University of Zurich[UZH]

BACHELOR THESIS – Communication Systems Group, Prof. Dr. Burkhard Stiller

Inventorying and Secure Life-Cycles of IoT Devices

Armin Richard Veres
Zürich, Switzerland
Student ID: 20-700-118

Supervisor: Dr. Eryk Schiller
Date of Submission: December 1, 2023

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland

ifi

Bachelor Thesis
Communication Systems Group (CSG)
Department of Informatics (IFI)
University of Zurich
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland
URL: http://www.csg.uzh.ch/

Abstract

Das ist die Kurzfassung...

Acknowledgments

Optional

Contents

Chapter 1

Introduction

1.1 Motivation

There are daily more and more Internet of Things, IoT, devices connected to the internet, with the need to gather and process massive amounts of real-time information, especially with 5th Generation networking, which allows for extensive information exchange.

Enhanced connectivity and adoption of IoT trigger cyber attacks, which are increasingly sophisticated and affect considerable amount of IoT-related infrastructures, raising security concerns with consumers, as well as businesses. This stresses the importance of appropriate IoT security management and enhancement of IoT life-cycle management. Considering the heterogeneity of IoT devices, the dynamism of the security landscape and number of IoT stakeholders make these tasks quite challenging, especially considering that a single insecure update can put a whole IoT system at risk.

As emphasized by the European Union Agency for Cybersecurity, ENISA, Cyber Security Act, CSA, the management of IoT infrastructures encompassing the entire life-cycle of products, as well as the continuous certification, are fundamental tools to guarantee a high level of security. [1] Also as pointed out by the Network and Information Security, NIS2, directive a pragmatic security framework must stimulate active collaboration between the IoT Stakeholders. [2]

Providing access to cybersecurity information is central for realization of a homogeneous perspective on cybersecurity. CSA and NIS promote strategic cooperation among stakeholders to support and facilitate information sharing, leading to an approach that helps respond to large-scale incident by creating more effective synergies against cybersecurity vulnerabilities.

1.2 Description of Work

This thesis will develop a service to support security information sharing between IoT stakeholders to support continuous security assessment throughout the IoT device life-

cycle. We will consider the use of Distributed Ledger Technology, DLT, as a possible approach to facilitate a trustworthy and transparent platform for sharing cybersecurity information without a trusted third-party. [3] It is important to integrate the presence of several entities with different responsibilities and roles in sharing cybersecurity knowledge. So while performing security monitoring activities, the user, i.e., a device, may detect vulnerabilities that will be shared with the manufacturer for further investigation, prompting for mitigation and or resolutions. This requires the device to be re-configurable throughout its life-cycle according to the changing threat landscape and to the device manufacturers publishing of secure updates, i.e., patches, and device profiles.

Established approaches of secure IoT deployment and bootstrapping have significant challenges. Using pre-shared credentials for every device is the simplest approach, but it prevents the identification of specific devices and the verification, whether the device is corrected to the correct network. [4] This thesis will develop a bootstrapping service to provide a lightweight bootstrapping protocol, supporting different authentication methods, depending on the characteristics of the device and providing key management.

The infrastructure of the developed bootstrapping mechanism will enable the inventorying of IoT devices. Said infrastructure will keep track of all embedded IoT devices and their respective security levels. To ensure security throughout the life-cycle of a device, an update/patching mechanism will be developed, where manufacturers and software providers will provide fixes to a security issue after an attack or vulnerability detection.

As most updating proposals are based on centralized models using e.g., client-server architectures, this thesis will design a scalable and secure approach for disseminating software updates in scenarios with selected IoT devices. The design shall entail decentralization, robustness and efficiency, bringing the upgrading functionality closer to the end devices. Blockchain based technologies will be leveraged by providing a transparent ledger to manage different versions of software elements composing and IoT device or system and share relevant security aspects, such as vulnerabilities or device information. As interoperability is crucial, this thesis will analyze the use of Bifröst [5] and Interledger [6] approaches to interconnect different blockchain implementations.

