ServiceDiscovery

Service Description

Abstract
This document describes the v4.3.0 of the ServiceDiscovery service as produced by the ServiceRegistry system.

ARTEMIS Innovation Pilot Project: Arrowhead
THEME [SP1-JTI-ARTEMIS-2012-AIPP4 SP1-JTI-ARTEMIS-2012-AIPP6]
[Production and Energy System Automation Intelligent-Built environment and urban infrastructure for sustainable and friendly cities]

Contents

1 Overview

This document describes the ServiceDiscovery service, which is 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.

2 Service Interface

This section describes the interfaces to the ServiceDiscovery service. In particular, each subsection names an abstract operation, 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.

Figure 1: SysML block description diagram of the ServiceDiscovery and its interface

The folowing interface operations are available.

2.1 function Register (ServiceRegistryRequest)

The Register operation is used to register services. The services will contain various metadata as well as a physical endpoint. The various parameters are representing the endpoint information that should be registered.

2.2 function Unregister (ServiceRegistryUnregisterRequest) : StatusCodeKind

The Unregister operation is used to de-register services. This is used to provide a graceful way of telling other systems that the service is no longer available.

2.3 function Query (ServiceQueryForm) : ServiceQueryList

The Query operation enable search for a specific service according to a search criteria.

2.4 function Echo () : StatusCodeKind

The Echo operation provides an is alive response from the ServiceDiscovery service.

3 Information Model

Here, all data objects that can be part of Service Discovery Register service calls 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 3.4, which are used to represent things like hashes and identifiers.

An overview of the data object types is illustrated in Figure 2.

3.1 struct ServiceRegistryRequest

This structure is used to register a service offering into the Service Registry.

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	Name	Name of the provider system.
secure	SecureType	Type of security the service uses.
serviceDefinition	Name	Service Definition.
serviceUri	URI	URI of the service.
version	Version	Version of the service.

3.2 struct ServiceRegistryUnregisterRequest

This structure is used to register a service offering into the Service Registry.

Field	Type	Description
address	Address	Address of the provider systems.
port	PortNumber	Port of the provider system.
system_name	Name	System name of the provider system
service_definition	Name	Service Definition of the unregistered service.

3.3 struct ServiceQueryForm

This structure is used to register a service offering into the Service Registry.

3.4 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.

