Fairhair
Alliance

Fairhair Specification

Version 1.0

April 2019

Elements of Fairhair Specifications may be subject to third party intellectual property rights, including without limitation, patent, copyright or trademark rights (such a third party may or may not be member of the Fairhair Alliance). The Fairhair Alliance is not responsible and shall not be held responsible in any manner for identifying or failing to identify any or all such third-party intellectual property rights. The furnishing of this Specification does not grant any license to any intellectual property of the Fairhair Alliance or its members.

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. The Fairhair Alliance and the Copyright Holder disclaim all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

For any further explanation of the contents of this document, or in case of any perceived inconsistency or ambiguity of interpretation, please contact the Fairhair Alliance:

Email: secretary-general@fairhair-alliance.org
Website: www.fairhair-alliance.org

Document History

Publication Date	Status	Comments
19 January 2019	Approved by TWG	
5 February 2019	Version for Members approval	Editorial changes: title page, information page, header and footer. Addition of document history and list of contributors
10 April 2019	Approved Fairhair Specification v1.0	Approved by Fairhair Alliance Members

Contents

Figures

Tables

1 Introduction

Today's building-automation systems are implemented independently, as separately managed and very often completely isolated systems, based on domain-specific transport and application protocols. Very little is done to enable information exchange between various automation systems in the building, or to allow building administrators to have cross-domain, streamlined control over the building infrastructure.

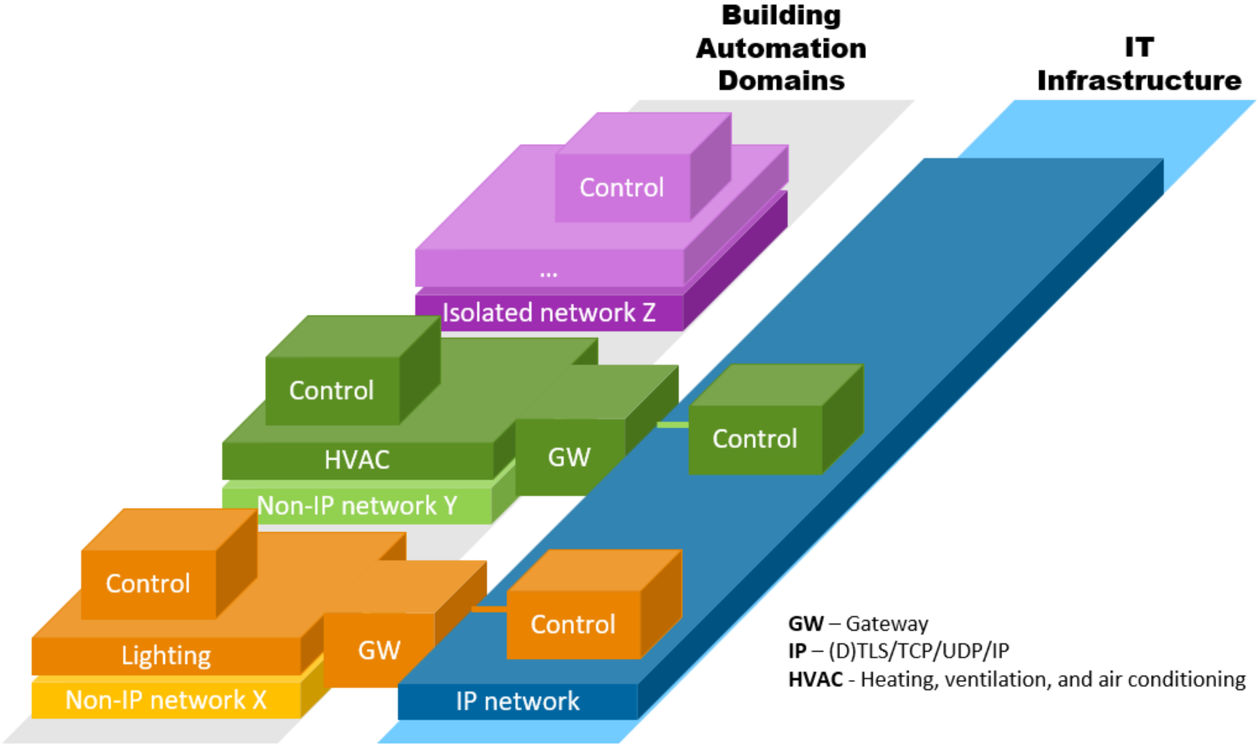


Figure 1: Current building automation domains.

This approach no longer meets the expectations of the building-automation market. Facility managers expect a system solution that offers a unified view of all building-automation domains that can provide real-time insights into the building infrastructure and environment status. They also expect to gain almost complete control over the entire system. The benefits they are looking for are greater efficiency, lower power consumption, efficient audits, easier maintenance but also increased security and safety. The ability to manage multiple buildings, spread all over the world, is no longer seen as a futuristic scenario but as a must.

This imposes a new set of requirements on the building-automation industry. The new solutions must no longer be built as self-contained, isolated domains but as a subset of devices that are part of the common building IT infrastructure. To reduce installation and operation costs, all components should speak the same set of communication protocols and work as one ecosystem able to exchange information.

The building-automation industry is now being confronted with these new requirements, and leading companies have addressed the challenge by establishing the Fairhair Alliance. This organization is developing a common approach in enabling a unified, multi-vendor building-automation solution meeting the requirements of a modern facility manager.

The approach defined by the Fairhair Alliance is to adopt an Internet Protocol (IP) stack as the common transport model for every device that is part of the building infrastructure. The unified IP networking layer provides a proven mechanism to establish end-to-end communication channels that can span across multiple networking domains, including public Internet. This enables integration of various previously isolated application domains into one system.

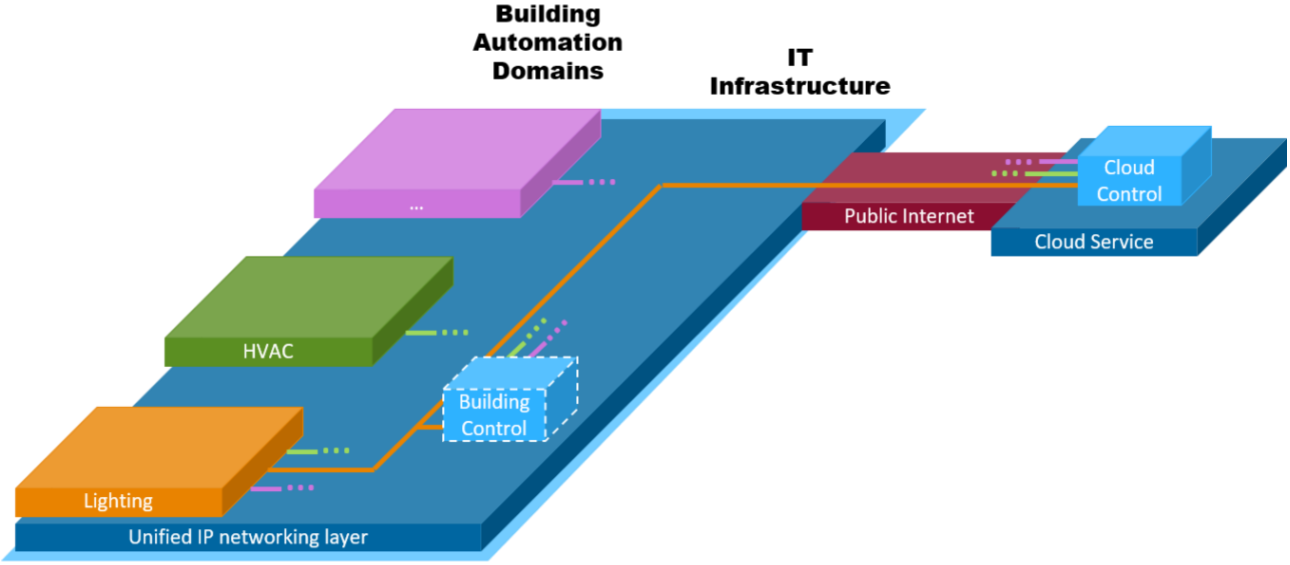


Figure 2: IP-based building automation network.

Such integration is not possible without adopting the established building-automation application protocols. Fairhair specifies the required extensions, such as metadata on resources and resource discovery to enable IP connectivity and to increase the level of interoperability between the application protocols. The goal of the Fairhair Alliance specifications is not only to adapt the existing protocols, but also to facilitate the deployment of a new set of protocols. The openness of this model will give building administrators flexibility to select and deploy any application protocol and solution they may choose, now or in the future.

1.1 Introduction to FA-System

The FA-System provides a set of application layer services that links application protocols to an IP based infrastructure as shown in the protocol stack below.

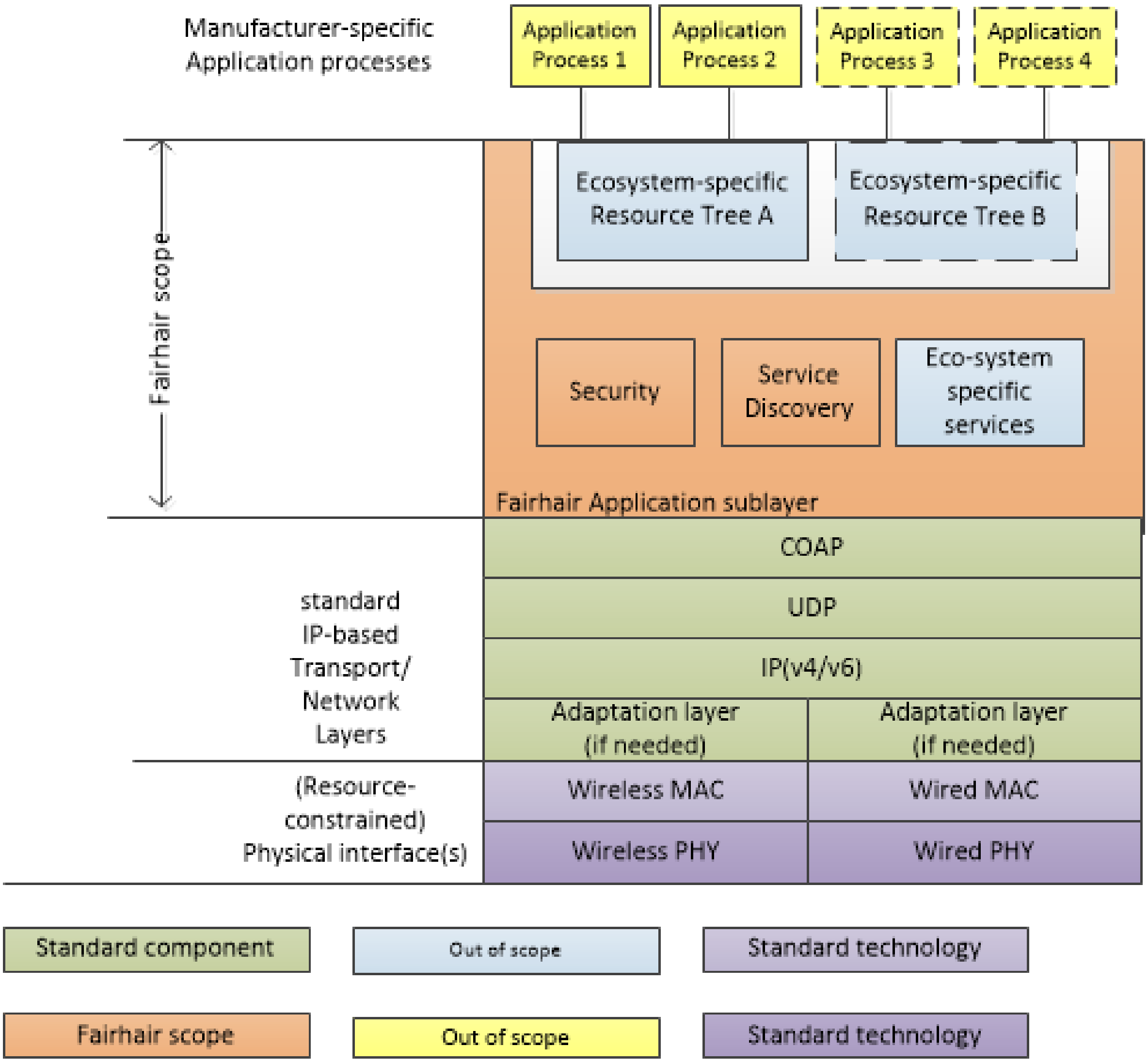

Manufacturer-specific Application processes
Application Process 1
Application Process 2
Application Process 3
Application Process 4
Fairhair scope
Ecosystem-specific Resource Tree A
Ecosystem-specific Resource Tree B
Security
Service Discovery
Eco-system specific services
Fairhair Application sublayer
COAP
UDP
IP(v4/v6)
Adaptation layer (if needed)
Adaptation layer (if needed)
standard IP-based Transport/ Network Layers
(Resource-constrained) Physical interface(s)
Wireless MAC
Wired MAC
Wireless PHY
Wired PHY
Standard component
Out of scope
Standard technology
Fairhair scope
Out of scope
Standard technology

Figure 3 FA-Device components.

The FA-Application framework sits on top of a generic UDP/IP service that provides a medium-independent transport over wired or wireless physical interfaces. Typically, one physical interface is provided per FA-Device. Although the FA-System does not prescribe any specific physical interface it is particularly designed to work with resource-constrained interfaces. When necessary, an adaptation layer MAY be used to interface between IP and the physical interface layer.

The Fairhair Application Service sublayer defines mechanisms for application services, namely resource discovery and security. Some application protocols may define additional ecosystem-specific services at this level; these are outside the scope of the specification.

To interface to the UDP/IP stack, Fairhair uses services provided by the IETF CoAP protocol [RFC7252] for resource-constrained devices.

FA-System assumes a generic framework for the application layer to which specific application protocols may be mapped. In this framework, the Application Process that represents the functionality of a physical (or logical) component, such as a switch, a sensor or an actuator is supported by an Ecosystem-specific Resource Tree. At least one but optionally more than one Ecosystem-specific Resource Trees are hosted on a FA-Device.

The Application sublayer uses services provided by the Application Service sublayer in order to support the component functionality across the transport network layer.

1.2 Introduction to Resource Discovery

FA-Resource Discovery is used to discover resources according their semantic attributes/metadata in commissioning and run-time scenarios of Fairhair deployments.
The main requirements addressed are:

- Seamless scalability from small (e.g. 10 devices) to large (100000+ devices) of networks;
- Support for different life cycle phases;
- Limit effects of compromised devices (security).

The FA-Resource Discovery is based on IETF CoAP discovery using Link Format [RFC 6690] resource descriptions. Both mechanisms are supported by the FA-System, a distributed discovery usually via multicast queries to the "/.well-known/core" resource of devices as well as a centralized discovery by means of unicast queries and registrations to a resource directory [RD]. Both complementary discovery modes support identical queries, i.e. also the distributed discovery supports multiple query arguments. The Discovery specification defines the mechanisms for discovery while the main part of the semantics (resource descriptions) is left to the ecosystems.

1.3 Introduction to Security

A building-automation system (BA-System) involves many employees for successful operation of a commercial building. Each employee has a specific role in managing the BA-System. There are the following roles relevant to security management in commercial buildings:

- Manufacturers of BA-Devices;
- Network administrators who are responsible for secure operation of the overall network infrastructure;
- System integrators who customize BA-Devices, integrate them into the BA-System and perform commissioning;
- Facility managers who monitor the system during their normal operation and respond to alarms; and
- Service technicians who are responsible for maintaining and repairing the BA-System.

In addition to this, these roles might be staffed from different organizations who integrate BA-Devices from different manufacturers into one BA-System. For example, the system integrators of the BA-System may be the manufacturer, the asset owner, or an external company. These roles are involved in operation of the BA-System including the BA-Device functionality and their management. Successful function of a BA-System is only possible when BA-Devices are properly commissioned, operated and maintained. One key aspect that the Fairhair Alliance specification brings forward is the need to strengthen the trust relationship between the manufacturer and the operator. This specification brings together two technologies to accomplish:

- Manufacturer Usage Descriptions are a means for manufacturers to communicate what sort of access a device needs, such that basic access controls can be deployed;
- ANIMA/Bootstrapping Key Infrastructure provides a means for the manufacturer to introduce the device and the local deployment to one another.

This document describes a common approach in enabling a unified, multi-vendor building-automation solution meeting the requirements of a [62443-3-3] security Level 3 compliant BA-System. The security model takes a layered approach based on network segmentation, federated security zones, and application-level authorization. This modular approach can be tailored to specific needs of an FA-System.

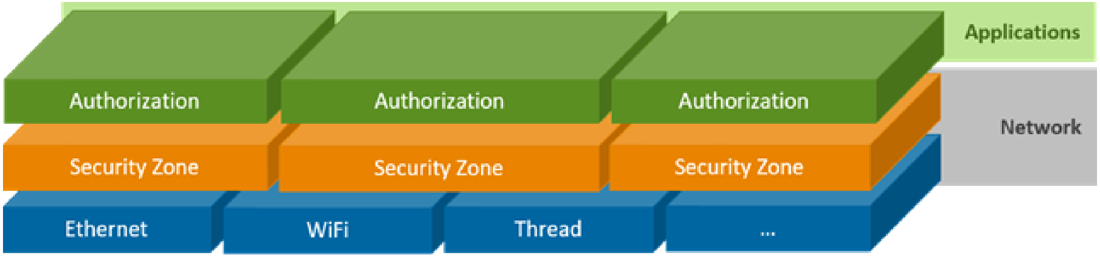


Figure 4 Fairhair Security Layers.

The solution can be deployed on top of any IP network deployment, providing uniform communication infrastructure independent from the underlying networking technologies (e.g. Ethernet, WiFi, or a Thread-based IEEE 802.15.4 network). At the transport level, a security zone concept is used to bridge between diversely administered systems. On top of that, the application-level authorization of the resource model limits the scope of what an FA-Device is entitled to do within the scope of a single security zone.

1.4 Requirements Language

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in [RFC2119].

1.5 Related Standards

[62443-1-1] ISA, “Security for Industrial Automation and Control Systems: Models and Concepts”, ISA-62443-1-1, October 2015 Draft 5

[62443-3-3] ISA, “Security for Industrial Automation and Control Systems: System Security Requirements and Security Levels”, ANSI/ISA-62443-3-3, August 2013

[ACP] T. Eckert, M. Behringer, S. Bjarnason, "An Autonomic Control Plane (ACP)", Internet-Draft Version 13, December 2017

[BRSKI] M. Pritikin, M. Richardson, M. Behringer, S. Bjarnason, K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", Internet-Draft Version 09, October 2017

[OAUTH] E. Hard, “The OAUTH 2.0 Authorization Framework”, RFC 6749, October 2012

[DNS NS] P. Mockapetris, "Domain names - concepts and facilities", RFC 1034, November 1987

[EST-CoAPS] S. Kumar, P. van der Stok, P. Kampanakis, M. Furuhed, S. Raza, "EST over secure CoAP (EST-coaps)", Internet-Draft, January, 2018

[802.1AR] IEEE Standard, "IEEE 802.1AR Secure Device Identifier", December 2009

[IIC-Sec] Industrial Internet Consortium (IIC), “G4: Security Framework”, September 2016 Version 0.27

[RFC 768] J. Postel, “User Datagram Protocol”, RFC 768, August 1980

[RFC 1035] P. Mockapetris, “Domain Names – Implementation and Specification”, RFC 1035, November 1987

[RFC 2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels”, BCP 24 14, RFC 2119, March 1997

[RFC 2460] S. Deering, R. Hinden, “ Internet Protocol, Version 6 (IPv6) Specification”, RFC 2460, December 1998

[RFC 2818] E. Rescorla, “HTTP Over TLS“, RFC 2818, May 2000

[RFC 3986] T. Berners-Lee, R. Fielding, L. Masinter, “Uniform Resource Identifier (URI): Generic Syntax”, RFC 3986, January 2005

[RFC 4122] P. Leach, M. Mealling, R. Salz, “A Universally Unique IDentifier (UUID) URN Namespace”, RFC 4122, July 2005

[RFC 4492] S. Blake-Wilson, N. Bolyard, V. Gupta, C. Hawk, B. Moeller, “Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)”, RFC 4492, May 2006

[RFC 5208] B. Kaliski, “Public-Key Cryptography Standards (PKCS)#8: Private-Key Information Syntax Specification Version 1.2”, RFC 5208, May 2008

[RFC 5216] D. Simon, B.Aboba, R. Hurst, “The EAP-TLS Authentication Protocol”, RFC 5216, March 2008

[RFC 5273] J. Schaad, M. Myers, “Certificate Management over CMS (CMC)”, RFC 5273, June 2008

[RFC 5280] D. Cooper, S. Santesson, S. Farrel, S. Boeyen, R. Housley, W. Polk, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”, RFC 5280, May 2008

[RFC 5652] R. Housley, “Cryptographic Message Syntax (CMS)“, RFC 5652, September 2009

[RFC 5967] S. Turner, “The application/pkcs10 Media Type”, RFC 5967, August 2010

[RFC 6012] J. Salowey, T. Petch, R. Gerhards, H. Feng, “Datagram Transport Layer Security (DTLS) Transport Mapping for Syslog”, RFC 6012, October 2010

[RFC 6690] Z. Shelby, “Constrained RESTful Environments (CoRE) Link Format”, RFC 6690, August 2012

[RFC 7030] M. Pritikin, P. Yee, D. Harkins, "Enrollment over Secure Transport", RFC 7030, October 2013

[RFC 7049] C. Bormann, P. Hoffman, “Concise Binary Object Representation (CBOR)”, RFC 7049, October 2013

[RFC 7159] T. Bray, “The JavaScript Object Notation (JSON) Data Interchange Format”, RFC 7159, March 2014

[RFC 7251] D. McGrew, D. Bailey, M. Campagna, R. Dugal, “AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS”, RFC 7251, June 2014

[RFC 7250] P. Wouters, H. Tschofenig, J. Gilmore, S. Weiler, T. Kivinen, ”Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)”, RFC 7250, June 2014

[RFC 7252] Z. Shelby, K. Hartke, C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, June 2014

[RFC 7292] K. Moriarty, M. Nystrom, S. Parkinson, A. Rusch, M. Scott, “PKCS#12: Personal Information Exchange Syntax v1.1”, RFC 7292, July 2014

[RFC 7346] R. Droms, “IPv6 Multicast Address Scopes”, RFC 7346, August 2014

[RFC 7381] K. Chittimaneni, T. Chown, L. Howard, V. Kuarsingh, Y. Pouffary, E. Vyncke, “Enterprise IPv6 Deployment Guidelines“, RFC 7381, October 2014

[RFC 7390] E. Rhaman, E. Dijk, “Group Communication for the Constrained Application Protocol (CoAP)“, RFC7390, October 2014

[RFC 7515] M. Jones, J. Bradley, N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, May 2015

[RFC 7517] M. Jones, "JSON Web Key (JWK)", RFC 7517, May 2015

[RFC 7641] K. Hartke, “Observing Resources in the Constrained Application Protocol (CoAP)”, RFC 7641, September 2015

[RFC 7959] C. Bormann, Z. Shelby, “Block-Wise Transfers in the Constrained Application Protocol (CoAP)”, RFC 7959, August 2016

[RFC 7967] A. Bhattacharyya, S. Banyopadhyay, A. Pal, T. Bose, “Constrained Application Protocol (CoAP) Option for No Server Response”, RFC 7967, August 2016

[RFC 8075] A. Castellani, S. Loreto, A. Rahman, T. Fossati, E. Dijk, “Guidelines for Mapping Implementations: HTTP to the Constrained Application Protocol (CoAP)”, RFC 8075, February 2017

[RFC 8132] P. van der Stok, C. Bormann, A. Sehgal, “PATCH and FETCH Methods for the Constrained Application Protocol (CoAP)”, RFC 8132, April 2017

[RFC 8187] J. Reschke, “Indicating Character Encoding and Language for HTTP Header Field”, RFC 8187, September 2017

[RFC 8323], C. Bormann, S. Lemay, H. Tschofenig, K. Hartke, B. Silverajan, B. Raymor, “CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets”, RFC 8323, February 2018

[RFC 8366] K. Watsen, M. Richardson, M. Pritikin, T. Eckert, “A Voucher Artifact for Bootstrapping Protocols”, RFC 8366, May 2018

[RFC 8520] E. Lear, R. Droms, D. Romascanu, “Manufacturer Usage Description Specification”, RFC 8520, March 2019

[X.509] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008

[Arcsight] ArcSight Inc., “Common Event Format (CEF)”, Revision 16, July 2010

[Links-JSON] K. Li, A. Rahman, C. Bormann, “Representing Constrained RESTful Environments (CoRE) Link Format in JSON and CBOR”, Internet-Draft, February 2018

[RD] Z. Shelby, M. Koster, C. Bormann, P. Van der Stok, C. Amsuess, “CoRE Resource Directory”, Internet-Draft, October 2018

[RD-DNS-SD] K. Lynn, P. van der Stok, M. Koster, C. Amsuess, ”CoRE Resource Directory: DNS-SD mapping”, Internet-Draft, March 2018

[draft-ietf-lwig-coap] K. Kovatch, O. Bergmann, C. Bormann, “CoAP Implementation Guidance”, July 2018

[voucher01] M. Richardson, P. van der Stok, P. Kampanakis, “Constrained Voucher Artefacts for Bootstrapping Protocols”, August 2018

1.6 Terminology

Table 1 Terminology.

