

System Description (SysD) DeviceRegistry System

12 april 2018

Contents

1	System Description Overview	2
2	Use-cases	3
2.1	Device Lookup	3
2.2	Device Registration	3
2.3	Device Deletion	4
2.4	Device On-boarding	4
3	Behaviour Diagrams	5
4	System services	5
4.1	Produced Services	5
4.2	Consumed Services	5
5	Security	7
6	Revision history	7
6.1	Amendments	7
6.2	Quality Assurance	7

Document title	Document type
DeviceRegistry System	SysD
Date	Version
April 12, 2018	1.1
Author	Status
Ani Bicaku	Draft
Contact	Page
ani.bicaku@fh-burgenland.at	2(8)

1 System Description Overview

This document is located in the GIT repository of the [Arrowhead Framework Project](#) in the folder: [3_Core SystemsandServices/2_SupportCoreSystemsandServices/4_DeviceRegistrysystem](#)

The DeviceRegistry system stores information and unique identities for devices registered within an Arrowhead local cloud based on DNS and DNS-SD standards [1, 2, 3, 4]. The registration into a local cloud is part of the bootstrapping process of a local cloud. This registry shall in addition to registering device identity, also store metadata about the device and shall hold data on systems that are deployed to each registered device.

The DeviceRegistry system is one of the support core systems of the Arrowhead Framework. It provides one service and consumes the mandatory core services, see Figure 1. This registry in combination with the SystemRegistry creates a chain of trust from a hardware device to a hosted software system and its associated services.

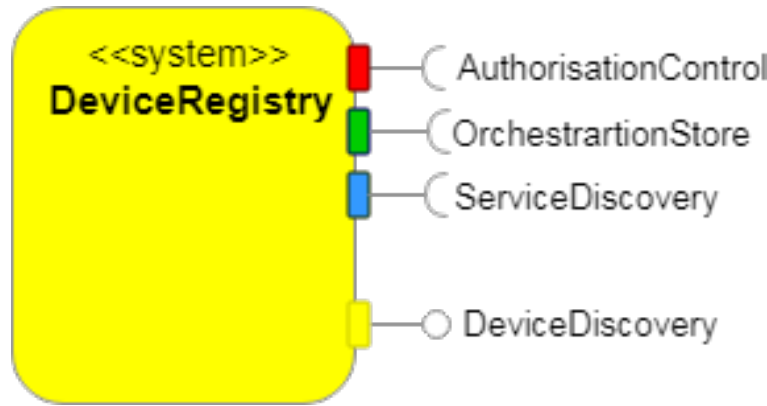


Figure 1: The DeviceRegistry system which produces the DeviceDiscovery service and consumes the AuthorisationControl, OrchestrationStore and ServiceDiscovery services.

The Arrowhead Framework devices within a local cloud should be registered within the DeviceRegistry by using the DeviceDiscovery service.

The DeviceDiscovery service is identified by using the following structure, based on RFC-6335 standard [5]: `_DeviceDiscovery._ahfc-DeviceRegistry._REST._tcp.domain`

The DeviceRegistry system shall be accessible using different SOA protocols like e.g. REST, CoAP, MQTT. The current implemented protocols for the DeviceRegistry system are documented in IDD. If an Arrowhead device is disconnected from the local cloud the device should be removed from the DeviceRegistry including the running systems in the SystemRegistry.

Document title	Document type
DeviceRegistry System	SysD
Date	Version
April 12, 2018	1.1
Author	Status
Ani Bicaku	Draft
Contact	Page
ani.bicaku@fh-burgenland.at	3(8)

2 Use-cases

This section provides use cases that represent the actors and their interaction with DeviceRegistry system.

2.1 Device Lookup

The use case Device Lookup is used by the service consumer to get the list of registered devices in the DeviceRegistry system as shown in Figure 2.

