SysD Onboarding Controller v1.0

25 september 2019

Contents

1 System Description Overview

Onboarding Controller system:

- A system at the edge of the Arrowhead local cloud, which is not part of the local cloud chain of trust
- It accepts all devices to connect via the Onboarding service, thus it is the first entry point to the local cloud
- It has a certificate for the *https* communication with the device
- (Optionally) the certificate is provided by a public CA (e.g. verisign)

On success, the system provides:

- the endpoints of the DeviceRegistry, SystemRegistry and ServiceRegistry systems
- the Arrowhead CA certificate
- an Arrowhead CA issued "onboarding" certificate

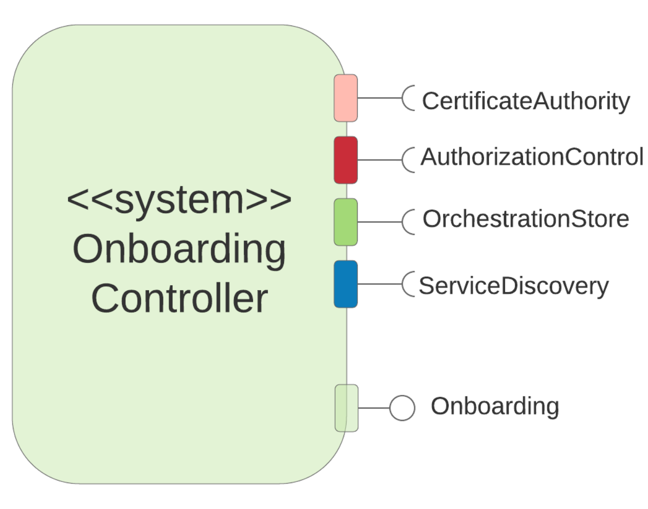


Figure 1: The Onboarding Controller system

The Onboarding Controller system consumes the ServiceDiscovery, AuthorizationControl and CertificateAuthority services and provides the OnboardingController service.

2 Use-cases

2.1 Onboarding Controller

This section provides the use cases that represent the actors and their interaction with the Onboarding Controller system. The actors can be devices with different credentials.

