Arrowhead Python Library IDD

**Abstract**

This document defines interface design of the Arrowhead Python Library within Arrowhead generation 4.0.

1. Introduction

This document describes the Arrowhead Python Library of G4.0 with the REST interface.

This interface uses HTTP.

1. Interfaces

As per the SD of this Service, there is one method implemented. Table 1 describes this method.

Table 1 Function description

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Input** | **Output** | **Description** |
| Provider | start | - | - | Publish service to service registry |
| Provider | stop | - | - | Unpublish service from the service registry |
| Consumer | consume\_service | providerUrl | Json Data | Consumes a provider |
| Orchestrator | register\_to\_orchestrator | Provider,  consumerSystemName,  consumerAddress,  consumerPort,  authenticationInfo | True False | Set a rule which consumers can use the provider |
| Orchestrator | orchestration\_request | Consumer, serviceDefinition | URL | Finds a provider with the serviceDefinition given |
| Authorization | register\_to\_authorization | Provider,  consumerSystemName,  consumerAddress,  consumerPort,  authenticationInfo | - | Sets a rule which consumer is allowed to use the provider. |
| Authorization | authorize | Consumer,  providerName,  providerAddress,  providerPort,  serviceDefinition | True,  False | Validates if a consumer is allowed to consume a specific service. |
| ServiceFinder | get\_insecure\_service\_registry | - | URL | Returns the URL and Port where the service registry is located. |
| ServiceFinder | get\_insecure\_orchestrator | - | URL | Returns the URL and Port where the orchestrator is located. |
| ServiceFinder | get\_insecure\_authorization | - | URL | Returns the URL and Port where the authorization is located. |

1. Information Model

# 3.1. Publishing service to service registry

|  |
| --- |
| import Provider  provider = Provider.Provider("CurrentTimeSweden",  "CurrentTime",  "/",  8080,  "127.0.0.1",  ["JSON"] ,  "127.0.0.1:8442",  {})  provider.start() |

# 3.2. Unpublishing service from service registry

|  |
| --- |
| import Provider  provider = Provider.Provider("CurrentTimeSweden",  "CurrentTime",  "/",  8080,  "127.0.0.1",  ["JSON"] ,  "127.0.0.1:8442",  {})  provider.start()  provider.stop() |

# 3.3. Consume service

|  |
| --- |
| import Consumer  consumer = Consumer.Consumer("test\_consumer", "127.0.0.1", 8081)  consume.consumeService(“127.0.0.1:8080”) |

# 3.4. Register service to orchestrator

|  |
| --- |
| import Provider  import Orchestrator  provider = Provider.Provider("CurrentTimeSweden",  "CurrentTime",  "/",  8080,  "127.0.0.1",  ["JSON"] ,  "127.0.0.1:8442",  {})  Orchestrator.register\_to\_orchestrator(provider,  "test\_consumer",  "127.0.0.1",  8081,  "null") |

# 3.5. Orchestration request

import Consumer

import Orchestrator

consumer = Consumer.Consumer("test\_consumer", "127.0.0.1", 8081)

Orchestrator.orchestration\_request(consumer, "CurrentTimeSweden")

# 3.6. Register to authorization

|  |
| --- |
| import Authorization  import Provider  provider = Provider.Provider("CurrentTimeSweden",  "CurrentTime",  "/",  8080,  "127.0.0.1",  ["JSON"] ,  "127.0.0.1:8442",  {})  Authorization.register\_to\_authorization(provider,  “test\_consumer”,  “127.0.0.1”,  8080,  {} ) |

# 3.7. Authorize

|  |
| --- |
| import Authorization  import Consumer  consumer = Consumer.Consumer("test\_consumer",  "127.0.0.1",  8081)  Authorization.authorize(consumer,  “CurrentTimeSweden”,  “127.0.0.1”,  8080,  CurrentTime) |

1. Revision history

# Amendments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Date | Version | Subject of Amendments | Author |
| 1 | 2017-10-02 | M2 | Initial | Csaba Hegedűs |
| 2 | 2018-02-14 | M3 | Updating document to M3 | Csaba Hegedűs |
| 3 | 2018-05-22 | G4.0 | Updated document to G4.0 | Zoltán Umlauf |

# Quality Assurance

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