Interface Design Description (IDD) Template

**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.

1. Interface Design Description Overview 3
2. Service Interfaces 3
   1. Interface 1 4
      1. Data Model 4
      2. Status and Error handling 5
      3. Interaction with consumers 5
   2. Interface 2 5
      1. Information Model 6
      2. Status and Error handling 6
3. Security 6
   1. Certificate 6
   2. Payload protection 7
   3. Audit of service exchange 7
4. References 7
5. Revision history 8
   1. Amendments 8
   2. Quality Assurance 8

## Interface Design Description Overview

This section shall contain path to associated SD document.

**Table 1 Pointers to SD documents**

|  |  |
| --- | --- |
| **Realised Service Description** | **Location** |
| wm-data | documentation/SD\_wm-data.docx |

This section shall outline the higher-level technical aspects of the interface design that must be known in order to have an Arrowhead system successfully consume it. Examples of such aspects include:

* its application-level transport protocol,
  + e.g. CoAP v x.y
* the message payload encodings it supports,
  + XML v x.y
* what message payload compressions schemas are supported,
  + e.g. EXI v x.y and schema …
* what metadata it advertises via the service registry
  + e.g. service\_type=temperature, response\_time=1s, …

## Service Interfaces

This section describes, for a produced service, the different interfaces of the Service, where each interface can be thought of as one kind of operation the service can be asked to perform, or one function it can execute on behalf of consuming systems. Every service interface shall be described in terms of how it’s implemented.

Each service interfaces must be described in a separate subsection such that a consuming system correctly can interpret the received information.

The use of the following diagrams is proposed for representing interface behaviour:

* SysML/UML Sequence diagram
* SysML/UML Activity diagram

If it is considered necessary to define the structure, these diagrams can be a choice:

* SysML Parametric diagram
* UML Class diagram
* UML Component diagram

## Interface 1

Description of service interface with details on how to consume this interface using the protocols and parameter semantics defined and implemented. The exact details that will need to be recorded here will vary for different application-level transport protocols.

The function executed by consuming this service interface shall be presented and explained. Sequence diagrams are recommended to give a clearer view on functions’ usage.

**Table 4 INTERFACE description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Interface name** | **Relative URL path** | **Method** | **Input** | **Output** |
| wm-data | /wm-data | HTTPS | next | ProviderDTO |

The above example table assumes that HTTP or CoAP is the transport protocol subject of this IDD. Note that a relative URL path is to be documented, as HTTP and CoAP services are assumed to always specify a certain URL base path in their metadata, which allows multiple services to be provided by the same system over the same port number.

Each operation which is described in Table 4 is required to be pointed to the correct interface for the protocol used.

Libraries etc. use for the implementation shall be documented. This includes for the implementation necessary metadata.

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

## Data Model

Data model for the service interface input and output shall be described.

* Data model,
  + Semantics definition eventually based ontology xxx
    - Any schema used, such as XML Schema or JSON Schema
  + Encoding, e.g. XML, JSON, CBOR.

Example of input and output data shall be provided as examples with tables explaining the data parameters used. What fields are included in the tables will vary depending on the concrete encoding used by the interface design described the the IDD. The below tables assumes that JSON is the encoding in question.

**Table 5 Data Object description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Object Field** | **Description** | **Format/limitations** | **Required** |
| timeStamp | Time of measurement | Double | Yes |
| Speed | Speed at point of measurement | Double | Yes |
| accelerometer | Timeseries of Gs measured | Double | Yes |

UML or SysML can be used to describe the relation of data format and specifications.

If any metadata is available, they shall be included in this section.

## Status and Error handling

Describe implemented status/error messages.

## Interaction with consumers

Provide UML/SysML sequence diagrams for how a consumer interacts with the produced service.

## Interface 2

…..

**Table 6 Function description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Interface function** | **Sub endpoint** | **Method** | **Input** | **Output** |
|  |  |  |  |  |

## Information Model

……..

**Table 7 Function description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Explanation** | **Required** |  |  |
|  |  |  |  |  |

## Status and Error handling

…..

.

.

.

## Security

What security mechanisms are implemented for the produced service.

* Authentication and Authorisation of consumer  
  Describe the interaction with the Authorisation system
* Audit of service exchange  
  Describe how the audit of service exchanges is expected to be made, e.g. usage of the DataManager to store a copy of the service exchanges made.
* Payload encryption and algorithm used

## Certificate

* + Which certificate type is valid for the service:
    - Application
    - Core
    - Management
  + How the certificate is generated and computed:
    - Which cryto-algorithm/s that is/are implemented, e.g. RSA

## Payload protection

* + Which payload protection is used.   
    Document used payload encryption algorithms used, e.g. TLS.

## Audit of service exchange

Document how the audit of service exchanges is expected to be made, e.g. usage of the DataManager to store a copy of the service exchanges made.

## References

Any references must be placed here.

## Revision history

## Amendments

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Date | Version | Subject of Amendments | Author |
| 1 | 20-2-2015 | 1.0 | Revision of text | Michele Albano / Luis Ferreira |
| 2 | 30-09-2015 | 1.1 | Refinement of the structure | Michele Albano / Luis Ferreira |
| 3 | 2020-06-10 | 2.0 | Major update | Jerker Delsing |
| 4 | 2020-06-15 | 2.1 | More verbose help text | Emanuel Palm |
| 4 | 2020-08-17 | 4.2 | Cleaning and minor update | Jerker Delsing |

## Quality Assurance

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