



Funded by
the European Union

Horizon Europe

EUROPEAN COMMISSION

European Climate, Infrastructure and Environment Executive Agency (CINEA)

Grant agreement no. 101136131



SHIFT to Direct Current

Deliverable D 1.5

IT requirements, information exchange and cybersecurity

Document Details

Due date	31-10-2024
Actual delivery date	31-10-2024
Lead Contractor	CIRCE
Version	1.0
Prepared by	Francisco José Arroyo Valle (CIRCE), Alberto Mur Rodrigo (CIRCE), Jonatan Peris Rivas (CIRCE), Andreas Muñoz Zuara (CIRCE)
Reviewed by	Hugo Morais (INESC ID) and Samy El Kamch (Watt&Well)
Dissemination Level	Public

Project Contractual Details

Project Title	Shift to Direct Current
Project Acronym	SHIFT2DC
Grant Agreement No.	101136131
Project Start Date	01-12-2023
Project End Date	31-05-2027
Duration	42 months

Document History

Version	Date	Contributor(s)	Description
0.1	25-09-2024	CIRCE	Initial version

0.2	22-10-2024	CIRCE	Draft of the document chapters
0.3	30-10-2024	CIRCE & Revisors	Revision of the document
1.0	31/10/2024	CIRCE & Revisors	Submitted version

Disclaimer

This document has been produced in the context of the SHIFT2DC project. Views and opinions expressed in this document are however those of the authors only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.

Acknowledgment

This document is a deliverable of SHIFT2DC project. SHIFT2DC has received funding from the European Union's Horizon Europe programme under grant agreement no. 101136131.



**Funded by
the European Union**

Executive Summary

This document represents the work that has been carried out in order to design the communications architecture that will be implemented in a later task (namely task 2.5).

The design of the architecture has taken into account several factors of different nature, such as the type of devices that are (or will be) installed in each demo site, the kind of information that they will provide and the type of connectivity that they allow and the information type.

The topology of the network has also played an important role in the definition of the architecture, mainly because not all the facilities are the same, nor their operation conditions.

The two previous paragraphs lead to another topic that has definitely shaped how the information in the overall system is exchanged among components, which has its own analysis in the corresponding part of this document.

Table of Contents

Executive Summary	4
Table of Contents	5
List of Figures.....	6
List of Tables.....	7
Keywords, Acronym.....	8
1 Introduction.....	9
1.1 Scope and Objectives	9
1.2 Structure.....	9
1.3 Relationship with other deliverables	9
2 Demonstrators	11
2.1 Port demonstrator.....	11
2.1.1 Information Exchange via Communication Protocols	12
2.1.2 IT requirements for Accessing Device Information	13
2.2 DC Data Centres demonstrator	14
2.2.1 Information Exchange via Communication Protocols	14
2.2.2 IT Requirements for Accessing Device Information	15
2.3 Commercial and residential building demonstrator	17
2.3.1 Information Exchange via Communication Protocols	17
2.3.2 IT Requirements for Accessing Device Information	18
2.4 DC Industrial demonstrator.....	19
2.4.1 Information Exchange via Communication Protocols	19
2.4.2 IT Requirements for Accessing Device Information	19
3 Architecture.....	21
3.1 Introduction.....	21
3.2 Design proposal.....	21
3.3 Cybersecurity.....	22
3.3.1 Summary.....	22
3.3.2 Using a VPN as main layer of cybersecurity	24
3.3.3 Using Blockchain as alternative/complement to VPN for cybersecurity	26
3.3.4 Conclusion using VPN as main layer of cybersecurity	27
3.4 Deployment requirements	29
4 Conclusions.....	30
4.1 Summary	30
4.2 Progress	30
4.3 Main Challenges	30
4.4 Next deliverables.....	31
5 References.....	32
ANNEX I	33



List of Figures

Figure 1- Global architecture 21
Figure 2 - Global architecture details..... 22

List of Tables

Table 1: Devices, protocols and information exchanged within Port demonstrator	12
Table 2: Devices, protocols and information exchanged within DC data centres demonstrator	15
Table 3: Devices, protocols and information exchanged within Building demonstrator	17
Table 4: Devices, protocols and information exchanged within Industrial demonstrator	19

Keywords, Acronym

API	Application Program Interface
ARM	Advanced RISC Machines
BESS	Battery Energy Storage System
CAN	Controller Area Network
DDoS	Distributed Denial of Service
DDR	Double Data Rate
DNS	Domain Name System
E2E	End-to-end
EMS	Energy Management System
ESS	Energy Storage System
EU	European Union
EV	Electric Vehicle
EVSE	Electric Vehicle Supply Equipment
GDPR	General Data Protection Regulation
HTTPS	Hyper Text Transfer Protocol Secure
IDS/IPS	Intrusion Detection/Prevention Systems
IP	Internet Protocol
IT	Information Technology
JSON	JavaScript Object Notation
MFA	Multi-Factor Authentication
MITM	Man-In-The-Middle
NVMe	Non-Volatile Memory Express
OS	Operating System
PDU	Power Distribution Unit
PLC	Programmable Logic Controller
PV	Photovoltaic
RBAC	Role-Based Access Control
RISC	Reduced Instruction Set Computer
RT	Real-Time
RTU	Remote Terminal Unit
SATA	Serial AT Attachment
TCP	Transmission Control Protocol
TLS/SSL	Transport Layer Security/Secure Sockets Layer
TPM	Trusted Platform Module
UPS	Uninterruptable Power Supply
VPN	Virtual Private Network
WP	Work Package

1 Introduction

1.1 Scope and Objectives

This document will serve as a guide to implement a communications architecture on a future stage of Shift2DC project.

There are several aspects that will be analysed and considered for the design of the communication network. So, all the devices of each demo site can connect and exchange information according to the use cases that have been defined in Task 1.3.

It is important to notice that, at this point of time, there are many parts of the system that can be described or designed without any specification of the implementation.

While there is a necessity of (for example) listing the type of devices that each demo site will have, there is a high chance that the architecture proposed in this document changes with respect to what ends up being implemented in Task 2.5.

To conclude, the main objectives of this deliverable are:

- Establish the **IT requirements** necessary for the network design.
- Define the communications **architecture** leveraging those IT resources.
- Set **cybersecurity** criteria to ensure that the monitored data from the various demonstrators is transmitted securely to the corresponding server.