Finally this thesis will consider mitigation for IoT devices using Threat Manufacturer Usage Description, MUD, proposed by NIST [7], which provides a flexible and dynamic way to alert about new threats and mitigation to apply before and update of patch is released. Threat MUD is intended as a complement to MUD file, dynamically reconfiguring a device in case of detection of a vulnerability.

1.3 Thesis Outline

Chapter 2

Background

2.1 Distributed Ledger Technology

2.2 Manufacture Usage Description

MUD has been developed by the International Engineering Task Force, IETF, with following goals and intents in mind: [8]

- Substantially reduce the threat surface on a device to those communications intended by the manufacturer.
- Provide a means to scale network policies to the ever-increasing number of types of devices in the network.
- Provide a means to address at least some vulnerabilities in a way that is faster than the time it might take to update systems. This will be particularly true for systems that are no longer supported.
- Keep the cost of implementation of such a system to the bare minimum.
- Provide a means of extensibility for manufacturers to express other device capabilities or requirements.

MUD does not entail address network authorization of general purpose computers, it simply creates a suggestion than can be followed. The architecture of Devices using MUD can be seen in Figure 2.1, which is the reference architecture by NIST [7] but it can be found in similar fashion inside the RFC specification.

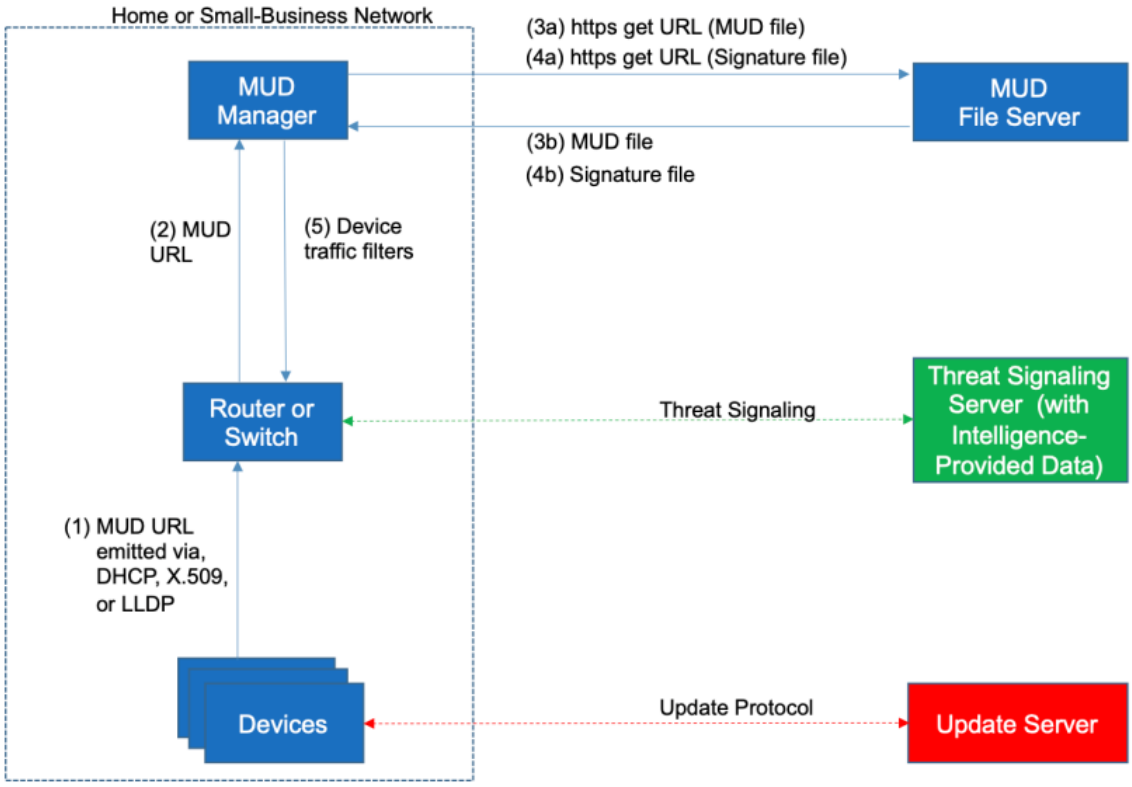

Figure 2.1: NIST MUD Reference Architecture

Chapter 3

Related Work

3.1 CERTIFY

This thesis is carried out in conjunction with the CERTIFY project.

