

# service-register

## Service Description

### Abstract

This is the template for Service Description (SD document) according to the Eclipse Arrowhead documentation structure.

## Contents

<b>1 Overview</b>	<b>3</b>
1.1 Significant Prior Art . . . . .	3
1.2 How This Service Is Meant to Be Used . . . . .	4
1.3 Important Delimitations . . . . .	5
1.4 Access policy . . . . .	5
<b>2 Service Interface</b>	<b>6</b>
2.1 interface <a href="#">HTTP/TLS/JSON</a> . . . . .	6
<b>3 Information Model</b>	<b>7</b>
3.1 struct <a href="#">ServiceRegistryRequest</a> . . . . .	7
3.2 struct <a href="#">ServiceRegistryResponse</a> . . . . .	7
3.3 Primitives . . . . .	7
<b>4 References</b>	<b>8</b>
<b>5 Revision History</b>	<b>9</b>
5.1 Amendments . . . . .	9
5.2 Quality Assurance . . . . .	9



ARROWHEAD

Document title  
**service-register**  
Date  
**2022-01-31**

Version  
**4.4.0**  
Status  
**RELEASE**  
Page  
**3 (9)**

# 1 Overview

This document describes the **service-register** service, which enables autonomous service registration. 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 Significant Prior Art

Describe significant prior art which provides the foundation for the service - May be omitted for simple services

## 1.2 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.

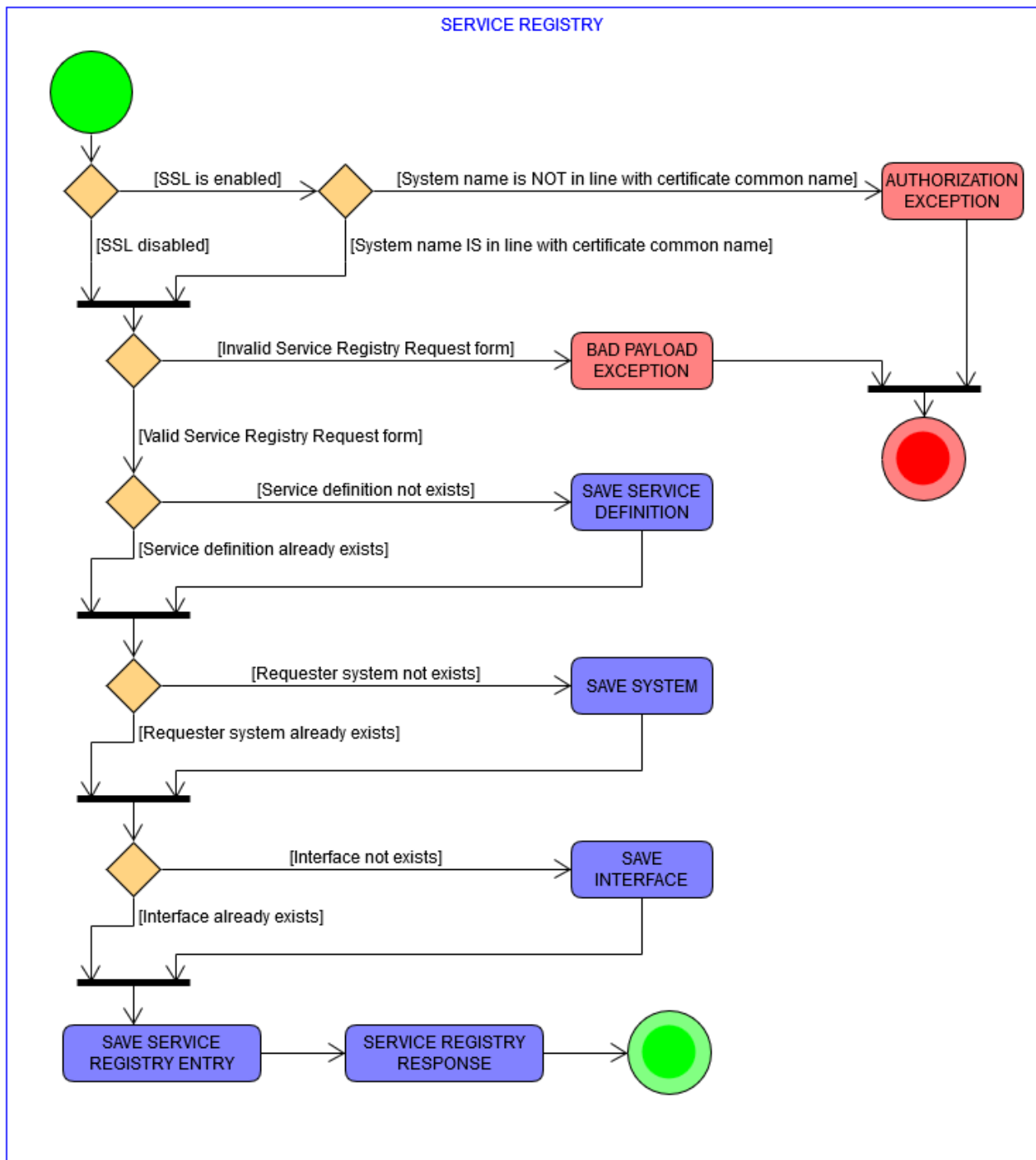


Figure 1: UML activity diagram of service registration process.

### 1.3 Important Delimitations

No delimitations.

### 1.4 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 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	e.g. JSON	RFC 8259 [1]
Compression	N/A	-
Semantics	e.g. SensML	RFC 9100 [2]
Ontology	N/A	-

Table 1: Communication and semantics details.

## 3 Information Model

Here, all data objects that can be part of the **ServiceRegister** 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	Description
endofValidity	DateTime	Service is available until this UTC timestamp.
interfaces	Array<Interface>	List of interfaces the service supports.
metadata	Metadata	Metadata
providerSystem	Object	Descriptor of the provider system.
secure	SecureType	Type of security the service uses.
serviceDefinition	Name	Identifier of the service.
serviceUri	URI	URI of the service.
version	Version	Version of the service.

### 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<Interface>	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.

### 3.3 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 <code>true</code> or <code>false</code> .
DateTime	Pinpoints a specific moment in time.
Object	Set of primitives and possible further objects.
Interface	Any suitable type chosen by the implementor of the service.
List<A>	An <i>array</i> 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
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>
- [2] C. Bormann, "Sensor Measurement Lists (SenML) Features and Versions," RFC 9100, Aug. 2021. [Online]. Available: <https://rfc-editor.org/rfc/rfc9100.txt>





ARROWHEAD

Document title  
**service-register**  
Date  
**2022-01-31**

Version  
**4.4.0**  
Status  
**RELEASE**  
Page  
**9 (9)**

## 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	