service-register

Service Description

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

Contents

1 Overview

This document describes the **service-register** service, which enables autonomous service registration, 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 has the capability to provide some kind of service. To enable other systems to use, to consume it, this service needs to be offered through the ServiceRegistry.

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-register** service at its startup in order the offered services are being discoverable for the possible consumer application systems within the local cloud. Figure 1 describes the processing of registration data submitted by the application system.

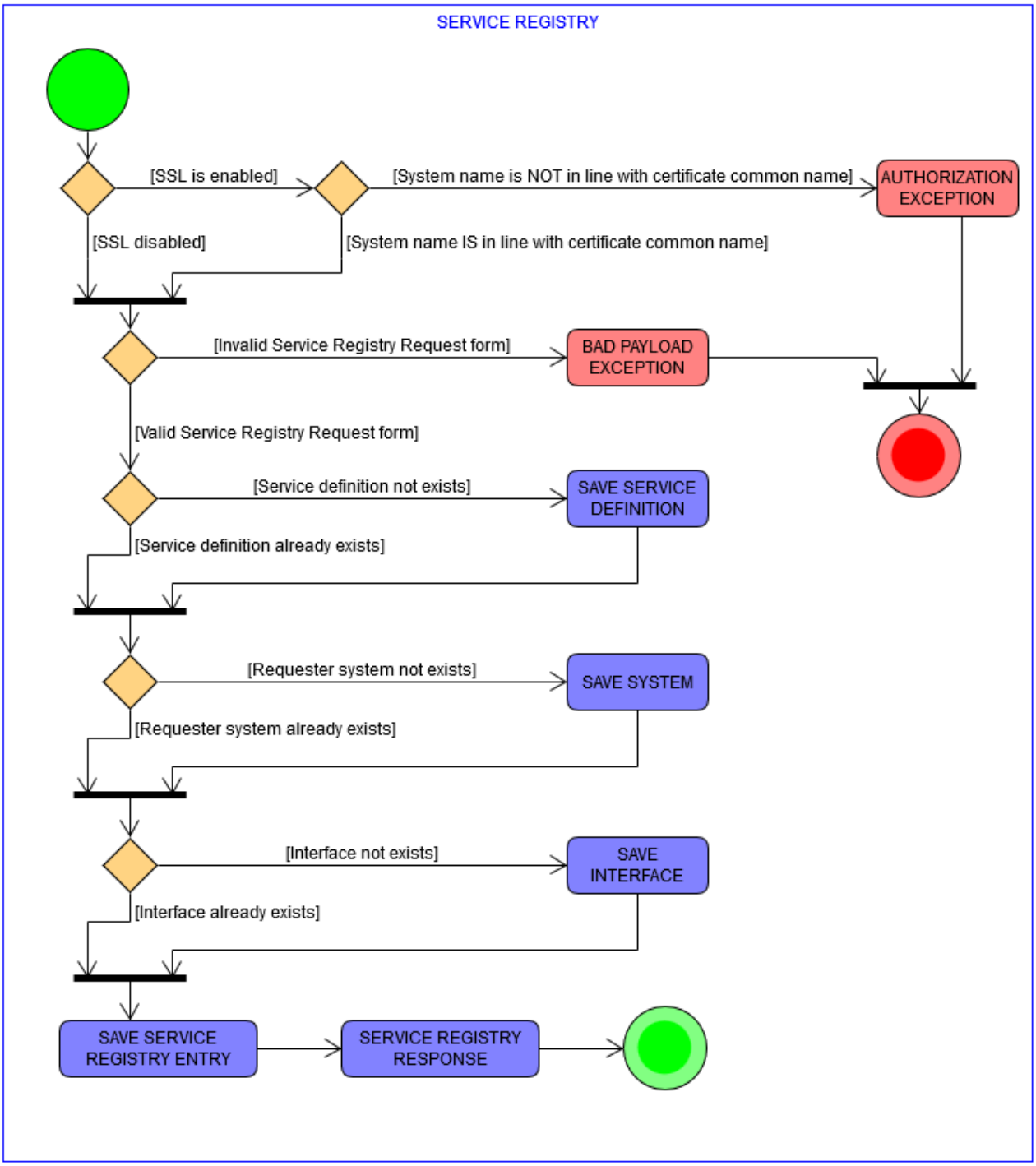

Figure 1: UML activity diagram of service registration process.

