ServiceDiscovery Service SD

**Abstract**

This document defines the ServiceDiscovery service functionality.

1. Service Description Overview

This document describes an Arrowhead service, including its interfaces, functions and information model.

1. Abstract Interfaces

This Service provides three functionalities.

# Register / Publish

The publish method 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.

# Unregister / Unpublish

The unpublish method is used to unregister service instances that were previously registered in the Registry.

The instance parameter is representing the endpoint information that should be removed.

# Query / Lookup

The Lookup method is used to find and translate a symbolic service name into a physical endpoint, for example an IP address and a port.

The query parameter is used to request a subset of all the registered services fulfilling the demand of the user of the service.

The returned listing contains service endpoints that have been fulfilling the query.

1. Abstract Information Model

A ServiceRegistry entry contains the following information, as presented in Table 1. This is the payload that needs to be sent when registering or removing an entry from the Registry.

Table 1 ServiceRegistryEntry type description

|  |  |
| --- | --- |
| **Field** | **Description** |
| ProvidedService: ArrowheadService | The Arrowhead Service object that is provided (SD and supported IDD-s). |
| Provider: ArrowheadSystem | ArrowheadSystem that is providing the Service. |
| Port: Integer | The port where the provided service can be consumed. |
| ServiceURI: String | The subpath of the Service within the host (address:port) of the Provider. This can be a REST URL or MQTT topic name for example. |
| Version: Integer | The version numbering of the Service provided. |
| UDP: Boolean | True, if the service is provided via UDP. By default, it is false. |
| Ttl: Integer | Service validity time period expressed in seconds. This gets converted to a date-time, and stored in the database. |
| Metadata: String | Metadata belonging to a service/provider pair. |

Metadata key-values are stored in a single string, with the following syntax:

***“<key1>=<value1>,<key2>=<value2>,<key3>=<value3>”***

Services that can only be consumed in a secure manner (via HTTPS) have a mandatory “security” metadata key, with possibles values: “certificate” or “token”. The presence of this metadata is required by this generation of the Arrowhead framework, to facilitate the orchestration process.

The other payload is related to querying (lookup) of the Registry contents.

Table 2 ServiceQueryForm type description

|  |  |
| --- | --- |
| **Field** | **Description** |
| Service: ArrowheadService | The Arrowhead Service object that is looked for (SD and supported IDD-s) |
| MetadataSearch: Boolean | True, if service metadata shall be taken into account during query. |
| PingProviders: Boolean | True, if the selected (query fitting) Application Systems shall be pinged to see if they’re alive. |
| Version: Integer | The minimum version of the Service we are looking for. |

When this Service is accessed via DNS, the following naming conventions ensure the interoperability with the Arrowhead framework:

1. Entry naming (SRV record):

***<systemName>.***

***\_ahf-<serviceDefinition>\_<interface>.***

***\_<transport>.***

***<cloudName>.<operator>.<topLevelDomain>***

2. TXT field is to filled out the following way:

* “ahsysname” = <systemName>
* “ahsysauth” = <authInfo>
* “path” = <serviceURI>
* Service metadata key-values: “ahsrvmetad\_<Key>” = “<Value>”
* “txtvers” = “<version>”

1. Non-functional Requirements

The Service Registry must be available first, during Core System start. This way, the other Core Systems can register their services in the Registry.

1. Revision history

# Amendments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Date | Version | Subject of Amendments | Author |
| 1 | 2017-09-29 | M2 | Initial | Csaba Hegedűs |
| 2 | 2018-05-23 | G4.0 | Updated to G4.0 | Zoltán Umlauf |

# Quality Assurance

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Date | Version | Approved by |
| 1 |  |  |  |
| 2 |  |  |  |