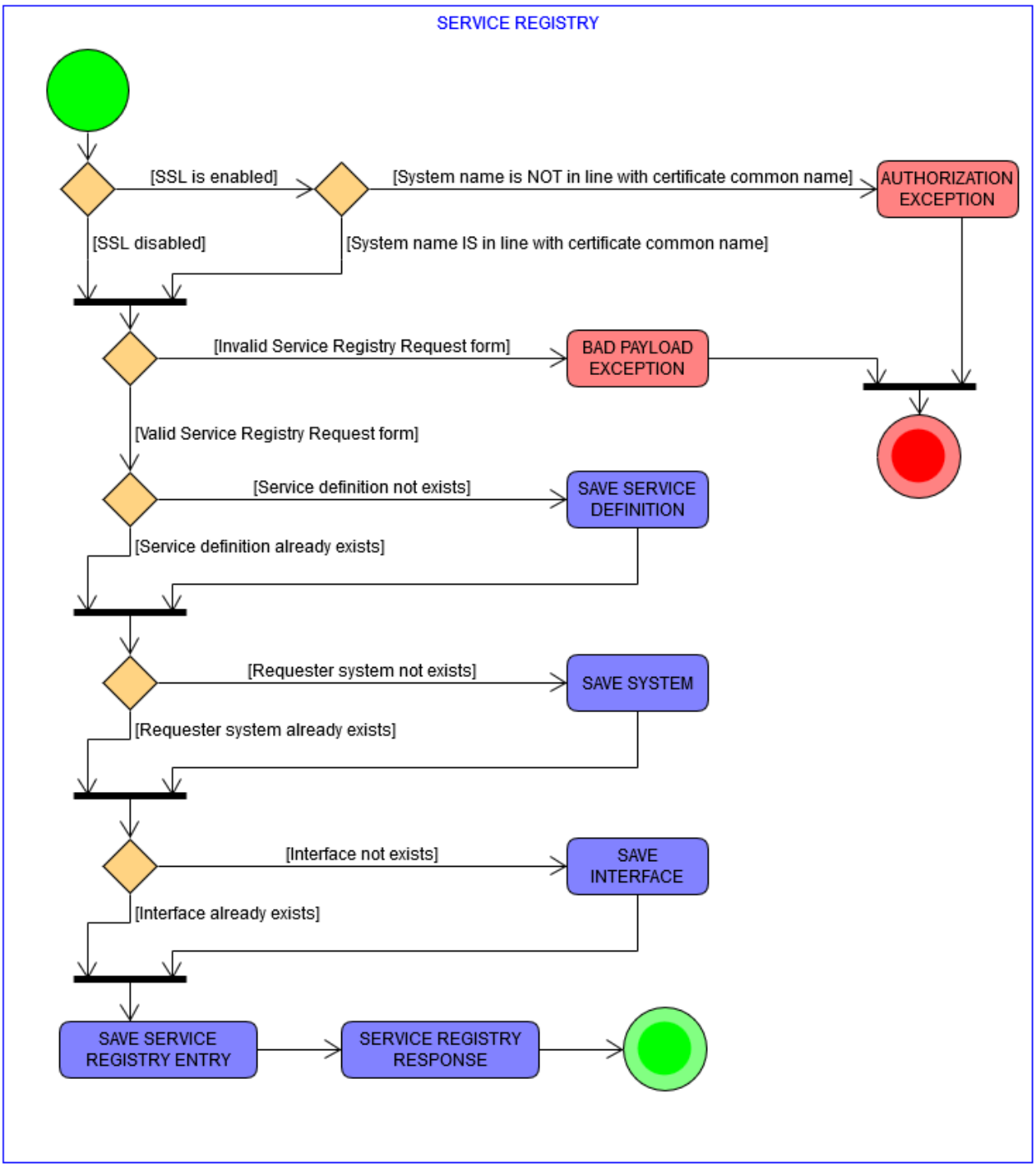

Figure 2: Information model as a UML activity diagram. Describes the process of service registration.

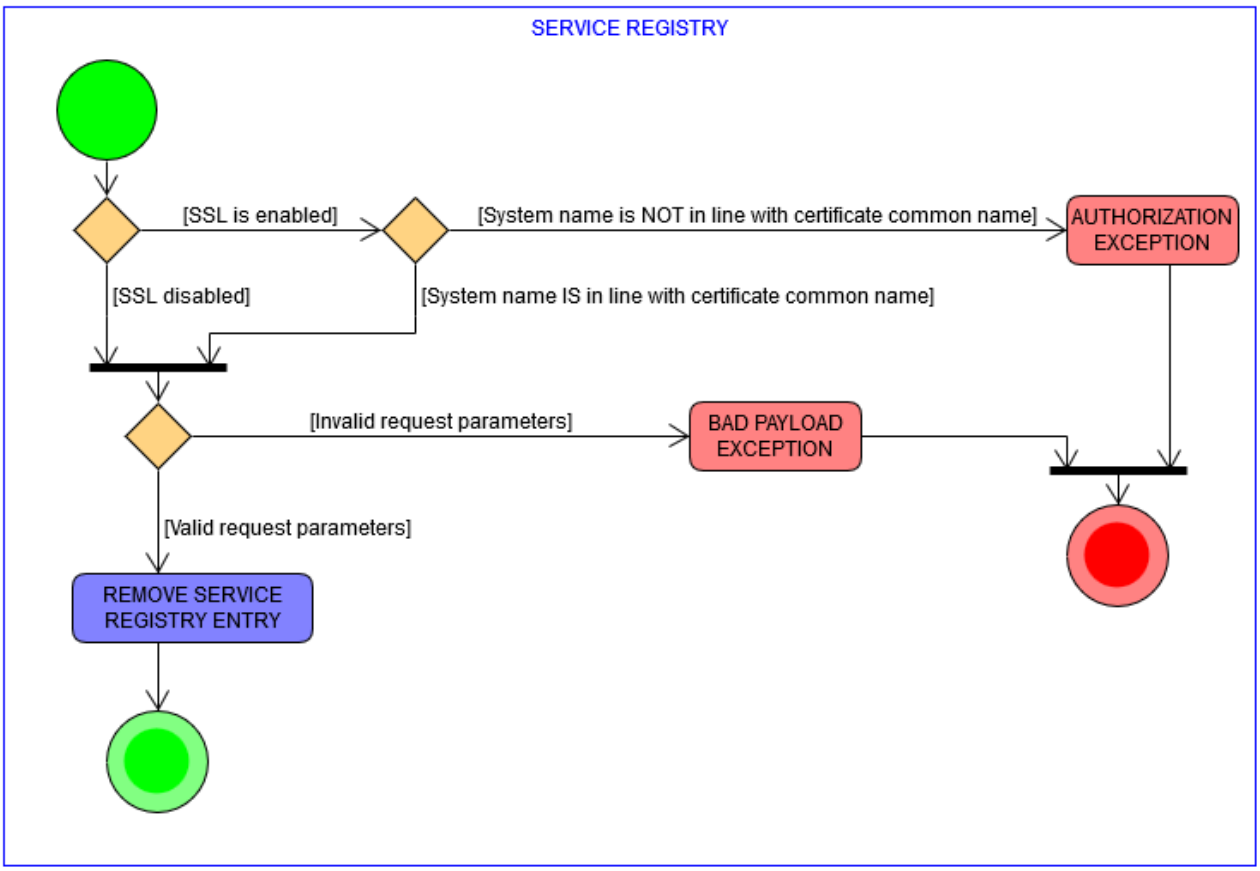

Figure 3: Information model as a UML activity diagram. Describes the process of service unregistration.