Term	Description
AAA	Authentication, Authorization, and Auditing
Administrator	Person authorized by device owner for supervisor access of the device
ANIMA	Autonomic Networking Integrated Model and Approach
BACnet	Building Automation and Control Networks [www.BACnet.org]
BA-Device	A building automation endpoint.
BA-System	The collection of devices, including network components operating together.
BRSKI	Bootstrapping Key Infrastructure
CoAP	Constrained Application Protocol [RFC7252]
Conduit	Communication path between two security zones
Conduit Controller	A device that resides within two security zones that effects a conduit.
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Denial of Service
dotdot	Standard as defined by Zigbee (www.zigbee.org)
EAP	Extensible Authentication Protocol
EAP-TLS	Transport Layer Security (TLS) method of EAP

Ecosystem	Ecosystem that makes use of the FA-System, examples: BACnet, KNX, Zigbee and possibly others in the future.
Ecosystem entry point	The specific top-level CoAP resource that provides an entry point into ecosystem-specific resources (e.g. objects, properties and commands). The entry point is the root resource of the resource subtree that models the ecosystem-specific representation of the FA-Devices functionality. The resource name and path are defined by the ecosystem. A FA-Device may contain multiple ecosystem entry points for example if its supports multiple ecosystems.
FA	Fairhair Alliance
fa.cs	Resource type for the Fairhair common services
fa.lnk	Resource type for the Fairhair Linkage resource
fa.rd-conf	Resource type for the Fairhair resource that configures and manages information about RDs on FA devices
fa.rd-id	Resource type for the Fairhair resource that Identifies the resource which configures and manages RD identity information.
FA-Client	An FA-Device behaving as a (CoAP) client, see also [RFC7252]: The originating endpoint of a request; the destination endpoint of a response.
FA-Device	A device adhering to this specification.
FA-Discovery	Device discovery according this specification
FA-Resource	A link with a predefined “rt” value, the actual “rt” value may be defined by Fairhair or another eco system.
FA-Server	An FA-Device behaving as a (CoAP) server, [RFC7252]: The destination endpoint of a request; the originating endpoint of a response.
FA-System	The collection of FA-Devices, including network components operating together
HMI	Human Machine Interface
HVAC	Heating, Ventilation & Air Conditioning
IACS	Industrial automated control system
IETF	Internet Engineering Task Force [www.ietf.org]
IP	Internet Protocol (v4 for version 4, v6 for version 6)
KNX	Standard as defined by KNX [www.knx.org]
L2	ISO-7498-1 Layer 2
L3	ISO-7498-1 Layer 3
LAN	Local Area Network
Link Identifier	Abstract address agnostic of the IP address
LRI	Logical Resource Identifier

MAC	Medium Access Control (layer)
MUD	Manufacturer Usage Description
NTP	Network Time Protocol
OAUTH	Open Authorization Protocol
PHY	PHYsical layer / Physical interface layer
QoS	Quality of Service
Resource Directory (RD)	A CoRE Resource Directory based on the definition in [RD]. The RD provides directory services where devices can register their resources and others can discover these through query functions. Having an RD helps discovery functions for nodes on constrained, bandwidth-limited mesh networks and for discovery of sleepy nodes.
RBAC	Role-Based Access Control
RD	Resource Directory
REST	Representational State Transfer
RM	Resource Model
SDO	Standards Development Organization
Security Zone	A group of FA-Devices that all make use of the same trust anchor
SL	Security Level
SLAAC	Stateless automatic address configuration
TLS	Transport Layer Security
UDP	User Datagram Protocol [RFC768]
URI	Uniform Resource Identifier [RFC3986]
UUID	Universally Unique IDentifier
VLAN	Virtual Local Area Network
X.509	Certificate format
ZCL	Zigbee Cluster Library

2 Business Requirements (informative)

2.1 Introduction

As enterprises consider how they will deploy new and advanced lighting, HVAC and other building automation systems, they will need to take into account whatever new risks those systems entail. Similarly Lighting and HVAC systems have to operate in a more open environment unlike the siloed proprietary networks of the past. This makes them more vulnerable as they are a part of an enterprise network, requiring them to protect themselves with application level security. While no system is impervious to attack, this requirements specification is intended to make clear how such systems can be secured against many forms of attack. The goals of the approach specified in this document are as follows:

1. The endpoint (lighting/HVAC sensor or actuator) will reasonably be safe from attackers on the network;
2. The network will be resilient against an endpoint being operated by an attacker, thus restricting the value of the attack to the smallest scope possible (e.g., to just the link to which the attacker is connected).
3. The system will provide economical auto configuration capabilities, such that the endpoint need only be pulled out of a box by an installer and plugged in for it to find any relevant controllers, perform any relevant discovery, and then operate.

These requirements are applicable only to devices that have an IP stack, including 6lo-based systems. Other network architectures will present their own risks. In particular, this specification is not applicable to TX-only devices. An IP-based approach is the best common denominator that enterprise administrators understand how to manage and support. These requirements further primarily assume that the endpoint is managed using CoAP on an IP-based network, and that endpoints require normal infrastructure services, such as DNS, DHCP, and NTP, that they may have a discovery facility to discover other such endpoints, and finally, that communications unrelated to lighting or HVAC will be restricted.

An endpoint may be situated on several types of networks:

- An isolated wired network with no Internet access, or access to any other system;
- An isolated wireless network with no Internet access, or access to any other system;
- A wired network that is connected to an enterprise infrastructure network, where there may be physical control of access to the wiring; or
- A wireless network that is connected to an enterprise infrastructure network, where there is less control of the airwaves.

There are mixed networks as well (wired, wireless, etc.).

2.2 Threats to be defended against

Two key axes are worth mentioning up front: remote attacks versus physical access; and attack for the sake of impacting the lighting or HVAC, versus an attack as a means to attack other enterprise infrastructure. An attacker in this case refers to someone who is either entirely unauthorized to use a network or its services, or someone who is authorized, but is attempting to exceed her or his authorization.

Therefore, foremost in our minds in requirements are the following threats:

- An attacker turns on or off the lights;
- An attacker prevents the lights or the HVAC system from being turned on or off;
- An attacker causes one or more luminaires or the HVAC system to malfunction;
- An attacker uses one or more luminaires to attack other endpoints on the network; and
- The attacker changes the configuration of the lighting or HVAC system.

We will not go into detail about why the above threats are serious but assume that the reader understands that they are.

2.2.1 Unauthorized Access

Unauthorized Access has occurred when someone has been able to manipulate the state of the endpoint without proper permission. For connected Lighting, in as much as a CoAP server is the management interface, this implies that an attacker has access to the network interface of the endpoint, and that he or she can present a credential to the endpoint that it finds as valid. In a wired environment where ports are shared, attackers who have access to the L2 network may themselves then attack the endpoint. In a wired environment where there are dedicated ports, the network is in a position to limit access to the endpoint, subject to the limits of physical security.

The simplest example of unauthorized access is poor management of passwords. The easiest approach to address that problem is not to use passwords, but instead to make use of asymmetric cryptography or pre-shared keys using strong pseudo-random number generation, or a combination of the two, in conjunction with some form of trusted introduction mechanism. In addition, the network is in a position to limit access to the endpoint if it knows what the endpoint is and what components are authorized to communicate with it. Unauthorized access is mitigated via various means and layers in the network.

2.2.2 Denial of Service

Denial of Service (DoS) has occurred when an attacker has prevented the lighting or HVAC service from being accessed by authorized individuals. This attack can take place in many forms. One example would be that the attacker sends so much traffic to the endpoint that legitimate traffic cannot get through. Another form of this attack is when the attacker has gained access to the endpoint and configured it to prevent what should be authorized access.

To facilitate the network protecting against flooding of a link, endpoint manufacturers should provide a profile of expected communications, and what sort of endpoints and services it expects to communicate with. From that point, the network is in a position to limit traffic to and from only those services.

2.2.3 Vulnerability Exploitation

An attacker has exploited a vulnerability in an endpoint when he or she has caused it to malfunction. Typically this is due to a programming error. Once a vulnerability is exploited it is assumed that the attacker can use the capabilities of the endpoint against the interest of its owners, perhaps to attack other endpoints as described above.

Endpoint manufacturers must assume that they will have vulnerabilities. The best ways to address vulnerabilities is to apply good coding practices, and to be able to upgrade the endpoint with an appropriate fix. It should only be possible for authorized administrators to upgrade endpoints. Furthermore, we must assume that it will take time for manufacturers and their customers to field patches. The network can sometimes provide protection against exploits if it is aware of the endpoint type and there is a means to tell the network what communication to block in order to defeat the exploit.

2.3 Different Roles Envisioned

In order to fully understand security requirements, it is helpful to have an understanding of the sorts of individuals that will need access to various components within the system.

2.3.1 Installer

An installation technician is responsible for physically mounting devices, wiring them, and verifying correct installation of devices. Typically, they are unlicensed and/or uncertified. Their work is complete as soon as they can verify that every device is operable. Installation locations may be high, inaccessible, or closed off after construction is finished, so lifts or other equipment may be required for this phase. For security concerns, installation technicians do not possess the network credentials, tools, or knowledge to commission devices securely onto the network.

2.3.2 Network Architect

A network architect is responsible for designing the topography of networks and interfacing the Fairhair network with the facility’s IT infrastructure. Their work may not take place on-site, and they may work with commissioning engineers or network engineers during the commissioning phase. Note that for some small installations a network architect may not be present and the role of the network architect may be performed by the commissioning or network engineer.

2.3.3 Network engineer

A network engineer is the on-site engineer responsible for implementing the design of the network architect. The network engineer is responsible for authenticating and authorizing devices onto the network and providing network access to the authorized devices. Their work may begin after construction and installation has completed - physical access to the installed devices can thus not be assumed - and is complete when all devices are provided appropriate network access. The network engineer may be required to work along with or before/after the commissioning engineer and both roles may be performed by the same person in some installations.

2.3.4 Commissioning engineer

A commissioning engineer is responsible for the commissioning phase of a project. The commissioner must configure the devices to behave according to the application needs and is often responsible for mapping physical location of devices to the network address or a unique device identifier. Their work may begin after construction and installation has completed - physical access to the installed devices can thus not be assumed - and is complete when all devices are commissioned and configured. The commissioning engineer may be required to work along with or before/after the network engineer and both roles may be performed by the same person in some installations.

2.3.5 Facility manager

A facilities manager is responsible for on-going maintenance of a facility, and all the devices contained therein. They are typically connected to the ownership of the facility, but might be contracted out. The main task of the facility manager is to plan, coordinate and delegate work to domain specific experts. Depending on the size of the installation, the domain specific expert roles may be taken up by one or multiple persons. Domain specific experts include:

- Service technician: A service technician reports to the facility manager and is responsible for moving devices, reconfiguring devices, performing regular sequence testing, and replacing devices; and
- Network technician: A network technician reports to the facility manager and manages connectivity during the operational phase of the system, and troubleshooting connectivity issues.

2.3.6 Planner/Building Architect

A planner is responsible for defining the components of a system. They define the device types and numbers being installed within a building. A planner may provide the basis for the network architects work, work with the network architect or incorporate the role of the network architect. Usually the planner creates a site or system plan that is handed over to the facility manager showing the position of devices and how devices are connected to each other. They do not necessarily have a direct interaction with the system. But the plan may be used during commissioning e.g. to reuse already defined device IDs during mapping of physical locations of devices to network addresses.

Note: A planner may have other roles that are not relevant for the scope of the work in Fairhair and are therefore not described here.

2.3.7 How much of this is specific to the lighting or HVAC solution?

The good news is that almost none of it is specific to lighting or HVAC. That's good for network vendors because it limits the amount of special code we have to develop and support for any one use of the network. By making use of existing standards, our partners can leverage work of those who have come before them.

The specific requirements that are necessary to address the threats are provided in Appendix C.

3 Security

3.1 System Security Level

Over the past few years, affordable computing power, IP6 connectivity on constrained devices and evolving data analytics techniques have opened the door to convergence of HVAC, lighting, business systems and the internet. Today and especially in future, BA-Systems are more and more exposed to attacks of increasing sophistication and the design assumptions of existing BA-Systems no longer apply. A successful attack on a BA-System might result in leaks of sensitive data, interruption of operations and destruction of systems. The consequences might be serious such as damage to brand and reputation, economic damage to critical infrastructure or injury or loss of human life.

Network administrators of enterprise networks must take these risks seriously. The use of sensors and actuators and the operation of a BA-System is not the typical information technology (IT) experience. Beyond reliability, IT and BA-Systems are based on different system characteristics with different priorities. IT organizations, for example, will need to place increased importance on safety and resilience beyond the levels expected in many traditional IT environments today. The highest priority of many BA-Systems is safety: do not cause injury or death, do not put public at risk and protect the environment from harm. The second and third priorities are often reliability and resilience.

On the other hand, not each BA-System has the same security requirements and the security level needs to be adapted. IEC 62443 offers a framework to balance the security-level.

3.1.1 IEC 62443-3-3 in a Nutshell

Security will be more and more regulated in the future. Current developments like the German initiative "IT Sicherheitsgesetz", the US Executive Order 13636 (Improving Critical Infrastructure Cybersecurity), and the French initiative from ANSII (Cybersecurity for Industrial Control Systems) are the first witnesses proving this statement. We use IEC [62443-3-3] as a means to validate our requirements, to ensure that we have not missed anything. Not all requirements in the IEC specification may be applicable.

IEC [62443-3-3] lists security requirements that must be met to reach a certain level of security, called security level (SL). Each security requirement consists of a baseline requirement and zero or more requirement enhancements (REs) to strengthen security.

IEC [62443-3-3] is intended for solutions, not for products. Thus, for a solution it must be ensured, that all products (preferably those that offer proper security measures that match the targeted security level SL-T for the solution) deployed in the solution are properly configured and additional measures are taken where needed to achieve a security level SL-A according to IEC [62443-3-3] that meets the SL-T.

IEC [62443-3-3] Security Level 1:
Protection against **casual or coincidental** violation

IEC [62443-3-3] Security Level 2:
Protection against **intentional violation** using **simple means** with low resources, generic skills and low motivation.
IEC [62443-3-3] Security Level 3:
Protection against intentional violation using **sophisticated means** with **moderate resources**, IACS specific skills and moderate motivation.
IEC [62443-3-3] Security Level 4:
Protection against intentional violation using sophisticated means with **extended resources**, IACS specific skills and high motivation.

When designing an FA-System to meet the set of system requirements associated with specific SL-Ts, it is not necessary that every component of the proposed FA-System support every system requirement to the level mandated in this standard. Compensating countermeasures can be employed to provide the needed functionality to other subsystems, such that the overall SL-T requirements are met at the BA-System level. Inclusion of compensating countermeasures during the design phase should be accompanied by comprehensive documentation so that the resulting achieved FA-System SL, SL-A (control system), fully reflects the intended security capabilities inherent in the design. Similarly, during certification testing and/or post-installation audits, compensating countermeasures can be utilized and documented in order to meet the overall control system SL.

This document does not adhere to IEC 62443 but instead borrows some principles from it.

3.2 Enterprise network deployments

3.2.1 IP Address Configuration

There are two forms of address assignment available to devices: DHCP (both for IPv4 and IPv6) and SLAAC. It is typically the choice of the local deployment as to which is used. FA-Devices connecting via 802.3 or 802.11 (non-mesh) technology MUST support both DHCP (v4 and v6) and SLAAC. FA-Devices connected via THREAD or other low power technology MUST use SLAAC and MAY support DHCP, and THREAD border routers (or future analogous devices) MUST support both. (C.1.1 - GEN4)

Note that an enterprise network with a more controlled environment will force FA-Devices to use DHCP only. The reason is that most network administrators want to manage network IP address provisioning centrally for ease traceability (i.e. which FA-Device has which IP at any given time) and in connection with a monitoring system.

FA devices MAY support DNS resolution. Network infrastructure MUST provide name servers and indicate them in appropriate announcements (DHCP or ND), when it is available (e.g., in any Internet-Connected deployment).

3.2.2 Network Time

Many security functions depend on time-sensitive credentials. Examples are operational device certificates or access tokens. For appropriate processing of operational device certificates and auditing functions it is important for FA-Devices to have accurate clocks. Lack of secure source of time can mean an attacker can modify the FA-Device time and fool the validation mechanism. Protocols such as NTP can provide rather accurate time sources from the network, but are not immune to attacks. A secure time source can be FA-System internal or external as long as the time source is signed by a trusted source and the FA-Device can validate this time source.

Hence, an FA-System SHALL have a means to receive time information for purposes of logging and certificate validation. The network infrastructure SHALL provide NTP for this purpose (IEC [62443-3-3] – SR 2.11 RE 1: SL3). In addition, FA-Devices SHOULD have a real time clock. When they do not, they MUST still receive time. It is the network's responsibility to prevent false NTP servers from distorting time.

3.2.3 Monitoring

A monitoring system in managed enterprise networks is standard. This includes tools for reporting abnormal traffic patterns (port scanning, SYN flooding, and related IP source addresses), monitoring tools (e.g. detection of abnormal bandwidth utilization) and syslogs (finding server and system errors). In general, a monitoring system examines IP traffic, records IP addresses, parses log files, and reports abnormal conditions. Integrity protection on logs and sources of log data is also important to detect unusual behavior (misconfigurations or attacks). Logs may be used in investigations, which depend on trustworthy data sources [RFC 7381].

An FA-Device SHALL provide the capability for authorized users and/or tools to access audit logs on a read-only basis (IEC [62443-3-3] – SR 6.1: SL1). In addition, an FA-Device can provide programmatic access to audit records using an application programming interface (API) (IEC [62443-3-3] – SR 6.1 RE 1: SL3). The log SHALL provide the capability to generate audit records relevant to exceptional events, such as configuration changes, access control errors, request errors, operating system events, backup and restore events, configuration changes, potential reconnaissance activity and audit log events.

FA devices SHOULD make use of SYSLOG with ArcSight Common Event Format as a means to assist logging utilities classify events.

Network monitoring and detection of security events also involves security incident management and the execution of proper responses to those events. One response of the system might be the isolation of security zones (IEC [62443-3-3] – SR 5.2: SL3 / C.1.3 - NAC7). Devices or groups of devices MAY be isolated when they exhibit anomalous behavior.

3.2.4 Network Segments

The main reasons for segmenting networks are to reduce the exposure, or ingress, of network traffic into an FA-System and reduce the spread, or egress, of network traffic from an FA-System. This improves overall system response and reliability as well as provides a measure of cyber security protection. It also allows different network segments within the FA-System, including critical control systems and safety-related systems, to be segmented from other systems for an additional level of protection. Different industrial security standards such as ISA/IEC [62443-1-1] or ISA/IEC [62443-3-3] **–** SR5.2 recommend separating networks into "security zones" or network segments, with each segment containing a set of connected equipment having similar security policies and communications requirements. They also recommend assigning each network segment a trust level, and protecting communications through the perimeters of network segments, especially for the cases of communications between segments at different trust levels.

Network segmentation can be fine-grained or coarse-grained, and individual FA-Systems must select their own degree of segmentation. For a particular FA-System, fine-grained segmentation is generally better but is also costlier to maintain. Network areas that are candidates for segmentation generally include public networks (such as the internet), business networks, control networks (e.g., engineering workstations, HMIs), device networks (e.g., automation station, actuators, sensors), protection networks (e.g., trust anchors) or safety (e.g. fire).

Manufacturer Usage Descriptions are used to facilitate network segmentation. See below for more details.

3.3 Security Zones

The results of the risk assessment are used to assign individual security levels [62443-3-3] to security zones and conduits. Security zones and conduits may be established by grouping assets based on functionality, location and responsible organization, or however a local administrator chooses.

A security zone:

- Is a domain that could possibly span across multiple networks segments
- Is represented by a group of devices that may be accessible to one another
- Uses a single trust anchor to issue operational device certificates for all devices participating in the security zone

Security zones differ from network segments in that a security zone may consist of multiple network segments that provide some form of isolation that is administered by a single entity.

3.3.1 Communications between Security Zones

From time to time it may be desirable for devices in one security zone to communicate with another security zone. Each of these zones may be operated by different entities. As such, the operational trust anchors within FA devices may be different. In order to simplify design of most end systems, communication across security zones is mediated through conduit controllers. These controllers are expected to have additional resources available, and may themselves be manually configured to establish trust beyond the domain in which they reside. For example, the controllers for the fire detection system and the security system may be configured to trust one another for the purposes of doors being unlocked if there were to be a fire.

Many FA-Systems are spread across numerous legal entities (typically expressed through different security zones) where data ownership rights and implementation choices may lead to legal liability concerns, and integration inconsistencies that complicate even the most straightforward architectural choices. Conduits [62443-3-3] are constructs that identify communications flow between security zones.

