ServiceRegistry Core System

System Description

Abstract
This document provides system description for the **ServiceRegistry Core system**.

Contents

1 Overview

This document describes the ServiceRegistry Core system, which exists to enable service discovery within an Eclipse Arrowhead Local Cloud (LC). An example of such interactions is a provider system offering some kind of service for use by other systems in the LC. This Core system provides the data storage functionality for the information related to the currently and actively offered services within the Local Cloud. It also stores information about the systems that offer and/or can use the previously mentioned services, and optionally data about the devices on which those systems are running.

The rest of this document is organized as follows. In Section 1.1, we reference major prior art capabilities of the system. In Section 1.2, we describe the intended usage of the system. In Section 1.3, we describe fundamental properties provided by the system. In Section 1.4, we describe delimitations of capabilities of the system. In Section 2, we describe the abstract services produced by the system. In Section 3, we describe the security capabilities of the system.

1.1 Significant Prior Art

The strong development on cloud technology and various requirements for digitization and automation has led to the concept of Local Clouds (LC).

"The concept takes the view that specific geographically local automation tasks should be encapsulated and protected." [1]

One of the main building blocks when realizing such Local Cloud is the capability of storing and maintaining information about the systems and services belonged to the given LC.

The previous versions of the ServiceRegistry (4.6.x) are very similar to this version, however there are some key differences, even in conceptual level:

- There was no data storage separation requirement: the ServiceRegistry's data storage was interconnected to other systems' storage. In the current version, data storage separation is mandatory.
- Every service definition contained exactly one operation. In the current version, a service definition can consist of several operations.
- The optional device information storage (and system information storage as well, to an extent) was handled by different systems. In the current version, the ServiceRegistry handles all three levels (service instances, systems, devices).
- Service access information (e.g. domain name or IP address) was tied to the system that provides that service, and only one such address could be specified. In the current version, service access information is tied to the interface of a service instance, and multiple addresses can be provided.
- Service interface representation was very rudimentary, a much sophisticated approach is used in the current version.
- X.509 certificate trust chains were used as authentication mechanism. The current version can support any type of authentication methods by using a dedicated Authentication Core system. However, a slightly modified version of X.509 trust chains is still supported.

1.2 How This System Is Meant to Be Used

ServiceRegistry is a recommended Core system of Eclipse Arrowhead Local Cloud and is responsible for the fundamental service discovery functionality. Systems and services (and devices) are being discoverable for other systems by consuming the services provided by ServiceRegistry Core system.

An application or other Core/Support system is required to register itself and its services into ServiceRegistry if it is designed to be a part of the Local Cloud and to unregister before shutdown or when it does not intend to continue providing its services.

1.3 System functionalities and properties

1.3.1 Functional properties of the system

ServiceRegistry solves the following needs to fulfill the requirements of service/system/device discovery.

- Enables the application and other Core/Support systems to register themselves in order to being part of the Local Cloud.
- Enables the application and other Core/Support systems to unregister themselves from the Local Cloud.
- Enables the application and other Core/Support system to register the device on which the system is running.
- Enables the application and other Core/Support system to unregister the device on which the system is running (only if no other system belongs to that device).
- Enables the application and other Core/Support systems to publish their services within the Local Cloud.
- Enables the application and other Core/Support systems to revoke their services from the Local Cloud.
- Makes the offered services/systems/devices queryable for other systems within the Local Cloud.

1.3.2 Non functional properties of the system

If an Authentication system is present in the Local Cloud, the ServiceRegistry will use its service(s) to verify a requester system before responding to its request. If X.509 authentication policy is used instead of a dedicated Authentication system, the ServiceRegistry will verify the provided certificate before any response.

1.3.3 Data stored by the system

In order to achieve the mentioned functionalities, ServiceRegistry is capable to store the following information set:

- **Device**: name, various types (e.g. MAC address) of addresses of the device and custom meta data.
- **System**: name, version, various types (e.g. IP address) of addresses of the system, an optional reference to the device on which the system is running, custom meta data.
- **Service definition**: name.
- **Interface**: name, communication protocol, optional validation data.

- **Service instance**: name, service definition and system reference, version, expiration date and time, instance level custom meta data, interface reference, security policy, and interface-related data (e. g. IP addresses, port, base path, content type, input and output data models).

1.4 Important Delimitations

- Names (device name, system name, service definition name, service instance name, interface name) must be unique in the Local Cloud.
- If the Local Cloud does not contain an Authentication system or does not using X.509 certificates, there is no way for the ServiceRegistry to verify the requester system. In that case, the ServiceRegistry will consider the authentication data that comes from the requester as valid.
- If ServiceRegistry's lookup is strict, only ”public” services (such as the *orchestration* service) can be queried directly from the ServiceRegistry by the application systems. For any ”non-public” services the application systems need to orchestrate. ”Public” services should be marked in the ServiceRegistry, for example using a service instance meta data entry.

2 Services produced

2.1 service serviceDiscovery

The purpose of this service is to lookup, publish and revoke provided services. The service is offered for both application and Core/Support systems.

2.2 service systemDiscovery

The purpose of this service is to lookup, publish and revoke systems that are part of (or want to be part of) the Local Cloud. The service is offered for both application and Core/Support systems.

2.3 service deviceDiscovery

The purpose of this service is to lookup, publish and revoke devices on which the Local Cloud's systems are running. The service is offered for both application and Core/Support systems.

2.4 service serviceRegistryManagement

Recommended service. Its purpose is to manage service definitions, service instances, interfaces, systems and devices in bulk. The different operations provide querying, registering and unregistering functionalities. The service is offered for administrative Support systems.

2.5 service monitor

Recommended service. Its purpose is to give information about the provider system. The service is offered for both application and Core/Support systems.

3 Security

For authentication, the ServiceRegistry can utilize an other core system, the Authentication system's service to verify the identities of the requester systems. If no Authentication system is deployed into the Local Cloud, the ServiceRegistry can accept X.509 certificates for identification. If none of these are available, the ServiceRegistry trusts the requester system's self-provided identity.

For authorization, in strict lookup, the ServiceRegistry only allows to lookup ”public” services. Information about all other services (including the *serviceRegistryManagement*) are only available through orchestration or using the *serviceRegistryManagement* service. Systems can only register/unregister themselves and their services/device. In open mode, information about all the services are available for lookup.

The implementation of the ServiceRegistry can decide about the encryption of the connection between the ServiceRegistry and other systems.

4 References

[1] J. Delsing and P. Varga, *Local automation clouds*. Boca Raton: Taylor & Francis Group, 2017, p. 28. [Online]. Available: https://doi.org/10.1201/9781315367897

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	