The National Institute for Standard and Technology has a few ongoing projects and white papers on security related mitigation methods for IoT devices.

3.2 DLT-based Asset-Tracking

Neisse et al. (2017) analyzed how blockchain-based approaches might be used for data accountability and provenance tracking under the then recently released GDPR legislation, highlighting challenges of scalability and considering sharding as a method to address it. [3] Further they also mentioned issues of clonability of the tracked assets, which we can also correlate to the physical assets that are tracked inside blockchain.

3.3 Device fingerprinting

For classification of device capabilities NIST has been considering the usage of MUDs, so that devices do not step out the bounds of their official and appointed capabilities. [7]

In order to be able to track IoT nodes in a blockchain, they need to be uniquely identifiable, in our case even in a distributed manner, using Distributed Identifiers, DIDs. Methods of creating identifiers that are unique to devices exist, such as SRAM-Based PUF Readouts [9].

3.4 Cybersecurity of IoT Devices

In order to maintain participation rights for only valid users/clients, Manufacturer Usage Descriptions, MUDs, are getting more and more relevant, as also the National Institute for Standards and Technology, NIST, have been considering their use cases. [7]

Chapter 4

Use Case Definition - Connected Cabin System

4.1 Background

Our use case will take the CCS scenario from Figure 4.1 into consideration and build up on their use cases.

Nowadays more and more IoT devices are being deployed to aircraft cabins to improve passenger experience and airline operations. Benefits span from remote PHM to reduced maintenance time, while also supporting a continuous (re)certification process.

4.2 Actors

We will consider following actors for our use case.

- Airline: Owns the aircraft and oversees interactions and system operations.
- Airplane maintainer: They could be e.g., Airplane manufacturer. Oversees maintenance of the aircraft, including the integration of systems designed by different manufacturers and their configuration.
- Product Owner: Oversees design and maintenance of systems deployed in the aircraft on assignment of the airplane maintainer.
- Maintenance operator: They work for the airplane maintainer. Their responsibilities include e.g., the replacement of devices or on-site software upgrades of e.g., portable data loaders.
- Passenger, Attendant, Pilot: They interact with the aircraft through sensors, actuators or HMI.