Hence, a consistent zoning concept not only consists of protection at network level with firewalls, VLANs, DNS Zones etc. but also contains access control on each FA-Device at application level. However,

a consistent security concept at multiple layers is complex and costly. In contrast, simple FA-Systems may only implement a security zoning concept in the device certificate [X.509] which reflects also certain access rights.
Each FA-Device is part of a security zone which reflects a logical location. This security zone id SHALL be represented in a subjectAltName/rfc822Name field in the operational device certificate according to [ACP] section 6.1.1. If a security zone has its own exclusive CA certificate then no additional security zone id is needed in the operational device certificate.

3.3.2 Conduits

A security zone may incorporate controller devices that are responsible to communicate with other security zones. Such controller devices must be commissioned with additional trust anchors (and device certificates) to become part of the external security zones the controller is supposed to communicate with. A Conduit is created by authorizing the pair of controllers belonging to two different security zones to communicate with each other.

The conduit controller:

- Acts as an application-layer-gateway to mediate between different security zones;
- Is responsible to “police” the data exchange between its own zone and the external zone(s) by applying appropriate set of rules (obtained via Manufacturer Usage Descriptions [RFC 8520] when dealing with L3 and L4 communications, or otherwise through application specific rules); and
- Is in control what to communicate with other devices within its own zone, or over the conduit to the external zone(s).

Conduit controllers may represent one or more domains, depending on the configuration of a deployment, and the deployment’s contractual or regulatory environment.

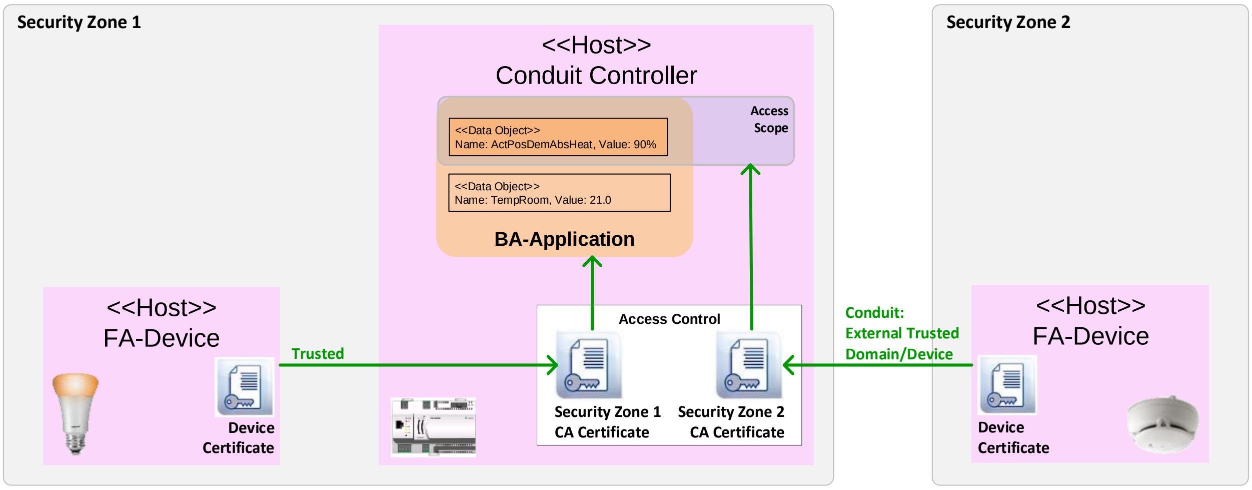


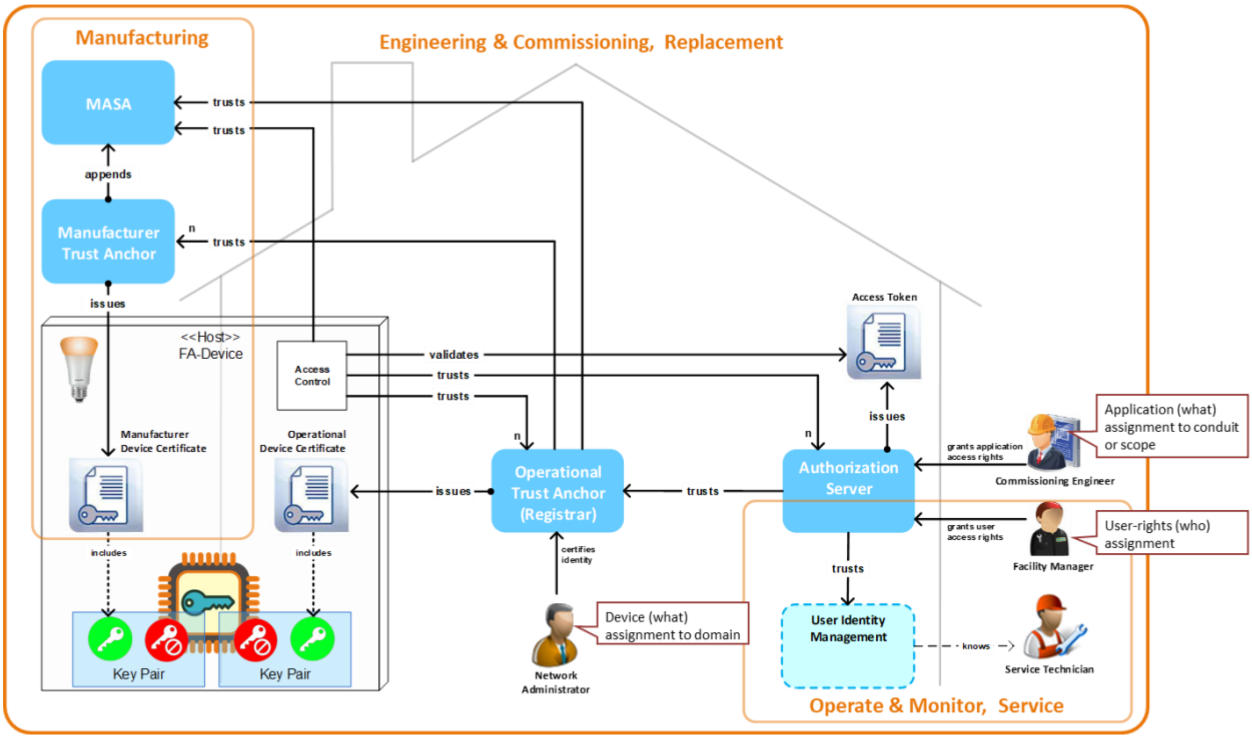
Figure 5 Conduit Example for (D)TLS (Transport Layer).

As an example, a fire detection system in security zone 1 has access to the lighting system in security zone 2 for emergency evacuation signaling. Conversely, the lighting system has no access to the fire detection system. In this example, one or more sensors residing in the fire protection zone would communicate with its control system to indicate elevated temperature and smoke, leading the controller to conclude that there was a fire. This system would serve as a conduit-controller to the lighting system, and have direct access to the lighting zone. It would have a trust anchor and device certificate that is understood within the lighting zone, and therefore directly issue appropriate commands to activate emergency lighting within the affected building.

3.4 FA-System Security

3.4.1 Actors and Trust Relationships

A typical FA-System is a complex assembly of components and devices. The trustworthiness of the FA-System depends on trust in all of the components and devices, how these elements are integrated and how they interact with each other. Permeation of trust is the hierarchical flow of trust within a system from its overall usage to all its components and devices [IIC-Sec]. The following figure illustrates the permeation of trust in an FA-System which usually starts with the manufacturer trust anchor followed by the operational trust anchor and the authorization server.



Manufacturing
Engineering & Commissioning, Replacement
MASA
trusts
trusts
appends
Manufacturer Trust Anchor
n
trusts
issues
<<Host>>
FA-Device
Access Control
validates
trusts
trusts
Manufacturer Device Certificate
Operational Device Certificate
includes
includes
Key Pair
Key Pair
Access Token
issues
n
n
Operational Trust Anchor (Registrar)
issues
trusts
Authorization Server
certifies identity
Network Administrator
Device (what) assignment to domain
grants application access rights
Commissioning Engineer
Application (what) assignment to conduit or scope
grants user access rights
Facility Manager
User-rights (who) assignment
trusts
User Identity Management
knows
Service Technician
Operate & Monitor, Service

Figure 6 Trust Anchors within a single security zone.

Each component of an FA-System has installers, commissioning engineers, service technicians, manufacturers, network administrator, etc., (actors) that execute the various roles in the engineering, commissioning, and usage of the hardware and software of an FA-System. These roles cut across multiple organizations, each with its own interests that must be aligned. Hence, each FA-System has a unique permeation of trust. Everything from supply chain, to commissioning, provisioning, regular usage and finally end-of-life decommissioning must be carefully monitored to ensure the initial trustworthiness is preserved through its lifecycle.

The manufacturer is expected to deliver trustworthy hardware, manufacturer usage descriptions, and software or firmware updates. The assignment of an FA-Device to a security zone is responsibility of the commissioning engineer. Later for operation at the application level, the facility manager takes over responsibility for the FA-Device. The operational trust anchor is in the responsibility of the network administrator. The network administrator adds an FA-Device to a Security Zone by issuing it an operational device certificate (device identity certificate).

3.4.2 Device Security

Each FA-Device MUST provide mechanisms to identify anomalous requests and attacks. At a minimum, the behavior MUST be logged. Where possible the device will mitigate the attack in a manner appropriate to it, taking into account safety and resource management (e.g., battery lifetime). These mitigations will be documented (C.1.3 NAC7). Devices MUST be delivered only with necessary services that will match what is provided in the MUD file (see below). Devices SHALL do appropriate bounds checking on any received application-level requests.

3.4.2.1 Support Lifetime and Device Software Management

The FA-devices SHALL have the means to securely update its software. Software packages SHOULD be signed by the manufacturer and verified by the device prior to installation. Manufacturers of FA-Devices MUST declare the intended lifetime of the devices, and MUST provide security-related updates for those devices during that lifetime.

There SHALL be sufficient storage to allow for an FA-device to receive an update for the foreseeable lifetime of the device. FA-manufacturers must provide a process by which software updates can be installed on FA-Devices.

3.4.3 Manufacturer Usage Descriptions

Manufacturer Usage Descriptions provide local deployments information necessary to limit the threat surfaces of devices. By providing a JSON based file on a manufacturer web site and by outputting a URL at onboarding time, manufacturers can provide deployment recommendations for appropriate access control settings. The MUD file contains abstractions such as “same-manufacturer” or “my-controller” that the enterprise administrator can fill in.

FA-Devices making use of [802.1AR] certificates SHOULD include the MUD extension that contains a URL, and create a corresponding MUD file. “My-controller” should be used to point to devices that control automation infrastructure. “same-manufacturer”, “local”, or “manufacturer” should be used for operational devices to communicate with one another (IEC [62443-3-3] – SR 7.6: SL1/3).

3.4.4 Authentication and Authorization

The FA-System SHALL make use of (D)TLS with both client and server-side certificates as described below for secure device-to-device communications for authentication in combination with standard RBAC practices, and MAY make use of the certificates for implicit authorization. The FA-System SHALL also make use of OAUTH as described below for authorizing specific access.

The explicit assumption is that FA-Devices which belong to the same Security Zone are initially trusted at least for process communication. As a consequence, the FA-Device must be certain that it can trust the operational trust anchor. It is obvious, however, that the scope of individual FA-Devices can be restricted if necessary.

3.4.4.1 Credential

The [X.509] certificate (or also known as public key certificate) is an electronic document used to prove ownership of a public key. The certificate includes information about the key, information about its owner's identity and the digital signature of an entity that has verified the certificate's contents are correct (trust anchor). If the signature is valid, and the entities examining the certificate trust the signer, then they know they can use that key to communicate with its owner.

FA-System access control depends on two related concepts: authentication and authorization.
The network and FA-Devices make use of [X.509] certificates to authenticate to one another. Applications make use of [OAUTH] tokens to authenticate and authorize each other, and users/clients may make use of a number of technologies, including passwords, smart cards, etc., to authenticate themselves.

Certificates are used to identify and authenticate FA-Devices to the network, and further as a means to issue [OAUTH] access tokens. [OAUTH] access tokens are then used to authorize application-to-application communication. To authenticate network infrastructure, FA-Devices make use of IEEE 802.1X and EAP-TLS or EAP-TEAP for 802.3 and 802.11 networks. FA-Devices on a THREAD network SHALL authenticate the THREAD border router via DTLS. In each case, the ANIMA [BRSKI] flow below MUST be followed. The ANIMA provisioning flow is described in section 3.5 .

The following is a sample operational flow via EAP-TEAP that also includes the ANIMA flow as an option appears below. However, EAP-TLS as specified in [RFC 5216] is sufficient once a device is provisioned.

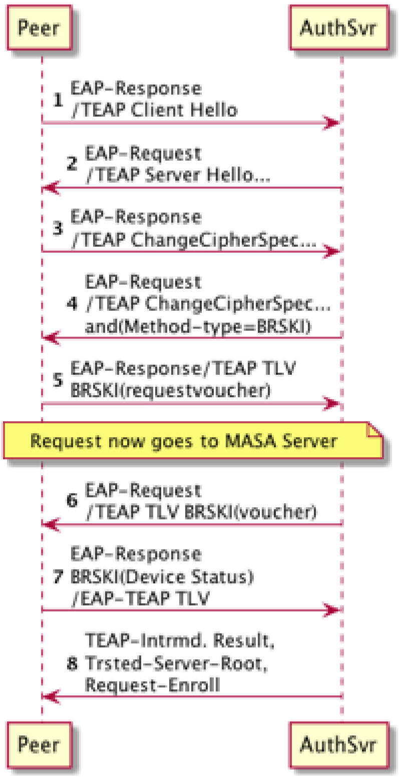

Peer
AuthSvr
1 EAP-Response /TEAP Client Hello
2 EAP-Request /TEAP Server Hello...
3 EAP-Response /TEAP ChangeCipherSpec...
4 EAP-Request /TEAP ChangeCipherSpec... and(Method-type=BRSKI)
5 EAP-Response/TEAP TLV BRSKI(requestvoucher)
Request now goes to MASA Server
6 EAP-Request /TEAP TLV BRSKI(voucher)
7 EAP-Response BRSKI(Device Status) /EAP-TEAP TLV
8 TEAP-Intrmd. Result, Trsted-Server-Root, Request-Enroll
Peer
AuthSvr

Figure 7 EAP provision flow.

3.4.4.2 Access Token

An access token is a digital document containing attributes associated to the holder (service technician or rather a user certificate) by the issuer (authorization server). The FA-Device presents the access token (e.g. JSON Web Token [RFC 7515]) with each message. The communication peer grants access to predefined scopes as soon as the access token is considered as trustworthy.

3.4.4.3 Immutable Identity

FA-Devices are installed in different environments with different security requirements for protecting the devices against a broad range of circumstances such as vandalism, theft, or tampering. This protection may be:

- Integral to the FA-Device (e.g. security module); or
- Physical enclosure of the FA-Device (e.g. lockable control cabinet).

Physical enclosures around the FA-Device serve as both access control mechanism to the physical device and prevent unauthorized access. However, physical enclosures are often not applicable since temperature sensors, light switches or luminaries, for example, require deployment in uncontrolled or even public environment. In such cases, and if SL3 is required, hardware mechanisms for secure key storage are necessary for storing credentials ([62443-3-3] – SR 1.5 RE1). These mechanisms are provided by a security module which could be an internal or external peripheral to the FA-Device's communication controller.

An FA-Device manufacturer MUST configure an IDevID (IEEE [802.1AR]) within the device for identification before it ships.

The FA-Device will then go through a manual, semi- or fully-autonomic on-boarding process on site before it is connected to the FA-System. The decommissioning, or end of life process, resets the FA-Device to the factory state in a secure manner. A security module is utilized by the FA-Device for cryptographic support – e.g. realized in form of a crypto peripheral or a smart card. The FA-Device may utilize services of a security module as a cryptographic service provider for different cryptographic functionalities as the generation and verification of digital signatures and key agreement as well as content data signature and content data encryption. The security module contains the cryptographic identity of the FA-Device, and in addition to a secure storage for cryptographic keys and certificates it may also serve as a reliable source for random numbers (C.1.2- ID3).
A standard crypto controller, for example, comes with a "built in" initial identity (manufacturer device certificate). The manufacturer device certificate defines a globally unique per-device secure identifier cryptographically bound to the device hardware as a special [X.509] certificate. Furthermore, a crypto controller has additional secure storage for on-site generated [X.509] certificates (operational device certificates).

3.5 Device Identity Enrolment

3.5.1 Registrar

The Registrar service is a representative of the domain that is configured to decide whether a new FA-Device is allowed to join the domain. The network administrator of the domain configures the Registrar service to control this process. Typically a Registrar service is part of the local FA-System and contains a mixture of blacklist rules, white lists (for known installed FA-Devices) and stateful tracking to protect the FA-System. A Registrar service can be part of a tool together with the Domain CA but also an independent service that connects FA-Devices with a remote Domain CA.

3.5.2 Device Operational Identity Provisioning

The Figure below shows the steps that shall be followed for a device in factory-new state to enroll it into a security zone.

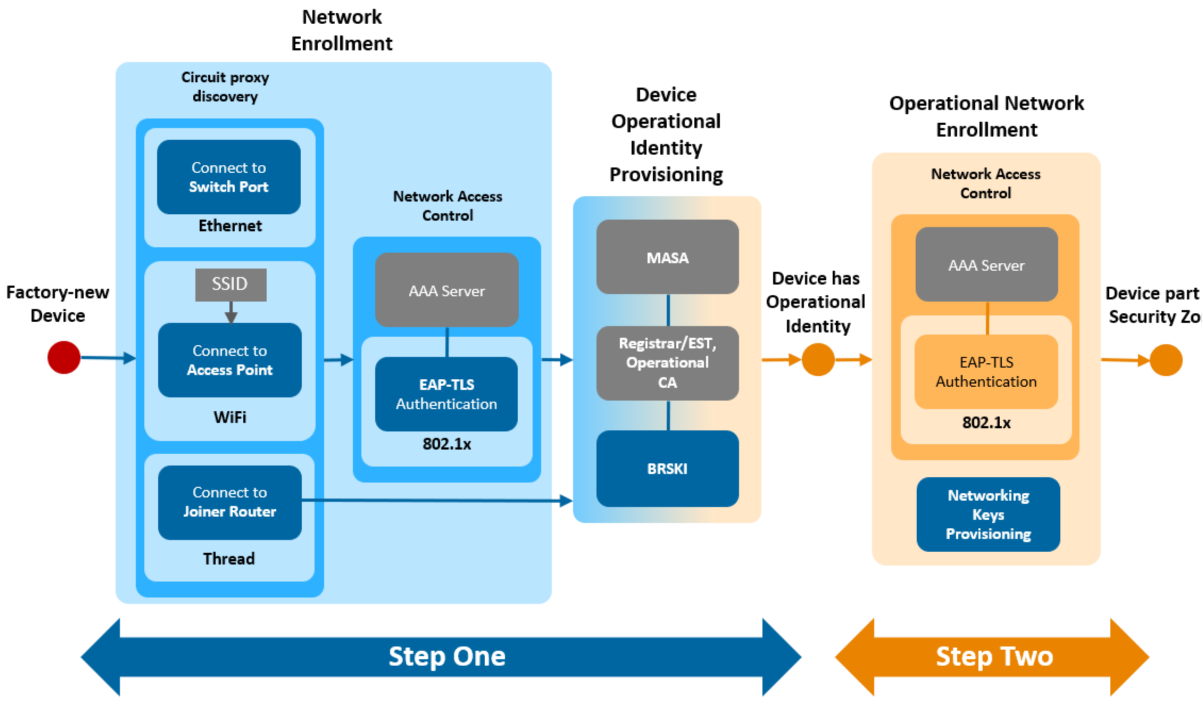


Figure 8 Device Identity Provisioning.

Prior to being authorized a device has only L2 connectivity, and can be expected to have access to only similarly unauthenticated devices or a registrar.

3.5.2.1 Autonomous Operational Identity Provisioning

When the factory-new device is turned on it looks for networks that are open to autonomous enrollment. Once a network is discovered the device initiates the enrollment process. Devices that are not capable of performing the autonomous enrollment shall be enrolled with alternative methods. Before a FA-device can join a security zone it has to be authenticated and provisioned with security zone CA trust anchor and the zone operational device certificate. The autonomous enrollment uses the ANIMA bootstrapping process as defined in “Bootstrapping Remote Secure Key Infrastructures” [BRSKI]. In the case of IEEE 802.3 or 802.11 networks, the format of the token and response is as specified in that document. In the case of 802.15.4/THREAD, the format of the request and response voucher format is as specified in [voucher01] section 6.1.

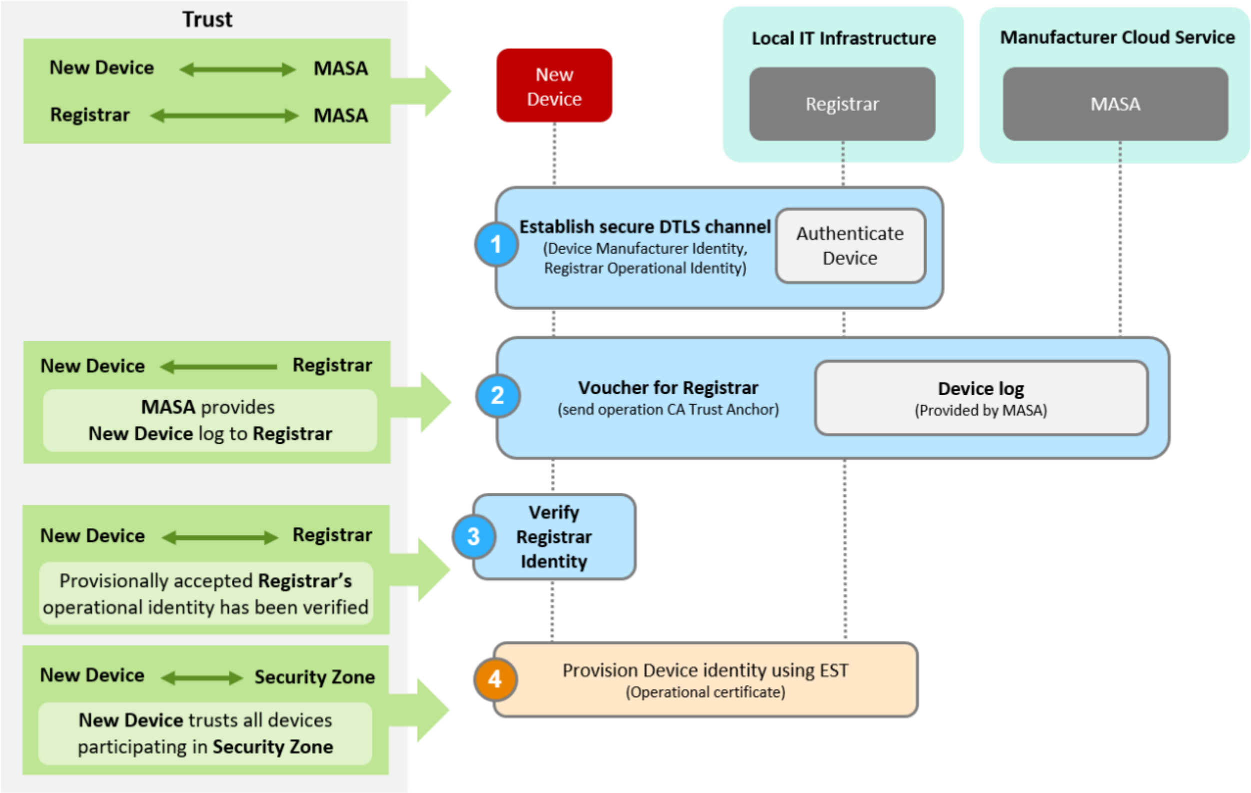

Trust
New Device
MASA
Registrar
MASA
New Device
Local IT Infrastructure
Registrar
Manufacturer Cloud Service
MASA
1
Establish secure DTLS channel
(Device Manufacturer Identity, Registrar Operational Identity)
Authenticate Device
New Device
Registrar
MASA provides New Device log to Registrar
2
Voucher for Registrar
(send operation CA Trust Anchor)
Device log
(Provided by MASA)
New Device
Registrar
Provisionally accepted Registrar's operational identity has been verified
3
Verify Registrar Identity
New Device
Security Zone
New Device trusts all devices participating in Security Zone
4
Provision Device identity using EST
(Operational certificate)

Figure 9 BRSKI bootstrapping flow chart.