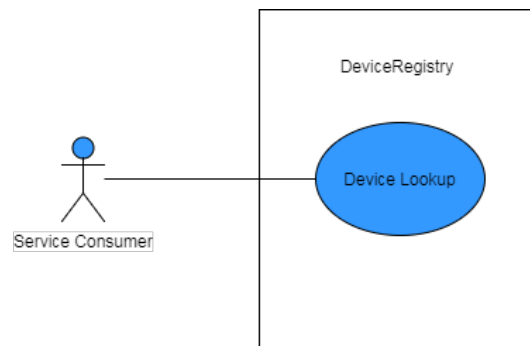


Figure 2: Device Lookup use case

2.2 Device Registration

The use case Device Registration is used by the service producer to register its device in the DeviceRegistry system as shown in Figure 3.

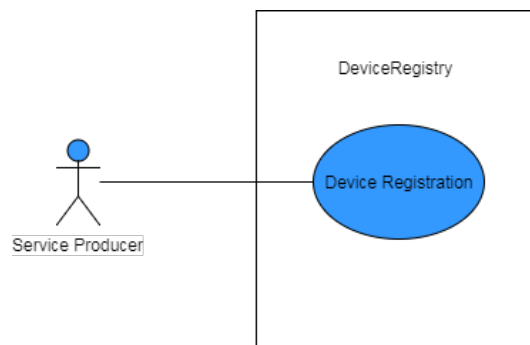


Figure 3: Device Registration use case

Document title	Document type
DeviceRegistry System	SysD
Date	Version
April 12, 2018	1.1
Author	Status
Ani Bicaku	Draft
Contact	Page
ani.bicaku@fh-burgenland.at	4(8)

2.3 Device Deletion

The use case Device Deletion is used by the service producer to delete its device from the DeviceRegistry system as shown in Figure 4.

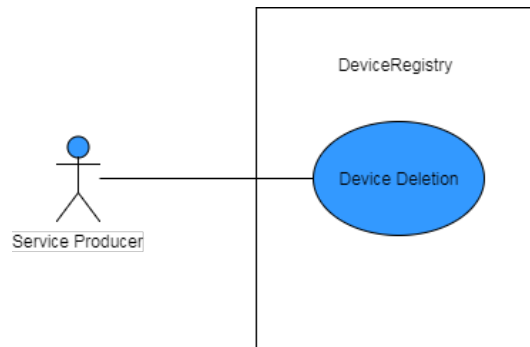


Figure 4: Device Deletion use case

2.4 Device On-boarding

A new device produced by a specific vendor (e.g., Siemens), which contains a Trusted Platform Module (TPM), wants to connect to the Arrowhead local cloud, as shown in Figure 5. An on-boarding procedure is needed between this new device and a local cloud if they have no previous knowledge of each other. The sequence diagram is shown in Figure 6.

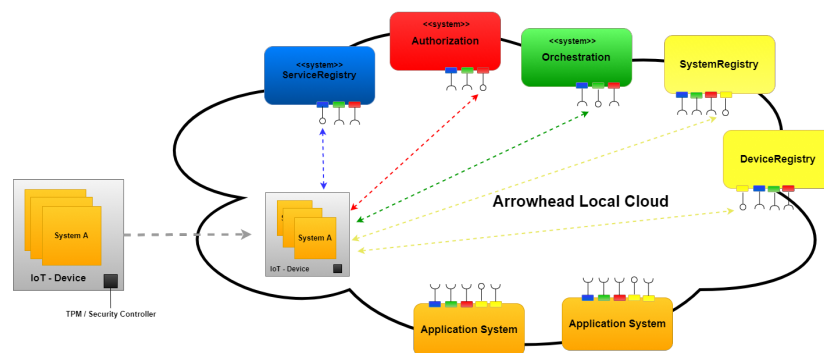


Figure 5: Device on-boarding use case

Document title	Document type
DeviceRegistry System	SysD
Date	Version
April 12, 2018	1.1
Author	Status
Ani Bicaku	Draft
Contact	Page
ani.bicaku@fh-burgenland.at	5(8)

