SMHUB product description

- 1. Introduction
- 2. Hardware Overview
- 3. Getting Started
- 4. Software & System
- 5. Network & Connectivity
- 6. Radios & Protocols
- 7. User Interface
- 8. Modules & Extensions
- 10. Glossary

1. Introduction

Enterprise-Ready Smart Home Platform

SMHUB is a **Linux-based smart home hub** designed for reliability, scalability, and openness. It runs a lightweight **Linux OS layer** with **Node.js preinstalled**, making it capable of running **major open-source smart applications** such as:

- **Zigbee2MQTT (Z2M)** → enterprise-proven Zigbee integration with MQTT backbone.
- **Matterbridge** → bridging Zigbee and Thread devices into the Matter ecosystem (Google Home, Apple Home, Amazon Alexa).
- **Node-RED** → a flow-based automation tool for building powerful automations and integrations.

This software stack ensures SMHUB can be deployed in **enterprise projects**, **system integrator environments**, and **large-scale smart installations** while remaining flexible for continuous growth.

Modular and Scalable

Unlike fixed-function consumer hubs, SMHUB is **hardware- and software-extensible**:

- Additional radios (Z-Wave, 4G/LTE) can be added via modules or USB dongles.
- System apps can be extended with **Node.js libraries or Python scripts**.
- Storage options include onboard eMMC and expandable SD card storage.
- Remote management is possible through **SSH, VPN (WireGuard), and DDNS services**.

This makes SMHUB suitable for **smart buildings, multi-site enterprise deployments, and integrators who need a universal controller**.

For Home Users & DIY Enthusiasts

While SMHUB is enterprise-ready, it is also **accessible to enthusiasts**:

- **Preinstalled apps** → Zigbee2MQTT, Matterbridge, Node-RED work out of the box.
- **Web interface** → intuitive dashboard for system monitoring and configuration.
- **USB passthrough** → attach external dongles (Zigbee, Thread, Z-Wave, etc.) for custom setups.
- **Ambilight LEDs, IR transmitter/receiver, buzzer, and audio** for interactive smart home features.

Whether you’re an integrator managing **hundreds of devices across buildings** or a DIY user running **Home Assistant in your living room**, SMHUB adapts to your use case.

2. Hardware Overview

SMHUB is a **Linux-based modular hub** designed to unify connectivity, automation, and control across multiple smart home and IoT standards. Its hardware architecture combines **high-performance networking, multiple radios, built-in peripherals, and modular power options**.

2.1 Core System

- **Processor:** 2 separate cores: C906@1GHz (for Linux) + C906@700MHz (for RTOS).
  C906 is an ultra-high-efficiency processor compatible with the RV64IMA[F]C[V] instruction set. It is one of the industry's first mass-produced processors supporting the RISC-V vector extension.
- **NPU (Neural Processing Unit):** 0.5 TOPS@INT8.
- **Memory:** 512 MB RAM.
- **Storage:** Onboard eMMC storage plus **microSD slot** for expansion.
- **Operating System:** Embedded Linux with Node.js runtime, supporting smart applications such as Zigbee2MQTT, Matterbridge, Node-RED, and custom Node.js/Python services.

2.2 Networking

- **Ethernet:** Integrated RJ-45 port for wired backbone connectivity.
- **Wi-Fi:**
  - Dual-band **2.4 GHz and 5 GHz** support.
  - Can operate in **Access Point** mode or **Client** mode.
  - CLI setup available, with web UI configuration in development.
- **Bluetooth:**
  - **Bluetooth 5.0** compliant.
  - Supports **Bluetooth Low Energy (BLE)** for pairing, sensors, and beacons.
  - Available for integrations and custom applications.

2.3 Radios

- **Zigbee (CC26xx)** → dedicated radio for Zigbee networks, fully compatible with Zigbee2MQTT.