These objectives are essential to guarantee the integrity, confidentiality, and availability of the data throughout the system's lifecycle, ensuring the successful deployment and operation of the communications infrastructure.

1.2 Structure

The present document will divide its content into chapters that will talk about the defined use cases, a description of the demo sites and the devices of each demo site (details about IT requirements and information exchange will be discussed here).

The result will be the proposal of one architecture that best fits the needs of the project according to the definition of the use cases.

Such proposal will divide itself into an introductory description, the proposed design itself, several sections about cybersecurity concerns and deployment requirements that will have to be taken into account in order for a successful startup process of the SHIFT2DC IT Platform (bear in mind that, while the list of requirements might seem considerable, there are aspects with a certain level of flexibility with the purpose of adapting the resulting platform to the communication with the devices of each demo site).

1.3 Relationship with other deliverables

According to what has been described in the executive summary and the introduction itself, this document will be constrained by one deliverable mostly: D1.3 which contains the definition of the different use cases.

D2.6 is very likely to receive influence from D1.5, because the architecture designed for Task 1.5 will have to interact with the Energy Management System (EMS) used in some of the demo sites (considerations will have to be taken into account to ensure maximum interoperability between the

devices of the demonstration sites, the architecture itself, the monitoring platform of the Task 2.5 and the EMS developed in Task 2.3c).

It will also shape what will be done in future Task 2.5 of WP2, so it will influence what is described in Deliverable 2.8.

2 Demonstrators

In this chapter, the different demo sites will be enumerated and described. For each demonstrator, there will be indications about the use cases that are related to the given demo site.

Regarding the technical part of each demonstrator, there will be an analysis of the devices (or at least the type of devices) that will compose the functionality of the demonstrator itself, because such details will help shape the requirements of the architecture that is being designed here and that will be implemented in a later stage.

Have in mind that giving a description of the use cases is out of the scope of this document. For such purpose, it is recommended to have a look at Deliverable 1.3 (on the other hand, Deliverable 1.1 could also be of help if general information the context of DC applications is required).

When having a look at each demonstrator description, a list of devices (or type of devices in case there are multiple ones of the same type) will be presented. It is important to notice that some of them might not be still ready, but that depends on each demonstrator and the current conditions of the demonstrator itself.

Although most of the purpose of devices in each demonstrator is clear from the beginning of the design, there is still some margin for change in relation to the communication protocol/s that will be used to communicate the bare-metal hardware of the demonstrator and the platforms with which it will need to communicate.

In the following four sections, the information gathered from the different demonstrators is presented in relation to the following aspects:

- Specific **devices** from which information is obtained.
- **Communication protocol** available for data extraction.
- Type of **information exchange** for monitoring purposes or control in the different demonstrators.

Subsequently, the necessary IT requirements to access that information for each demonstrator are detailed, considering the information gathered.

2.1 Port demonstrator

The port demonstrator (located in Funchal, Madeira Island – Portugal) will be the scenario of the following use cases:

- **UC1:** DC Port Energy Storage System (ESS) Management (usage of Battery Energy Storage System - BESS to promote battery-buffered charging strategies to mitigate the impact of connecting new ships to the Port)
- **UC2:** DC Port | Grid Coordination (contributing to the grid stabilization)
- **UC3:** DC Port as Microgrid (capable to operate in islanded mode)
- **UC4:** DC Port as an Energy Hub (H₂) (considering possibility to implement hydrogen systems to generate alternative electricity generation respects PV and ESS)

2.1.1 Information Exchange via Communication Protocols

Thanks to the coexistence of two highly developed solutions in this demonstrator, such as:

- The energy monitoring system (primarily composed of the eGauge measurement system and a gateway) (TRL 8/9)
- Cloud-based Data Storage for data collection in the cloud (TRL 7/8)

It is possible to monitor information from the various devices directly and in a standardized manner, which simplifies the IT requirements for this demonstrator. These solutions provide detailed information on the different devices (via HTTPS and encapsulated in JSON) as shown in the following Table 1.

Also, in this table are described the devices and the information available for each device in the demonstrator.

Table 1: Devices, protocols and information exchanged within Port demonstrator

Device	Communication protocol	Information exchanged
eGauge Meter	HTTPS (JSON) Note: JSON data is available in server from Raspberry Pi Gateway	Current in different phases, active power, reactive power, energy consumption over the time, voltage in each phase and power factor
BESS (Battery Energy Storage System)	HTTPS (JSON)	Real- time energy generation, input voltage, output voltage, input current, output current, power delivered, converter efficiency, device temperature, metric power setpoint (battery), power PV -> MPPT (Maximum Power Point Tracking)
		From BUCs (Bidirectional Unit Converters), it will available: connection type, frequency, voltage, harmonic distortion, total shore-power, peak consumption, BESS capacity, BESS power charge rate, BESS power discharge rate, active energy consumption and reactive energy consumption
SUC1-AssetControl	HTTPS (JSON)	Power scheduling, state of charge (Soc), active power in alternating current, reactive power in alternating current, voltage in alternating current, current in alternating current, frequency (Hz), active power in

		direct current, voltage in direct current, current in direct current and energy (Wh)
SUC2-Ahead scheduling	HTTPS (JSON)	Ship ID, latitude, longitude, arrival time, departure time, power, voltage, microgrid start period (service contracting), microgrid period duration (service contracting), ship power demand, port power demand, ship power flexibility estimation and ship schedule flexibility estimation
EVSE (Electric Vehicle Chargers)	HTTPS (JSON)	Charging status, output current (charging), output voltage (charging), estimated charging time remaining, total energy delivered, total time in charging, final price of the charging and charger temperature
Panel solar	HTTPS (JSON)	Panel temperature, voltage from each module of the panel, current from each module of the panel and solar irradiation

2.1.2 IT requirements for Accessing Device Information

The demonstrator's devices all utilize **HTTPS communication** and **JSON format** to transmit data, making the IT requirements consistent across the system. A server capable of handling HTTPS requests and processing JSON-formatted data is essential for accessing the information from each device. The specific requirements for each device category are outlined below:

2.1.2.1 eGauge Meter

- **Data available:** Current, active/reactive power, voltage, energy consumption, and power factor across different phases.
- **Requirements:** A server capable of handling HTTPS requests and processing JSON-formatted data.