3 Behaviour Diagrams

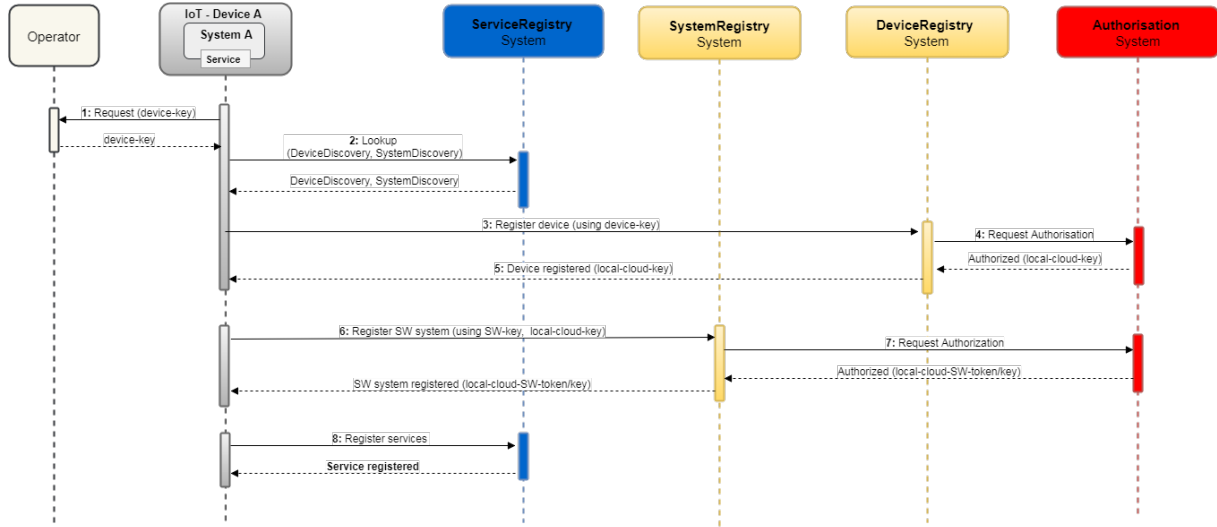


Figure 6: Behaviour diagram for the device on-boarding use case

4 System services

The services produced and consumed by the DeviceRegistry system are provided respectively in Table 1 and Table 2.

4.1 Produced Services

The DeviceDiscovery service is used to register and unregister devices, their systems, as well as find devices among the registered devices in the DeviceRegistry system. Table 1 provides the produced service of DeviceRegistry and the location of the IDD document in the Arrowhead Framework git repository: [arrowhead-f/arrowhead-f.git/3_CoreSystems/2_SupportCoreSystems/4_DeviceRegistrysystem/Documentation/IDD_DeviceRegistry](https://github.com/arrowhead-f/arrowhead-f.git/3_CoreSystems/2_SupportCoreSystems/4_DeviceRegistrysystem/Documentation/IDD_DeviceRegistry)

4.2 Consumed Services

Table 2 provides the consumed services and the location of the Interface Design Description (IDD) document for each service in the Arrowhead Framework git repository.

- The AuthorisationControl service provides the possibility of enabling fine grained access control to any resource/service for external requests; also information about the external consumer. The

Document title	Document type
DeviceRegistry System	SysD
Date	Version
April 12, 2018	1.1
Author	Status
Ani Bicaku	Draft
Contact	Page
ani.bicaku@fh-burgenland.at	6(8)

Table 1: Produced services by the DeviceRegistry system

Service	IDD Document Reference
DeviceDiscovery	https://forge.soa4d.org/plugins/scmgit/cgi-bin/gitweb.cgi?p=arrowhead-f/arrowhead-f.git;a=tree;f=3_Core+Systems/2_Support+Core+Systems/4_DeviceRegistry+system/Documentation/IDD_DeviceRegistry;hb=fba7d498691a51ae22cb33ccf77ebb0eab48d9f5