- **Thread (EFR32MG series)** → native support for Thread and OpenThread Border Router (OTBR).
- **EFR32ZG23 Expansion Module** → provides **Sub-GHz communications including Z-Wave**, integrated via UART3 with dedicated reset, flash, and LED lines.

2.4 Expansion Modules

- **UPS Module (SM-HUB UPS)**
  - Powered by **18650 Li-ion battery cells**.
  - Charging via TP5100.
  - Step-up/step-down converters deliver **5 V, 3.3 V, and 1.8 V rails**.
  - **INA226 sensor** monitors voltage and current for telemetry.
  - Solar/DC charging input supported.
- **PoE Module (SM-HUB PoE)**
  - 802.3af compliant (Power over Ethernet).
  - Converts Ethernet-supplied power to regulated 5 V.
  - Surge and ESD protection integrated.
  - “Power-check” signals exposed for system monitoring.
- **4G/LTE Module (SM-HUB 4G)**
  - Based on **SIM7672G modem** with **GNSS (GPS/GLONASS)**.
  - NanoSIM slot with SIM detect.
  - Interfaces: main UART, USB, debug UART.
  - LTE and GNSS antenna connectors.
  - Power and network LED indicators.

2.5 Built-in Peripherals

- **LEDs:**
  - **12 × WS2812B addressable RGB LEDs** – configurable for system visualization, animations, or Ambilight effects.
  - **4 × service LEDs** – indicating power, network, and radio activity.
- **Buttons:**
  - **Reset button**.
  - **Pairing/Function button**.
  - **User-configurable button**.
- **Audio Output:**
  - **3.5 mm audio jack**.
  - Supports system notifications, alerts, or media playback.
- **Buzzer:**
  - Integrated piezoelectric buzzer.
  - Usable for system alerts, notifications, or user automation rules.

- **IR Modules:**
  - **IR transmitter** for controlling TVs, AC units, and other IR devices.
  - **IR receiver** for capturing signals from existing remotes.

2.6 Power Architecture

- **Primary Power:** USB-C PD input.
- **Optional Power Sources:**
  - PoE (via PoE module).
  - UPS battery module.
  - Solar/DC input (through UPS module).
- **Smart USB Power Switching:** Automatic host/device detection on USB-C CC lines with integrated FET switching. Prevents reverse powering and ensures safe operation whether SMHUB is supplying power to peripherals or being powered itself.

2.7 Expansion & DIY Interfaces

For developers and hardware enthusiasts, SMHUB exposes additional interfaces:

- **GPIO headers** with UART, SPI, and I²C.
- **DIY pinouts** for custom modules, sensors, or experimental expansions.

3. Getting Started

This section explains how to unbox, power, and access your SMHUB for the first time.

3.1 Unboxing

Inside the SMHUB package, you will find:

- **SMHUB main unit**
- **2x 5dB antennas**
- **Quick start guide**
- (Optional, depending on order) **PoE module**, **UPS module**, or **4G/LTE module**

⚠ Some accessories such as **18650 batteries (for UPS)**, **SIM card (for LTE)**, or **external antennas** may need to be purchased separately.

3.2 Powering SMHUB

SMHUB supports multiple power options. Choose the one that fits your setup:

1. **USB-C Power (default)**
   - Connect the included USB-C cable to the **USB-C (upstream/power) port** on the back of the device.
   - Use a 5V/2A USB-C adapter.
2. **PoE (Power over Ethernet)** *(requires PoE module)*
   - Install the PoE add-on module.
   - Connect SMHUB to a PoE-enabled Ethernet switch (802.3af standard).
3. **UPS Module with Batteries** *(optional)*
   - Insert supported **18650 Li-ion cells** into the UPS board.
   - Connect the UPS board to the main unit.
   - SMHUB will automatically switch to battery power during outages.
4. **Solar/DC Input via UPS** *(optional)*
   - If using the UPS module, SMHUB can also be powered from an external DC or solar supply.