2.1.2.2 Battery Energy Storage System (BESS)

- **Data available:** Real-time energy generation, voltages, currents, power, efficiency, and BESS capacity. Additionally, connection type, frequency, harmonic distortion, and power metrics from Business Use Cases (BUCs).
- **Requirements:** Server capable of real-time monitoring of energy storage and converter data.

2.1.2.3 SUC1 - AssetControl

- **Data available:** Power scheduling, state of charge (SoC), voltage, current, frequency, and energy in both AC and DC systems.
- **Requirements:** Infrastructure to monitor power flow and asset control in real time.

2.1.2.4 SUC2 - Ahead Scheduling

- **Data available:** Ship ID, location (latitude, longitude), arrival/departure times, power demand, and flexibility estimations for ships and ports.
- **Requirements:** Server infrastructure to handle scheduling data and coordinate microgrid service contracting.

2.1.2.5 Electric Vehicle Chargers (EVSE)

- **Data available:** Charging status, output current/voltage, remaining charging time, total energy delivered, charging duration, and pricing.
- **Requirements:** Systems to monitor EV charger performance and manage pricing data.

2.1.2.6 Solar Panel

- **Data available:** Panel temperature, module voltages/currents, and solar irradiation levels.
- **Requirements:** Real-time access to monitor panel performance and environmental conditions.

In summary, the main **IT requirement of Port demonstrator** is a server infrastructure capable of **handling HTTPS communications and processing JSON data** to access real-time monitoring information across all devices. This ensures a streamlined and standardized approach to gathering data from the system.

2.2 DC Data Centres demonstrator

2.2.1 Information Exchange via Communication Protocols

The data centres demonstrator will be deployed at the premises of Bachmann in Stuttgart, Germany, and will be the scenario of the following use cases:

- **UC1:** DC grid resilience (more complex load management should be integrated to improve the DC grid resilience)
- **UC2:** Data centre scalability (the power architecture will be evaluated to be built up for the full power)
- **UC3:** Sector coupling (How to manage the excess heat of the servers is important respects climate change even is important trying to reuse it in the future)

Next Table 2 outlines the devices, communication protocols, and data exchanged for a data centre demonstrator.

The listed devices, such as the EMS, controllable server, ESS, and others, communicate using protocols like Modbus TCP, CAN, and Ethernet to monitor and control energy-related parameters.

While some specific topics are yet to be confirmed, the table provides an overview of the current setup, facilitating effective energy management and system control within the data centre.

Table 2: Devices, protocols and information exchanged within DC data centres demonstrator

Device	Communication protocol	Information exchanged
EMS (Energy Management System)	Modbus TCP/ MQTT (Pending to confirm)	Pending to confirm the specifics topics available
Controllable server	Via EMS	Power consumption and other operating parameters. Will be controlled by the EMS
ESS	Modbus/CAN	Voltage, Current, SoC
Fan speed controller	Modbus/CAN	Speed setpoints, on/off status
Smart DC PDU	Web GUI Transport protocol: TCP/IP Application protocol: to be defined	Load, voltage, power, energy
V2X DC station	CAN (CANopen)	Charging session states request/feedback, max voltage/current/power, RT output current/voltage/power, delivered energy, temperature.
AIC	Modbus/CAN	Current, power and direction of power flow

2.2.2 IT Requirements for Accessing Device Information

The devices in this data centre demonstrator use a variety of communication protocols such as Modbus TCP, CAN, and Ethernet (TCP/IP), with some devices also utilizing web interfaces for data exchange.

To access the information from these devices, a server infrastructure must be capable of handling these protocols and formats.

Below are the specific requirements for each device category:

2.2.2.1 EMS (Energy Management System)

- **Data available:** Pending confirmation of specific topics.
- **Requirements:** A server supporting Modbus TCP/MQTT to retrieve power and energy data, once specifics are confirmed.

2.2.2.2 Controllable Server

- **Data available:** Power consumption and other operating parameters. Controlled via the EMS.
- **Requirements:** A system integrated with the EMS to manage and control power-related data for the server.

2.2.2.3 ESS (Energy Storage System)

- **Data available:** Voltage, Current, SoC.
- **Requirements:** A server supporting Modbus/CAN for accessing energy storage data.

2.2.2.4 Fan Speed Controller

- **Data available:** Speed setpoints, on/off status.
- **Requirements:** A server supporting either Modbus or CAN protocols to monitor and control fan speed parameters.

2.2.2.5 Smart DC PDU (Power Distribution Unit)

- **Data available:** Load, voltage, power, energy.
- **Requirements:** A server infrastructure capable of TCP/IP communications and a web interface (Web GUI) for real-time data monitoring. The application protocol is yet to be defined.

2.2.2.6 V2X DC Station

- **Data available:** Charging session states request/feedback, max voltage/current/power, RT output current/voltage/power, delivered energy, temperature.
- **Requirements:** A system supporting CANopen communication to monitor and control the power distribution unit's performance.

2.2.2.7 AIC (Active Inverter Controller)

- **Data available:** Current, power, and direction of power flow.
- **Requirements:** A server supporting either Modbus or CAN protocols to manage and monitor inverter performance and power flow direction.

To support real-time monitoring and control of all devices, the main IT requirement for this data centre demonstrator is a server infrastructure capable of handling Modbus TCP, CAN, and TCP/IP communications.

This will ensure smooth and efficient access to the various operational data from each device.

Some devices have pending specifications, which will need further clarification to fully integrate into the system.

2.3 Commercial and residential building demonstrator

2.3.1 Information Exchange via Communication Protocols

The commercial and residential building demonstrator will be located in France, and will be the scenario of the following use cases:

- **UC1:** DC grid resilience (Capable to work apart from upstream AC supply grid and also scenarios with connect/disconnect events)
- **UC2:** DC grid energy management (optimize consumption and prioritize DC consumption regard AC consumption)
- **UC3:** DC services to the main grid
- **UC4:** DC grid self-healing capacity (improving continuously operation)

The Table 3 below details the communication methods and data topics available for energy management, conversion, and distribution within the building's demonstrator.