The process involves three steps including FA-Device authentication, secure zone CA trust anchor provisioning and security zone device certificate enrollment.

3.5.2.1.1 FA-Device Authentication

In the first step (D)TLS channel between FA-Device and the Registrar is established.

1. FA-Device initiates (D)TLS handshake with Registrar via Proxy acting as a relay. The proxy function will vary based on L2 technology. For THREAD-based devices, it might be (for instance) a border router. For wired Ethernet it might be either null or the switch.
2. The (D)TLS handshake performs mutual authentication based on the device manufacturer and Registrar certificates. The device lacks yet the domain CA to verify the Registrar certificate, therefore it provisionally accepts the Registrar certificate to complete the (D)TLS handshake.

At this stage, the Registrar may decide to not continue further with the process if the FA-Device is not authorized to join the security zone (e.g. verification of the device Manufacturer certificate failed).

3.5.2.1.2 Domain CA Certificate Provisioning

The Operational Trust Anchor (Domain CA) can be local as part of a tool or a remote service from a manufacturer or an enterprise network. The Domain CA is an entity that issues operational device certificates for a particular entity and certifies the ownership of a public key by the named subject of the FA-Device certificate to a security zone or rather a site.

The correct domain CA certificate is added to the FA-Device described in the steps:

1. The FA-Device sends a signed request for an audit token from Registrar on the provisionally authenticated (D)TLS connection.
2. The request is forwarded by the Registrar to the MASA service of the device manufacturer with additional information about its own and domain CA certificate.
3. The audit token is generated by the MASA service containing domain CA certificate and signed with the manufacturer's key to ensure the new FA-Device can trust it.
4. The token is relayed by the Registrar to the FA-Device.
5. The FA-Device verifies the token and stores security zone CA certificate.
6. The FA-Device verifies provisionally accepted Registrar certificate used during the TLS handshake.

3.5.2.1.3 Operational Device Certificate Enrollment (Pull Certificate)

In the autonomous scenario, a new FA-Device discovers the Registrar either by using DNS, mDNS, a resource directory or the FA-Device has a configuration setting where to find the Registrar. [RFC 5967] is the mandatory format for a CSR. A CSR includes all of the key details of the requested certificate such as subject, organization, state, as well as the public key of the certificate to get signed. The CSR MUST include the specific subject distinguished name from the manufacturer device certificate. After sending the CSR, the Registrar may decide not to continue further with process if the new FA-Device cannot be authorized automatically. The Registrar receiving this CSR SHOULD validate the proof-of-identity and then, check the subject distinguished name included in the CSR with the manufacturer device certificate. The CSR only gets signed by the trust anchor if all checks were successful. A successful response MUST be a certs-only CMC Simple PKI Response (PKCS#7), as defined in [RFC 5273]. The authorization at the Registrar is deployment specific and may need manual confirmation from a network administrator. As soon as the new FA-Device is authorized to join the domain then the Registrar writes the operational device certificate to the FA-Device. The returned operational device certificate is a FA-System specific [X.509] device certificate. The enrollment process for a new operational device certificate is based on EST (Enrollment over Secure Transport) [EST-CoAPS].

Devices may re-enroll via EST-CoAPS at any time prior to certificate expiration, but MUST re-enroll with their IDevID **after** certificate expiration.

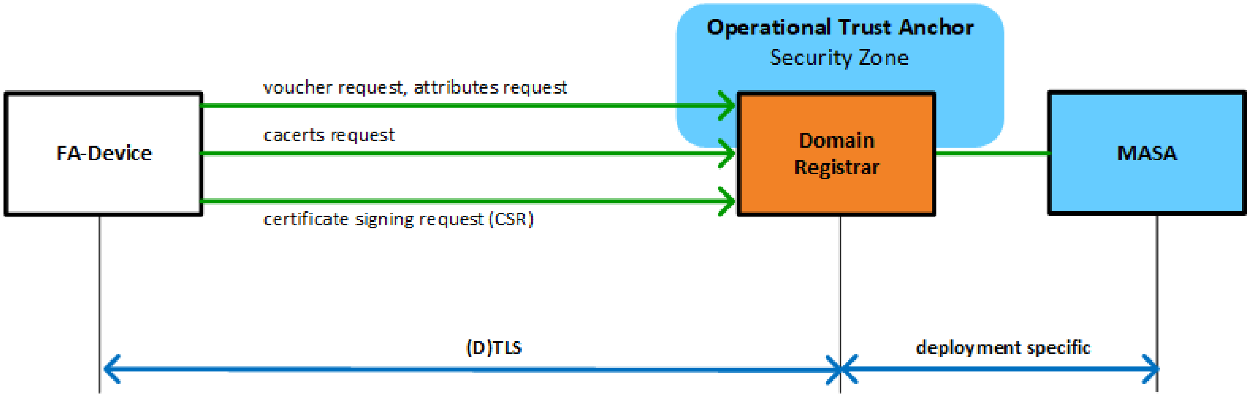


Figure 10 EST-CoAPs CSR.

3.5.2.2 Commissioning Tool as Registrar (Push Certificate)

When an FA-System domain registrar is not available, a vendor-provided commissioning tool may serve as in this function. Once an enterprise Registrar becomes available, devices will enroll with the IDevIDs as described above. For transition and mobility purposes, FA-Devices may continue to use any actively enrolled certificate for communication until that certificate expires or is removed. Devices already using one LDevID SHALL not make use of new LDevIDs issued from different registrars until configured to do so, so as to avoid confusion in certificate selection.

The commissioning tool always acts as a client and controls the communication flow with FA-Devices. A FA-Device in factory state SHALL accept client D(TLS) sessions provisionally. The FA-Device may begin generation of a key pair as a result of the CSR request. If the FA-Device cannot immediately respond due to time required to generate a key pair, the FA-Device shall return "5.03 Service Unavailable" but uses the Max-Age option to indicate the number of seconds after which to retry. The FA-Device SHALL implement the following interface.

Before accepting a voucher request, the Registrar MUST be authenticated on the FA-Device:
Request: GET brski/v **Response:** { ietf-voucher-request }
The commissioning tool writes the voucher (coming from a MASA server) to the FA-Device:
Request: POST brski/v **Response:** { ietf-voucher [RFC 8366] }
Certificate signing request (CSR) attributes can be configured as follows:
Request: POST ldevid/csr/att { [RFC 7030] section 4.5.2 ASN.1 encoded }
The following request is used to read a PKCS#10 (including proof-of-possession) certificate signing request (CSR) from a FA-Device.
Request: GET ldevid/csr/sen **Response:** { PKCS#10 }
The following request is used to write a device certificate signing response (PKCS#7) to a FA-Device:
Request: POST ldevid/csr/sen { PKCS#7 }
The following request is used to write a private key and a certificate to a FA-Device:
Request: POST ldevid/csr/skg { PKCS#7, PKCS#8 }

3.5.2.3 Out-of-Band Certificate Provisioning (Push Certificate)

In an out-of-band scenario, the commissioning tool may act as a proxy for providing a CSR to a remote Registrar or directly to a remote Operational Trust Anchor (Domain CA). However, in some cases this connection is not reliable or trustworthy (e.g. Email, Bluetooth etc.). Hence, the commissioning tool writes a nonceless voucher [RFC 8366] to the FA-Device first, followed by cacerts (trusted CA certificates) and the operational device certificate (CSR). The Registrar certificate from the voucher MUST be used to check the signature of the enveloped cacerts payload. How the commissioning tool gets the voucher is out-of-scope of this document. The FA-Device MUST store all CA certificates (voucher and cacerts) into the trust list (see Appendix B.1.1).

The CSR needs also an additional signature that can be validated in the Registrar with the manufacturer CA certificate (proof-of-identity). The manufacturer's CA certificate is provided out-of-band.
If the FA-Device has not a direct mutual authenticated connection (e.g. DTLS) with the Registrar then the certificate signing request SHOULD be signed (proof-of-identity) with the pre-installed manufacturer device certificate's private key. In this case, the [RFC 5967] request to the Registrar SHOULD be enveloped using either a [RFC 5652] (default) or [RFC 7515] or [COSE] structure.

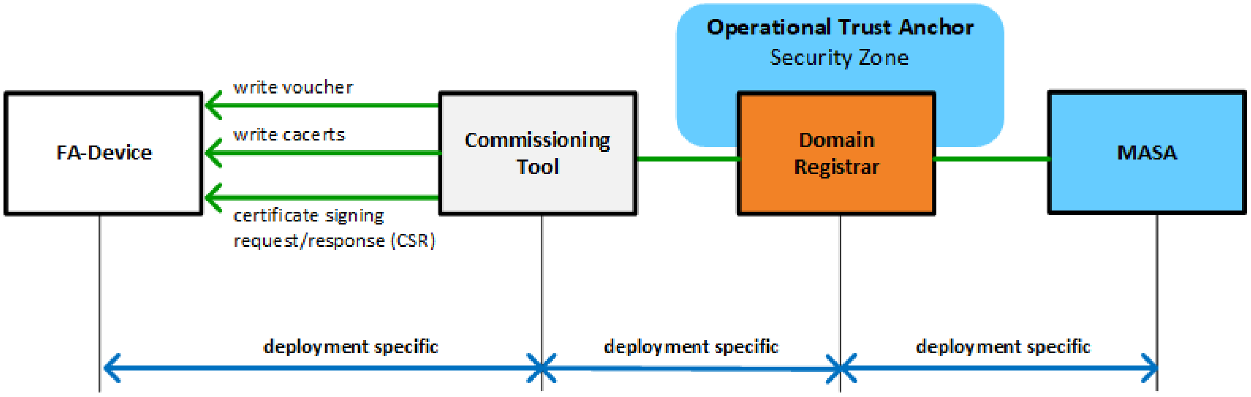


Figure 11 Out of Band CSR.

The FA-Device can choose to accept vouchers [RFC 8366] using less secure methods. These methods enable offline deployment use cases. In any case, the FA-Device MUST accept nonceless vouchers. This allows for a use case where the Registrar cannot connect to the MASA at the deployment time. The FA-Device MAY also support "trust on first use" for physical interfaces such as a local console port or physical user interface but MUST NOT support "trust on first use" on network interfaces.

If the commissioning tool acts as a proxy to a Registrar then the tool makes following requests to FA-Device resources.

In cases where the Registrar is not available at deployment time the commissioning tool SHALL use the following request in order to write CA certificates to a FA-Device (see also Trust List):
Request: POST ldevid/csr/crts { PKCS#7 according to [RFC 7030] section 4.1.3 + (CMS) registrar signed payload (cmsVersion=1 ≙ PKCS#7) }
The following request is used to read a CMS (including proof-of-identity) certificate signing request (CSR) from a FA-Device:
Request: GET ldevid/csr/cms **Response:** { PKCS#10 + (CMS) IDevID signed payload (cmsVersion=1 ≙ PKCS#7) }
The following request is used to write a device certificate signing response (PKCS#7) to a FA-Device:
Request: POST ldevid/csr/sen { PKCS#7 }

3.5.2.4 JWS Signed CSR

The [RFC 7515] CSR MUST contain an "iss" (issuer) claim, the application MUST validate that the cryptographic keys used for the cryptographic operations in the message belong to the issuer. If they do not, the application MUST reject the CSR. The [RFC 7515] CSR SHOULD contain an "aud" (audience) claim that can be used to determine whether the CSR is being used by an intended party (e.g. Registrar) or was substituted by an attacker at an unintended party. If present the Registrar MUST validate the audience value and if the audience value is not associated with the Registrar it MUST reject the [RFC 7515] CSR.

The "kty" (key type) parameter identifies the cryptographic algorithm family used with the key and MUST be present in a [RFC 7517]. The "use" (public key use) parameter identifies the intended use of the public key. The "x5c" public key or rather the [802.1AR] manufacturer device certificate is used to sign the payload.
The "x5c" (X.509 certificate chain) parameter contains a chain of one or more [X.509] certificates. The certificate chain is represented as a JSON array of certificate value strings.

The following JSON Signature [RFC 7515] message example shows the header and the payload structure of a CSR. The [X.509] manufacturer device certificate containing the public key value MUST be the first certificate and the issuer field (iss) MUST correspond to the subject distinguished name in the manufacturer device certificate.

Header:

```
{
        "kty":"EC",
        "use":"sig",
        "x5c":  " <X.509 manufacturer device certificate base64 encoded>"
}
```

Payload:

```
{
        "iss": "<serialNumber of X.509 manufacturer device certificate subject>",
        "aud": "<registrar>",
        "csr": " <PKCS#10>"
}
```

The following request is used to get a [RFC 7515] signed (proof-of-identity) certificate signing request (CSR):

Request: GET ldevid/csr/jws
Response: { PKCS#10/[RFC 7515] signed payload }

3.5.3 Manufacturer Trust Anchor

The manufacturer CA issues vendor pre-installed manufacturer device certificates (identity certificate). The offline root CA signs the issuing CA and is hosted in a high secure, offline environment.

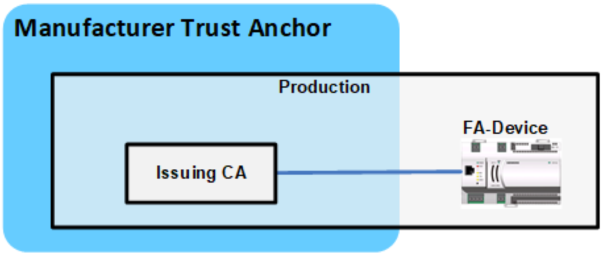


Figure 12 Manufacturer Trust Anchor.

The manufacturer trust anchor MUST be pre-installed on the FA-Device. The FA-Device will trust the manufacturer for purposes of introduction of the device to an operational network.

3.5.3.1 Manufacturer Device Certificate

The IDevID [802.1AR] manufacturer device certificate is an assertion of the FA-Device manufacturer as to the device's unique identity. The IDevID subject field MUST contain the "serialNumber" attribute with the device's unique serial number ([802.1AR] section 4.1.2.4). Typically, the manufacturer device certificate is permanently stored in a security module. This certificate is only used for commissioning until certificate bootstrapping has completed. The manufacturer device certificate MUST not be used for proof of identity after successful certificate bootstrapping but MUST only be reactivated after factory reset of an FA-Device. FA-Devices may make use of the manufacturer device certificate to create additional identities for other purposes. How that is done is beyond the scope of Fairhair.

The manufacturer DER encoded device certificate can be read without permission:
Request: GET idevid **Response:** { X.509 binary DER encoded }

The manufacturer device certificate is used for bootstrapping when the new FA-Device is not initialized. Manufacturer device certificates MUST have indefinite expiration dates, and hence MUST use the GeneralizedTime value [X.509]. Validating entities SHOULD ignore validity period information in the certificate. This ensures that FA-Device authentication can always be verified during [X.509] path validation. The Registrar and its network administrator are responsible for determining the trustworthiness of a manufacturer certificate and its signer (see Registrar).

A manufacturer MAY include a URL that points to a certificate revocation list (CRL) in the signing certificate for the device. A local network MAY check the CRL prior to registering a device.

3.5.3.2 Manufacturer Device Certificate Fields

The manufacturer device certificate SHALL support the following fields according to IEEE [802.1AR]:

Version field defines which [X.509] version applies to the certificate:
V3
Each certificate contains a **Certificate Serial Number** (unsigned integer of up to 20 octets) that distinguishes it from other certificates:
Example: 00:01:AB:CD
The **Algorithm information** used by the issuer to sign the certificate
See section 3.5.6.1
The **Issuer** field contains domain CA specific information:
O=[Manufacturer] CN=[Manufacturer CA]

Example: O=Siemens CN=Siemens Manufacturer CA
The certificate contains **Not Before** and **Not After** dates, which set the boundaries of the **Validity** period. **Not Before** SHALL be the time the certificate is created. **Not After** is the latest time the certificate is expected to be used. However, manufacturer device certificates are expected to operate indefinitely into the future and SHOULD use the GeneralizedTime [X.509]
Not Before: Aug 1 00:00:00 2016 GMT Not After : Dec 31 23:59:59 9999 GMT (GeneralizedTime)
The **Subject** contains FA-Device specific information:
O=[Vendor] CN=[Product ID] serialNumber=[PID:xxxx SN:yyy] Example: O=Siemens BT CN=PXC3.E75-100A serialNumber=123456
Subject Public Key is the public key associated with the identity
See section 3.5.6.1
Non-Critical: The **Subject Alternative Name** SHALL contain a unique name of the security hardware module:
otherName = [OID] [Serial Number] Example: 2.23.133.1.0 caffe987654321
Critical: The **Key Usage** extension defines the purpose of the key contained in the certificate. The KeyUsage field SHALL have the following value: keyAgreement.
Example: keyAgreement, Digital Signature
Non-Critical: The **Certificate Policy** extension defines the issuance and management of certificates. The field MUST contain at least one device type Identifier OID.
Example: 1.3.6.1.4.1.4329
Non-critical: Manufacturer Usage Description (MUD) URL
Example: https://example.com/.well-known/mud/v1/devicev1

The manufacturer usage description (MUD) URL points to a YANG-based JSON file that the manufacturer maintains, in which the recommended communication profile is communicated.

3.5.4 Domain CA

A Domain CA can be local as part of a tool (see Registrar) or a remote service from an enterprise network. The Domain CA is an entity that issues identity certificates (e.g. operational device certificates) for a particular entity. The Domain CA of the operational trust anchor certifies the ownership of a public key by the named subject of the FA-Device certificate to a security zone.

3.5.4.1 Domain CA Certificate for a Security Zone

This CA certificate acts as a trust anchor and has to be distributed to all FA-Devices that belong to the same security zone or have a trust relation to this system (such as conduit controllers). The domain CA certificate MUST have a lifetime that exceeds any certificate that it is used to sign.

Identifying how critical assets are connected within an overall network security architecture crossing various FA-Systems, network segments and security zones is a necessary step for risk analysis and management as well as the overall management of the security program for the owner deploying and operating the system. These considerations have a direct impact on how namespaces are structured (see Security Zones and Conduit).

FA-Devices within the same security zones are initially considered as trustworthy with no restrictions. FA-Systems are manifold and differ from site to site. In general, it is supposed that a security zone typically represents how functional or legal entities are organized in a building or campus.

3.5.4.2 Operational Intermediate Certificates

A device MUST support the ability to be provisioned with a single intermediate certificate that is signed by the domain CA, and that has signed the device's operational certificate. This certificate is provided to other devices to assist their validation of the device's certificate. Domains MUST NOT require more than a single intermediate CA within a device.

3.5.5 Operational Device Certificate

The LDevID [802.1AR] operational device certificate is a [X.509] certificate for a particular FA-Device that is, especially in large FA-Systems, in the responsibility of the network administrator. These certificates are used to identify the device to the network. They are also used to establish [RFC 7515] tokens, as required, and may be used and may be used for different purposes (e.g. at link layer, network layer or application layer) for application-layer authentication. An operational device certificate contains authentication information in the Subject Alternative Name (SAN), to associate it to a particular security zone. Hence [X.509] certificates MUST NOT contain multiple security zones (e.g. more than one rfc822Name field). In addition, the wildcard character ('*') is neither allowed in the Subject Alternative Name extension, nor in the common name. How an operational device certificate is issued depends on the certificate bootstrapping method, and on the particular certificate management lifecycle during operation. The FA-Device MUST have a new key pair before requesting a new operational device certificate. The FA-Device either generates the key pair locally if possible or obtains the key pair from a trusted Registrar or commissioning tool. All stored operational device certificates and its associated private keys MUST be deleted on factory reset of the FA-Device. Operational device certificates are expected to be periodically updated. The length of time a certificate is valid is up to the deployment. A recommended certificate lifetime is one year. However, if the deployment does not have mature processes to update certificates, then the lifetime should be infinite. Vice versa, if the deployment does not have a mature process to manage revocation of certificates, then the lifetime MAY be significantly less than one year (e.g. 30 days).

Operational certificate trust anchors may contain a pointer to a CRL. However, FA-Devices are not expected to support OCSP. See section 4.2.6 for appropriate interfaces.