⚠ **Important:** Thanks to integrated **USB power switching logic**, SMHUB automatically detects whether it should act as a **host** (powering peripherals) or **device** (drawing power).

3.3 Network Connection

You can connect SMHUB to your network in different ways:

1. **Ethernet (recommended)**
   - Connect an Ethernet cable to the LAN port.
   - SMHUB will obtain an IP address via DHCP.
2. **Wi-Fi Client Mode**
   - Connect via CLI/SSH (see section 3.4).
   - Use the `nmcli` command to scan and connect to your Wi-Fi network.
   - Web-based Wi-Fi setup is in development.
3. **Wi-Fi Access Point Mode**
   - On first boot, SMHUB may start in AP mode.
   - SSID: `SMHUB-XXXX` (last 4 characters of MAC).
   - Default password: `smhub1234`.
   - Connect and then open the web UI.

3.4 First Access

Once powered and connected, you can access SMHUB in several ways:

Web Interface

> On the first SMHUB models (with old firmware) the web interface will be on port **3000**
> To it access use: `http://smhub.local:3000`
> **If you have such a device, we strongly recommend that you immediately update it via USB according to this guide**
> Old firmware has critical issues that have been fixed in the new version!

- Open a browser and go to:
  - `http://smhub.local` (mDNS)
  - or the IP address assigned via DHCP.

SSH Access

- Connect from a terminal: `ssh smlight@smhub.local`
- Default credentials are user: **smlight**, password: **smlight**
- Use SSH for advanced configuration and Wi-Fi setup until the web Wi-Fi interface is fully implemented.

USB-C Upstream Mode

- If connected via the **USB-C upstream port**, SMHUB may appear as a network device to your host machine.
- This allows direct connection without a router.

3.5 First Boot Behavior

- The system automatically starts **Zigbee2MQTT** service. Other services like **Matterbridge** and **Node-RED** could be run manually afterwards.
- Service LEDs and RGB LEDs will indicate boot sequence and status.
- On first login, you should:
  1. Change the default password (via web UI or SSH).
  2. Configure your preferred network connection (Ethernet or Wi-Fi).
  3. Update system software (see Section 4).

3.6 Safety Notes

- Always use certified power supplies when running from USB-C.
- If using the UPS module, only install **good quality18650 cells**.
- Ensure proper ventilation - SMHUB is designed for continuous 24/7 operation.

4. Software & System

SMHUB is powered by a **Linux-based operating system** with an integrated **Node.js runtime** and a suite of **smart home applications**. This software foundation provides both **enterprise-grade stability** and **DIY flexibility**.

4.1 Operating System

- **Base:** Lightweight Linux distribution optimized for embedded controllers.
- **Runtime:** Node.js preinstalled for native execution of Node.js applications.
- **Package Management:** Extendable via Linux packages and Python scripts.
- **Update System:**
  - Secure firmware updates.
  - Individual application updates independent of OS.

4.2 Preinstalled Applications

Zigbee2MQTT (Z2M)

- **Preinstalled by default**.
- Works with onboard CC26xx or EFR32XX Zigbee radios.
- Publishes device data to MQTT.
- Compatible with Home Assistant, OpenHAB, Node-RED, etc.
- Local dashboard for Zigbee pairing and monitoring.

Matterbridge

- **Preinstalled by default**.
- Bridges Zigbee/Thread devices into the Matter ecosystem.
- Compatible with Google Home, Apple Home, Amazon Alexa, SmartThings.

Node-RED

- **Preinstalled by default**.
- Visual, flow-based automation editor.
- Includes libraries for MQTT, Z2M, and Matterbridge.
- Accessible via web browser.

Mosquitto MQTT Broker

- **Preinstalled and running by default**.
- Provides the MQTT backbone for Zigbee2MQTT, Node-RED, and Matterbridge.
- Allows additional MQTT clients to integrate seamlessly with SMHUB.

