Translator

System Design Description (SysDD)

Abstract
This document describes a system capabale of providing translation services for Eclipse Arrowhead-compliant systems. The Translator system provides features for producers and consumers of data to communicate in a heterogenoues networking environment consisting of multiple prototocols and technologies.

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 **3**
1.1 Status of this Document 3

2 Important Delimitations **4**

3 System Role **4**
3.1 Data models 4

4 Services **5**
4.1 Consumed Services 5
4.2 Produced Services 5

5 References **6**

6 Revision History **7**
6.1 Amendments 7
6.2 Quality Assurance 7

1 Overview

This document describes the Translator of the Eclipse Arrowhead [1] system, which provides services for translation to enable produces and cosumers from different protocols to interact. Common use-cases are;

- Direct translation from CoAP to HTTP
- Direct translation from HTTP to CoAP
- Integration of FIWARE Entities into Arrowhead Service Registry
- Arrowhead Consumers can request data from FIWARE Entities
- Arrowhead Producers and their values can be injected into FIWARE request

All these scenarios are illustrated in Figure 1.

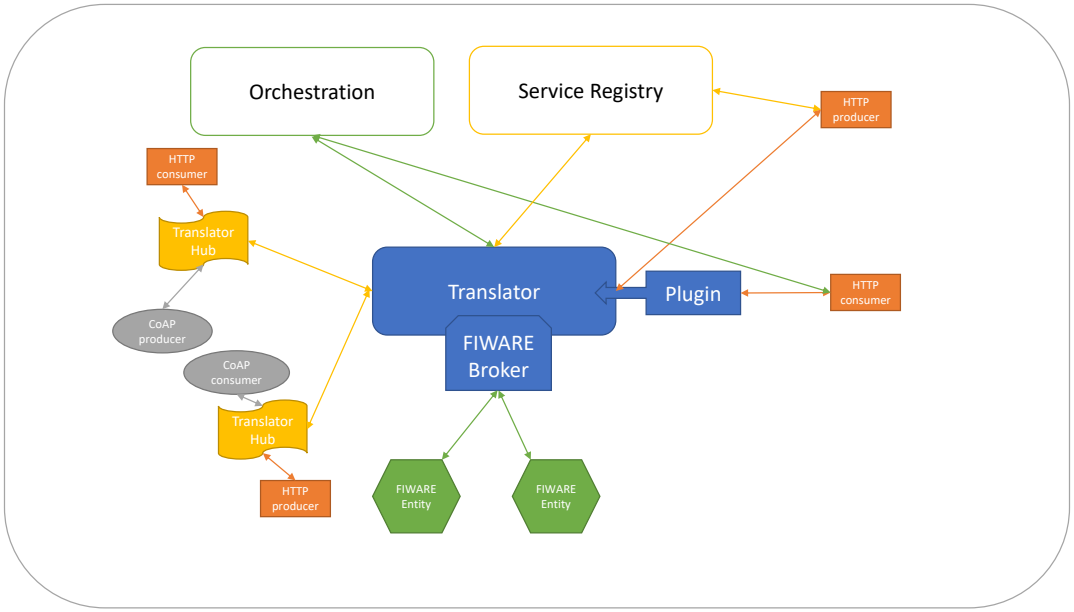

Figure 1: An example of an Arrowhead local cloud which contains the Core systems, and multiple producers and cosumers with different protocols.

The rest of this document is organized as follows. In the remainder of this section we comment on the status of this document. In Section 2, we outline major delimitations of the system, which is a work-in-progress. Section 3 presents how the Translator is used in a local cloud. Finally, In Section 4, there is a description of the multiple services that Translator consumes and produces.

1.1 Status of this Document

This document presents the current state of the Translator. However, since Eclipse Arrowhead is an actve open source project, changes will most like happen in the future. Features will be added, modified or in other ways altered. This document will thus be updated when needed.

This document, an all other such part of the same Eclipse Arrowhead Core proposal, are still to be considered early drafts and might to have to undergo several significant revisions before becoming sufficient for most kinds of industrial deployments.