The operational DER encoded device certificate can be read without permission:
Request: GET ldevid **Response**: { X.509 binary DER encoded}

3.5.5.1 Operational Device Certificate Fields

An operational device certificate SHALL support the following fields:

Version field defines which [X.509] version applies to the certificate:
V3
Each certificate contains a **Certificate Serial Number** that distinguishes it from other certificates:
Example: AB:CD:00:01:03
The **Algorithm information** used by the issuer to sign the certificate

See section 3.5.6.1
The **Issuer** field contains domain CA specific information:

C=[Country] O=[Vendor or installer] CN=[CA]
 	
Examples for site local offline CA (or managed enterprise network while commissioning):
C=CH O=Installer-Company CN=Installer-CA-Tool Installer-ID
C=CH O=Siemens BT CN=ca.bt.siemens.com/Email=info@ca.bt.siemens.com

The certificate contains **Not Before** and **Not After** dates, which set the boundaries of the **Validity** period. **Not Before** is the earliest time a certificate may be used. This SHALL be the time the certificate is created. Once a certificate's validity period has passed, a new certificate MUST be requested.
Example for enterprise networks with defined expiration date: Not Before: Aug 1 00:00:00 2016 GMT Not After : Dec 31 23:59:59 2020 GMT Example for unmanaged FA-Systems with indefinite expiration date: Not Before: Aug 1 00:00:00 2016 GMT Not After : Dec 31 23:59:59 9999 GMT (GeneralizedTime [X.509])
The **Subject** contains FA-Device specific information that are typically taken over from the manufacturer device certificate:
C=[Country] O=[Vendor] CN=[Product ID] Example: C=CH O=Siemens BT CN=PXC3.E75-100A serialNumber=123456
Subject Public Key is the public key associated with the identity
See section 3.5.6.1
Critical: The **Subject Alternative Name** MUST match either the Fully Qualified Domain Name (FQDN), security zone ID or the IP address of the host or, if present, all. Note that this document does not recommend use of IP addresses in certificates nor does it discuss the implications of placing IP addresses in certificates.

dNSName = [FQDN]
 	
rfc822Name = [Device ID + Security Zone ID] iPAddress = [ip]
 	
FQDN example: temperature-sensor-1.building-1.campus.net
Security Zone ID example: 123456+dev.sec-zone-56789@example.com

Critical: The **Key Usage** extension defines the purpose of the public key contained in the certificate. The KeyUsage field SHALL have the following value: keyAgreement.
Example: KeyAgreement
Critical: The **Extended Key Usage** is typically used on leaf certificates, to indicate the purpose of the public key contained in the certificate.
Example: extendedKeyUsage = serverAuth, clientAuth
The **CRL Distribution Points** extension identifies how certificate revocation list (CRL) information is obtained. This non-critical extension is only needed if the FA-System supports a CRL (e.g. on the Registrar).
Example: http://registrar. building-1.campus.local/crl/master.crl

3.5.6 Certificate Validation

FA-Devices MUST follow the procedure defined in RFC 5280 [X.509] to verify certificates. Certificate verifiers MUST reject certificates that contain one or more unsupported critical extensions. Any extension not listed by name within this document SHOULD NOT be included within a compliant certificate and, if included, MUST NOT be marked critical. In an authentication exchange, the FA-Device

SHALL supply a complete and valid chain comprising its own certificate and any intermediate CA certificate between the FA-Device and the Root CA.

FA-Devices uses [X.509] certificates for mutual authentication for device-to-device communication based on (D)TLS. Server identities MUST be checked as described in [RFC 2818] section 3.1. If a security zone is configured, then the client and server side MUST check the [X.509] subjectAltName (e.g. rfc822Name) field in the operational device certificate with the settings in the FA-Device Access Control List.

A certificate policy applies to the complete certification process, from root down to FA-Device, and this is reflected in the certificate by inclusion of the policy OID in all the certificates in the chain. Hence, policy-mapping is not supported, and certificates containing policy mappings MUST be rejected. Therefore, issuers SHOULD NOT include policy qualifiers in operational device certificates. However, verifiers SHOULD NOT reject certificates containing policy qualifiers unless there are other reasons to do so.

3.5.6.1 Device Certificate Cipher Suites

Elliptic curve cryptography (ECC) provides the cryptographic basis for secure communication with resource-constrained FA-Devices due to its small key size and comparably low arithmetic requirements. All FA-Devices supporting identity certificates [X.509] SHALL support the following cipher suite for communication: TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 [RFC 7251].

Specifically, the device identity certificate [X.509] will use ECC NIST P-256 (secp256r1 or prime256v1) [RFC 4492]. The signer of the certificate MUST also use NIST P-256 (secp256r1).

Cryptographic suites must be updated from time to time. As such, future releases of this specification may require more than one algorithm for both backward compatibility and improved security. Algorithms may also be deprecated.

3.6 User Management

3.6.1 User Identity Management

The user identity management is required to ensure that devices can be accessed by different users in the lifecycle with different rights. Identifying and authenticating users in an enterprise is often based on pre-existing mechanisms like LDAP or Active Directory. Once the user has established the identity, the user needs to securely establish his or her access rights towards the devices he or she communicates with. This is managed using an Authorization Server that can issue access tokens to users once they have established their identity and their role is known in the system. Devices need to trust the Authorization Server signed access tokens and provide the user with appropriate rights to access its resources.

3.6.2 Proof of Possession Access Token Based on Public keys

The user’s public key, respectively the associated private key, is used for proof of possession of the access tokens. The user public private key pair may be the existing user’s certificate issued by a user management CA or newly issued public keys that are signed by the authorization server.

3.6.3 User Credentials

People related user identity credentials (e.g. for Service Technicians) are typically issued from a facility manager. In some cases these credentials may be user certificates that provide authorization information directly and can be used to access device resources.

User certificates are stored as [RFC 7292] format which is protected with a password-based symmetric key. The [RFC 7292] defines a container format that includes the public certificate, the entire certificate chain including public key, and root certificates.

3.6.3.1 User Certificate Fields

A user certificate SHALL support the following fields:

Version field defines which [X.509] version applies to the:
V3
Each certificate contains a **Certificate Serial Number** that distinguishes it from other certificates:
Example: AB:CD:00:01:03
The **Algorithm information** used by the issuer to sign the certificate
See section 3.5.6.1”
The **Issuer** field contains domain CA specific information:
C=[Country] O=[Organization] OU=[General info, order or site id] CN=[Operational Trust Anchor ID] Examples: C=CH O=IAM-Provider OU=Campus-6789 CN=ca.access-trust-anchor
The certificate contains **Not Before** and **Not After** dates, which set the boundaries of the **Validity** period. **Not Before** is the earliest time a certificate may be used. Once a certificate's validity period has passed, a new certificate MUST be requested.
Example with defined expiration date: Not Before: Aug 1 00:00:00 2017 GMT Not After : Aug 4 23:59:59 2017 GMT
The **Subject** contains user specific information:
O=[Organization] G=[Name] SN=[Surname] CN=[Common Name] Example: O=Siemens G=Oskar SN=Camenzind CN=Oskar Camenzind
Subject Public Key is the public key associated with the identity
See section 3.5.6.1”
Critical: The **Subject Alternative Name** contains the Email address of the user:
rfc822Name=[Email] Example: rfc822Name=oskar.camenzind@example.com
Critical: The **Key Usage** extension defines the purpose of the public key contained in the certificate.
Example: keyEncipherment, KeyAgreement

4 Discovery

4.1 Key Design Principles

The FA-System has been shaped to its current form by the following key design principles:

1. Re-use of the IETF CoRE standards for CoAP discovery;
2. Seamless scalability from initially small networks (10 nodes) to large networks (100000s of nodes);
3. Cope with the different phases of installation, commissioning, operation, re-commissioning (with changing levels of system openness in this lifecycle);
4. Support for discovery on highly constrained (6LoWPAN) mesh networks;
5. Sleepy devices discovery support;
6. Security: limit the options for a compromised node to modify or spoof discovery information of a large number of other nodes;
7. Discovery of groups or group membership needs to be enabled;
8. Ability to deal with multiple Commissioning Tools operating in a system at the same time;
9. Cope with partial availability of network infrastructure during installation, as well with increasing size of the installation.

The design for FA-Discovery is based on some high-level choices:
Provides discovery for the resources of the RM (e.g. resources representing devices, objects, or groups if present in an ecosystem.
The RD itself implements the /.well-known/core interface. This interface might be used by FA-Devices to learn about the lookup and registration interfaces of the RD. It does not provide information about resources of other nodes that are registered at the RD.
FA-Discovery in general and especially registration at a resource directory is subject to security issues. These are covered in section 4.8.

4.2 Lookup and Registration Resources

The following sections defines the resources used for the "/.well-known/core" interface and the RD registration and lookup interfaces. Access to these resources might be restricted for security reasons. This is part of section 4.8.

4.2.1 Resource for the "/.well-known/core" Lookup Interface

All FA-Devices SHALL support discovery queries on the "/.well-known/core" resource.

4.2.2 Resource for the RD Lookup Interface

The Fairhair centralized discovery lookup interface is defined according to the "RD Lookup Function Set" [Chapter 8, RD], FA-System makes use of the functions of the resource related parts of the RD specification. The default paths to the respective resources and their discovery will be specified in section 4.6.2 .

4.2.3 Resource for the RD Registration Interface

The Fairhair centralized discovery registration interface is defined according the "Resource Directory Function Set", see section 4.6.2. The default paths to the respective resources and their discovery are be specified in section 4.6.5

4.3 Discovery Queries

This section explains the basic mechanisms, query formats, query response formats, and other requirements that are common over all the FA-Discovery operations. More details per discovery type are given in later sections.
Note that the distributed and centralized discovery use the same generic mechanism of link format querying for attribute values, which simplifies a constrained implementation. The actual resource path for lookups will be different for distributed and centralized discovery variants. To express this, we denote the lookup resource path as <lookup-path>. This resource path is either "/.well-known/core" on a FA-Device (see section 4.2) or is identified on a RD according to the process described in section 4.6.1.3 and 5.5.

4.3.1 Basic Query Format

An FA-Device MUST support the query filtering which is designated as optional in [RFC 7252] and [RFC 6690].
The below CoAP request format is used to do a one-attribute query:

```
GET coap://hostname/<lookup-path>?attribute=value
```

Where hostname is typically an IPv6 literal between square brackets, either a unicast address or multicast address. For example: [ff02::fd].
The specified content format for discoveries is "application/link-format+cbor". Thus, CoAP queries for the FA-Discovery SHALL be sent with the corresponding CoAP Accept Option. FA-Devices and RDs SHALL send their responses to such queries with the content format "application/link-format+cbor".

4.3.2 Multiple Query-Attributes Format

More attributes may be added using query parameters, for example:

```
GET coap://hostname/<lookup-path>?attribute=value&attr2=val2
```

The attributes MAY appear in any order, this order has no specific significance. All the query attributes are Boolean AND-ed by the receiving FA-Device to provide the query result.
Allowing multiple attributes also for the distributed discovery mechanism deviates from the [RFC 6690] specification which only allows to query one attribute. This might lead to unexpected behavior for non FA-Devices, such as errors or answers that do not follow the logic required by Fairhair. As best practice, only FA-Devices should be on the network. This could be enforced for example by a proper security setup.
The number of attributes for a query is not limited by the specification. An ecosystem MAY impose limits on the number of query attributes to reduce complexity in constrained implementations. Linkage and resolution (section 5.5) utilizes 2 arguments. Depending on the description of links, even more arguments might be desirable – concrete requests lead to better and smaller responses and reduce the parsing effort.

4.3.3 Wildcard Usage and Hierarchy

As defined by [RFC 6690] the asterisk '*' wildcard can be used in any query attribute value:

```
GET coap://hostname/<lookup-path>?attribute=partialValue*
```

Were the partialValue is a partial value prefix string. This is used to search on multiple values that start with a given string prefix.
The IETF [RFC 6690] allows that the partial value string is empty and uses this syntax to check for the existence of attributes. As a consequence of this, attributes may also take the form of a tag, i.e. an attribute without value. Such attributes are queried according [RFC 6690] with the query:

```
GET coap://hostname/<lookup-path>?attribute=*
```

In case of multiple attributes without values the query takes the following form:

```
GET coap://hostname/<lookup-path>?attribute_1=*&attribute_2=*
```

The ecosystem specifications MAY define a basic hierarchy in their attribute value namespace, such that queries on overarching categories are enabled. This holds for all attributes, including the [RFC 6690] 'rt' attribute (and its FA-Systems usage, as defined in section 5.2, as well as for other ecosystem-specific attributes or (value) tags see section 5.4.

4.3.4 Basic Response Format

The default response to a FA-Discovery query SHALL consist of a unicast 2.05 CoAP response with the resulting CoRE Link Format [RFC 6690] payload encoded as "application/link-format+cbor" [Links-JSON]. This payload may also contain CoRE link attributes (in the following often abbreviated as 'attributes') without value, as shown in the second link in the example below. For better readability, the examples in this document are given in link-format, as indicated below:

```
2.05 Content

<coaps://[nodeIPv6Addr]/uriPath_a>;attr_1=value1;attr_2=value2,
    <coaps://[nodeIPv6Addr]/uriPath_b>;attr_no_val_1;attr_no_val_2,
    <other link 2>
```

The CoAP Token in the response MUST be the same as the request Token.
The response to a FA-Discovery query SHALL contain all matching resources that fulfil the query. The resources in a response may contain the full URI or be relative if the resources are reachable over the default secure "coaps://…" scheme and hosted by the queried node. This deviates from [RFC 7252] but is more efficient due to the fact that the specification (see 3.4.2) recommends the secure scheme as default. For performance reasons, relative links SHOULD be preferred. A client SHALL be able to process absolute and relative URIs.
The following example shows the above shown response with relative paths, the response will be sent from the `[nodeIPv6Addr]`:

```
2.05 Content

</uriPath_a>;attr_1=value1;attr_2=value2,
    </uriPath_b>;attribute_no_val_1;attribute_no_val_2,<other link 2>
```

The attributes returned in a response to a FA-Discovery query are defined by the responding node resp. RD, i.e. the response MAY contain only the matching resources and no attributes at all, or some attributes only, or the matching resources and all registered attributes. A client SHALL NOT rely on receiving the full metadata information via the FA-Discovery mechanism.
Note: the definitions in [RFC 6690] do not indicate the attributes returned in responses to FA-Discovery queries. Existing implementations (e.g. Californium) return the full list of attributes, though this list

might be large and lead in the case of queries with multiple matches to a high load on a constrained network. It is beyond the scope of this document to prescribe how systems efficiently handle FA-Discovery. The specification mandates support for multiple query parameters, which allows to restrict the length of responses to queries.
Note: alternative content encodings might be added in future. This will also require a definition of the usage of the CoAP accept option.

4.3.5 Attributes and Attribute Values

FA-Devices will follow [RFC 6690] in using attribute names and attribute values to characterize resources for discovery purposes. The FA-System defines some specific attribute names (and values) and their proposed usage in section 5.2. In general, the mechanism is kept extensible, and it is expected that the ecosystems and potentially vendors and system engineers will make use of own definitions of specific attribute names and values. The following sections give the rules (in accordance with the respective IETF RFC) for the specification of attribute names and values.

4.3.6 Attribute Names Syntax

Besides the Link Format attributes defined in this specification, an ecosystem MAY define its own specific Link Format attributes used for discovery.
The following syntax rules hold for Link Format attribute names:

- It SHALL comply with the ABNF syntax for ‘parmname’ defined in [RFC 8187].
- Reserved characters [RFC 3986] SHOULD NOT be used in names.

The above rules imply that an attribute name can contain alphanumeric characters and the characters hyphen “-“, dot “.” ,underscore “_” and tilde “~”. The latter special characters are in the unreserved [RFC 3986] set.

4.3.7 Attribute Values Syntax

The following syntax rules hold for Link Format attribute values:

- It MUST comply with ABNF syntax for ‘link-param’ defined in [RFC 6690] for those IETF-defined attributes on page 7 of RFC 6690 listed under the ‘link-param’ ABNF list.
- It MUST comply with the ABNF syntax defined in [RFC 6690] for either one of
  - ‘reg-rel-type’ – which is lowercase letters, numbers, dot and dash.
  - ‘quoted-string’ – which is any characters within quotes except unescaped quote (“) itself.
- Values SHOULD NOT use ‘quoted-string’ wherever possible to conserve bandwidth since quotes are not really needed in practice.
  - For example, unquoted URNs are allowed.

Some of the attributes defined by the IETF or other standards (e.g. the ‘rt’ attribute) limit the syntax for their values stronger than the general rules. In such case, the most stringent limitations are applied.

4.4 Multicast Operation

The definitions in this section apply to the distributed discovery on the “/.well-known/core” interface. The RD lookup interface SHALL NOT support multicast.

4.4.1 All CoAP Nodes Multicast Address

The IPv6 address range ff0x::fd has been reserved by IANA as the “All CoAP Nodes” variable scope multicast address [RFC 7252]. The “x” designates the 4 bits that declare the multicast address scope; all scopes can be used in principle.
This multicast address (ff0x::fd) and the default port 5683 (according [RFC 7252] section. 7.1) is used in CoAP for distributed discovery.
A FA-Device SHALL support CoAP distributed discovery by listening to this multicast address on the following scope:

- Link-local (2)
    - Typically used to query in a single Wi-Fi or Ethernet network segment

An FA-Devices connected via a 6LoWPAN based mesh network that supports Realm-local scope [RFC 7346] SHALL support multicast for the distributed discovery also on the following scope:

- Realm-local (3)
    - Typically used to query in a single mesh topology IPv6 network
    - Note: realm-local is defined for 6LoWPAN and currently only implemented in certain IPv6 mesh network standards, e.g. Thread.
    - A LAN or Wi-Fi FA-Device SHOULD NOT send realm-local multicast messages.

An FA-Devices SHOULD NOT support any higher scopes for discovery, for security reasons.

4.4.2 Response Suppression

If there is no matching result i.e. the query result set has zero elements, and the request was sent as multicast, then a response to the query request MUST NOT be sent. This behavior complies with [RFC 6690] section 4.1.
If a multicast request was sent, any error responses (CoAP 4.xx or 5.xx class) MUST be suppressed by the responding CoAP server.

4.4.3 Response Timing

Any response to a multicast query is to be delayed by the CoAP server by a randomly chosen duration between 0 and DEFAULT_LEISURE (default is 5 seconds, but might be adjusted, [RFC 7252]).

4.5 Unicast Operation

This section applies to all unicast operations in the context of FA-Discovery, i.e. unicast interactions on the “well-known/core”, “RD lookup” and “RD registration” interfaces.

4.5.1 Error Responses to Queries

Error responses are generated when a unicast request was sent, and an error occurred in processing the request. The applicable response codes including error responses are defined in [RFC 7252] section 12.1.2. Specifically the case where a query contains too many elements for the responding FA-Server to handle is relevant for FA-Discovery. This includes, for example:

- Too many query parameters;
- Too long strings in query parameters, or other parts of the URI;
- Too long URI in total.

In such situations, the FA-Server SHOULD respond with a 4.13 Bad Request. This might indicate to the FA-Client that it should adapt its query before sending it again.
It might happen, that the response to a discovery request (holds for all requests) would be too large to construct or send for the server. In absence of a specific error code for this situation, the server could indicate this with an error code 4.00 bad request to indicate that the client might change / restrict the discovery query.
Since discovery queries might lead to long answers, especially if filtering is not implemented fully, FA-Clients need to support the option of CoAP Block-Wise Transfers [RFC 7959].

An exception is however when the CoAP server is only temporarily occupied and will soon (within some seconds) be available to process larger queries. Then, the FA-Server SHOULD respond 5.03 Service Unavailable. (Note: this definition is also relevant for the handling of other FA-System communication and might be moved to a different part of the specification addressing the general usage of CoAP within FA-System.)
A response to a unicast query SHOULD NOT be delayed on purpose by the FA-Server.

4.6 Resource Directory (RD)

The Resource Directory (RD) is a repository providing information about FA-Devices and their resources in the network. It is accessed per unicast, and thus allows discovery beyond the limitations of multicast scopes and multicast performance issues, such as scalability and collisions. An RD might also be used to discover devices that are not permanently available, e.g. sleepy devices. Albeit, the RD is not intended to serve as data proxy for sleepy devices.
The RD coexists with the multicast discovery on the "/.well-known/core/" interface supported by all FA-Devices. The level of information (i.e. attributes and attribute values) provided is not always necessarily identical, e.g. in setups, where the RD is used during operation while the "/.well-known/core" interface is only used in bootstrapping.
The RD provides in accordance with the IETF RD [RD] separate function sets for registration of resources and the lookup (discovery) of resources. The IETF RD [RD] lookup function set (interface) provides additional capabilities to discover endpoints, domains or groups. To maintain the compatibility of the lookup functionality in line with queries in the distributed discovery via the "/.well-known/core" interface, the lookup of resources is the only lookup functionality specified for the FA-System.
The RD provides also a /.well-known/core" interface. This interface can be used to discover the RD itself, but does not provide information on other resources but the RD itself.
The IETF RD [RD, Chapter 5] mandates the implementation of the "Simple Registration". RDs SHALL disable this for security and scalability (e.g. such as extensive requests into a constrained network) reasons.
The RD draft [RD] is in principle applicable not only over CoAP but also over HTTP. FA-RDs SHALL implement the CoAP interfaces.
FA-System supports deployments with multiple RDs (see section 4.7).
The FA-RD and especially the registration come with security concerns - those are given in the security aspects section 4.8.

4.6.1 FA Discovery / Configuration of the RD

FA-Devices need either to discover the FA-RD or be configured with the address (and registration and lookup resources) of the RD. Several options, depending on the network configuration, exist for this task:

- Discovery and interface identification via “/.well-known/core”;
- Configuration;
- IPv6 ND option;
- DHCPv6 option;
- If DNS is supported, a DNS service (and DNS Service Discovery) might be used.

A RD SHALL provide the identification of its lookup interfaces via its “/.well-known/core” interface over unicast and multicast (section 4.6.1). FA-Devices SHALL support the configuration of RDs (see section 4.6.1.4). The other options will be configurable and implemented according a conformance statement. Interface identification with a multicast discovery is limited to the corresponding multicast scope. The other methods can be applied also in case the network extends over multiple subnets that are not covered by a multicast scope.

4.6.1.1 Precedence of RD Information on FA-Devices

RD information can be configured or learned (all methods but the configuration method) on FA-Devices. Configured configuration SHALL have precedence over learned configuration, i.e. the configured RD information is used and any other learned information that might be available is ignored.

4.6.1.2 Storage of RD Information on FA-Devices

FA-Devices SHALL store configured RD information in permanent storage (i.e. in non-volatile memory (NVM)), so that this information is present after a reboot of the device. FA-Devices MAY also store learned RD information in NVM, to speed up the process after a reboot of a device.

4.6.1.3 RD Interface Identification

Within a supported multicast scope (according section 4.4.1), the RD function sets are discovered via the “/.well-known/core” interface according to IETF RD [RD], i.e. using the distributed discovery lookup interface as defined in section 4.2.1 and filtering:

- The query “rt=core.rd” is used to discover the registration function set;
- The query “rt=core.rd-lookup-res” is used to discover the lookup set.

The RD SHALL support this form of discovery for multicast and unicast queries.
The interface identification needs also to be applied in any case where a FA-Device is aware of the IP address of RDs. Then the query is done in form of a unicast query. In case of a multiple RD deployment, some RDs may only implement and provide the lookup interface.
The response to this discovery request has the same content format encoding as other FA discoveries, i.e. application/link-format+cbor [Links-JSON] (see section 6.2.4).

4.6.1.4 Configuration of FA-Devices with RD Locations

Configuration of information about RDs on FA-Devices are managed and configured per default by the resource which is annotated with “rt=fa.rd-conf”.
CoAP Interaction with the resource with resource type “rt=fa.rd-conf”.is as follows:

- PUT to the resource URL replaces the information;

- GET to the resource returns the information;
- Deleting information is done by writing (PUT) with empty payload.