4.3 System Services

- **Networking**
  - Ethernet (default).
  - Wi-Fi 2.4 GHz and 5 GHz (AP and Client modes).
  - Bluetooth 5.0 with BLE.
- **VPN Support**
  - WireGuard preinstalled.
  - Secure remote access tunnels.
- **USB Passthrough**
  - Full passthrough of USB dongles to the host stack.
  - USB protection logic for safe host/device switching.

4.4 Built-In Visual System

- **Ambilight System**
  - Runs continuously by default.
  - Uses 12 × WS2812B RGB LEDs for system visualization, notifications, or ambient effects.
  - User-configurable patterns and colors.
- **Service LEDs**
  - Indicate power, network status, and radio activity.

4.5 Features in Development

- **Buzzer Control**
  - System and user notifications.
- **Audio Output (3.5 mm jack)**
  - System alerts, notifications, and future media playback.
- **IR Transmitter & Receiver**
  - IR learning and replay for remote-controlled devices.
- **Extended LED Controls**
  - Custom animations and media sync in Ambilight.

4.6 Developer Options

- Full **SSH access**.
- Node.js environment ready.
- Support for Python scripting and Linux daemons.
- Exposed GPIO, UART, I²C, and SPI for DIY hardware extensions.

5. Network & Connectivity

SMHUB provides robust and flexible networking options to ensure reliable operation in both **enterprise environments** and **home setups**. It can connect via **Ethernet, Wi-Fi (2.4/5 GHz), Bluetooth**, and optionally via **VPN tunnels and DDNS services** for secure remote access.

5.1 Ethernet

- **Primary interface** for stable operation.
- Supports **100Mb Ethernet** via onboard RJ-45 port.
- Recommended for enterprise and multi-device deployments.
- Provides lowest latency and highest reliability compared to wireless options.

5.2 Wi-Fi

SMHUB includes **dual-band Wi-Fi** with both **2.4 GHz** and **5 GHz** support.

- **2.4 GHz**
  - Longer range, higher wall penetration.
  - Ideal for IoT-heavy environments with many legacy devices.
- **5 GHz**
  - Higher throughput.
  - Less interference in crowded environments.
  - Suitable for high-speed data transfers and local automations.

Operating Modes:

- **Client Mode**
  - SMHUB connects to an existing Wi-Fi network.
  - Configurable via CLI (`nmcli`) or SSH.
  - Web UI configuration will be available in upcoming updates.
- **Access Point Mode (in development)**
  - SMHUB creates its own Wi-Fi network.
  - Default SSID: `SMHUB-XXXX` (last 4 MAC digits).
  - Default password: `smhub1234`.
  - Useful for first-time setup or when no LAN is available.

5.3 Bluetooth

- **Bluetooth 5.0** with **BLE (Bluetooth Low Energy)** support.
- Future updates will expand Bluetooth integration into the web interface.

5.4 VPN (WireGuard)

SMHUB includes **WireGuard VPN** for secure tunneling.

- Provides encrypted remote access to SMHUB and connected devices.
- Suitable for enterprise remote deployments or secure home use.
- Configuration available via CLI or SSH.
- Can be combined with DDNS for access without a static IP.

5.5 USB Networking

When connected via **USB-C upstream**, SMHUB can act as a **network device** to the host PC:

- Appears as an Ethernet-over-USB adapter.
- Provides direct access without needing an external router or switch.
- Useful for initial setup, debugging, or portable use cases.

6. Radios & Protocols

SMHUB integrates multiple radios to support a **broad range of smart home and IoT protocols**, making it a true **multi-radio coordinator and Matter bridge**.