Table 3: Devices, protocols and information exchanged within Building demonstrator

Device	Communication protocol	Information exchanged
"White" EMS tool for AC/DC Hybrid Systems	Modbus TCP/ MQTT (pending to confirm)	Inputs: <ul style="list-style-type: none"> - Devices parameters: limits, boundaries, capacities. - Device state parameters: state of switch, state of charge, warning/faults/flags. - Voltage, current, and power measurements. - User inputs. Outputs: <ul style="list-style-type: none"> - Setpoints for devices (reference voltage and droop coefficients), state of switches – on/off or threshold voltage level.
LVAC-LVDC Interlink Converter	Modbus/CAN	Pending to confirm the specifics topics available
V2X DC station	CAN (CANopen)	Charging session states request/feedback, max voltage/current/power, RT output current/voltage/power, delivered energy, temperature.
DC/DC Converter - Smart Power Distribution Unit	CAN (CANopen)	Voltage, current, power, system state request/feedback and temperature.
FlexiVerter (Micro solar DC systems)	Wifi MQTT (ESP module)	Current, voltage and efficiency (MPPT)

2.3.2 IT Requirements for Accessing Device Information

The building demonstrator relies on multiple communication protocols for exchanging data across various energy management and power distribution systems.

To access data from these devices, the IT infrastructure must support these protocols and handle the relevant data formats.

Below is a detailed breakdown of the devices and their specific requirements:

2.3.2.1 “White” EMS tool for AC/DC Hybrid Systems

- **Data available:**
 - **Inputs:**
 - Device parameters: limits, capacities.
 - Device state: switches, state of charge, warnings/faults.
 - Voltage, current, and power measurements.
 - User inputs.
 - **Outputs:**
 - Setpoints: reference voltage, droop coefficients.
 - Switch state: on/off or threshold voltage levels.
- **Requirements:** A system capable of handling Modbus TCP / MQTT communications (pending confirmation) and integrated with measurement and control systems.

2.3.2.2 LVAC-LVDC Interlink Converter

- **Data available:** Pending confirmation of specific topics.
- **Requirements:** A system capable of handling Modbus or CAN communications to manage data exchange once available.

2.3.2.3 V2X DC Station

- **Data available:** Charging session states request/feedback, max voltage/current/power, RT output current/voltage/power, delivered energy, temperature.
- **Requirements:** A system supporting CANopen communication to monitor and control the power distribution unit's performance.

2.3.2.4 DC/DC Converter (Smart Power Distribution Unit)

- **Data available:** Voltage, current, power, system state request/feedback and temperature.
- **Requirements:** A system supporting CANopen communication to monitor and control the power distribution unit's performance.

2.3.2.5 FlexiVerter (Micro Solar DC Systems)

- **Data available:** Current, voltage, and efficiency (MPPT).
- **Requirements:** A server capable of handling Wi-Fi MQTT communications (via ESP module) to monitor solar system performance in real-time.

The IT infrastructure must support multiple communication protocols including Modbus TCP, CAN, MQTT, and Ethernet (TCP/IP).

This will enable real-time monitoring and control of energy flows, while some devices await further clarification of specific data topics for full integration.

2.4 DC Industrial demonstrator

2.4.1 Information Exchange via Communication Protocols

The industrial demonstrator will be in two different locations in Germany (one being an existing facility and the second one being a new facility built from scratch), and will be the scenario of the following use cases:

- **UC1:** Dynamic DC grid stability behaviour in switching on and operating industrial applications (dynamic power peaks)
- **UC2:** Proving the Handling and Reliability of DC Field Device Connectors on Individual Devices
- **UC3:** Adapting BESS Capacity and Power Rating to Meet Industrial DC Grid Demand Changes
- **UC4:** Validating the Installation and Retrofitting of DC Measurement Devices for Grid Monitoring & Control

The Table 4 below details the communication methods and data topics available for energy management, conversion, and distribution within the industrial's demonstrator.

Table 4: Devices, protocols and information exchanged within Industrial demonstrator

Device	Communication protocol	Information exchanged
EMS (Energy Management System)	Modbus TCP/ MQTT (Pending to confirm)	Pending to confirm the specifics topics available
Scalable and modular BESS (Battery Energy Storage System)	Modbus TCP	Pending to confirm the specifics topics available
Retrofit DC (T3.4e measurement device)	Modbus RTU (future also Modbus TCP)	DC current measurement
V2X DC station	CAN (CANopen)	Charging session states request/feedback, max voltage/current/power, RT output current/voltage/power, delivered energy, temperature.

2.4.2 IT Requirements for Accessing Device Information

The industrial demonstrator incorporates multiple communication protocols to enable data exchange across energy management systems and power distribution devices.

To access data from these systems, the IT infrastructure must support various communication protocols and handle specific data formats.

Below is a breakdown of the devices and their specific requirements:

2.4.2.1 EMS (Energy Management System)

- **Data available:** Pending confirmation of specific topics.
- **Requirements:** A system capable of handling Modbus TCP / MQTT communications (pending confirmation), to allow for monitoring and control of energy flows and system performance.

2.4.2.2 Scalable and Modular BESS (Battery Energy Storage System)

- **Data available:** Pending confirmation of specific topics.
- **Requirements:** A system capable of handling Modbus TCP communication, enabling monitoring and control of the battery energy storage system once data topics are confirmed.

2.4.2.3 Retrofit DC (T3.4e Measurement Device)

- **Data available:** DC current measurement.
- **Requirements:** A system supporting Modbus RTU (with future support for Modbus TCP) for integrating DC current data from the retrofit measurement device into the overall energy management system.

2.4.2.4 V2X DC Station

- **Data available:** Charging session states request/feedback, max voltage/current/power, RT output current/voltage/power, delivered energy, temperature.
- **Requirements:** A system supporting CANopen communication to monitor and control the power distribution unit's performance.

3 Architecture

3.1 Introduction

Once the demonstrators and their hardware have been presented, architecture analysis will take place.

The design of a communications architecture obeys to these principles:

- The type of devices connected to the network (being network any necessary medium in order to send or receive information to/from the devices)
- The information that will be exchanged
- Cybersecurity

Depending on the approach taken to come up with an architectural design that covers all the requirements and functionality specified in the different use cases, other aspects different from these ones could arise, but in this case these ones will be the ones that will drive how the overall is operated.

Along this architecture part of the document, several designs will be presented.

Such designs will be of these types:

- Designs of different granularity: where applicable, simpler or more complex designs will be presented depending on how much context or how much precision wants to be given. Diagrams of this kind will try to abstract from the possible implementations that could be performed at a later stage
- Designs of different alternatives: for any given level of granularity (that is to say, two or more diagrams that represent elements with the same precision), multiple designs can be proposed to represent that some part of the architecture can be implemented in multiple ways

Each diagram will be accompanied by the corresponding description so any choice (whether it is related to the type of devices, to the way the information is exchanged or to any security aspect) is made according to the most reasonable criteria available for that part of the architecture.