The information about RDs is encoded in link-format, i.e. identical in form and content as if discovered via the "/.well-known/core" resource of RDs.
Access to the resource with resource type "rt=fa.rd-id" SHOULD be subject to authorization

Example:
The following example shows the configuration of a RD with registration and lookup resource and a second RD with a lookup resource only. The RDs accept application/link-format (ct=40) and application/link-format+cbor (the "ct" will be defined by IANA in [Links-JSON]*)

```
Req:     PUT coap://[a_node_ip]/.fa/rd/conf
         <coap://[rd_ip1]/rd>;rt="core.rd";ct=40;ct=xyz,
         <coap://[rd_ip1]/rd-lookup/res>;rt="core.rd-lookup-res";ct=40;ct=xyz,
         <coap://[rd_ip2]/rd-lookup/res>;rt="core.rd-lookup-res";ct=40;ct=xyz

Res:     2.04 modified
```

4.6.1.5 Configuration of FA-Devices with RD Identities

Configuration of information about RDs on FA-Devices are managed and configured by the resource which is annotated with "rt=fa.rd-id".
CoAP Interaction with the resource with resource type "rt=fa.rd-id" is as follows:

- PUT to the resource URL replaces the information
- GET to the resource returns the information
- Deleting information is done by writing (PUT) with empty payload.

The information about the IDs of RDs is a JSON / CBOR document containing IDs of potential RDs that might be present in a system and discovered / learned. Detailed information might be found in the security specification [FA-Security].
Access to the resource with resource type "rt=fa.rd-id" SHOULD be subject to authorization.

4.6.1.6 IPv6 ND Option

The IPv6 ND option for announcing RDs is defined in the RD draft [RD]. Configuration of the router for the respective subnetwork is needed.
The option can be present multiple times in a router announcement FA-Devices SHOULD evaluate all occurrences of the option to support multiple RDs.

4.6.1.7 DNS Service Discovery

A RD may be configured in the DNS server as DNS service and in turn discovered by DNS queries. If it is configured, it SHALL use the following name:

"rd._sub._coap._udp"

4.6.2 Registration at the RD

The RD registration interface defined in [RD] can be used by:

- FA-Devices to register their own resources including any FA-System entry point(s); and
- Commissioning Tool (CT) to register such items on behalf of an FA-Device.

This interface is secured according the requirements in section 4.8.

The registration is done following the definition in [RD] section 6.3.
According to [RD] a lifetime of 86400 seconds (24 hours) is used as default. If no update is done within this period, the registration is removed from the RD. For registrations performed with a CT, it is recommended to increase the lifetime of the registration to the maximum number of 4294967295 s (~136 years) to be larger than the expected lifetime of the deployment.
Registrations by a commissioning tool are implemented via the “base” parameter defined in [RD] section 5.3. This implies, that a registration has to be done individually for each combination of scheme, host and port.
The registration interface SHALL support the content format “application/link-format+cbor”.

4.6.3 Update of Registrations at the RD

The update is done following the definition in [RD] section 6.4.
The update will in most cases just renew the lifetime, keeping the registration valid. It also allows to change existing registrations, e.g. to add attributes.
This interface is secured according the requirements in section 4.8.

4.6.4 Deletion of Registered Resources at the RD

Endpoints and their entries that are no longer available SHOULD be deleted from the RD. This is implemented either by the soft state of the registrations or by explicit deletion of endpoints of devices that are aware of the removal, e.g. in case of a device shutdown. The process follows the definition in [RD] section 5.4.

4.6.5 RD Lookup

The RD lookup interface, as defined in [RD] section 8, can be used by:

- FA-Devices to discover resources in the network; and
- Commissioning Tool(s) to discover resources in the network.

The queries to the RD lookup interface are identical to the queries supported on the “/.well-known/core” interface, i.e. are formed according the definitions in section 4.3. For this purpose, this specification makes the lookup of resources according the resource lookup mandatory part of the RD implementation [RD] section 7.1.

4.7 Multiple RDs

A FA-system deployment may implement multiple RDs to avoid single points of failure and to allow balancing load.
In the case that a device is aware of multiple RDs, all those RDs SHOULD eventually be consistent and provide the same information on the lookup interface and respectively same registration functionality on the registration interface (if such interface is available). As consequence, all those RDs need to be fully synchronized. Any change to the content of any of the respective RDs is synchronized to all other respective RDs.
Note: The FA-System does not prescribe strict rules for the timing of the synchronization. It might take some time for all RDs to be in sync. The time for synchronization is expected to be in the range of seconds to a few minutes.
In a system with multiple RDs, some RDs may only implement the lookup interfaces. Registration is done then with RDs that implement also a registration interface. All registration interfaces available to a FA-

Device should provide the same functionality, i.e. it is sufficient for the FA-Device to register to any one of them.

For the time being, the FA-System does not define the mechanisms to synchronize RDs. This synchronization needs to make sure, that the links returned after a registration are valid for the full setup and are not in conflict. Synchronization mechanisms could be added later to provide interoperability between RDs of different vendors.

The usage of multiple RDs is local matter.

The FA-System does not define the mechanisms for redundancy or load balancing. The related strategies are local matter.

Example configurations might consist of a single registration interface and multiple lookup interfaces, where the first interface is selected per default and the others are only utilized in case of errors. This would put the focus mainly on redundancy.

4.8 Security Aspects

Malfunctioning or compromised RD have effects that affect potentially the full installation in the building. The main risks seen are:

- Rogue entity registers, updates or deletes information pertaining to other FA-Devices
- Rogue entity impersonates as the RD and either responds to RD lookup or registration queries by providing falsified information or losing information
- Rogue entity modifies queries to the RD lookup interface thus falsifying the information provided back (man in the middle)
- Rogue entity reads out the RD, thus acquiring info on the availability of FA-Devices and resources in the network
- DoS attack, making the RD unavailable

The approach to secure FA-Discovery follows the guidelines of section 3. Implementations of the FA-System need to support the full range of security mechanisms, the activation may depend of the choices of the deployment.

The solution assumes, that the FA-Devices are provided with the needed certificates and all other information needed to communicate, i.e. have been security bootstrapped.

5 Resource Identification

This section gives rules on the FA-Resources that need to be exposed for discovery and provides guidelines how the ecosystems could use predefined attributes as well as ecosystem or installation defined attributes.

5.1 Resource Model Concepts

The RM provides some concepts that SHALL be exposed to be able to discover devices for the purpose of management and commissioning. Those resources are:

- Ecosystem entry points; and
- Common services resource.

Further, it SHALL be possible to identify the ecosystem to which each exposed resource belongs.

Besides the basic rules below, the definition of attributes and respective values should be in the responsibility of the ecosystems.

5.2 Resource Type ‘rt’ Attribute

FA-Discovery makes frequent use of the resource type, represented by the ‘rt’ attribute (see [RFC 6690] section 3.1) for selected CoAP resources hosted on FA-Devices. Every resource that is discoverable using the FA-Discovery MUST have an ‘rt’ attribute associated.

The resource type value for a resource SHALL contain an identification of the respective ecosystem and is to be interpreted as an identifier for the type of the resource.

The value format and actual values of the ‘rt’ Link Format attribute are defined by the ecosystem(s).

Note: though being a definition by the ecosystems, it is expected that the $type metadata will somehow be represented in the rt attribute value.

For example, a resource representing a Zigbee Level Control Cluster would be annotated with a ‘rt’ attribute value of “urn:zcl:c:s.42” or “urn:zcl:LevelControl” or simply “zcl.42”.

According to [RFC 6690] and the FA-System guidelines for attribute values (section 4.3.5) the following types of resource type (rt) attribute values are all allowed:

1. CoRE resource type(s)
   These values MUST be registered in the CoRE Parameters Registry for Resource Type (rt=) Link Target Attribute values [RFC 6690].
   Examples:
   a. Zigbee alliance could register the entire namespace of resource types zcl.* in this registry, provided that IANA allows this;
   b. zcl.42 or za.cl.42 (note that these values can be written always without quotes as they are of reg-rel-type [RFC 6690]).
2. URIs
   For example, URIs encoding semantic identifiers according to some ecosystem defined ontology or taxonomy, such as http://sweet.jpl.nasa.gov/2.0/phys.owl#Temperature
3. URNs (as special case of a URI)
   URN namespaces at top level MUST be registered at IANA in the “Official IANA Registry of URN Namespaces”.
   Example:
   a. urn:zcl:* or urn:za:* namespace could be registered to the Zigbee Alliance. A specific ‘rt’ URN for a cluster could then look like, for example, urn:zcl:c:c.42 or urn:za:cl:42.

The choice of above categories and syntax/semantics of the attributes, including resource type (rt) values, are defined in the respective ecosystems. However, URIs type 2 SHOULD NOT be used by an ecosystem as using URIs tends to lead to inefficient, long Link Format descriptions.

5.2.1 Resource Type for the Ecosystem’s Entry Point Resource

Each ecosystem entry point on a FA-Device MUST have a specific resource type (‘rt’ attribute) that is chosen by the respective ecosystem.

For example (multiple alternative values are given sometimes, varying in size):

- The ecosystem entry-point resource for Zigbee might have the resource type
  - rt=urn:zcl;
- The ecosystem entry-point resource for KNX might have the resource type
  - rt=knx, or
  - rt=urn:knx, or
  - rt=knx;
- The ecosystem entry-point resource for BACnet might have the resource type
  - rt=bac, or

- o rt=urn:ashrae:bacnet, or
- o rt=urn:bac0.

Each ecosystem must define the specific values it wants to use while following the guidelines of this and the following section; above names and values are only examples.
Note: probably the ecosystem entry point needs a more specific attribute value to discern the entry point from the eco system identifier. Though, it is reasonable to keep the ecosystem identifier short. This might be e.g. rt=bac.ep, knx.ep or urn:zcl.ep.

5.2.2 Resource Type for Common Services Resource

FA-Resources are managed over the Fairhair resource tree (see section A.1) Therefore, the common services resource needs to be discoverable. The common services resource might have any name and be at any place in the resource tree.
The common services resource is defined with the resource type:

- ● rt=”fa.cs”.

5.3 IETF defined attributes

The IETF defines a number of attributes that could be useful for describing resources. Overloading those with originally not intended meaning might lead to unwanted side effects. The usage of those attributes is in responsibility of the ecosystems.
The following list contains the IETF defined attributes, their origin and intention as well as remarks from the FA-System point of view.

Table 2 IETF defined attributes.

Attribute	Origin	Intention	Remarks
if	[RFC 6690]	Interface	Restrictions (similar to “rt”), IANA registry exists for values.
ep	[RD]	Endpoint name	Mandatory parameter for registration at RD, must be unique.
et	[RD]	Endpoint type	Used in the [RD] to indicate types for endpoints – the only example with the attribute value of ‘core.rd-group’ is used to indicate a group endpoint. There is no equivalent for the ‘/.well-known/core’. Regular resources are of the type ‘core.rd-ep’, this is not used in any example.
ins	[RD-DNS-SD]	Instance	

5.4 Tags and Metadata

In some ecosystems it is required that FA-Devices and resources expose (parts of) their metadata for discovery. For this reason, FA-System supports annotation of all resources including ecosystem entry

points with arbitrary attributes and respective attribute values. Running queries on these attributes is also supported in all discovery operations. Those attributes may have the form of a tag, i.e. a link format attribute without value or as a full attribute (with key and value).
FA-System is not defining vocabularies, in principle. It will only define the syntax rules for attributes and how to discover attributes (in form of tags or complete attributes with name and value). The ecosystems or other organizations need to define those vocabularies and their mapping to attributes. If needed, the ecosystems should also register their vocabularies at the IANA.
Each ecosystem SHALL avoid clashes in tag namespace, either by

1. Defining tags within its own separate tag namespace (prefix string); or
2. Cooperation with other ecosystems or SDOs to use (industry-standard or de-facto standard) tagging schemes.

If an ecosystem uses its own separate tag namespace the prefix string SHOULD be as short as possible (e.g. 4 characters or less) and SHOULD end with a dot '.' (or a hyphen "-", underscore "_", tilde "~") character to denote namespace hierarchy. For example, ZCLIP may use "zcl." for attribute and tag names prefix.

5.5 Linkage and Resolution

FA-Resources may be identified by a Link Identifier and a relative path component.

To identify a specific resource a Logical Resource Identifier (LRI) is used, this LRI can be generated as UUID by the FA-Device.
A Link Identifier is defined by the scheme: "iot://<LRI>".
A FA-Resource reference can be identified by its Link Identifier and relative path component.
An example to reference to a resource by means of the Link Identifier is
`iot://[88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49]/color-temperature`

The FA-device SHALL register the resource with the LRI included as "ep" attribute value in the RD.
Example with the LRI as "ep" attribute in link format representation:

```
<coap://[ff35:30:2001:db8::1]/lamp>;if="core.p";
       rt="incandescent-lamp";
       ep=" 88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49 "
```

The use of the ep-parameter for the LRI is not necessarily aligned with the future use of ep for other links.

5.5.1 Resolution of the Link Identifier

The Link Identifier is used to store references to other resources within FA-Devices. Since the Link Identifier is independent of IP addressing it can be used to resolve relations between FA-Devices after changes to the network addresses.

The following example illustrates the use of the Link Identifier:

Resource reference: `iot://afabc221-09bb-48c3-999b-90dd0589e580/cert`
with Link Identifier "`iot://afabc221-09bb-48c3-999b-90dd0589e580`" and relative path component "`/cert`".

A resolution of the LRI in the Link Identifier will lead to a URI (‘base URI’ in [RD]), that might include a ‘base’ path:

```
coaps://device-1a2b3c4f.local/mgmt/fa
```

The full resource URI is then obtained by appending the relative path component to the URI:

```
coaps://device-1a2b3c4f.local/mgmt/fa/cert
```

In use with the RD and .well-known/core, the URI will contain an IP address without need for further resolution.

5.5.2 Implementation with the RD and “.well-known/core”

Linkage and resolution is performed via the ep attribute. The LRI is set as attribute value of the “ep” attribute.

Registering at an RD might use the “base” attribute (as defined in [RD]) to register the base path.

To keep the identical behavior of RD and “.well-known/core” interfaces, the “rt=fa.lnk” attribute is used to identify Linkage Resources.

Alternatively, the RD might be queried with the ep-lookup interface. This will not require the "rt=fa.lnk" attribute but will enforce different queries for RD and ".well-known/core".

The following example shows the interaction with an RD:

- registration at RD (for FA-Device: `coap://[2001:db8:3::129]:61616`)

```
Req: POST coap://rd.example.com/rd?ep=afabc221-09bb-48c3-999b-90dd0589e580
                                &base=coap://[2001:db8:4::3]:61616/sensors
     Content-Format: 40
     Payload: </s/t>;ct=41;rt="temp-c";if="sensor",
              </s/l>;ct=41;rt="lgt-lux";if="sensor",
              </>;rt=fa.lnk

Res: 2.01 Created Location: /rd/4521
```

- ‘rd-lookup’ on ep will give all registered links

```
Req: GET /rd-lookup/res?ep=afabc221-09bb-48c3-999b-90dd0589e580

Res: 2.05 Content
     <coap://[2001:db8:3::129]:61616/sensors/s/t>;ct=41;rt="temp-c";if="sensor“;
           ep=afabc221-09bb-48c3-999b-90dd0589e580,
     <coap://[2001:db8:3::129]:61616/sensors/s/l>;ct=41;rt="light-lux";
           if="sensor";ep=afabc221-09bb-48c3-999b-90dd0589e580,
     <coap://[2001:db8:3::129]:61616/sensors>;rt="fa.lnk";
           ep=afabc221-09bb-48c3-999b-90dd0589e580
```

- The ‘rd-lookup’ might also be filtered using the "fa.lnk" rt value to return only the Link Identifier including the base path

```
Req: GET /rd-lookup/res?ep=afabc221-09bb-48c3-999b-90dd0589e580&rt=fa.lnk

Res: 2.05 Content
     <coap://[2001:db8:3::129]:61616/sensors>;rt="fa.lnk";
           ep=afabc221-09bb-48c3-999b-90dd0589e580
```

- Example with "/.well-known/core", "ep". The "rt" must be provided as an attribute of the resources

```
Req: GET [multicast]/.well-known/core?ep=afabc221-09bb-48c3-999b-90dd0589e580
&rt=fa.lnk
Res: 2.05 Content
     <sensors>;ep=afabc221-09bb-48c3-999b-90dd0589e580
```

5.6 Addressing

Each FA-Device is uniquely addressed by its IP address.
An ecosystem should use the default CoAP port 5683.
Ecosystems should not assume the ecosystem entry point is hosted as a direct child resource of the CoAP server root on the port, or at any fixed resource path on the CoAP server. An FA-Client should anticipate that an arbitrary path could identify an ecosystem entry point. This allows the implementation of multiple ecosystem entry points in one FA-Device.

5.7 CoAP Usage

This section specifies how the Constrained Application Protocol (CoAP) [RFC 7252] and its related specifications from IETF are used in a FA-Device. Implementers MUST follow all the mandatory functions listed in CoAP [RFC 7252] and its related specifications as listed in section 5.7.1, except where stated otherwise in this CoAP usage section. Recommended functions listed in CoAP [RFC 7252] and the related specifications SHOULD be applied, unless stated otherwise in this section. For optional functions, the requirements are defined in this section.

5.7.1 Supported IETF Specifications

A FA-Device MUST support Constrained Application Protocol (CoAP) [RFC 7252] in the CoAP server role. To generate any CoAP requests itself, it MUST implement the CoAP client role also.

A FA-Device MAY support the CoAP block-wise transfers specification [RFC 7959] in its role as a CoAP server. This implies that a non-Fairhair CoAP client interacting with a FA-Device, such as a discovery tool or other off-the-shelf CoAP client, is expected to support CoAP block-wise transfers to be able to access resources that are served in a block-wise fashion.

A FA-Device MAY support block-wise transfers as a CoAP client.

A FA-Device typically does not support the observation of CoAP resources specification [RFC 7641] in a CoAP server role and/or in a CoAP client role, because the ecosystems using the FA-System define alternative mechanisms for this. However, a vendor or ecosystem still MAY add observation functions to the CoAP client or server in a FA-Device, without impacting overall interoperability.

A FA-Device SHOULD support the No-Response Option [RFC 7967] to be able to control the response generation for multicast requests with fine granularity. This is used for (automated) diagnostics of group communication, or to suppress superfluous success (2.xx) responses for Non-Confirmable CoAP commands when these commands are sent in quick succession (e.g. "dim up").

5.7.2 REST Methods

A FA-Device MUST support all the CoAP methods specified in [RFC 7252]: GET, PUT, POST, DELETE.

In addition, a FA-Devices CoAP server MAY support other CoAP methods defined by IETF but these are not used in this specification and not encouraged.

5.7.3 Protocol Functions

5.7.3.1 Multicast

Multicast CoAP request processing MUST be supported by a FA-Device CoAP server. Any CoAP resources for which multicast usage is not explicitly defined by an ecosystem specification or by this specification MUST NOT respond to multicast requests in any way. The rules for multicast request acceptance and response suppression of [RFC 7390] section 2.7 MUST be adhered to, with the default configuration set to suppression of any empty responses and any error responses. In a multicast request the No-Response Option [RFC 7967] can be used to modify this default behaviour, e.g. to disable default suppression rules.

In case a multicast request leads to a response from a CoAP server on a FA-Device, the server by default applies the “leisure” period as described in [RFC 7252] section 8.2 with the parameter DEFAULT_LEISURE as defined in [RFC 7252] section 4.8. An ecosystem MAY specify a different leisure value for specific multicast-supporting resources, or even another value for DEFAULT_LEISURE for all multicast-supporting resources.

5.7.3.2 Message deduplication

A FA-Device CoAP server SHOULD apply CoAP message deduplication as described in [RFC 7252] section 4.5 for non-idempotent POST requests. It MAY apply deduplication for the idempotent GET, PUT and DELETE requests.

Informative: If deduplication is not implemented for unsecured (coap://) POST requests, there is a fairly high probability that ecosystem commands – typically implemented as POST requests – sent using confirmable (CON) CoAP messages over wireless networks will be executed twice by the receiver due to CoAP-level retransmissions, in cases where ACK messages from the server are lost. Therefore, deduplication is important. The memory usage of a deduplication function can be dramatically lowered by using sequential MIDs (as in [draft-ietf-lwig-coap] section 3.5).

5.7.3.3 ETags and Validation requests

A FA-Device CoAP server MAY use an ETag Option in a response for the purposes described in [RFC 7252] section 5.10.6. However, a FA-Device will typically not need to use this, as Fairhair metadata provides alternative ways of timestamping resources. A CoAP client in a FA-Device MAY use an ETag Option in a request, but only using ETag values that where actually produced by the target server in an earlier request to the same resource.

5.7.3.4 Proxy Operation

A FA-Device MAY operate as a CoAP Proxy as defined in [RFC 7252] section 5.7. However, this function is typically not used and any CoAP client accessing a FA-Device by default should assume proxy operation is not supported. To test whether a server supports proxying, a CoAP client can include the Proxy-Uri Option in a CoAP request and check whether that request is accepted or returns either one of 4.02 Bad

Option or 5.05 Proxying Not Supported. In the error case the target server does not support Proxying.

HTTP-CoAP proxying as defined in [RFC 7252] section 10.2 and in is for the current specification out of scope. It MAY however be implemented in a proxy server or even in a FA-Device, in order to allow HTTP access to (CoAP) FA-Devices. Such proxies are strongly advised to follow [RFC 8075] guidance.

5.7.3.5 Message ID (MID) Generation

Following the recommendation of [draft-ietf-lwig-coap] section 3.5, a CoAP client SHOULD generate its Message ID values (MIDs) in a sequential fashion, increasing an internal MID counter by one for each outgoing CoAP message and wrapping back to 0 after reaching 0xFFFF. This will help constrained CoAP servers to perform message deduplication more efficiently.

5.7.3.6 Token Usage

There are no specific requirements on Token generation for a CoAP client, except that the Token SHOULD be as short as possible and the security guidelines of [RFC 7252] on Tokens SHOULD be followed.

5.7.3.7 Rate-limiting

The recommended response transmission rate limiting on CoAP servers in the final note in [RFC 7252] section 4.7. MAY be implemented on a FA-Device. An ecosystem or device vendor is free to choose its own mechanisms for this. One typical good implementation of rate limiting is to not process a new request from any CoAP client until a previous ongoing request has been fully processed and responded to.

5.8 Response Codes

Below table lists the CoAP response codes that the CoAP server in a FA-Device MUST (‘M’) or MAY (‘O’, for optional) support. The ‘X’ means SHOULD NOT use this response because there is a better alternative or no reason to use it in Fairhair context. The usage rules for the code are given by [RFC 7252] or the RFC listed in the Notes column; plus in addition the Requirements for Use stated in this column, if any. The required support for a CoAP client is not specified in detail; however a client MUST at least be able to distinguish the basic CoAP response code classes 2.xx (success), 4.xx (client error) and 5.xx (server error).

Table 3 CoAP Response Codes supported by an FA-Device.