6.1 Zigbee

- **Chipset:** TI CC26xx series.
- **Port:** /dev/ttyS1
- **Baudrate**: 115200
- **RTS/CTS**: false
- **Protocol:** Zigbee 3.0 and legacy devices.
- **Software:** Zigbee2MQTT (Z2M).
- **Integration:**
  - Pairs and manages Zigbee devices directly.
  - Publishes states/events via the preinstalled **Mosquitto MQTT broker**.
  - Compatible with Home Assistant, OpenHAB, Node-RED, and other MQTT platforms.
- **Coordinator Mode:** Can operate as a standalone Zigbee coordinator, without additional servers.

6.2 Thread

- **Chipset:** Silicon Labs EFR32MG series.
- **Port:** /dev/ttyS2
- **Baudrate**: 115200
- **RTS/CTS**: false
- **Protocol:** Thread 1.1/1.2.
- **Integration:**
  - Acts as an **OpenThread Border Router (OTBR)**.
  - Works with **Matterbridge**, exposing Thread/Zigbee devices into Matter.
- **Use Cases:**
  - Bridging Zigbee sensors to Matter.
  - Running mixed Zigbee + Thread networks.

6.3 Z-Wave

- **Radio Module:** EFR32ZG23 expansion module.
- **Port:** /dev/ttyS3
- **Baudrate**: 115200
- **RTS/CTS**: false
- **Protocol:** Z-Wave 700/800 series, including **Z-Wave Long Range (LR)**.
- **Software:** Managed via Z-Wave JS (installable).
- **Integration:**
  - Local Z-Wave or Z-Wave LR networks.
  - Automation via Node-RED and MQTT.
- **Hardware:** Dedicated LED and reset/flash pins for diagnostics and firmware updates.

6.4 4G/LTE (in development)

- **Module:** SIM7672G LTE modem.
- **Port:** /dev/ttyS4
- **Baudrate**: 115200
- **RTS/CTS**: false
- **Capabilities:**
  - LTE Cat 1 connectivity.
  - Interfaces: USB, debug UART.
  - Antenna connectors for LTE and GNSS.
  - SIM slot with SIM detection.
- **Status:** Hardware fully integrated, **software support is in active development**.
- **Planned Functions:**
  - LTE fallback connectivity.
  - Remote site deployments without Ethernet/Wi-Fi.

6.5 Wi-Fi

- **Bands:** Dual-band **2.4 GHz** and **5 GHz**.
- **Modes:** Access Point (AP) and Client.
- **Use Cases:**
  - 2.4 GHz: long range, high compatibility.
  - 5 GHz: higher bandwidth, reduced interference.
- **Integration:** Used for device connectivity and as a network backbone when Ethernet is not available.

6.6 Bluetooth

- **Version:** Bluetooth 5.0.
- **Support:** Bluetooth Low Energy (BLE).

6.7 Multi-Radio Operation

- Zigbee + Thread + Wi-Fi + Bluetooth operate simultaneously out of the box.
- Z-Wave and 4G/LTE are available via expansion modules.
- Hardware and firmware are designed to minimize interference and ensure stable concurrent radio operations.

6.8 Matter Bridge

SMHUB can function as a **Matter bridge**, allowing non-Matter devices to join modern ecosystems:

- Bridges Zigbee and Thread devices into Matter.
- Compatible with Google Home, Apple Home, Amazon Alexa, SmartThings.
- Entirely **local bridging** — no third-party cloud services required.

7. User Interface

The SMHUB web interface is the primary way to configure and manage the system. It provides access to dashboards, built-in applications, and detailed system settings.

7.1 Dashboard

The **Dashboard** is the starting page of the interface and shows an overview of the hub:

- **System Information**
  - Device model, firmware version, uptime.
  - CPU load, memory usage, and storage status.
- **Network Status**
  - Ethernet/Wi-Fi interfaces with IP addresses.
  - VPN (WireGuard) status.
- **Radio Status**
  - Zigbee, Thread, Z-Wave (if installed), Bluetooth.
  - Service LED indicators and Ambilight effects.
- **Service Overview**
  - Running state of **Zigbee2MQTT, Matterbridge, Node-RED, Mosquitto MQTT broker**.
  - Quick buttons to start/stop services.