3.2 Design proposal

The next diagram will give a brief overview of what will connect as a whole in the resulting SHIFT2DC IT Platform.

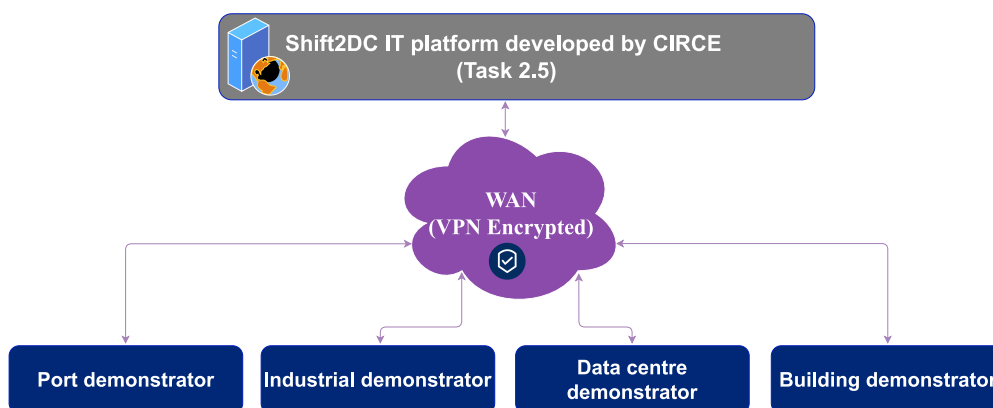


Figure 1- Global architecture

This schema does not point its focus in the implementation details of the entire network, nor it does center its attention to the details of a not yet implemented tool as CIRCE’s monitoring platform from Task 2.5.

The goal is to show how the information flow will work once the final system is implemented in Task 2.5 (which doesn’t only include the monitoring tool, but also the implementation of the communications network itself).

Next in the design roadmap, a prototype of a demonstrator is presented to give an idea of how the different technological challenges could be tackled.

ARCHITECTURE (Draft)

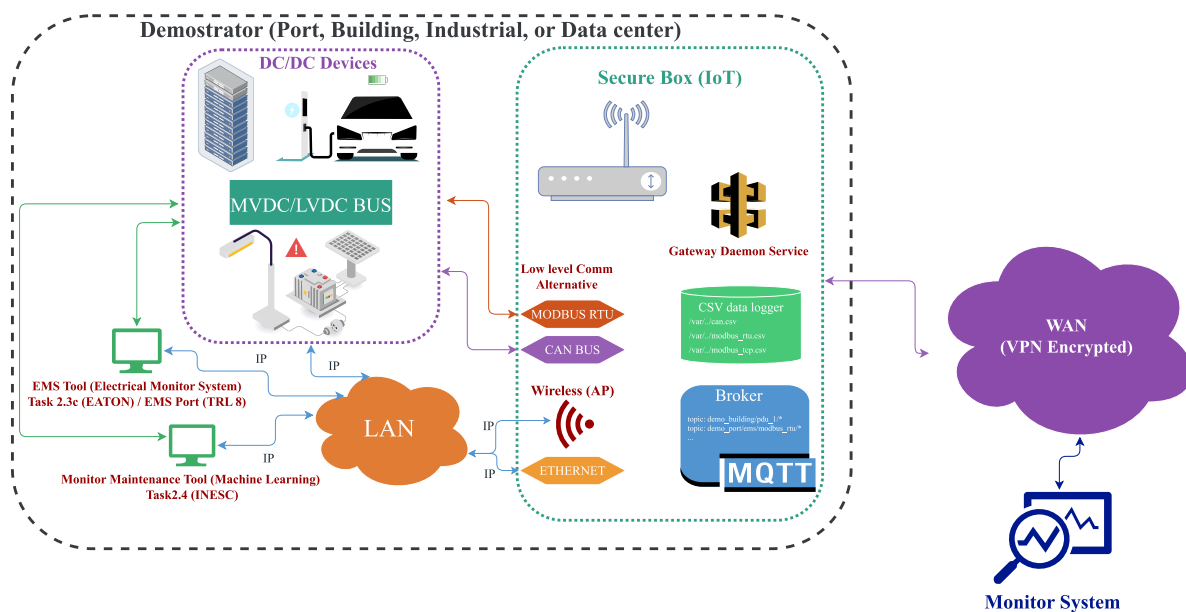


Figure 2 - Global architecture details

3.3 Cybersecurity

3.3.1 Summary

A comprehensive approach to securing a sensor-based data capture system must include measures at the hardware level, data encryption in transit and at rest, network security (VPNs, firewalls), robust authentication, role-based access control, and continuous monitoring.

Securing the remote server is critical to maintaining the confidentiality and integrity of the collected data.

To apply a cybersecurity layer to a system based on the capture of data from sensors and DC devices (voltage, current, power, alarms, etc.), ensuring secure transmission of variables to a remote web server from different demonstrators, it is essential to implement security measures across hardware, software, network, and encryption.

Below is an integral approach as example:

3.3.1.1 Hardware Security

- **Securing capture devices:** Sensors and controllers should be physically and electronically secured against tampering or unauthorized access. Authentication mechanisms on the devices themselves should prevent unauthorized data collection.
- **Digital signatures and hardware-based authentication:** Use hardware that supports **TPM** (Trusted Platform Module) or other security modules to ensure only authenticated devices can collect and send data.

3.3.1.2 Data Encryption in Transit

- **Secure transmission:** Encrypt data during transmission using secure protocols such as **TLS/SSL** to maintain confidentiality and integrity of the captured data.
- **End-to-End Encryption (E2EE [3]):** Implement E2E encryption to ensure data is encrypted from the point of capture to the remote server, and only the server can decrypt the information.

3.3.1.3 Network Security

- **VPN (Virtual Private Network):** Use a **VPN** to establish a secure connection between the sensors and the remote server, ensuring data travels through a private, encrypted channel, reducing interception risks.
- **Firewalls and Intrusion Detection/Prevention Systems (IDS/IPS):** Implement firewalls and IDS/IPS to monitor and block unauthorized access or suspicious activity on both the demonstrators and the remote server.