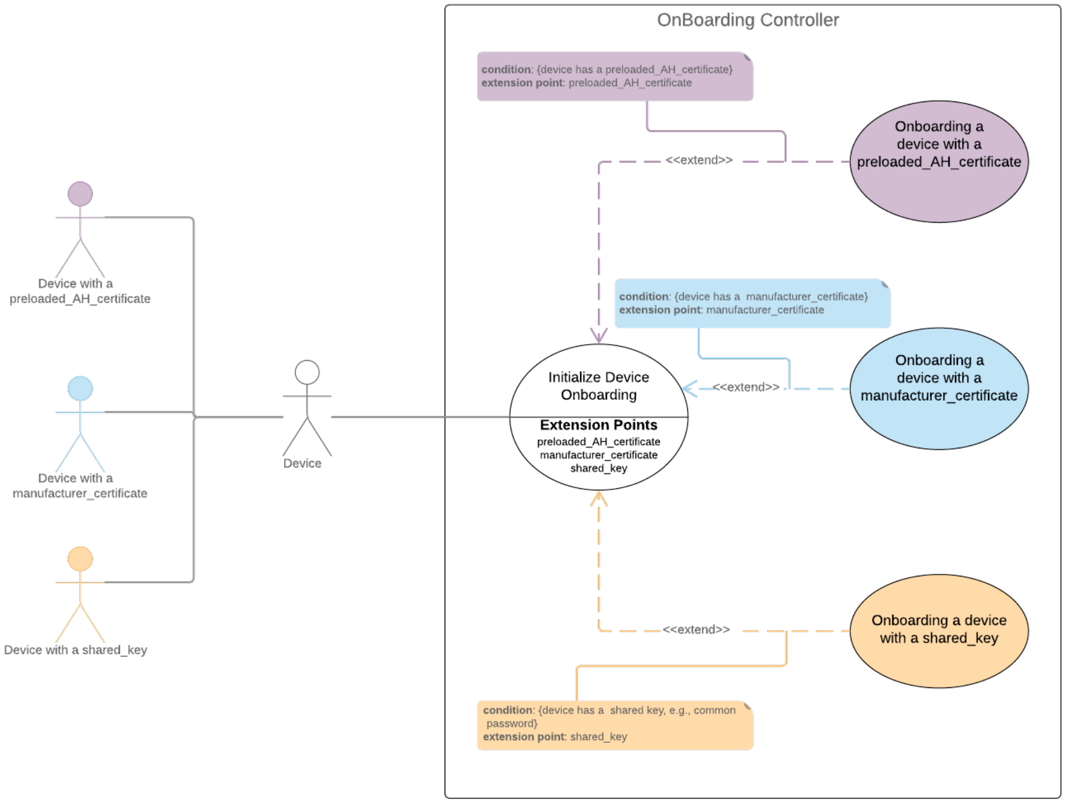

OnBoarding Controller
condition: {device has a preloaded_AH_certificate}
extension point: preloaded_AH_certificate
Onboarding a device with a preloaded_AH_certificate
<<extend>>
Device with a preloaded_AH_certificate
condition: {device has a manufacturer_certificate}
extension point: manufacturer_certificate
Initialize Device Onboarding
Extension Points
preloaded_AH_certificate
manufacturer_certificate
shared_key
<<extend>>
Onboarding a device with a manufacturer_certificate
Device with a manufacturer_certificate
Device
Device with a shared_key
<<extend>>
Onboarding a device with a shared_key
condition: {device has a shared key, e.g., common password}
extension point: shared_key

Figure 2: Onboarding Controller use cases

2.2 Onboarding Procedure

The onboarding procedure is needed when a new device produced by any vendor (e.g. Siemens, Infineon, Bosch, etc.), containing a security controller (e.g. TPM), wants to interact with the Arrowhead local cloud. To assure that the cloud is not compromised upon the arrival of this new device, it is important to establish a chain of trust from the new hardware device, to its hosted application systems and their services. Thus, the onboarding procedure makes possible that the device, systems and services are authenticated and authorized to connect to the Arrowhead local cloud.

The use cases in which the external actor interacts with the Arrowhead local cloud during onboarding include:

- Initialize Device Onboarding (via the Onboarding Controller system)
- Register a Device in the DeviceRegistry (via the DeviceRegistry system)
- Register a System in the SystemRegistry (via the SystemRegistry system)
- Register a Service in the ServiceRegistry (via the ServiceRegistry system)
- Start normal operation (e.g., service lookup, service consumption, etc.)

3 System Services

The Onboarding Controller system produces one service, as shown in Table 1, and consumes three services, as shown in Table 2. All documents and code related to Onboarding Controller system can be found in the repository of the Arrowhead Framework project, `https://forge.soa4d.org/docman/?group_id=58`.

3.1 Produced Services

The Onboarding service ....