1.2 Important Delimitations

The registration data must meet the following criteria:

- Service definition can contain maximum 63 character of letters (english alphabet), numbers and dash (-), and has to start with a letter (also cannot ends with dash).
- System name can't be what is reserved for core systems.
- System name can contain maximum 63 character of letters (english alphabet), numbers and dash (-), and have to start with a letter (also cannot end with dash).
- Interface name has to follow the `Protocol-SecurityType-MimeType` format.
- Security types could be only `NOT_SECURE`, `CERTIFICATE` or `TOKEN` .

1.3 Access policy

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

Exception: Translator Supporting Core Sytem is allowed to register other services too.

2 Service Interface

This section describes the interfaces to the service. The **service-register** service is used to register services. A service could contain various metadata as well as a physical endpoint. The various parameters are representing the necessary system and service input information. In particular, 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 (ServiceRegistryRequest) : ServiceRegistryResponse

Profile ype	Type	Version
Transfer protocol	HTTP	1.1
Data encryption	TLS	1.3
Encoding	JSON	RFC 8259 [1]
Compression	N/A	-

Table 1: HTTP/TLS/JSON communication details.

3 Information Model

Here, all data objects that can be part of the **service-register** 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 3.3, which are used to represent things like hashes and identifiers.

3.1 struct ServiceRegistryRequest

Field	Type	Mandatory	Description
endOfValidity	DateTime	no	Service is available until this UTC timestamp.
interfaces	Array<Interface>	yes	List of interfaces the service supports.
metadata	Metadata	no	Metadata
providerSystem	Object	yes	Descriptor of the provider system.
secure	SecureType	yes	Type of security the service uses.
serviceDefinition	Name	yes	Identifier of the service.
serviceUri	URI	no	URI of the service.
version	Version	yes	Version of the service.

3.1.1 struct Metadata

A JSON Object which maps String key-value pairs.

3.1.2 struct providerSystem

Field	Type	Mandatory	Description
address	String	yes	Network address.
authenticationInfo	String	no	Public key of the client certificate.
metadata	Metadata	no	Metadata
port	PortNumber	yes	Port of the system.
systemName	Name	yes	Name of the system.

3.2 struct ServiceRegistryResponse

Field	Type	Description
createdAt	DateTime	Service instance record was created at this UTC timestamp.
endOfValidity	DateTime	Service is available until this UTC timestamp.
id	Number	Identifier of the service instance
interfaces	Array<Object>	List of interfaces the service supports.
metadata	Metadata	Metadata
providerSystem	Object	Descriptor of the provider system record.
secure	SecureType	Type of security the service uses.
serviceDefinition	Object	Descriptor of the serviceDefinition record.
serviceUri	URI	URI of the service.
updatedAt	DateTime	Service instance record was modified at this UTC timestamp.
version	Version	Version of the service.

3.2.3 struct interfaces

Field	Type	Description
createdAt	DateTime	Interface instance record was created at this UTC timestamp.
id	Number	Identifier of the interface instance
interfaceName	Interface	Specified name of the interface.
updatedAt	DateTime	Interface instance record was modified at this UTC timestamp.

3.2.4 struct Metadata

A JSON Object which maps String key-value pairs.

3.2.5 struct provider

Field	Type	Description
address	String	Network address.
authenticationInfo	String	Public key of the client certificate.
createdAt	DateTime	System instance record was created at this UTC timestamp.
id	Number	Identifier of the system instance
metadata	Metadata	Metadata
port	PortNumber	Port of the system.
systemName	Name	Name of the system.
updatedAt	DateTime	System instance record was modified at this UTC timestamp.

3.2.6 struct serviceDefinition

Field	Type	Description
createdAt	DateTime	Service definition instance record was created at this UTC timestamp.
id	Number	Identifier of the service definition instance
serviceDefinition	Name	Name of the service definition.
updatedAt	DateTime	Service definition instance record was modified at this UTC timestamp.

3.3 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
Boolean	One out of `true` or `false`.
DateTime	Pinpoints a specific moment in time.
Object	Set of primitives and possible further objects.
Interface	Any suitable type chosen by the implementor of service
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
SecureType	Any suitable type chosen by the implementor of service
Version	Specifies a service version.

4 References

[1] T. Bray, “The JavaScript Object Notation (JSON) Data Interchange Format,” RFC 8259, Dec. 2017. [Online]. Available: https://rfc-editor.org/rfc/rfc8259.txt

5 Revision History

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	YYYY-MM-DD	4.4.0		Xxx Yyy

5.2 Quality Assurance

No.	Date	Version	Approved by
1	YYYY-MM-DD	4.4.0	