3.3.1.4 Authentication and Access Control

- **Device authentication:** Ensure that only authenticated and trusted devices are allowed to connect and send data to the remote server. This can be achieved through **digital certificates** or **authentication tokens**.
- **Role-Based Access Control (RBAC):** Implement RBAC to limit privileges, ensuring only authorized users can view or modify specific information.

3.3.1.5 Data Integrity

- **Digital signatures or cryptographic hashes:** Use **hashing** or **digital signatures** to verify the integrity of the data during transmission, ensuring any alterations can be detected.

3.3.1.6 Remote Server Security

- **Patches and updates:** Keep the remote server up to date with the latest security patches. Use reliable operating systems and software with continuous support.
- **Encryption at rest:** Data stored on the remote server should be encrypted, ensuring that even if an attacker accesses the database, the information is unreadable without decryption keys.
- **Database passwords:** Saving in the data base the hash of the password, not specific password.

3.3.1.7 Monitoring and Auditing

- **Continuous monitoring:** Use monitoring tools to detect unusual activities or unauthorized access on both the capture devices and the remote server.
- **Logging and auditing:** Maintain detailed logs of system activities to track incidents and perform security audits.

3.3.1.8 API Security (if applicable)

- If data is transmitted through APIs, ensure the APIs are secured using **authentication** and **encryption**.

3.3.1.9 Backup and Redundancy

- **Encrypted backups:** Regularly back up data in an encrypted format and store it securely to prevent unauthorized access.
- **Redundant data transmission:** Implement redundancy systems to prevent data loss or compromise in the event of network or device failure.

3.3.1.10 Cybersecurity Training and Awareness

Train personnel managing the devices and the remote server on cybersecurity best practices, common threats (like phishing or malware), and how to protect the system from external attacks.

After outlining the various approaches that can be applied to enhance the security layers of the communication system between different demonstrators, **VPN** could be identified as the primary solution due to its ability to create a secure, encrypted tunnel over potentially insecure network.

This approach ensures that data exchanged between demonstrators and the control platform remains protected from unauthorized access and tampering, providing a foundational layer of cybersecurity essential for the system's resilience. The following sections describe the VPN's implementation in greater detail, outlining its configuration, benefits, and potential challenges.

Finally, an alternative or complementary approach using **blockchain** technology and smart contracts is proposed, adding another dimension of security to the system.

3.3.2 Using a VPN as main layer of cybersecurity

Using a VPN (Virtual Private Network) is an excellent example of an authentication and authorization measure to secure data transmission.

How it works:

3.3.2.1 Authentication

Before a device or user can access the network, authentication is required. This may involve the following:

- **User Credentials:** Users must enter a username and password to access the VPN.

- **Multi-Factor Authentication (MFA):** MFA can be implemented to add an additional layer of security. For instance, after entering credentials, the user may receive a code on their mobile phone, which must also be entered.
- **Public/private keys:** as an alternative to the usage of plain credentials, a system of public and private keys could be used to identify a client entity inside the VPN network.

3.3.2.2 Authorization

Once authenticated, the VPN can manage authorization:

- **Roles and Permissions:** Users can be assigned specific roles that determine which resources or data they can access within the network. For example, a sensor operator may have access to voltage and current data, while an administrator would have full access.
- **Network Segmentation:** Through the VPN, network segments can be created to limit access to certain devices or areas within the network, ensuring that only authorized parts of the system can communicate with each other.

3.3.2.3 Benefits of Using a VPN

- **Data Encryption:** The VPN encrypts communication between the device and the server, preventing third parties from intercepting data during transmission.
- **Risk Reduction:** By establishing a secure tunnel, the risk of attacks such as “Man-In-The-Middle” (MITM), where an attacker could attempt to intercept or alter communications, is reduced.
- **Secure Remote Access:** It enables devices in remote locations to securely connect to the central server, facilitating the collection of data from distributed sensors.

3.3.2.4 Additional Advantages of a VPN for Cybersecurity

In addition to the benefits already mentioned, several other advantages make VPNs a powerful tool for cybersecurity:

- **Use in EU countries:** The use of VPNs is permitted and legal in all EU countries, with no general bans preventing their use in these territories. In fact, many companies and private users rely on this technology to safeguard online privacy and security. Additionally, compliance with GDPR is essential when operating within the EU. It is important to ensure that chosen solutions meet the requirements of the General Data Protection Regulation (GDPR), including adequate guarantees from service providers regarding data handling and protection.
- **Anonymity and Privacy Protection:** A VPN hides the true IP address of the user or device, making it difficult for third parties, such as hackers or unauthorized entities, to trace communication back to the original source. This enhances privacy and reduces the risk of tracking. By routing traffic through servers in different geographic locations, it also prevents attackers or unauthorized services from determining the physical location of the user or device, effectively obfuscating location.
- **Bypassing Geographic and Network Restrictions:** VPNs allow users or devices to bypass network restrictions that might otherwise limit access to certain resources (Accessing Restricted Resources). This can be useful in environments where firewalls or geographic restrictions prevent necessary data flow or communication with central servers. VPNs can

circumvent network throttling measures, i.e. avoiding network throttling, as they encrypt traffic, making it harder for service providers to identify and limit specific activities.

- **Enhanced Security in Untrusted Networks:** VPNs are particularly useful when connecting to public or untrusted networks, such as those in airports or cafes. These networks are often vulnerable to attacks, but the encryption provided ensures that, even if data is intercepted, it remains unreadable. This technology helps reduce the risk of common network-based attacks like DNS spoofing, packet sniffing, or session hijacking, maintaining communication integrity even on unsecured networks.
- **Centralized Management and Policy Enforcement:** VPNs enable organizations to centrally manage and enforce security policies across all connected devices, ensuring that all traffic adheres to security standards and reducing the risk of individual users bypassing these controls. They can also integrate with firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS), adding an extra layer of defense by monitoring traffic as it passes through.
- **Scalability and Flexibility:** VPNs are highly scalable, making them suitable for expanding systems with more devices, users, or demonstrators across multiple locations. This infrastructure can be easily expanded as an organization's needs grow without compromising security. It also supports multiple platforms and devices, such as desktops, mobile devices, and IoT sensors, ensuring consistent security across the entire ecosystem.
- **Cost-Effective Solution:** VPNs enable secure communication over the internet, reducing the need for expensive leased lines or dedicated communication channels, making them a cost-effective solution for securing data transmission over long distances. Additionally, they can be managed with minimal infrastructure investment compared to deploying physical security measures or complex private networks.
- **Compliance with Data Protection Regulations:** Many industries must comply with strict data protection and privacy regulations, such as GDPR and HIPAA. VPNs assist organizations in meeting these requirements by securing the transmission of sensitive data, protecting it from unauthorized access during transit. Additionally, they can be configured to log detailed access and activity, facilitating audits and ensuring compliance with industry standards.
- **Defence Against Distributed Denial of Service (DDoS) Attacks:** VPNs can help shield networks from DDoS attacks by masking the IP addresses of servers or devices, making it harder for attackers to target them directly. This capability is especially beneficial for critical systems that require high availability. Many providers offer load balancing services, distributing traffic across multiple servers to reduce the impact of potential attacks. Additionally, VPNs can serve as a foundational component of a zero-trust security model, which assumes no part of the network is inherently secure. By providing secure tunnels for traffic, they ensure that only authenticated users and devices can access specific resources, even within the network perimeter.