Table 2: Consumed services by the DeviceRegistry system

Service	IDD Document Reference
AuthorisationControl	https://forge.soa4d.org/plugins/scmgit/cgi-bin/gitweb.cgi?p=arrowhead-f/arrowhead-f.git;a=tree;f=3_Core+Systems/1_Mandatory+Core+Systems/3_Authentication/Arrowhead_IDD_Authorisation+Control+REST_WS-TLS-XML-SPAUTH.pdf;hb=fba7d498691a51ae22cb33ccf77ebb0eab48d9f5
OrchestrationStore	https://forge.soa4d.org/plugins/scmgit/cgi-bin/gitweb.cgi?p=arrowhead-f/arrowhead-f.git;a=tree;f=3_Core+Systems/1_Mandatory+Core+Systems/1_Orchestration/Arrowhead_IDD_Orchestration+Store+REST_WS-TLS-XML-SPORCH.pdf;hb=fba7d498691a51ae22cb33ccf77ebb0eab48d9f5
ServiceDiscovery	https://forge.soa4d.org/plugins/scmgit/cgi-bin/gitweb.cgi?p=arrowhead-f/arrowhead-f.git;a=tree;f=3_Core+Systems/1_Mandatory+Core+Systems/2_ServiceRegistry/Arrowhead_IDD_Service+Discovery+REST_WS-XML-SPSDTR.pdf;hb=fba7d498691a51ae22cb33ccf77ebb0eab48d9f5

IDD document is located in the Arrowhead Framework git repository: [arrowhead-f/arrowhead-f.git/3_CoreSystems/1_MandatoryCore Systems/3_Authentication/](https://github.com/arrowhead-f/arrowhead-f/tree/master/3_CoreSystems/1_MandatoryCore%20Systems/3_Authentication/)

- The OrchestrationStore service provides functionality for storing and retrieving orchestration requirements, which is a set of rules for describing the ideal service required by a consuming system. The IDD document is located in the Arrowhead Framework git repository: [arrowhead-f/arrowhead-f.git/3_CoreSystems/1_MandatoryCore Systems/](https://github.com/arrowhead-f/arrowhead-f/tree/master/3_CoreSystems/1_MandatoryCore%20Systems/)
- The ServiceDiscovery service is used to register and unregister services, as well as find services among the registered serviced in the ServiceRegistry system

Document title	Document type
DeviceRegistry System	SysD
Date	Version
April 12, 2018	1.1
Author	Status
Ani Bicaku	Draft
Contact	Page
ani.bicaku@fh-burgenland.at	7(8)

5 Security

Registration of a device to the DeviceRegistry system of a local cloud should allow:

- Authentication of devices requesting device registration
- Authentication of devices requesting device discovery
- Authentication of devices requesting device de-registration
- Authorization of device registration
- Authorization of device discovery
- Authorization of device de-registration

References

- [1] R. Elz and R. Bush, "Rfc 2181 clarifications to the dns specification, internet engineering task force, <http://tools.ietf.org/html/rfc2181>, 1 page," 1997.
- [2] P. Mockapetris, "Rfc 1034: Domain names: concepts and facilities (november 1987)," *Status: Standard*, vol. 6, 2003.
- [3] P. Mockapetris, "Rfc 1035 domain names. implementation and specification, november 1987," <http://www.ietf.org/rfc/rfc1035.txt>, 2004.
- [4] S. Cheshire and M. Krochmal, "Rfc 6763: Dns-based service discovery," *Internet Engineering Task Force*, 2013.
- [5] M. Westerlund and S. Cheshire, "Iana procedure for the management of the service name and transport protocol port number registry," *RFC6335*, vol. 201, no. 1, 2011.

6 Revision history

6.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	2017-09-10	1.0	Initial Version	Ani Bicaku
2	2017-11-15	1.1	Initial Version	Ani Bicaku

6.2 Quality Assurance



Document title	Document type
DeviceRegistry System	SysD
Date	Version
April 12, 2018	1.1
Author	Status
Ani Bicaku	Draft
Contact	Page
ani.bicaku@fh-burgenland.at	8(8)

No.	Date	Version	Approved by
1			
2			

