4
2.3 operation revoke	4
3 Information Model	5
3.1 struct SystemRegistrationRequest	5
3.2 struct Identity	5
3.3 struct Metadata	5
3.4 struct SystemRegistrationResponse	5
3.5 struct AddressDescriptor	5
3.6 struct DeviceDescriptor	6
3.7 struct ErrorResponse	6
3.8 struct SystemLookupRequest	6
3.9 struct MetadataRequirements	6
3.10 struct SystemLookupResponse	7
3.11 struct SystemLookupResult	8
3.12 Primitives	8
4 References	9
5 Revision History	10
5.1 Amendments	10
5.2 Quality Assurance	10

1 Overview

This document describes the **systemDiscovery** service, which enables both application and Core/Support systems to register itself to and revoke itself from Local Cloud. It also enables to lookup for systems. System representation is mandatory for the base functionalities of a Local Cloud therefore it is an integral part of the implementation of the requirements in ServiceRegistry Core System. An example of this interaction is when a system registers itself because it is a requirement to interact to other systems in the Local Cloud. To enable other systems to use, to consume it, this service needs to be offered through the ServiceRegistry.

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

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

An application system can use the *register* operation of **systemDiscovery** service to register itself. Before that, the application system can use the *lookup* operation to check whether it is already registered or not. When a system is removed from the Local Cloud, the *revoke* operation can 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 **systemDiscovery** service is used to *register*, *lookup* and *revoke* systems. 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** (**SystemRegistrationRequest**) : **SystemRegistrationResponse** / **ErrorResponse**

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

- System names are case sensitive, must following the PascalCase naming convention and have to be unique within the Local Cloud.
- System names can contain maximum 63 character of letters (english alphabet), and numbers, and have to start with a letter.
- System has to have at least one address (including those that come from the referenced device, if any). It is recommended to use IP address.
- Keys in the metadata structure can not contain dot (.) character.
- Device, if specified, has to be present in the Local Cloud.
- Systems can register multiple times, but only if they are identical (same version, addresses, metadata and device).

2.2 operation **lookup** (**SystemLookupRequest**) : **SystemLookupResponse** / **ErrorResponse**

Operation *revoke* removes a system from the Local Cloud. 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** (**Identity**) : **OperationStatus** / **ErrorResponse**

Operation *lookup* lists the systems that match the filtering requirements. The input operation data must meet the following criteria:

- With this operation a system can only revoke itself.

3 Information Model

Here, all data objects that can be part of the **systemDiscovery** 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 SystemRegistrationRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
metadata	Metadata	no	Additional information about the system.
version	Version	no	Version of the system.
addresses	List<Address>	yes	Different kind of addresses of the system.
deviceName	DeviceName	no	Unique identifier of the device on which the system is running.

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 SystemRegistrationResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
name	SystemName	Unique identifier of the registered system.
metadata	Metadata	Additional information about the registered system.
version	Version	Version of the system.
addresses	List<AddressDescriptor>	Different kind of addresses of the registered 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.

3.5 struct AddressDescriptor

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

3.6 struct DeviceDescriptor

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.

3.7 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.

3.8 struct SystemLookupRequest

Field	Type	Mandatory	Description
authentication	Identity	yes	The requester of the operation.
verbose	Boolean	no	If true detailed device information also returns (only if the provider supports it).
systemNames	List<SystemName>	no	Requester is looking for systems with any of the specified names.
addresses	List<Address>	no	Requester is looking for systems with any of the specified addresses.
addressType	AddressType	no	Requester is looking for systems with the specified type of address.

metadataRequirementsList	List<MetadataRequirements>	no	Requester is looking for systems that are matching any of the specified metadata requirements.
versions	List<Version>	no	Requester is looking for systems with any of the specified versions.
deviceNames	List<DeviceName>	no	Requester is looking for systems that are running on any of the specified devices.

3.9 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.

3.10 struct **SystemLookupResponse**

Field	Type	Description
status	OperationStatus	Status of the operation.
entries	List<SystemLookupResult>	List of system results.
count	Number	Number of returned systems.

3.11 struct **SystemLookupResult**

Field	Type	Description
name	SystemName	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.

3.12 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.
Boolean	One out of <code>true</code> or <code>false</code> .
DateTime	Pinpoints a specific moment in time.
DeviceName	A string identifier that is intended to be both human and machine-readable. Must following the UPPER_SNAKE_CASE naming convention.
ErrorType	Any suitable type chosen by the implementor of service.
List<A>	An <i>array</i> 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.
SystemName	A string identifier that is intended to be both human and machine-readable. Must following PascalCase naming convention.
Version	Specifies a system version. Version must follow the Semantic Versioning.



ARROWHEAD

Document title
systemDiscovery
Date
2025-06-06

Version
5.0.0
Status
DRAFT
Page
9 (10)

4 References



ARROWHEAD

Document title
systemDiscovery
Date
2025-06-06

Version
5.0.0
Status
DRAFT
Page
10 (10)

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	