If the reader has ay comments or suggestions regarding the design or implementations, please contact Pablo Puñal Pereira <pablo.punal@thingwave.eu>, who the maintainer of the Translator system.

2 Important Delimitations

The primary purpose of the Translator system is to provide services and features to enable the communication between different protocols. Currently, the following features are out of scope;

- Direct translation from CoAP to HTTP
- Direct translation from HTTP to CoAP
- Integration of FIWARE Entities into Arrowhead Service Registry
- Arrowhead Consumers can request data from FIWARE Entities
- Arrowhead Producers and their values can be injected into FIWARE request

3 System Role

As stated in Section 1, the Translator system performs three main roles. Firstly, enables direct translation CoAP-HTTP, secondly, injects services and data into FIWARE request, and thirdly, injects into Arrowhead FIWARE entities creating virtual services replicating the FIWARE data.

One direct translation use case can be the following:

1. The Orchestrator or an authorized user, can request a new Tranlation Hub for a consumer producer with different protocols (CoAP or HTTP).
2. The Translator creates a Hub that acts as a proxy between both entities.
3. The consumer consumes the resource of the hub.
4. The Hub consumes the resource of the producer and replicates the data back to the consumer.

One FIWARE injection use case can be the following:

1. A FIWARE client request the list of Entities.
2. The FIWARE Broker subservice forwards the request to the real FIWARE Broker and stores the response. At the same time request to the Orchestrator the list of Systems that offer services that matches the type of the request.
3. When the Translator receives the list of Systems and Services from the Orchestrator, it converst each System into a FIWARE entity, request its values and injects the result into the FIWARE response from the real FIWARE broker.

One injection into Arrowhead from FIWARE use case can be the following:

1. The Translator, after booting and periodically, request the list of FIWARE Entities to the real FIWARE broker.
2. Each Entity that has not been register into the Service Registry, a virtual service is created on the translator and it is registered on the Service Registry.
3. When the Orchestrator gives any of the registered Services as the one to use, the consumer will connect to the virtual service that will read the values of the FIWARE Entity and will return it as plain text, SenML or JSON (the consumer can select the Media Type).

3.1 Data models

The table below shows the different encodings that can be used with SenML [2]. Today, only plain text, SenML and JSON is used.

Encoding	Description
JSON	Standard JSON. Pros; easy to work with. Cons: relatively verbose.
XML	Standard XML. Pros; Lots of tools, Cons: very verbose.

4 Services

The Translator system produces and consumes the following services, as described in Figure 1.
More details regarding the consumed and produced services are given in the following subsection.

4.1 Consumed Services

This section presents an overview of consumed services.

4.1.1 Service Registry

This service is consumed to register FIWARE Entities into Arrowhead.

4.1.2 Orchestration

This service is consumed to request available producers that can be injected on the FIWARE requests.

4.2 Produced Services

This section presents an overview of provided services.

4.2.1 Echo

The Echo service is a simple service that can be used to determine if a system is available or not.

4.2.2 Translator

The Translator service provides three (3) subservices Translation, Plugin and FIWARE Broker.

- Translation subservice offers a direct translation between CoAP to HTTP and HTTP to CoAP.
- Plugin subservice offers to consume FIWARE Entities data from a standard HTTP Arrowhead Consumer.
- FIWARE Broker is a proxy broker that can inject Arrowhead data into the responses from the real FIWARE broker.

5 References

[1] J. Delsing, “Iot automation : Arrowhead framework,” 2017.

[2] C. Jennings and Z. Shelby, “Sensor Measurement Lists (SenML),” RFC 8428, 2018, RFC Editor. [Online]. Available: https://doi.org/10.17487/RFC8428

6 Revision History

6.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	2019-03-27	1.0	Initial	Pablo Puñal Pereira
2	2019-05-02	1.1	Models and Interfaces update	Pablo Puñal Pereira
3	2020-12-11	2.0	Template Update	Pablo Puñal Pereira

6.2 Quality Assurance

No.	Date	Version	Approved by
1			