Resp. Code	Description	Support Level	Notes and Requirements for Use
2.01	Created	**M**	[RFC 7252]
2.02	Deleted	**M**	[RFC 7252]
2.03	Valid	O	Only used when ETag Option was in request
2.04	Changed	**M**	[RFC 7252]
2.05	Content	**M**	[RFC 7252]
2.31	Continue	O	[RFC 7959]
4.00	Bad Request	**M**	[RFC 7252]

4.01	Unauthorized	**M**	Used if the client could not (yet) be properly authenticated, in order to check if it is authorized to access this resource. The client should perform action(s) to authenticate itself before retrying the request.
4.02	Bad Option	**M**	Diagnostic payload MUST be included and SHOULD start with the first unrecognized Critical-option number as 4-byte UTF-8 string
4.03	Forbidden	**M**	Used if the client could be authenticated but does not have the authorization to access this resource. The client should take action(s) to improve its authorization status, before retrying this request.
4.04	Not Found	**M**	[RFC 7252]
4.05	Method Not Allowed	**M**	[RFC 7252]
4.06	Not Acceptable	**M**	[RFC7252]
4.08	Request Entity Incomplete	O	[RFC 7959]
4.09	Conflict	X	See [RFC 8132], typically not expected to be supported.
4.12	Precondition Failed	O	If 4.12 not implemented, then 4.02 MUST be returned if a conditional request [RFC 7252] is made.
4.13	Request Entity Too Large	O **R**	‘O’ as defined in [RFC 7252] for request size only. ‘R’ in addition: 4.13 SHOULD be returned if the response to a request-without-payload would become too large to be handled by the CoAP server and the server does not support Blockwise transfer.
4.15	Unsupported Content-Format	**M**	[RFC 7252]
4.22	Unprocessable Entity	X	See [RFC 8132], typically not expected to be supported.
5.00	Internal Server Error	**M**	[RFC 7252]
5.01	Not Implemented	X	[RFC 7252] requires use of 4.05 instead for all cases.
5.02	Bad Gateway	O	Only relevant for CoAP Proxy; [RFC 7252]
5.03	Service Unavailable	O	[RFC 7252]
5.04	Gateway Timeout	O	Only relevant for CoAP Proxy; [RFC 7252]

5.05	Proxying Not Supported	O	If proxying is not supported for any resource, a CoAP request with Proxy-Uri Option SHOULD trigger 4.02 response instead of 5.05.

5.9 CoAP Options

Below table lists for a CoAP server in a FA-Device the MANDATORY ('M'), RECOMMENDED ('R') or OPTIONAL ('O') support for all CoAP Options. Hyphen ('-') means not applicable. The 'X' means the Option can be used in CoAP but SHOULD NOT be used in the indicated context. All the Options are specified in [RFC7252] except where noted otherwise in the "Notes" column.

Any CoAP Option in a Confirmable request that is not recognized by a CoAP server where the option number has the "Critical" flag [RFC 7252] set MUST lead to a 4.02 Bad Option response. These are the uneven Option numbers.

Option support for a CoAP client is not specified in detail. However, it MUST support at least the Options marked mandatory ('M') in the CoAP Server "In Response" column below.

Table 4 CoAP options supported by an FA-Device.

Opt Nr	Option Name	CoAP Server support for the Option		Notes
		In Request	In Response	
1	If-Match	O	-	
3	Uri-Host	O	-	Normally not included in a CoAP request; a Server MUST respond 4.02 if it does not support multiple virtual servers per Section 5.10.1 of [RFC 7252].
4	ETag	O	O	
5	If-None-Match	O	-	
6	Observe	O	-	[RFC 8323]
7	Uri-Port	O	-	Normally not included in a CoAP request; a Server MUST respond 4.02 if it does not support multiple virtual servers per [RFC 7252] section 5.10.1.
8	Location-Path	-	**M**	Typically used when resources are created using POST.
11	Uri-Path	**M**	-	
12	Content-Format	-	**M**	
14	Max-Age	-	O	By default, excluded in response.
15	Uri-Query	**M**	**-**	
17	Accept	**M**	-	
20	Location-Query	-	O	

23	Block2	O	O	[RFC 7959]
27	Block1	O	O	[RFC 7959]
28	Size2	O	O	[RFC 7959]
35	Proxy-Uri	O	-	
39	Proxy-Scheme	O	-	
60	Size1	O	**R** O	'R' for Section 5.9.2.9 of [RFC 7252] defined Size1 semantics for 4.13 response; 'O' for [RFC 7959] Size1 semantics.
258	No-Response	**R**	-	[RFC 7967]

5.10 Transmission Parameters

The default transmission parameters from [RFC 7252] section 4.8 are used by a FA-Device.

Note: with the purpose to optimize an FA-System MAY define specific leisure values per resource that differ from the default leisure.

5.11 CoAP Content-Format

The media type i.e. CoAP Content-Format [RFC 7252] used to encode resources into a payload when accessed via a RESTful method SHALL use CBOR [RFC 7049]. The examples given in this document are shown in CBOR diagnostic notation for human-readability ([RFC 7049] section 6).

Annex A Fairhair Application sublayer (informative)

The Fairhair Application sublayer is the basis for representing the state and functionality of FA-Devices. The Fairhair Application sublayer includes both Fairhair-defined resources that provide access to common Fairhair Application Services like Device and Resource Discovery as well as ecosystem-defined resources which expose the FA-Device state and functionality in the semantics which are specific for that ecosystem.

Data is accessible by RESTful methods for creating, reading, setting, and deleting data.

Depending on the ecosystem, metadata (i.e., any information used to annotate data) may also be accessible by RESTful methods, may be accessible through other means (e.g., via a reference to a definition), or may not be accessible via an interface at all (e.g., because it is defined in an ecosystem specification which is not available in machine-readable format and can only be used offline). In this document the term metadata is used to refer to metadata which is accessible via RESTful interfaces. It is acknowledged that there might exist data that annotates another resource but that is not used or exposed in other ways. The latter are out of scope of this specification.

Furthermore, it is important to highlight that metadata is optional, i.e. an ecosystem may decide not to use or expose any data that would conceptually qualify as ‘metadata’. However, any ecosystem that does expose metadata through a RESTful interface can use metadata in a Fairhair-defined way as described in section A.1.2.1 .

An FA-Device is associated with a resource tree, being the root of the tree. Within the -resource tree there are typically multiple CoAP resource trees. Subordinate resources are generally addressed by concatenating the resource names and using the "/" name separator.

A resource C that is directly subordinate to a resource P is called the child of P or the sub-resource of P, and P is called the parent of C. The term “below” is used in this specification to indicate a sub-resource, e.g. for C and P we say that C is below P or that C is a sub-resource of P.

Some of the resources are defined by Fairhair and provide access to common Fairhair Application Services. Resources may also be defined by the ecosystems and represent elements of the respective ecosystem modelling of the FA-Device state and functionality. An ecosystem-specific resource model instance is identified by a specific resource referred to as ‘ecosystem entry point’. A FA-Device can support resource model instances of multiple ecosystems simultaneously, where two generic ecosystem entry points (ecosystem A and ecosystem B) are represented. Ecosystem entry points and FA resources can be embedded within (sub)resources as long as they are discoverable by means of one of the standard discovery mechanisms provided in the Discovery section 4.

In addition, arbitrary arrangements of data may be present, where the structure and naming is generally a local matter.

All resources are of one of the defined Base Data Types, described in detail in section A.1.2.4. Base Data Types can be Primitive Base Data Types or Complex Base Data types.

Within the ecosystem subtree, aggregation of resources may be used by ecosystems to represent complex structures for the ecosystem (e.g. Objects, Clusters, Devices).

A resource may be described by a collection of zero or more metadata, each being of one of the defined Base Data Types. In addition, metadata may themselves be described by metadata.

Fairhair defines a number of common Fairhair Application Services, such as discovery, that are available to all ecosystems.
Each Fairhair Application Service is defined in an ecosystem-independent manner and is made accessible via one or more resources referred to as Fairhair Resources.
Resources and metadata are accessed in a RESTful manner using the respective methods of CoAP.

A.1 Resource Trees and ecosystem entry points

Ecosystem entry points and the entry point to Fairhair-defined common resources can be discoverable by means of rt.
Note that some Fairhair defined common services extend mechanisms which are defined in other specifications and as a result use different resource paths. An example is the discovery service which uses the path /.well-known/core.

A.1.1 URIs

Resources and metadata are addressed by URIs. The definition of the structure of URIs that address ecosystem defined resources or arbitrary resources defined by a vendor or configured for an installation for logical representations is out of scope of the Fairhair specification. It is the responsibility of the ecosystems to define the structure of URIs that address ecosystem defined resources.

A.1.2 Metadata

Metadata is used to annotate and describe resources and metadata.
Metadata provides additional information about a resource, for example, physical location, data type or permitted value range. Metadata can in general be present at multiple levels within a resource tree.
Although metadata is conceptually different from (regular) properties, metadata is often accessed in a similar way.

Every resource may have a set of metadata. All metadata is of one of the Base Data Types defined in section A.1.2.4.

Metadata is accessible as a referenced resource below the resource or metadata. An example path to a metadata, providing additional information on the resource "temp":

coap://<IP_address_node>/bldg-5/floor-2/room-5/temp/$metadata_example

An URI path example to a metadata of metadata:
coap://<IP_address_node>/bldg-5/floor-2/room-5/temp/$metadata_example1/$metadata_example2

A.1.2.1 Common Metadata

With the goal to enhance interoperability, metadata is introduced that can be applied to any resource, irrespective of whether the resource is an FA resource or has been defined by an ecosystem that describes the Data type of the Resource.

Table 5 Common metadata definitions.

Metadata name	Description	Data type
$base	The mnemonic of the data type as in section “Primitive	String

	Data Types" and "Complex Data Type"	

Examples

The following examples show how access to metadata could be performed in Zigbee dotdot

CoAP Request:

CON
GET /zcl/e/1/s3/$base

CoAP Response:

2.05 Content
"list"   // common metadata

where zcl is the Zigee specific entrypoint, "e" identifies a Zigbee endpoint followed by the Zigbee specific endpoint number "1". The previous request accesses the metadata $base of the resource "s3", server side of the Zigbee Cluster "Identify".

A.1.2.2 Non-Fairhair metadata

Any organization using the Fairhair resource model may specify and use additional metadata.

A.1.2.3 Metadata query

As described in section A.1.1 metadata of resources and metadata of metadata are accessible directly specifying the metadata name in the URI.

Example: coap://<IP_address_node>/bldg-5/floor-2/room-5/temp/$base

Considering that metadata are optional and that a client device may not have prior knowledge of the metadata related to a resource, Fairhair defines a mechanism for requesting all metadata associated to a specific resource by means of the query "?meta", as shown in the following example:

coap://<IP_address_node>/bldg-5/floor-2/room-5/temp/**?meta=***

Another example shows the same query applied to request all the metadata associated to the Zigbee attribute 1, within a specific cluster (s3)

CoAP Request:

CON GET /zcl/e/01/s3/a/1?meta=*

In some ecosystems, resources might be complex structures including multiple sub-resources. In these cases, it is beneficial to be able to retrieve a specific metadata for the selected resource and its first level children, with the syntax shown in the following example:

coap://<IP_address_node>/{ecosystem-prefix}/bldg-5/floor-2/room-5/container/**?meta=$base**

This mechanism could be used for example in Zigbee to retrieve the $base metadata of all the attributes of a specific cluster as shown below:

CoAP Request:

CON GET zcl/e/01/s3/a/?meta=$base

The Fairhair standard targets resource constrained devices. It is therefore recommended to avoid using complex query filters to retrieve multiple metadata via a single command. Wherever possible, metadata should instead be accessed individually.

A.1.2.4 Base Data Types

The choice of data types is ecosystem specific. However, a guideline is presented in this chapter to achieve the best mapping to common media types like CBOR [RFC 7049], and enable a limited interoperation on primitive base data types.

These lists of types are not exhaustive and might be extended by the ecosystems to represent data types which do not find a direct map to one of the data type here proposed.
The data type of a CoAP resource can be one of the following:

- Primitive Base Data Type:
  Set of base data types that can't be derived from other data types;
- Complex Base Data Type:
  Set of base data types composed of Primitive Data Types.

1.1.1 Primitive Base Data Types

Table 6 Primitive Base Data Types.

Fairhair Data Type	**Description**	**Mnemonics** ($base value)	**CBOR Representation**
Null	No value. The resource has no value set.	null	CBOR null (major type 7, additional information 22)
Boolean	Boolean value (true/false)	bool	CBOR true (major type 7, additional information 21) CBOR false (major type 7, additional information 20)
Unsigned Integer	Represents unsigned integer values of different sizes.	uint	CBOR unsigned integer (major type 0)
Signed Integer	Represents signed integer values of different sizes.	int	CBOR unsigned integer (major type 0) or CBOR signed integer (major type 1), depending on the actual value
Half Float	Half precision floating point	float16	major type 7, additional information 25
Float	Single precision floating point	float32	major type 7, additional information 26
Double	Double precision floating point	float64	major type 7, additional information 27
String	Unicode character string	string	CBOR text string (major type 3)
Enumeration	Value from a set of assigned names	enum	CBOR unsigned integer (major type 0)

Bits	Set of flags identified by position	bits	CBOR byte string (major type 2)
Binary	Binary data, i.e. a sequence of octets	binary	CBOR byte string (major type 2)

A.1.2.5 Complex Base Data Types

Complex Base Data Types are composed of one or multiple elements. Each element is of type Primitive or Complex Data Type.

Table 7 Complex Base Data Types.

Fairhair Data Type	**Description**	**Mnemonics** ($base value)	**CBOR Representation**
Collection	A collection of Primitive and Complex Base Data Types, where the elements are named.	coll	CBOR map (major type 5)
List	An unordered array of elements.	list	CBOR array (major type 4)
Array	An ordered array of elements.	array	CBOR array (major type 4)

Definition of ordered vs. unordered

A resource of type Complex Base Data Type is ordered, if the child elements are always sorted in the same way.

This means that an ordered Complex Base Data Type always results in the same serialization (e.g. CBOR or JSON) as long as the values don’t change. An unordered Complex Base Type may result in different serializations, even if the values didn’t change but the order of values.

Considerations regarding naming / identifying of child nodes:

Child elements of a Complex Base Data Type may be named or unnamed.

Annex B Application security (informative)

B.1 Device Access Control

Complementing the core principles of the defense-in-depth protection strategy and the overarching principles of information security, access management itself has a series of core guiding principles, as follows:

- Categorization and classification: Clearly categorize and value all data.
- Least privilege: Provide the least amount of access necessary for a given entity to complete their system role.
- Need to know: Provide access to systems and information only where there is a need for the recipient of such access to have it.
- Controlled access: Define procedures to monitor, enable and disable access methods, and enforce security policy at all access points.

Effectively applying these principles to a FA-System throughout processes, networks and users in the FA-System will ensure that access related risks are appropriately controlled, allowing authorized access when required and unauthorized access to be prevented (IEC [62443-3-3] – SR 5.4: SL1).

B.1.1 Trust List

A trust list is a list of device certificates or CA certificates which are trusted for authentication. The FA-Device MUST reject connections from peers whose device certificate or CA certificate is not in the trust list or if the certificate is expired. Entries in the trust list will be added and removed as an explicit administrative action reflecting changes in trust relationships in conjunction with the access control list.

Certificates are stored in the trust list as binary DER encoded [X.509] format. The [PKCS#7] defines a container format that includes just the public certificate, or may include an entire certificate chain or a signature of the encapsulated enveloped data structure. The [PKCS#7] shall be used to add new trust relations to an FA-Device. An FA-Device shall support following URI operation paths:

To read/write configured CA certificates from a FA-Device, the client would use the following request:

Request: POST ldevid/crts
{ PKCS#7 according to [RFC 7030] section 4.1.3 }

Request: GET ldevid/crts
Response:

```
{       "crts": [
                { "kid": "00:a5:43..:55", "type": "ca-builtin" },//indirect trust: default ca certificates
                { "kid": "00:a5:43..:66",  "type": "ca" },        //indirect trust: configured ca cert.
                { "kid": "00:a5:43..:77", "type": "dev-cert" },  //direct trust: configured device cert.
                { "kid": "00:a5:43..:88", "type": "rpk-cert" }   //direct trust: config. raw pub. key cert.
        ]
}
```

B.1.2 FA-Device Access Control List

The device access control list is a mandatory functionality in order to achieve the access management core principles and helps the commissioning engineer with key management and permission rules. The list contains permissions, which controls access to FA-Device resources. Each permission in the device access control list identifies an authorized entity and specifies the access rights for that entity (e.g. allowed or denied). It is up to the commissioning engineer in collaboration with the network administrator to define permission rules for an application or scope.

As a precondition for any access control or authorization decisions, the FA-Device must authenticate the peer by verifying its [X.509] certificate. Verification includes that

- The operational device certificate is correctly signed by a trusted CA; and
- The peer's identity in the [X.509] operational device certificate (e.g. Subject and/or SubjectAltName name) matches an entity that is authorized to access the required resources.

If a security zone has not its own exclusive CA certificate (e.g. intermediate CA) then the FA-Device access control list MUST be added with settings equivalent to name constraints defined in RFC 5280 section 4.2.1.10 [X.509]. The following is an example of possible access control list items:

Following request is used for writing access permissions for security zone issued by a trusted CA (indirect trust). All devices that belong to the security zone have same minimum access rights:

Request: POST szone/crts/acl

```
{       "kid": "00:a5:43..:66",
        "sub": [
                { "dnsName": ".building-1.campus.local"},
                { "rfc822Name": ".sec-zone-56789@example.com" }
        ],
        "scope": "szone",
        "perm": "<permission settings>"
}
```

Following request is used for writing access permissions for device certificate (direct trust):

Request: POST szone/crts/acl

```
{       "kid": "00:a5:43..:77",
        "scope": "conduit-fire-protection",
        "perm": "<permission settings>"
}
```

B.1.3 Revocation List

A revocation list is a blacklist of digital certificates that have been revoked by a trustworthy entity before their scheduled expiration date. Digital certificates in the revocation list are no longer trusted and a request from suspended certificates must be refused. Digital certificates are revoked for many reasons. For example, if a certificate is discovered to be counterfeit, the trust anchor will revoke it and add it to the revocation list of each FA-Device in the FA-System. A common reason for revocation occurs when the owner of the Domain CA no longer owns FA-Devices, or the original certificate being replaced with a different certificate from a new trust anchor (e.g. after commissioning).

The problem with revocation lists, as with all blacklists, is that they are difficult to maintain and are an inefficient and unreliable method of distributing in real time. Furthermore, the size of the revocation list is limited especially on constrained FA-Devices and, as a consequence, the list should be updated only under exceptional circumstances. The process of updating revocation lists is vendor specific and not part of this document. An option might be that the registrar supports such a functionality.

B.2 Role-Based Access Control (RBAC)

Role Based Access Control (RBAC) is an access control model that grants access to a resource based on the role a user or an FA-Device holds in an FA-System. The RBAC model is the most widely used and actually dominant model today. Most access control security products available in the market are based on this model because its objectives are architectural. The model is based on the concept of "separation of duties". Privileges are assigned to particular roles that have been globally specified by administrators (e.g. commissioning engineer). Users or FA-Devices are mapped to one or several roles and their privileges are combined of all roles assigned to them. The following figure shows the mappings used in the RBAC model. Users or FA-Devices are associated with one or more roles and roles are associated with access rights for particular resources.

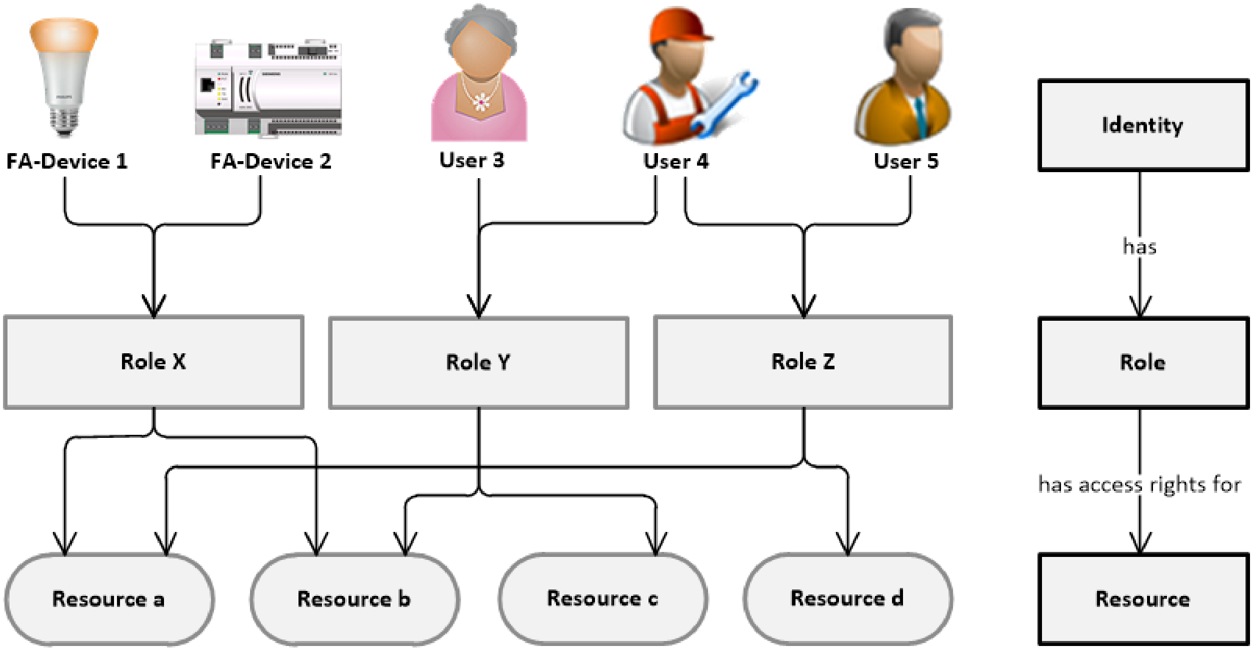


Figure 13 Role Based Access Control Model.

B.2.1 Authorization Model

The RBAC model (figure) depicts the relationships between an identity, its access rights and the resource server. The identity is characterized by its credential that is intended for authentication of the client at the authorization server. User and FA-Device profiles are stored on the authorization server and provide different kinds of information relevant for authorization of resource requests. A user profile, for example, contains references to one or several roles which may be adopted by a client. Each client or rather FA-Application has either an associated user or FA-Device. An FA-Device client in a machine-to-machine communication use case uses typically its operational device certificate (credential) as a means for authentication. On the other hand, user centric BA-Applications may use other credentials such as user certificates or passwords.