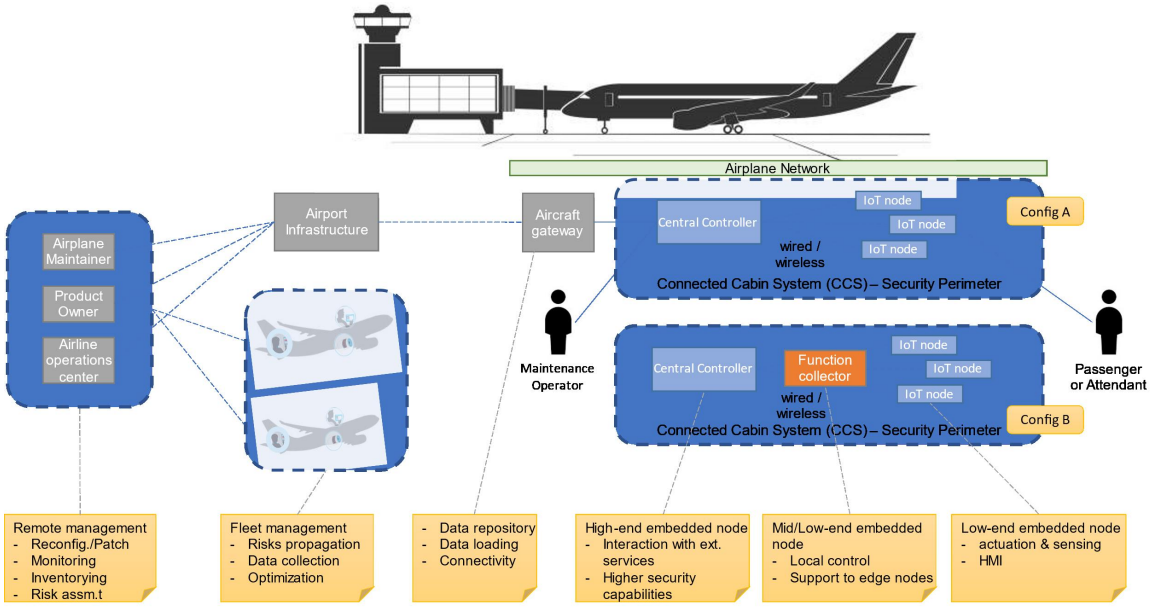

Figure 4.1: Collins CCS

4.3 System Components

We will consider an aircraft to have multiple networks, covering various aspects.

- In-flight entertainment system
- Aircraft System
- Flight Maintenance

For our use case we will assume config 'A' as the main configuration of the networks, where edge nodes are connected to a central controller that manages the edge nodes as a subnet.

- IoT / Edge Nodes: low-end devices, including actuation, sensing or HMI capabilities, with limited room for hardware and software based cybersecurity, that requires offloading to a more capable instance.
- Central Controller: High-end devices with ability to host full-fledged security functionalities.

External communication will take place through aircraft gateway offering services for data repository, data loading and connectivity with external environment. The airline operations center, product owner and airplane maintainer can interact through the airport infrastructure. A technician may directly access the aircraft if necessary.

Table 4.1: Actors involved

Airline	Airplane Maintainer	Product Owner	Maintenance Operator	Passenger, Attendant, Pilot
X	-	X	-	X

Table 4.2: Lifecycle stages involved

Bootstrapping	Operation	Update	Repurposing	Decommissioning
X	-	X	-	X

4.4 Scenarios

4.4.1 Installation of Connected Cabin Systems

4.4.1.1 Goals

The goals of this scenario include bootstrapping and customization of devices for specific deployment, updating and decommissioning of previous systems, guaranteeing a reset to a known and fresh, wiped data, state. Table 4.1 highlights the involved actors and Table 4.2 shows the stages involved in this scenario.

4.4.1.2 Pre-condition

In order for this scenario to be valid, following pre-conditions need to be met:

- Actors involved can establish a secure connection with the aircraft, wireless or wired, through airport infrastructure
- Airport and Aircraft network infrastructure can receive authorization requests for needed connections from the external environment.
- The Maintenance Operator is provided access to the airplane and to maintenance ports of the target CCS.

4.4.1.3 Sub-Scenario 1: Component Installation

4.4.1.3.1 Flow of Events The flow of events can be tracked in Figure 4.2 and is verbalized as follows:

- Airline requests installation of new component to the Airplane Maintainer also issuing an authorization request to Airport and Airplane gateways

