Version: 0.2
Status: for approval
Date: 202012-10
Author: Gustav Hansson
Contact: gushan-6@student.ltu.se

Interface Design Description (IDD) - Producer

Abstract

This document defines the template for the Interface Design Description, IDD, of Arrowhead compliant Interfaces. The document outlines the structure that should be followed when documenting the design of the interfaces exposed by a concrete realization of an Arrowhead Service Description (SD).

An IDD provides a detailed description of how the service is implemented/realized by including the communication profile and the chosen technologies.

An IDD is distinct from its corresponding SD in that it is a white-box rather than block-box description, outlining how each of the abstract interfaces of its SD are realized using a particular Communication Profile (CP). In other words, an IDD describes its interfaces in terms of a single transport protocol stack, encoding and, potentially, standardized semantics (?), while an SD describes its interfaces in abstract terms.

All Arrowhead Interface Designs should be specified using this template and stored on the common repository (github.com/arrowhead-f), in order to document and formalize the Arrowhead systems.

Author: Gustav Hansson
Contact: gushan-6@student.ltu.se

1. Interface Design Description Overview

Table 1 Pointers to SD documents

Realised Service Description	Location
Producer	SD Service Description Producer.docx

The demo producer service uses JSON encoding.

Layer	Protocol
Application Layer	HTTP
Transport Layer	TCP
Internet Layer	IP

2. Service Interfaces

This service provides a single function using a REST api endpoint for demonstrating the fetch of data from the Producer.

2.1. REST API

The REST API is implemented using the Grapevine REST server library for C#.

Table 4 INTERFACE description

Interface name	Relative URL path	Method	Input	Output

Get Demo data	/demo	GET	n/a	*message*

The Input and Output fields are references to data objects described in the data model.

2.1.1.Data Model

The interface does not take any input. The response message is a JSON encoded message containing the timestamp of the request as well as a random value for demonstration purposes.

Response message example:

```
{
    ”timestamp”: string,
    ”value”: integer
}
```

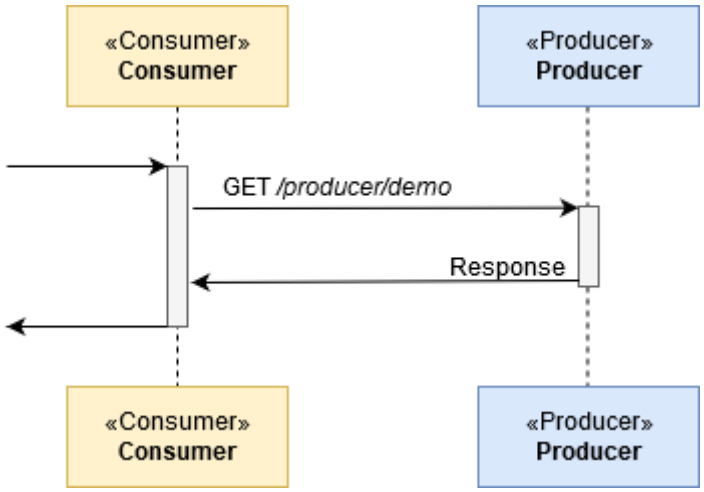
Table 5 Response *message Description*

Object Field	Description	Format/limitations
Timestamp	The timestamp of when the Producer received the request from the Consumer	CHECK THIS SUTFF
Value	Random integer for demo purposes	Integer between 0 and 100

2.1.2.Status and Error handling

Status	Description
200	A successful GET request to the REST API

2.1.3.Interaction with consumers



3. Security

For a Consumer to be able to start orchestration the produced service intracloud rules must be correctly configured in the Authenticator. No other security mechanisms are implemented.

3.1. Certificate

For registering the Provider Service in the Service Registry, a correctly created client certificate must be used. This certificate is generated using RSA and signed by the cloud certificate.

4. References

5. Revision history

5.1. Amendments

No.	Date	Version	Subject of Amendments	Author
1	2020-12-10	0.1	Initial Draft	Gustav Hansson
2	2020-12-29	0.2	Fix Table of Contents	Gustav Hansson

5.2. Quality Assurance

No.	Date	Version	Approved by
1			
2			