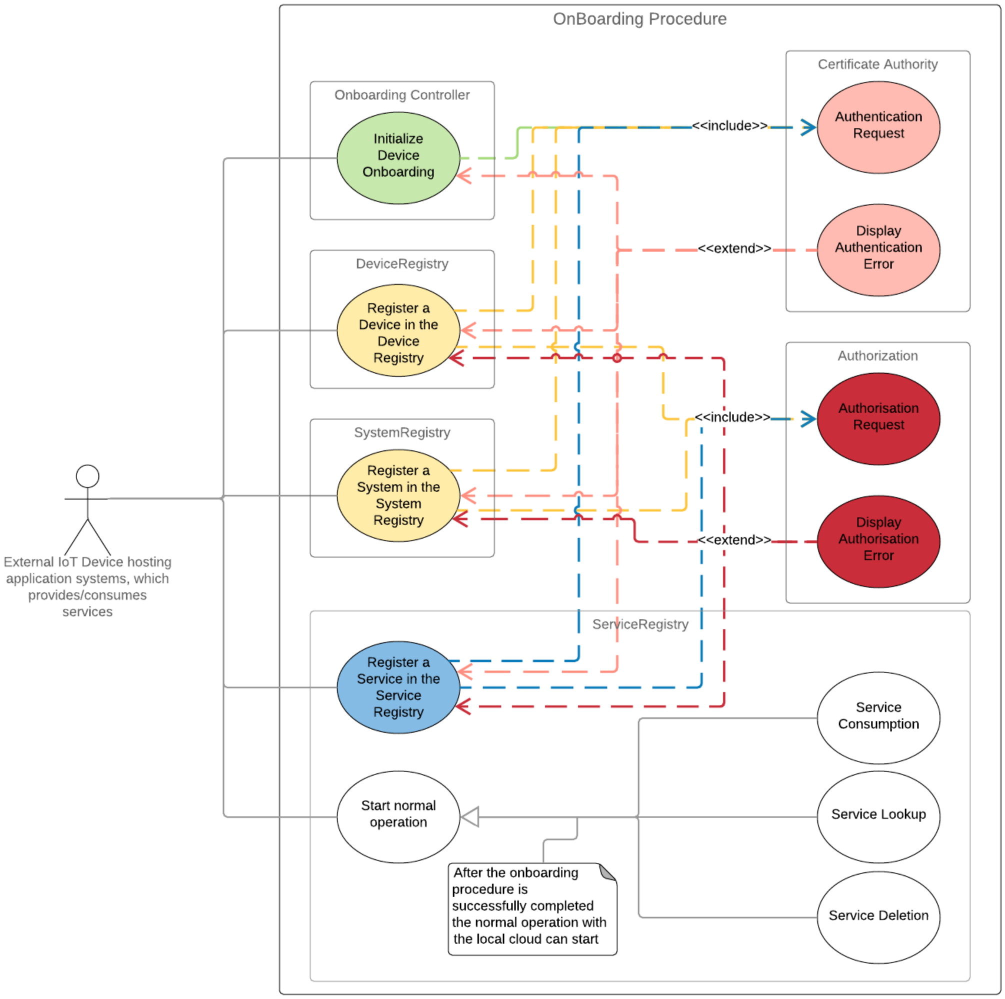
Table 1: Produced services by the Onboarding Controller system

Service	IDD Document Reference
Onboarding	. . .

3.2 Consumed Services

The Onboarding Controller system consumes the three mandatory core services, briefly described below.

- The *AuthorisationControl* service provides the possibility of enabling fine grained access control to any resource/service for external requests; also provides customized information about the external consumer.
- 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 *ServiceDiscovery* service is used to register and unregister services, as well as find services among the registered serviced in the ServiceRegistry system.



OnBoarding Procedure
Onboarding Controller
Initialize Device Onboarding
Certificate Authority
Authentication Request
Display Authentication Error
<<include>>
<<extend>>
DeviceRegistry
Register a Device in the Device Registry
Authorization
Authorisation Request
Display Authorisation Error
<<include>>
<<extend>>
SystemRegistry
Register a System in the System Registry
External IoT Device hosting application systems, which provides/consumes services
ServiceRegistry
Register a Service in the Service Registry
Service Consumption
Service Lookup
Service Deletion
Start normal operation
After the onboarding procedure is successfully completed the normal operation with the local cloud can start

Figure 3: Onboarding Procedure use cases

4 Security

4.1 Authentication

4.1.1 Certificate based authentication

The certificate based authentication works through asymmetric cryptography. Each party needs to have a valid certificate which is signed by a trusted (and known) parent certificate. The trusted

Table 2: Consumed services by the SystemRegistry system

Service	IDD Document Reference
AuthorisationControl	`https://forge.soa4d.org/docman/view.php/58/215/auth-40.zip`
OrchestrationStore	`https://forge.soa4d.org/docman/view.php/58/216/Orch-40.zip`
ServiceDiscovery	`https://forge.soa4d.org/docman/view.php/58/217/sr-40.zip`

certificate confirms the identify of the presented certificate.

4.1.2 Chain of trust

The chain of trust is a concept where a certificate is signed by another certificate, which is more known and trusted. Figure 4 shows a chain of trust with three certificates. The certificate with the highest authority is known as root certificate. The root certificates signs its descending certificates, confirming their identity. Any certiffcate between the root certificate and the end-entity certificate is called an intermediate certificate.