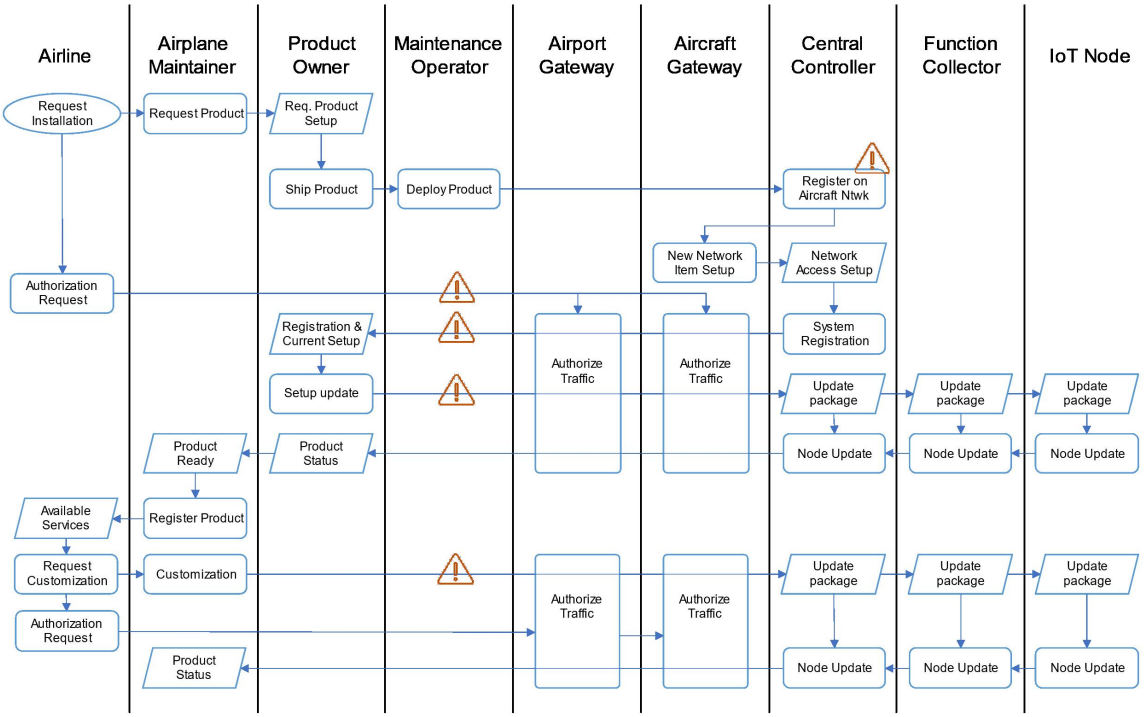

Figure 4.2: Collins Scenario 1 Installation

- The request is forwarded to the Product Owner and then to the Maintenance Operator, who oversees the physical deployment of the product.
- Once connected, Central Controller registers on the Aircraft Network and receives required setup to complete network access and system registration.
- Product Owner is now able to reach the CCS, push configuration and security updates to the Central Controller, as well as the Function Collector and IoT nodes.
- After the update, the product is registered and the Airplane Maintainer can offer remote services to the Airline
- The Airline requests a customization of the CCS. It is performed by the Airplane Maintainer by pushing an update package and/or modifying specific configurations as allowed by the Product Owner API for Maintenance.
- The new product status is confirmed with a feedback message.

4.4.1.4 Sub-Scenario 2: Component Replacement

4.4.1.4.1 Flow of Events

- Airline requests replacement of a component to the Airplane maintainer, also issuing an authorization request to the Airport and Airplane Gateways