3.3.3 Using Blockchain as alternative/complement to VPN for cybersecurity

It is possible to use technologies like **blockchain** and **smart contracts** as an alternative or complement to **VPNs** for implementing cybersecurity in a system of sensors and devices transmitting data to a remote server.

Although blockchain and smart contracts do not replace VPNs in terms of establishing secure connections, they provide mechanisms for **authentication, authorization, data integrity, and decentralization** that can be extremely useful.

3.3.3.1 Blockchain for Authentication and Authorization

- **Decentralized authentication:** Blockchain enables a decentralized authentication model. Instead of relying on a central authority, the blockchain can securely store cryptographic keys or unique device identifiers. Devices sign their requests with private keys, and the blockchain network verifies their identity.
- **Access control via blockchain:** Blockchain can store and manage access permissions. Instead of a centralized database controlling which users or devices can access certain resources, blockchain ensures transparency and security in managing this information.

3.3.3.2 Smart Contracts for Security Policies

Smart contracts are programs that automatically execute when certain conditions are met in a blockchain network. They can be used to manage security and device interactions:

- **Automated authorization:** A smart contract can define the rules for allowing certain devices or users to access data or resources. If a device does not meet the necessary conditions, the smart contract automatically denies the request.
- **Immutable access logs:** Each device interaction can be recorded immutably on the blockchain, creating a transparent and auditable access log.
- **Managing permission updates:** When permissions need to be updated, smart contracts can be reprogrammed to enforce new security policies without requiring central administration.

3.3.3.3 Encryption and Privacy with Blockchain

- **Data integrity and non-repudiation:** Since blockchain is immutable, it ensures that data transmitted by sensors has not been tampered with. Transactions can back up each piece of data, verifying its authenticity.
- **Private blockchain networks:** While public blockchains are transparent, private or consortium blockchains offer more control over who can access the stored or exchanged data.
- **Decentralized key management:** Blockchain can serve as a decentralized key exchange platform, securely distributing cryptographic keys without relying on third-party key management services.

3.3.3.4 Data Transmission Security

Blockchain is not typically used to transport high volumes of data but can help ensure data authenticity:

- **Data signing:** Sensor data can be digitally signed and verified through blockchain, ensuring the data has not been altered.
- **Integrity proofs:** Hashes of sensor data can be stored on the blockchain, allowing verification that the data remains unchanged during transmission.

3.3.4 Conclusion using VPN as main layer of cybersecurity

VPNs offer a robust and versatile set of cybersecurity features, including privacy protection, secure access on untrusted networks, centralized management, and scalability.

They also help organizations comply with data protection regulations and defend against cyber threats such as **DDoS and MITM attacks**.

Additionally, **VPNs** integrate well with **firewalls, monitoring systems, and zero-trust architectures**, making them an essential tool for securing communication across distributed systems and remote locations.

By encrypting data, reducing attack risks, and providing secure remote access, VPNs prove to be an effective solution for protecting sensitive data and maintaining the integrity of communications in modern network environments.

In summary, using a **VPN** as part of your **IoT** cybersecurity strategy is completely viable and authorized in the **EU**.

It is a common practice to enhance data transmission security and is aligned with the region's policies on protecting data privacy and integrity.

Blockchain can serve as an alternative or **complement** to **VPNs** in IoT-based systems. While VPNs provide encrypted tunnels for secure data transmission, blockchain offers a decentralized, tamper-resistant ledger that ensures data integrity and transparency.

By using blockchain, each transaction between IoT devices can be securely recorded and verified without relying on a central authority, reducing the risk of data breaches or attacks.

This makes blockchain a valuable addition to enhance security, particularly in environments where trust and data authenticity are critical.

Combining both technologies can further strengthen the overall security framework of IoT systems.

3.4 Deployment requirements

This section will present a set of minimum software/hardware requirements for the deployment of the future EMS that will be installed, for example, in the port demonstrator.

While no hardware vendor or models will be mentioned here, it is important to meet these requirements so the local platform of every demonstrator (whereas it is the EMS of task 2.3c or any other platform installable by each demo site responsible partner).

- **OS compatibility:** the EMS (or any other local control platform) should be compatible with a Linux based Operating System, whether it ends up running on the host system or inside Docker containers (or any other containerization system)
- **CPU:** while no clock speed will be specified, it is very reasonable to assume that the deployment host will mount an ARM 64 bits processor. ARM processors, while much more power efficient than any other desktop architectures (such as amd64 processors), are not mainly intended to run heavy loads of work
- **Memory:** most devices like latest models of Raspberry Pi (or other known SBC) have memory limitations compared to desktop computers and laptops (not to mention servers of any kind). Such constraints are related to the technology of the memory chips (LPDDR4 for example in comparison to DDR4 or DDR5) and the amount of memory (while a server can include huge amounts of RAM, devices like the Raspberry Pi only include 8GB at maximum). This means that the platform of each demonstrator needs to be lightweight regarding the usage of memory (this relates more to the quantity of memory used more than to the technology of the memory chips)
- **Storage:** it is intended that the information read from the DC devices of each demonstrator is stored in a dedicated server for such purpose. Despite this, the gateway device proposed for the deployment of the local platform of each demonstrator will have a certain amount of space designated mainly for the deployment of a local control platform. Storage in these devices is usually limited by a slower storage technology (usually eMMC instead of a NVMe or SATA III disk) and a lower amount of available disk space. The platform should take into account that the available disk space won't be as high as it would be in an industrial-grade server

Research is being conducted to determine which device is more suitable for this purpose. While there are some alternatives that could be suitable, specifying a brand or a model is out of the scope of this deliverable, and will only be done when deliverable 2.8 is written on a mid-term future for the completion of task 2.5 about implementation of the architecture designed here.