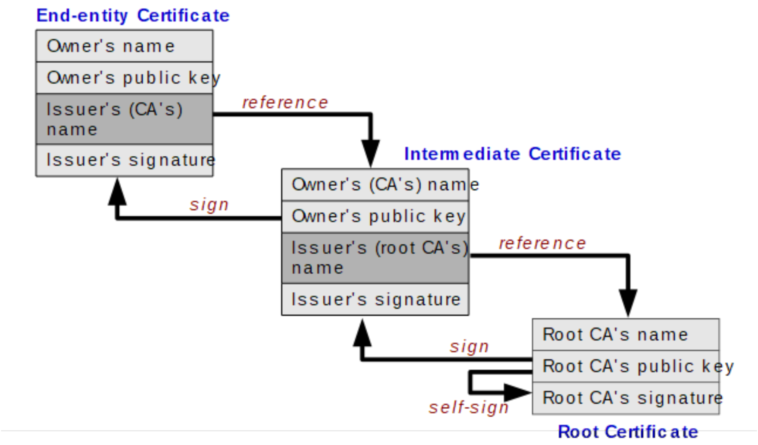
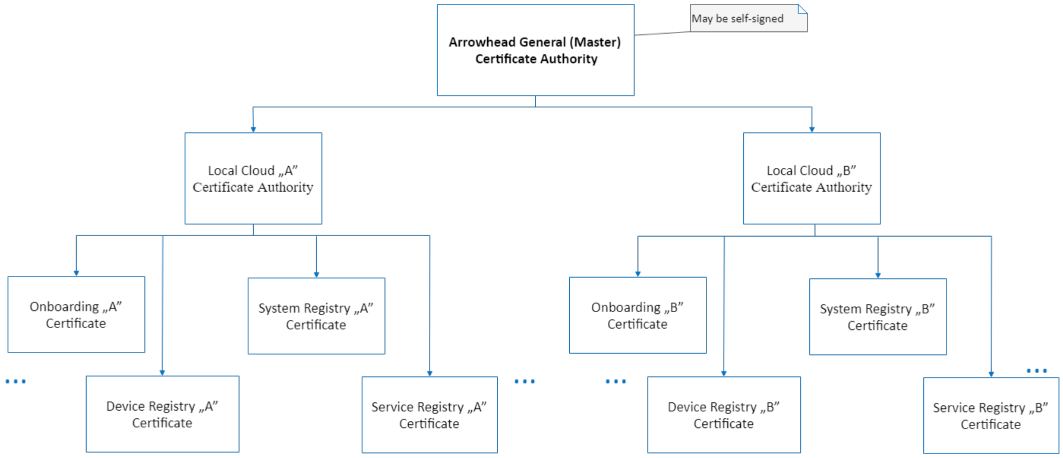


Figure 4: Chain of trust

Figure 5 shows the certificate hierarchy in the Arrowhead framework.

4.1.3 Arrowhead Certificate Authority

Arrowhead contains a core system called Certificate Authority (CA). The CA is the highest authority in the local cloud and responsible for signing any descending certificate. All services in the local cloud must trust the CA of the local cloud. The CA itself may be signed by a central arrowhead consortium, establishing a chain of trust and allowing different arrowhead clouds to interconnect with each other.



Arrowhead General (Master) Certificate Authority
May be self-signed
Local Cloud „A" Certificate Authority
Local Cloud „B" Certificate Authority
Onboarding „A" Certificate
System Registry „A" Certificate
Device Registry „A" Certificate
Service Registry „A" Certificate
Onboarding „B" Certificate
System Registry „B" Certificate
Device Registry „B" Certificate
Service Registry „B" Certificate
...

Figure 5: Certificate Hierarchy in Arrowhead

4.2 Authorization

Authorization is the function of allowing or disallowing specific actions. In the arrowhead system, the AuthorizationControl service is responsible for enforcing all authorization related policies. A policy may define that no systems with a specific OS may be allowed in the cloud. Another policy may define that a service (e.g., temperature collector service) may only query the service registry about existing temperature services but not e.g. power regulating services.

4.3 Assets

No defined yet.

4.4 Non-technical Security Requirements

No defined yet.

5 Revision history

5.1 Amendments

No.	Date	Version	Subject of Amendments	Author
1	07-12-2018	1.0	Initial Version	Silia Maksuti, Mario Zsilak
2	04-02-2019	1.1	Updated Version	Silia Maksuti, Mario Zsilak
3	23-09-2019	1.2	Updated Version	Silia Maksuti, Mario Zsilak