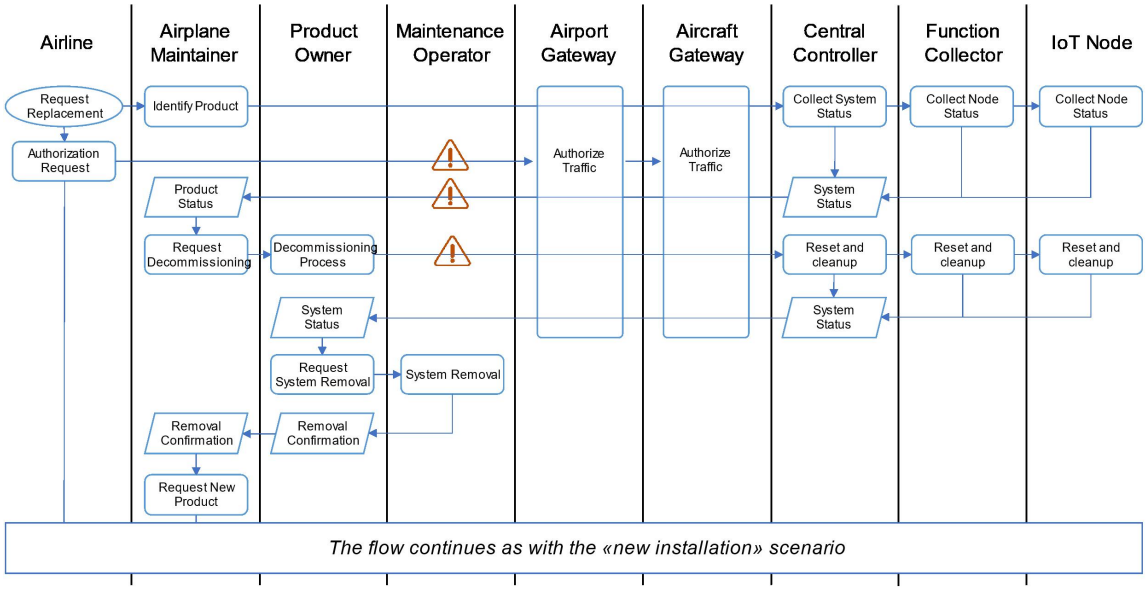

Figure 4.3: Collins Scenario 1 Replacement

- The Airplane Maintainer identifies the target product (location) and collects latest system status.
- The Airplane Maintainer issues a decommissioning request to the Product Owner and starts the decommissioning process, which causes a reset and cleanup of all the nodes that will be replaced
- After remote reset and clean-up, product owner requests the Maintenance Operator to physically remove the system from the cabin.
- The product is then unregistered and can be dismissed.
- The remaining part continues with the 'New installation process flow'

4.4.1.5 Post-Condition

After completion of the Installation Scenario, following post-conditions need to be met:

- New component is deployed in the CCS, integrated into the network, updated with latest security patches and configured by the Airline for their specific needs.
- Component is securely onboarded in the CCS, unique identity and certificates are dispatched for authentication.

4.4.1.6 Attack Scenario

As an alternative flow of events, i.e., in an attack scenario, highlighted by the yellow triangles in Figure 4.2 and Figure 4.3 following points were identified:

- Attacker can inject malicious payloads in place of the intended one, (confidential) credentials provided to the Central Controller for network access and authentication ca be stolen.
- IP sensitive data can be lead by the Airplane Maintainer when retrieving system status
- maintenance/reset/cleanup procedures integrity can be compromised

4.4.2 System Operation and Monitoring

Chapter 5

Device Life-Cycle

Chapter 6

Evaluation

Chapter 7

Summary and Conclusions

Bibliography

[1] "The eu cybersecurity act." [Online]. Available: https://digital-strategy.ec.europa.eu/en/policies/cybersecurity-act

[2] "Directive on measures for a high common level of cybersecurity across the union (nis2 directive)." [Online]. Available: https://digital-strategy.ec.europa.eu/en/policies/nis2-directive

[3] R. Neisse, G. Steri, and I. Nai-Fovino, "A blockchain-based approach for data accountability and provenance tracking," in *Proceedings of the 12th international conference on availability, reliability and security*, 2017, pp. 1–10.

[4] (2023) Trusted iot device network-layer onboarding and lifecycle management. [Online]. Available: https://www.nccoe.nist.gov/projects/trusted-iot-device-network-layer-onboarding-and-lifecycle-management

[5] E. J. Scheid, T. Hegnauer, B. Rodrigues, and B. Stiller, "Bifröst: a modular blockchain interoperability api," in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*. IEEE, 2019, pp. 332–339.

[6] V. A. Siris, P. Nikander, S. Voulgaris, N. Fotiou, D. Lagutin, and G. C. Polyzos, "Interledger approaches," *Ieee Access*, vol. 7, pp. 89 948–89 966, 2019.

[7] D. Dodson, D. Montgomery, W. Polk, M. Ranganathan, M. Souppaya, S. Johnson, A. Kadam, C. Pratt, D. Thakore, M. Walker *et al.*, "Securing small-business and home internet of things (iot) devices: Mitigating network-based attacks using manufacturer usage description (mud)," National Institute of Standards and Technology, Tech. Rep., 2021.

[8] E. Lear, R. Droms, and D. Romascanu, "Manufacturer Usage Description Specification," RFC 8520, Mar. 2019. [Online]. Available: https://www.rfc-editor.org/info/rfc8520

[9] S. Vinagrero, H. Martin, A. de Bignicourt, E.-I. Vatajelu, and G. Di Natale, "Sram-based puf readouts," *Scientific Data*, vol. 10, no. 1, p. 333, 2023.

