

System-of-Systems Description (SoSD)

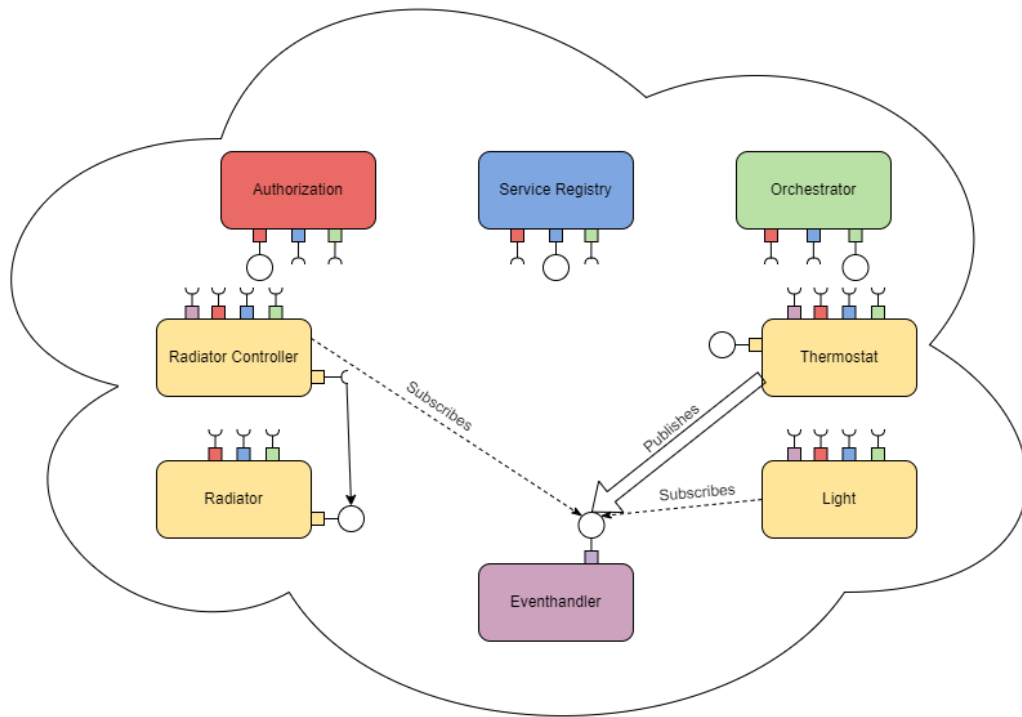
4 januari 2024

1	System of Systems Overview	2
2	Systems	2
3	Use-cases	3
4	Security	3

1 System of Systems Overview

This document describes the Automatic Temperature System, which exists to automatically control the temperature in a space and give indications through a light whether or not a temperature is above a certain degree. The system is built on Arrowhead framework which provides core and support systems: authorization, service registry, orchestrator and eventhandler. The system contains four application systems: thermostat, light, radiator controller and radiator. The thermostat provides a service to get data (temperature) and publishes the result on a topic. The light subscribes to the topic and performs a task based on the event type. The radiator controller subscribes to the topic and performs a task to control the radiator based on the event type. The radiator provides two services, to turn on and to turn off.

Figure 1. System of systems local cloud diagram.



2 Systems

Table 1 Pointers to the SysD documents

Authorization	https://github.com/eclipse-arrowhead/core-java-spring/tree/master/authorization/docs/v4.6.0
Service Registry	https://github.com/eclipse-arrowhead/core-java-spring/tree/master/serviceregistry/docs/v4.6.0
Orchestrator	https://github.com/eclipse-arrowhead/core-java-spring/tree/master/orchestrator/docs/v4.6.0
Eventhandler	https://github.com/eclipse-arrowhead/core-java-spring/tree/master/eventhandler/docs
Thermostat	https://github.com/Lucke0011/iot-arrowhead/tree/master/thermostat-publisher/documentation
Light	https://github.com/Lucke0011/iot-arrowhead/tree/master/light-subscriber/documentation
Radiator Controller	https://github.com/Lucke0011/iot-arrowhead/tree/master/radiator-controller-subscriber/documentation
Radiator	https://github.com/Lucke0011/iot-arrowhead/tree/master/radiator-provider/documentation

3 Use-cases

Table 2 Use-case description table

Automated temperature
ID: 1
Brief description: The temperature is controlled automatically
Primary actors: Prosumer
Preconditions: Have the system set up in a space
Main flow: 1- Prosumer starts the system 2- Prosumer notices that it's cold and the light is turned off and the radiator turned on 3- Prosumer notices it's warm and the light is turned on and the radiator turned off
Postconditions: The system set up

4 Security

The security of Eclipse Arrowhead - and therefore the security of each system - is relying on X.509 certificate trust chains. The Arrowhead trust chain consists of three level:

- Master certificate: **arrowhead.eu**
- Cloud certificate: **my-cloud.my-company.arrowhead.eu**



- Client certificate: `my-client.my-cloud.my-company.arrowhead.eu`
- Example: `lightsubscriber.arrowhead.ltu.arrowhead.eu`

The system sets up a filter that is responsible for validating whether the received HTTPS requests are coming from one of the local cloud's clients based on its certificate.