7.2 Zigbee2MQTT

- Direct integration with the **Zigbee2MQTT frontend**.
- Allows pairing, removing, and managing Zigbee devices.
- Displays a network map, device states, and logs.
- Zigbee devices are automatically published to the local **Mosquitto MQTT broker**.

7.3 Matterbridge

- Provides direct access to the **Matterbridge frontend**.

- Manages bridging between Zigbee/Thread devices and Matter ecosystems (Google Home, Apple Home, Alexa, SmartThings).
- Allows configuration of bridge modes and Matter controllers.

7.4 Node-RED

- Opens the **Node-RED frontend**, a visual automation editor.
- Users can create and deploy flows combining Zigbee2MQTT, Matterbridge, MQTT, HTTP APIs, or custom integrations.
- Pre-installed flows include templates for MQTT and device automation.

7.5 Apps

The **Apps** section acts as a package manager for SMHUB applications.

- **App Management**
  - Install, uninstall, or update applications.
  - Configure settings per application.
  - Start/stop applications with one click.
- **Preinstalled Apps**
  - Zigbee2MQTT
  - Matterbridge
  - Node-RED
  - Mosquitto MQTT broker
- **Additional Apps**
  - Future apps (e.g., Z-Wave JS, custom services) can be added here.
  - Apps can be distributed as packages and installed via this page.

7.6 Settings

The **Settings** section contains all configuration options for SMHUB. It is divided into modules, each with tabs for **configuration, telemetry, and debug** where available.

System

- Hostname, time zone, system info.
- User management (accounts, passwords).
- Updates (firmware & app updates).
- Backup & restore.
- Factory reset.

Network

- Ethernet (DHCP/static).
- Wi-Fi (2.4/5 GHz, AP/Client mode).
- VPN (WireGuard).
- DDNS configuration (DuckDNS, others).
- USB-over-network.

Radios

- Zigbee: coordinator settings, pairing mode, diagnostics.
- Thread: OTBR management, Matterbridge integration.
- Z-Wave (if module installed): inclusion/exclusion, network management.
- Bluetooth: enable/disable, scanning, BLE integrations.
- 4G/LTE (in development): SIM/APN settings, LTE telemetry, GPS data.

Audio & Buzzer

- Volume control (3.5 mm output).
- Enable/disable system notification sounds.
- Buzzer toggle and test alerts.

LEDs

- Service LED control.
- Ambilight modes (status, animations, user-defined patterns).

IR

- IR transmitter: configure/send commands.
- IR receiver: learn remote commands and store them.

Storage

- SD card and eMMC info.
- Format, mount/unmount.
- Telemetry: usage stats.

Security

- User authentication settings.
- SSH toggle.
- VPN certificates and DDNS keys.
- Firewall and remote access rules.

8. Modules & Extensions

SMHUB is designed to be modular. The **main board** provides essential radios, networking, and I/O, while additional **hardware modules** extend functionality for power resilience, connectivity, and advanced integrations.

8.1 USB Passthrough

- **Purpose:** Enables SMHUB to act either as a **USB host** (powering connected peripherals) or as a **USB device** (connected to a PC or another host).
- **Logic:**
  - Automatically decides whether to supply or draw power.
  - Prevents reverse powering through integrated FET switching.
- **Use Cases:**
  - Attach Z-Wave or extra Zigbee dongles, printers, audio devices, storage etc.
  - Connect SMHUB to a PC for development/debugging.
  - Power downstream devices safely.

8.2 SD Card

- **Slot:** Standard microSD slot on the main board.
- **Purpose:** Expand storage for logs, app data, or backups.
- **Integration:**
  - Mounted automatically by the Linux OS.
  - Users can format, mount/unmount, and monitor usage from the UI.
  - Support up to 2Tb volume.