Abbreviations

AAA	Authentication, Authorization, and Accounting
ACL	Access Control List
CCS	Connected Cabin System
CTIS	Cyber Threat Information Sharing
EVM	Ethereum Virtual Machine
GDPR	General Data Protection Regulation
HMI	Human Machine Interface
IDS	Intrusion Detection System
IFE	In-flight Entertainment System
IPS	Intrusion Prevention System
IoT	Internet of Things
LRU	Line Replacable Unit
MUD	Manufacturer Usage Description
NIST	National Institute of Standards and Technology
PHM	Prognostics and Health Management
PUF	Physically Unclonable Function
SCADA	Supervisory control and data acquisition
SRAM	Static Random-Access Memory
TEE	Trusted Execution Environment
TOE	Target of Evaluation
VC	Verifiable Credential
IETF	International Engineering Task Force

Glossary

Trust Model In the trust model the issuer issues credential to a holder while the holder can prove identity by showing the credential to a verifier.

Manufacturer Usage Description A component-based architecture specified in Request for Comments (RFC) 8520 that is designed to provide a means for end devices to signal to the network what sort of access and network functionality they require to properly function.

Cloud Computing Cloud computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user.

Fog Computing As an extension of Cloud computing, Fog Computing brings the computation closer to IoT Edge devices.

Edge Computing Edge computing is the placement of storage and computing resources closer to source, where the data is generated.

Trusted Execution Zone

Line-Replaceable Unit modular component of airplane, designed to be replaced quickly

List of Figures

List of Tables

List of Code Snippets

Appendix A

Installation Guidelines

Appendix B

Code Snippets

```
{
  "ietf-mud:mud": {
    "mud-version": 1,
    "mud-url": "https://iot-device.example.com/dnsname",
    "last-update": "2019-01-15T10:22:47+00:00",
    "cache-validity": 48,
    "is-supported": true,
    "systeminfo": "This is an example of a device that just wants to talk
                   to its cloud service",
    "mfg-name": "Example, Inc.",
    "documentation": "https://iot-device.example.com/doc/dnsname",
    "model-name": "dnsname",
    "from-device-policy": {
      "access-lists": {
        "access-list": [
          {
            "name": "mud-96898-v4fr"
          },
          {
            "name": "mud-96898-v6fr"
          }
        ]
      }
    },
    "to-device-policy": {
      "access-lists": {
        "access-list": [
          {
            "name": "mud-96898-v4to"
          },
          {
            "name": "mud-96898-v6to"
```

```
          }
        ]
      }
    }
  },
  "ietf-access-control-list:acls": {
    "acl": [
      {
        "name": "mud-96898-v4to",
        "type": "ipv4-acl-type",
        "aces": {
          "ace": [
            {
              "name": "cl0-todev",
              "matches": {
                "ipv4": {
                  "ietf-acldns:src-dnsname": "cloud-service.example.com"
                }
              },
              "actions": {
                "forwarding": "accept"
              }
            }
          ]
        }
      },
      {
        "name": "mud-96898-v4fr",
        "type": "ipv4-acl-type",
        "aces": {
          "ace": [
            {
              "name": "cl0-frdev",
              "matches": {
                "ipv4": {
                  "ietf-acldns:dst-dnsname": "cloud-service.example.com"
                }
              },
              "actions": {
                "forwarding": "accept"
              }
            }
          ]
        }
      },
      {
        "name": "mud-96898-v6to",
```

```
      "type": "ipv6-acl-type",
      "aces": {
        "ace": [
          {
            "name": "cl0-todev",
            "matches": {
              "ipv6": {
                "ietf-acldns:src-dnsname": "cloud-service.example.com"
              }
            },
            "actions": {
              "forwarding": "accept"
            }
          }
        ]
      }
    },
    {
      "name": "mud-96898-v6fr",
      "type": "ipv6-acl-type",
      "aces": {
        "ace": [
          {
            "name": "cl0-frdev",
            "matches": {
              "ipv6": {
                "ietf-acldns:dst-dnsname": "cloud-service.example.com"
              }
            },
            "actions": {
              "forwarding": "accept"
            }
          }
        ]
      }
    }
  ]
 }
}
```

Code 1: Example MUD file