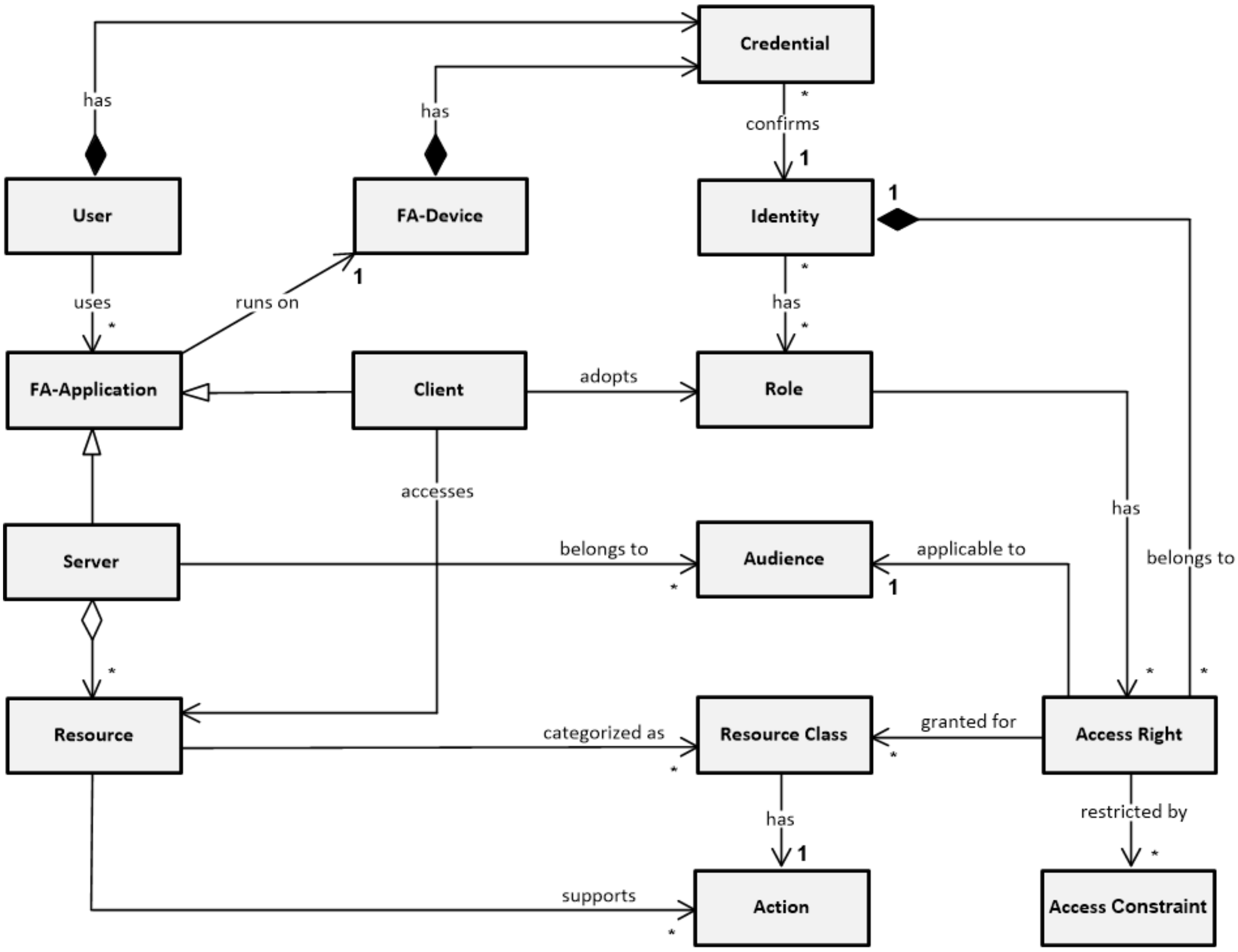


Figure 14 Role Based Access Control Model

B.2.2 Role

A role is the owner of an arbitrary number of access rights. In an FA-System with constraint FA-Devices a client should assume only one role at time and explicitly switch to another role if other privileges are required.

Roles may be organized in a hierarchical manner where roles higher in the hierarchy inherit access rights from roles connected to them that are lower in the hierarchy. The authorization model shall only support a limited hierarchy of roles. For instance, the commissioning engineer higher in the hierarchy combines access rights (inherits) from the service technician and the facility manager. However, multiple inheritances should be explicitly excluded in a reasonable authorization model (less error prone).

B.2.3 Access Right

Access rights specify under which conditions an access to a certain class of resources is granted. It contains the following elements: One single audience, a set of resource classes with an assigned action and a set of access constraints.

B.2.4 Resource Class

A resource class is used to characterize the meaning/purpose/type of a resource. Resource classes are used in access right specifications and as annotations to resources in resource servers. A resource class classifies a resource according to different categories (e.g. automation discipline, functional aspects, safety and security aspects or organizational aspects etc.). If several resource classes are associated with an access right, the authorization is valid for all resources that have a subset of these resource classes assigned. A resource class that is higher in the hierarchy (super-class) obtains access rights from all resource classes that are lower in the hierarchy (sub-classes).

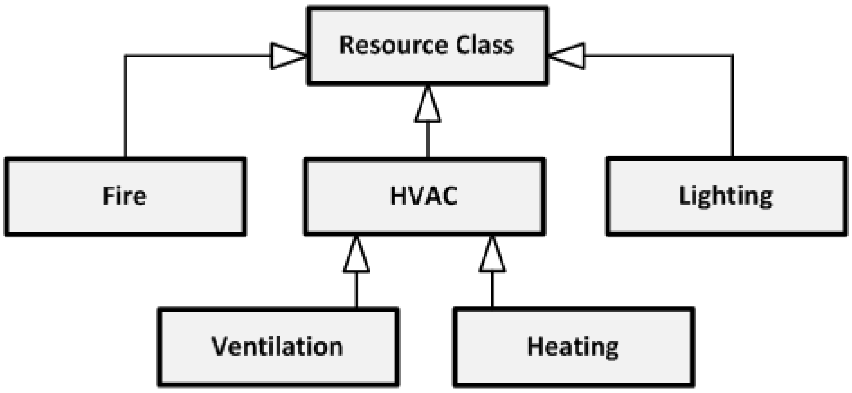


Figure 15 Resource Classes.

Using a hierarchy of resource classes to specify access right implies that a resource server must know the structure of the hierarchy in order to be able to decide if a particular scope is actually matching the list of resource classes associated with the resource.

B.2.5 Action

An action identifier is used to specify the kind of operation that the access right provides authorization for. In agreement with the CRUD model proposed by the REST paradigm, one of four different kinds of actions can be chosen: Read, Read&Write, Create, and Delete.

B.2.6 Audience

An audience specifies either a single resource server (individual audience) or groups of resource servers (group audience) for which the access right applies. An audience contains either a device ID (e.g. serial number or FQDN) or an application specific string (e.g. semantic tags).

B.2.7 Access Constraint

An access constraint restricts the access right containing the constraint to certain conditions. Following types of access constraints are possible:

- Temporal Constraint: specifies the temporal validity of the access right, i.e. is restricting the time frame when the client is allowed to access the resource (e.g. expiration time); and
- Client Constraint: specifies a restriction regarding the client that accesses the resource (subject).

The subject of an access token should be bound to an identity which can be validated by a resource server without having direct access to an authorization server. The subject in the access token is used for proving possession of a credential (private key) corresponding to a client identity. One such identity binding would be to configure for the access token subject a value that the operational device certificate must contain in the [X.509] subject field or the subject alternative name. For peer-to-peer (D)TLS communication, the subject of an access token shall contain either the FA-Device serial number (e.g. PXC3.E75-100A SN:123456) or the FQDN if present.

The lifetime in an access token is optional if the FA-Device has permanent access rights and the token is used in a (D)TLS session. In this case, the lifetime of the access token is the same as the lifetime of the operational device certificate.

B.3 Application Access Control

To mitigate the risk that, at some point in time, a device participating in the security zone can be compromised, as well as to lower the incentive to attack and compromise devices, application-level authorization is used to limit the scope of what a device belonging to a security zone can do. This is done by means of authorization tokens that are linked to operational identities. These define what type of request can be issued by client devices towards resource-server devices.

In this way, the impact of the attack can be limited to only a subset of devices. Any attempt by the compromised device to communicate with a device outside the scope defined by authorization, issued by the administrator of the system, can be detected, logged and reported.

It is up to the enterprise administering the security zone to define what resources should be protected by authorization, and to select what devices should be granted with access rights.

The authorization and access control use the OAuth ACE framework [OAuth].

B.3.1 Authorization Bootstrapping

The authorization bootstrapping is a process in which the devices belonging to a single Security Zone get authorization to register, remove, lookup and access resources from the Authorization Server, see the following Figure.

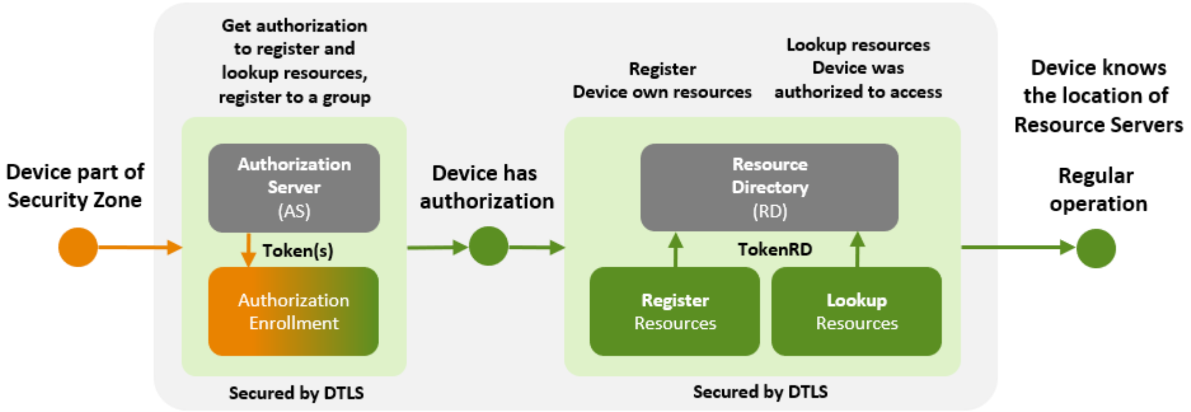

Get authorization to register and lookup resources, register to a group
Register Device own resources
Lookup resources Device was authorized to access
Device knows the location of Resource Servers
Device part of Security Zone
Authorization Server (AS)
Token(s)
Authorization Enrollment
Device has authorization
Resource Directory (RD)
TokenRD
Register Resources
Lookup Resources
Regular operation
Secured by DTLS
Secured by DTLS

Figure 16 Application Commissioning Flow.

The authorization bootstrapping process is initiated during the device enrollment. Every new device enrolled into the Security Zone gets the locations and operational identities of the Authorization Server (AS) and Resource Directory (RD) from the Registrar.

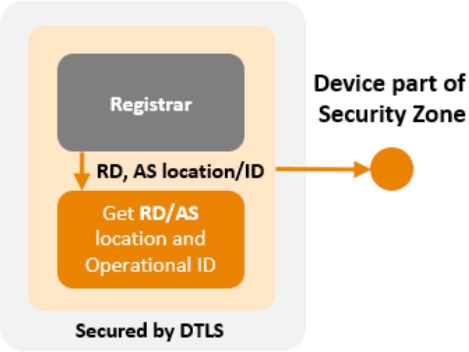

Registrar
RD, AS location/ID
Get RD/AS location and Operational ID
Secured by DTLS
Device part of Security Zone

Figure 17 Getting Resource Directory and Authorization Server locations and IDs.

Therefore, every Security Zone member device can locate RD and AS and verify their operational identities. A device may validate the subject name of a given authorization server by retrieving an extension to EST (GET .../aslist) that contains a list of valid authorization servers for a given security zone.

The device operational identity used to establish DTLS channel with the authorization server is used by the server to determine what authorization the device must be granted with. At the same time operational identity of the authorization server is used by the device to authenticate the server and verify the server identity corresponds to the identity provided by the Registrar during the device bootstrapping process.

The authorization granted by the authorization server belonging to the enterprise controlling the security zone is represented by tokens. A token issued to a client device includes:

- Scope defining what resource can be accessed by the device;
- Reference to the operational identity of the device; and
- Signature of the security zone CA issuing the operational identities.

The authorization also includes tokens granting the device with required access to the resource directory (TokenRD) enabling the device to register its own resources and discover the location of the resources it was enabled to access.

As a result of the authorization bootstrapping process, the device:

- Has all required tokens to access resources (TokenRS); and
- Knows location of all resources it was authorized to access.

The device is therefore authorized to access resources.

B.3.2 Resource request

The resource requests are sent from a client device to a resource server device. A client device can, for instance, request a resource or send a control command to change the status of the resource. A resource server device can verify any incoming request based on the operational identity of the client device and the authorization token scope linked to the identity.

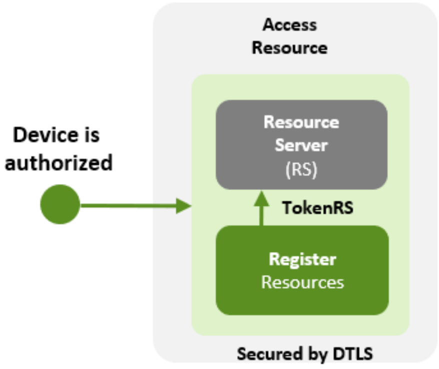


Figure 18 Device regular operation.

Optionally, an introspection request can be sent from the resource server device to the authorization server, to check the active state of the token to determine if the token is valid. In this way, any unauthorized request will be rejected, logged and reported by the resource server device.

Annex C Requirements (informative)

C.1 Introduction

Fairhair specifies six types of requirements, including General, Identity, Network Access Control, Application Layer, Software Management, and Device Hardening. Each requirement is intended either for an endpoint or for the network. Requirements are split into three components: the actual requirement itself, in many cases, a specific approach that is required to meet the requirement, and an explanation as to why the requirement is required and specified in the way it is. In some cases, the requirement will be noted as "future", indicating that although there is no current mandate, the intention of the group is to make something mandatory as and when underlying standards mature.

C.1.1 General Security Requirements

C.1.1.1 Zero Touch Deployment

[GEN1] Endpoint security features MUST provide zero touch deployment capabilities (highly desirable) or require absolute minimal configuration by the installer (the person). There MUST be zero touch for the installer (no configuration). There will be minimal touch during commissioning phase with the network and/or commissioning engineer.

As mentioned above, endpoint installers will generally not be certified to configure or manage networking systems. Required installation experience is a) mount the endpoint b) power the endpoint c) the endpoint auto discovers required services and configuration to bring itself into an operational state. Given the number of devices being attached, enterprise address plans are becoming more and more complex.

C.1.1.2 Scaling

[GEN2] The system as a whole MUST be able to scale from as few as a single endpoint to as many as O(10^6) actuators and sensors, and shall be able to handle cold start of all of those devices in two minutes. If external AAA servers are present, they MUST be nXm redundant for both scale and resiliency.

Power failures happen and the system must be able to recover in a reasonable time and in an orderly fashion.

C.1.1.3 Multiple Deployment Situations (Endpoints)

[GEN3] Endpoints MUST be able to function within deployment scenarios ranging from an unmanaged isolated cluster/LAN to a fully managed enterprise network. Put another way, these systems must function in both pizza parlors and offices.

C.1.1.4 Multiple Deployment Situations (Network)

[GEN4] The network MUST provide for basic connectivity services, including addressing, AAA, and routing in managed, connected, and disconnected / standalone environments

C.1.1.5 Seamless Transitions (Endpoints)

[GEN5] Endpoints MUST support a seamless transition from unmanaged to managed (such as after commissioning or a tenant move).

Because buildings will go through stages during their construction, it must be possible for installers to test endpoints, even if there is no Internet access. Additional configuration may occur later through whatever commissioning tools come on line, but such tools should not be expected to be running when the first endpoints are installed.

C.1.1.6 Seamless Transitions (Network)

[GEN6] The system as a whole MUST support a seamless transition between various states, from initial installation through a fully provisioned and operational set of devices, covering the entire building life cycle. The system MUST support disconnected, managed (e.g., enterprises), and unmanaged networks (e.g., pizza parlors).,

C.1.1.7 Multi-tenancy

[GEN7] Both devices and network will support multi-tenancy. This is necessary so that control of the lighting and HVAC can be separated between the facility manager and various different tenants.

C.1.1.8 Keying Material

[GEN8] It MUST be possible to establish multiple application groups within a single security group. When these groups must be separately authorized, then new security groups MUST be created. An example: a group lights in a room may be in the same security group, but may respond differently to different requests (e.g., turn off the lights near the projector).

C.1.1.9 Amount of Keying Material

[GEN9] The keying material required to secure group communication on any device MUST scale with the number of security groups (see GEN8) and NOT number of group members.

C.1.1.10 QoS

[GEN10] The network MUST support differentiated Quality of Service (assured delivery/AF) to meet the requirements of emergency and low latency endpoint messaging.

C.1.1.11 Replacement

[GEN11] Replacement of commissioned devices should require minimal manual configuration of the new device or the network management system. It should be possible to simply indicate that one device is a replacement for another.

C.1.2 Identity Requirements

C.1.2.1 Secure Immutable Identity

[ID1] Endpoints MUST implement a secure, immutable, unique ‘device’ identity.

This basic requirement provides a means for networks to understand what sort of device is connecting. Based on this information the device can be classified and its network access determined. If this information is absent, a manual configuration step must take place.

C.1.2.2 Secure Operational Identities

[ID2] Endpoints MUST implement a unique secure identity within an enterprise that is configurable by the network administrator.

Local operational identities provide a means for a local network to identify a device without any external dependencies.

C.1.2.3 Quality Entropy

[ID3] Endpoints MUST have a source of entropy from which to generate random numbers and any necessary salts, nonces, etc. NIST SP 800-90A Rev 1 specifies one means of producing reasonable entropy. There may be other industry-approved approaches. Implementers should take care not to reuse entropy that has been gathered.

Entropy is the basis for strong cryptography.

C.1.3 Network Access Control Requirements

C.1.3.1 Endpoint Authentication to the Network

[NAC1] An endpoint MUST authenticate itself to the network in which it wishes to be admitted. It must also be able to authenticate the network. This must be supported when the device does not have Internet connectivity (e.g., perhaps where cell phone (or some commissioning tool) is acting as application-level proxy).

C.1.3.2 Network authentication to the endpoint

[NAC2] The network MUST authenticate itself to an endpoint and authenticate the endpoint itself before the endpoint is admitted to the network. It MUST then properly authorize appropriate network access for the device. This function MUST be supported when the device does not have Internet access (See NAC1).

Note: we do not specify how to get an appropriate enterprise trust anchor to the device at this time. We expect that in the future this will occur through the use of ANIMA key bootstrapping, as described in draft-ietf-anima-bootstrapping-keyinfra.

C.1.3.3 Restrict Traffic from Unauthorized Endpoints

[NAC3] The network will restrict access to devices that do not meet these requirements or are otherwise unauthenticated to L2 access, and may further implement rate-limiting, limiting by types of message, or a mix of all of the above. The network will block general communication between unauthenticated and authenticated systems. The goal is to allow for bootstrapping from an unauthenticated state to an authenticated state, and to allow for basic local functions of the endpoints, without risking the rest of the infrastructure.

By doing this the network is protecting those systems that are already authenticated and provisioned.

C.1.3.4 Avoid Counterfeit Systems

[NAC4] When it can detect them, the network MUST strictly limit the ability of counterfeit endpoints to be admitted to the network. This requirement can be satisfied when the network has some path of communication to a manufacturer to determine whether the device has been registered in more than one location at one time.

Counterfeit systems include those where the keying material from a legitimate system has been stolen. If this has happened, then **any** system could present itself as a legitimate endpoint, and then attack other systems within the infrastructure.

The mechanisms to implement this requirement may differ or evolve over time.

C.1.3.5 Endpoints provide product information

[NAC5] Endpoints MUST securely provide a description of TCP/UDP ports and the direction communication and MAY also provide additional classing information at finer granularity. This information may change after commissioning, over time (although classes of devices will be maintained by enterprises).

Consistent with GEN1, the intent is that there be as little configuration as possible. Absent this information, the network may not have sufficient information to understand the device’s profile.

C.1.3.6 Profile Communication

[NAC6] The network MUST support isolation for endpoint messaging. The network will maintain profiles of the various endpoint systems in order to protect those endpoints, and to detect when they may have been compromised.

C.1.3.7 DoS and Coordinated Attack Protection

[NAC7] The network MUST mitigate denial-of-service (DoS) attacks on the access ports (both directions).   The network MUST identify and defend against coordinated attacks on the infrastructure.

These attacks may not be due to malice but could instead be due to a device malfunctioning. In either case, through the use of profiles or other means, the network should detect and limit such attacks.

C.1.3.8 Authorization by network attach point

[NAC8] The network MUST be able to authorize a device by its topological location (e.g., switch port) for purposes of accessing Building Automation Control (BAC) infrastructure. It should not be possible, for instance, for someone to insert a PC into the lighting or HVAC network to gain access to the building automation infrastructure.

C.1.3.9 Bootstrapping of Authentication Information

[NAC9] The device SHALL have a means to initialize credential information so that it will be able to authenticate to the network and to authenticate the network itself.

C.1.3.10 Traceability

[NAC10] There SHALL be a means to trace communications to their source.

C.1.4 Application Requirements

C.1.4.1 Group Based Communications

[APP1] Members of security groups MUST be individually authorized to receive group keying material. This makes it possible for only them to send and receive encrypted messages within a group.

Receivers of group messages MUST be able to verify the integrity of received messages as being generated within the group. Message communication and processing MUST happen with a low latency and in parallel manner.

C.1.4.2 Endpoints Use RBAC

[APP2] Endpoints MUST support role-based resource access controls.

As described in the introduction, different people will need different levels of access, some to simply operate the device, and others to configure it. Devices are required to provide some means to differentiate access to various resources. RBAC provides this capability.

C.1.4.3 Logging

[APP3] Applications MUST have the capability to log network or security-related information to a central server in a standard and structured way all supervisor access and anomalous behaviors.

C.1.4.4 Signatures

[APP4] Application messages MUST be integrity protected and authenticated. Sensitive application messaging SHOULD be signed. Other messages MAY be signed.

When something goes wrong there is a need to understand which device issued a command and under what authority. At the same time, signatures presents additional CPU requirements for real time processing.

C.1.4.5 Encrypt Sensitive Information

[APP5] Application messaging containing sensitive or private information MUST be encrypted.

A method MUST be provided to allow an authorized proxy to perform a forwarding task even if the message is encrypted.

Information containing passwords or other credentials should be protected from public view (for instance). It is left to endpoint manufacturers to determine what is sensitive.

C.1.5 Software Management Requirements

Many requirements stated in this section are not necessary for interoperability but are necessary for security of the device.

C.1.5.1 Secure Upgrade

[SW1] A secure mechanism MUST be provided to upgrade executable images on an endpoint. In particular, only administrators should be able to activate new binaries; and the identity of those administrators MUST somehow be securely verified. Images delivered to endpoints MUST be signed by the manufacturer and verified by the end point. Images endpoints MUST be delivered to endpoints over an encrypted channel.

This requirement is important to avoid a system being coopted by an attacker for purposes of harming the other endpoints.

C.1.5.2 Network Support

[SW2] The network SHOULD support the secure distribution, of signed firmware images. The network SHOULD support devices that are upgrading firmware images. In particular, the network will need to provide sufficient access to a management system to receive such images, and then to distribute them amongst the lighting and HVAC infrastructure.

C.1.5.3 Image Selection

[SW3] When endpoints contain multiple images, an endpoint MUST have a secure and reliable way to enumerate, select and activate different executable images

C.1.5.4 Image Growth Over Time

[SW4] The endpoint MUST be able to support the growth of its OS image for security patches over its lifetime. In particular, endpoint manufacturers SHOULD anticipate that images will grow between 150% and 250% over time. This may have ramifications on the certification process.