4 Conclusions

4.1 Summary

This deliverable presents an overall view of the architecture that will be developed in the Task 2.5.

The document has focused its attention on how the connection of the different devices of each demonstrator can be made, not only locally within each demo site, but also between demonstrators.

The two main reasons for such interconnection are the following ones: the first one, a network architecture has to be provided so control of each demo site can be achieved; and the second one, a monitoring platform will be developed in Task 2.5 so the global status of the system can be supervised.

4.2 Progress

The devices that will be present in each demonstrator have been detailed as much as possible.

The information available at the moment is related not only to the variety of assets for each demo site, but also to the kind of information that will be exchanged at a later stage when Task 2.5 starts.

Research about devices that could be suitable as deployment/gateway device for each demo site has been performed, although it is reasonable not to have gone as deep as expected in this topic due to this document being about design (and not about implementation). When the proper task begins, one will be chosen among a selection detailing as much as possible the choice from a technical perspective.

The decision of using a VPN has been made taking into account several aspects analysed in their own section (mainly because cybersecurity is something that transcends any particular demo site). This VPN will ensure interconnection and security between the main gateway devices of each demo site and CIRCE's monitoring platform from Task 2.5.

4.3 Main Challenges

Due to technical reasons, there is a lack of information that is reasonably believed to be mitigated in the next Deliverable 2.8.

This information is mostly related to what each device will be able to exchange within the communications network of the global system (and its format).

In the case of other demonstrators, the location of the demo site itself is what has conditioned the provisioning of information coming out of the devices.

Another aspect that has had an impact on the architecture design is the choice of devices for the communication itself. This does not refer to assets of the demo sites as such (EV chargers, PV inverters and so on), but to the gateway devices that will be used to export the information of each demo site.

Because the local platform of each demo site (in case there is a local platform) is intended to be installed in the gateway device itself, this represents some challenges: the local platform of each demo site has to be light enough to run on a device that is not expected to have (by design) much processing power (related to CPU, memory and storage capabilities). The gateway device will need to be versatile enough to potentially connect to multiple physical assets located in its corresponding demo site (with industrial grade protocol such as Modbus TCP, Modbus RTU or CAN), as well as having connection to the internet whether it is through a 4G/5G modem, Wi-Fi or a LAN cable connected to a locally installed router.

4.4 Next deliverables

This document will be de determinant of what will be made in Task 2.5 and, therefore, it will greatly impact what will appear in the corresponding Deliverable 2.8.

It has been a design document, having it potentially presented some situations where many choices are suitable for a later implementation. This means that when the programming and implementation phases take place, it is possible that the resulting system is not completely equal to what has been designed here with all the information that has been gathered.

This is because of the following reasons:

- When thinking about a possible design (while trying to be abstract from implementation as much as possible), it is very likely that the real implementation raises issues and concerns that hadn't appeared in the previous design stage.
- Leaving aside the development activities of the system, it is important to recall that it has not been possible to retrieve all the information of the devices from all the demonstrators. While a considerable amount of information has been gathered in this document about what assets will be installed, how they will be interconnected and the information they will exchange, there are still some aspects that will need to be discussed along the implementation phase itself during the Task 2.5.

Deliverable 2.6 (EMS for Hybrid AC/DC systems) will also have influence from this document. The catalogue of available AC/DC devices that will be installed in the demonstrators (or at least the ones from the building demonstrator, that will be the one with which the EMS will be tested) won't be conditioned by how the devices are connected but, what will be conditioned by the deployment is the EMS platform.

Such platform will have to be designed and implemented paying attention to two details, the gateway device where it will run and the assets from the demo site that it will need to be connected to.

5 References

- [1] Amanda Tucker, [“What is RSA Asymmetric Encryption? How Does it Work?”](#), Securew2, 30th of January 2024.
- [2] [“How does SSL work? | SSL certificates and TLS”](#), Cloudflare.
- [3] Randy Battat, [“End-to-End Encryption: What is it & How it Works”](#), Preveil, 30th of August 2024.

ANNEX I

Section 1: How does a public/private key system work?

In the cybersecurity context, a pair of public/private RSA keys [1] is an element used to uniquely identify an entity, being such entity a server, a person or any other element.

This encryption system is not only used within communications itself, but generally to ensure that someone (or something) is what it is supposed to be.

The basic concept behind the usage of this methodology is the following one:

- Two or more entities have a public/private key pair
- The sender entity encrypts a message (let's assume an array of bytes) using the public key of the receiver
- Only the legitimate receiver will be able to decrypt the incoming message because only its private key will work with the public key used to encrypt the message

It doesn't mind if there is a malicious entity intercepts the original message as long as the private key of the receiver hasn't been filtered or stolen.

Section 2: How is SSL/TLS related to the usage of RSA public/private keys?

Related to the explanation of the previous section, the SSL/TLS [2] system for securely communicating a server and a client relies on the use of certificates and public/private keys.

This process involves two purposes:

- Encryption
- Authentication

By using the public-private keys mechanism, the data exchange between a client and a server that use SSL/TLS is ensured to remain private and only usable by these two actors.

Not only the data has to be secure, but also the actors themselves must be trusty. This is what certificates achieve, authentication. A connection to a server (no matter which kind of operations it performs) can be trusted thanks to a chain of certificates (having it a root certificate). A certificate represents the identity of an actor within a communications network. This identity of course could be fraudulent so, the main way this mechanism has of tackling this is by the representation of a certification authority (CA). In the already mentioned chain of certificates, a CA is a superior entity that is supposed to be trustful, whether it is a final CA or even if it has more authorities above in the chain of certificates.

Depending on the country/region, there might be different organizations or public entities that oversee validating which other entities are entitled as Certification Authorities. Depending on multiple criteria, like the usage of the certificates, their origin and so on, OS vendors (or any other kind of vendor) end up choosing a preset of certificates to install in their systems (this applies to desktop devices but also to mobile phones, for example).

Apart from the set of certificates included as a base in a device, the end user might want or need to install additional certificates. Such action should only be done with certificates from a legitimate

source, because a malicious entity could for example try to supplant a trusty entity by issuing a certificate apparently well formed.