8.3 PoE Module

- **Standard:** IEEE 802.3af-compliant.
- **Input:** Power delivered via Ethernet cable from a PoE-enabled switch or injector.
- **Output:** Stable **5V regulated** supply to the hub.
- **Protection:** Surge and ESD protection integrated.

- **Monitoring:** “PoE Power Check” exposed in the system for telemetry.
- **Use Case:** Ideal for enterprise deployments where powering over Ethernet is standard.

8.4 UPS Module

- **Battery:** Supports **18650 Li-ion cells** (user-installed).
- **Charger:** advanced circuit with overcharge/undervoltage protection.
- **Conversion:** Step-up/step-down converters to maintain **5V, 3.3V, 1.8V** rails.
- **Telemetry:**
  - Monitors voltage/current
  - Provides battery health and charge state in the UI.
- **Inputs:** Can also accept **DC power** or **solar panels**, making SMHUB deployable off-grid.
- **Use Case:** Ensures uninterrupted operation during outages.

8.5 4G/LTE Module

- **Connectivity:** LTE Cat 1 with fallback modes.
- **Interfaces:**
  - Main UART.
  - USB.
  - Debug UART.
- **SIM Slot:** NanoSIM with detection circuitry.
- **Antennas:** Dedicated connectors for LTE and GNSS.
- **UI Integration:**
  - Settings → Radios → 4G/LTE.
  - SIM/APN configuration, LTE signal telemetry.
- **Status:** Hardware available, **software support in development**.

8.6 DIY Pinouts & Interfaces

For advanced users and developers, SMHUB exposes additional hardware interfaces:

- **GPIO pins** for general use.
- **UART, SPI, and I²C** buses.
- **Custom module connectors** for prototyping or specialized hardware.

- **Use Case:** Attach sensors, control boards, or experimental expansions not covered by official modules.

8.7 Integrated Peripherals

While not separate modules, the following are considered part of SMHUB’s extension capabilities:

- **IR Transmitter & Receiver**
  - Control TVs, AC units, and IR-based devices.
  - Capture IR codes for learning mode.
- **Audio Output (3.5 mm)**
  - System alerts and future media/streaming support.
- **Buzzer**
  - Local audible notifications.
- **LED Systems**
  - **12 WS2812B RGB LEDs (Ambilight)** – running by default, user configurable.
  - **4 Service LEDs** – network, power, radio status.

10. Glossary

Ambilight – Integrated RGB LED ring used for status indication and visual effects.

Apps – Modular software packages (e.g., Zigbee2MQTT, Node-RED) that can be installed/uninstalled independently on SMHUB.

BLE (Bluetooth Low Energy) – Energy-efficient version of Bluetooth for sensors, tags, and small IoT devices.

DDNS (Dynamic DNS) – A service that maps a dynamic IP to a permanent domain name for remote access.

Matter – A connectivity standard that allows devices from different ecosystems (Google, Apple, Amazon, SmartThings) to interoperate.

Matterbridge – An application that bridges Zigbee/Thread devices to the Matter ecosystem.

MQTT (Message Queuing Telemetry Transport) – A lightweight messaging protocol used for IoT communications. SMHUB includes a built-in **Mosquitto MQTT broker**.

Node-RED – A flow-based programming tool for IoT automation, preinstalled on SMHUB.

OTBR (OpenThread Border Router) – Service that connects Thread devices to IP networks.

PoE (Power over Ethernet) – A method of delivering power and data over the same Ethernet cable.

UPS (Uninterruptible Power Supply) – Add-on module with 18650 cells providing backup power during outages.

WireGuard – A modern VPN protocol, preinstalled on SMHUB for secure remote access.

Zigbee2MQTT (Z2M) – Application that connects Zigbee devices to MQTT brokers, enabling integration with multiple smart home platforms.

Z-Wave JS – An open-source software stack for managing Z-Wave devices.