service-unregister

Service Description

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

Contents

1 Overview

This document describes the **service-unregister** service, which enables autonomous service unregistration, therefore it is an integral part of the implementation of service discovery requirements in Service Registry Mandatory Core System. Examples of this interaction is a system that offers a service through the Service Registry, but for certain reason the service should be terminated. To prevent that this service instance could be offered for other systems, the service provider system is required to remove its service from Service Registry.

The rest of this document is organized as follows. In Section 2, we describe the abstract message functions provided by the service. In Section 3, we end the document by presenting the data types used by the mentioned functions.

1.1 How This Service Is Meant to Be Used

The given service provider application system is required to use the **service-unregister** service right before, when the service instance is no longer provided. Figure 1 describes the processing of unregistration data submitted by the application system.

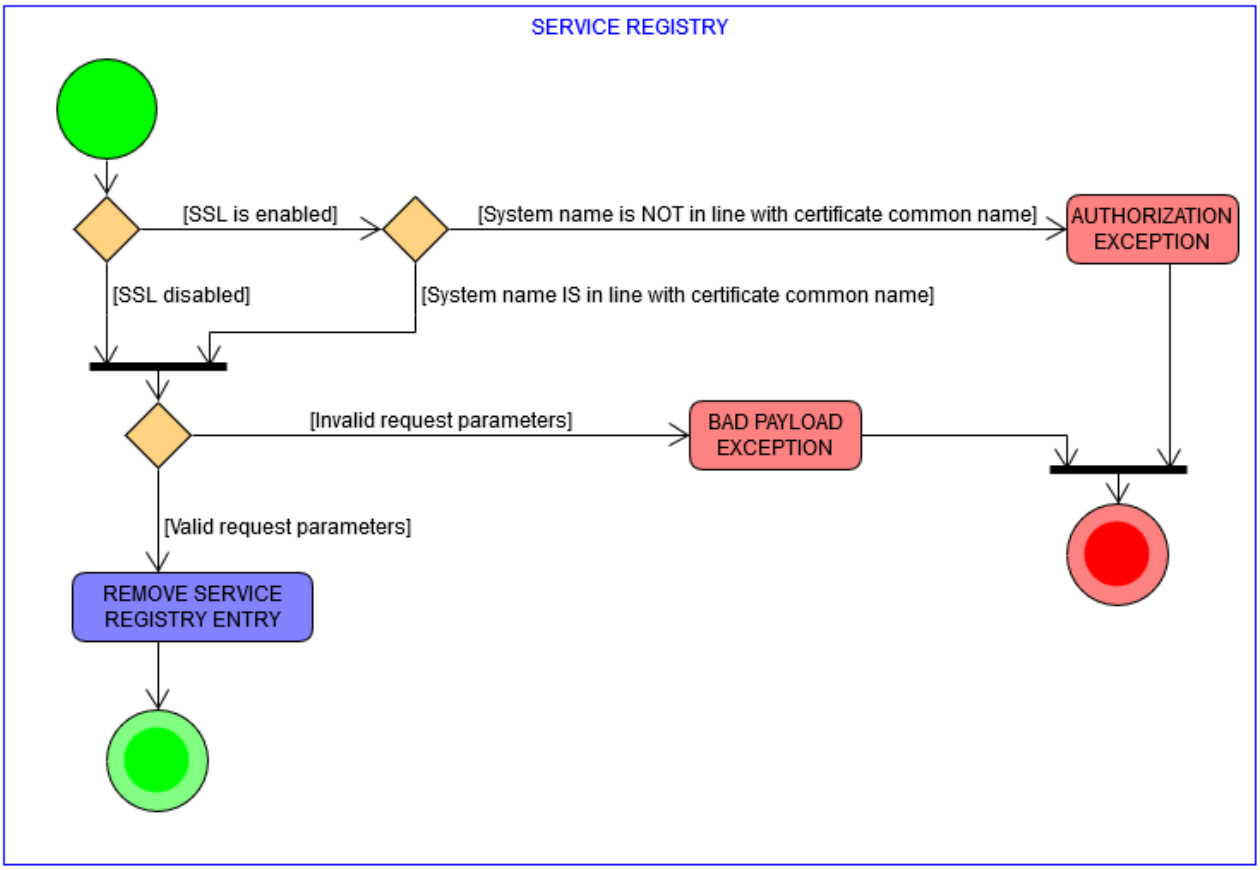

Figure 1: UML activity diagram of service unregistration process.

1.2 Important Delimitations

No delimitations.

1.3 Access policy

Available for anyone within the local cloud, but in case of secure mode service provider is allowed to unregister only its own services. It means that provider system name and system part of certificate common name must match for successful unregistration.

2 Service Interface

This section describes the interfaces to the service. The **service-unregister** service is used to unregister services. In the following, each subsection names an interface, an input type and an output type, in that order. The input type is named inside parentheses, while the output type is preceded by a colon. Input and output types are only denoted when accepted or returned, respectively, by the interface in question. All abstract data types named in this section are defined in Section 3.

The following interfaces are available.

2.1 interface HTTP/TLS/JSON (QueryParams) : void

Profile ype	Type	Version
Transfer protocol	HTTP	1.1
Data encryption	TLS	1.3
Encoding	URL	RFC 1738
Compression	N/A	-

Table 1: HTTP/TLS/JSON communication details.

3 Information Model

Here, all data objects that can be part of the **service-unregister** service provides to the hosting System are listed in alphabetic order. 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 3.2, which are used to represent things like hashes and identifiers.

3.1 struct QueryParams

Field	Type	Mandatory	Description
serviceDefinition	Name	yes	Identifier of the service.
providerName	Name	yes	Identifier of the provider system.
providerAddress	Address	no	Network address.
providerPort	PortNumber	yes	Port of the system.
serviceUri	URI	yes	URI of the service.

3.2 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
Name	A string identifier that is intended to be both human and machine-readable.
Number	Decimal number
PortNumber	Decimal number in the range of 0-65535

4 References

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	YYYY-MM-DD	4.6.0		Xxx Yyy

5.2 Quality Assurance

No.	Date	Version	Approved by
1	YYYY-MM-DD	4.6.0	