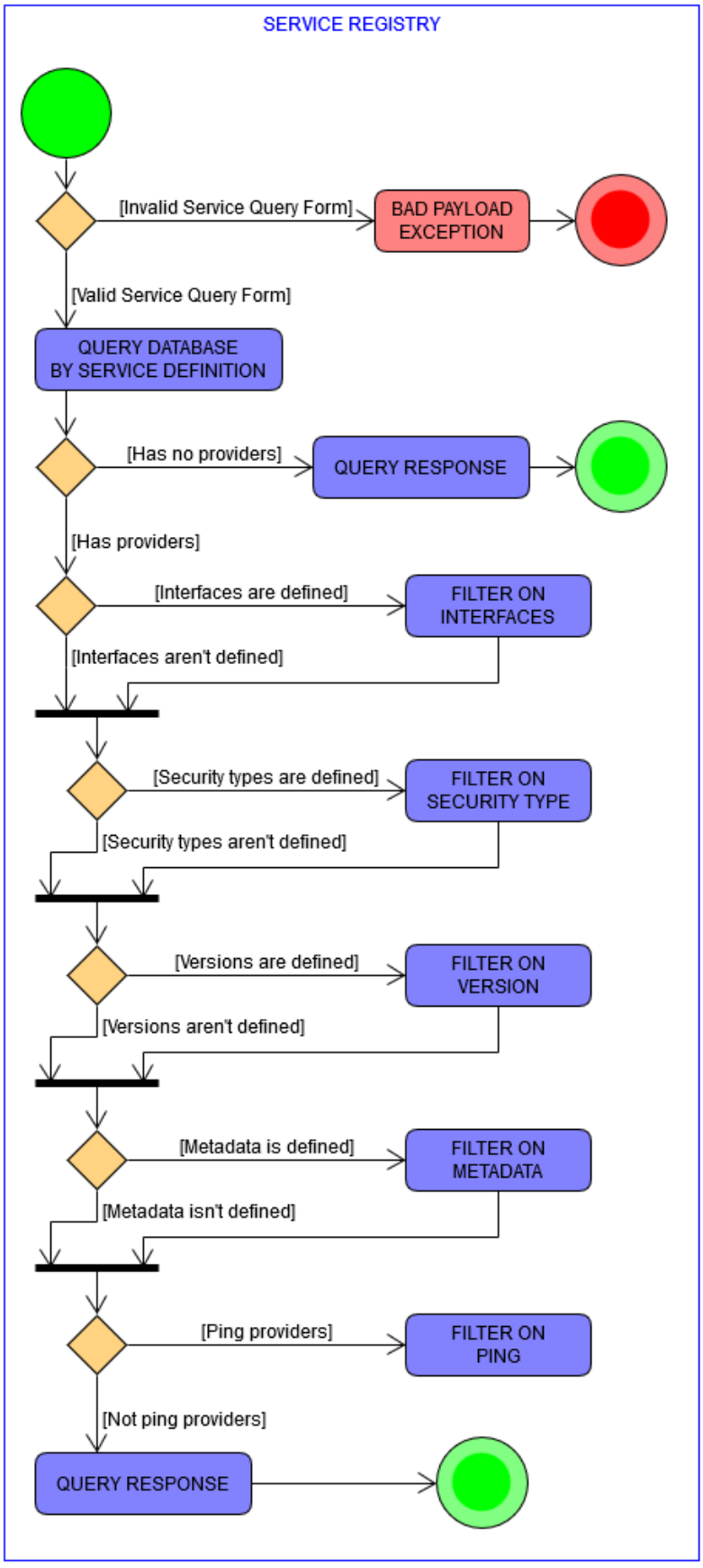

Figure 4: Information model as a UML activity diagram. Describes the process of service querying.

Object Field	Value Type	Description
”interfaceRequirements”	Array<Interface>	List of the required interfaces.
”maxVersionRequirement”	Version	Maximum version.
”minVersionRequirement”	Version	Minimum version.
”metadataRequirements”	Metadata	Metadata.
”pingProviders”.	Boolean	Checks the availability of the providers if true
”securityRequirements”	Name	Type of security.
”serviceDefinitionRequirement”	Name	Service Definition.
”versionRequirement”	Version	Version of the service.

Type	Description
Address	A string representation of the address
Boolean	One out of `true` or `false`.
Interface	Any suitable type chosen by the implementor of the service.
DateTime	Pinpoints a specific moment in time.
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.
PortNumber	Decimal number in the range of 0-65535
Version	Specifies a service version.

4 Revision History

4.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	2020-12-05	1.0.0		Tanyi Szvetlin

4.2 Quality Assurance

No.	Date	Version	Approved by
1			