IoT Internet of Things

Hartmut Seitter

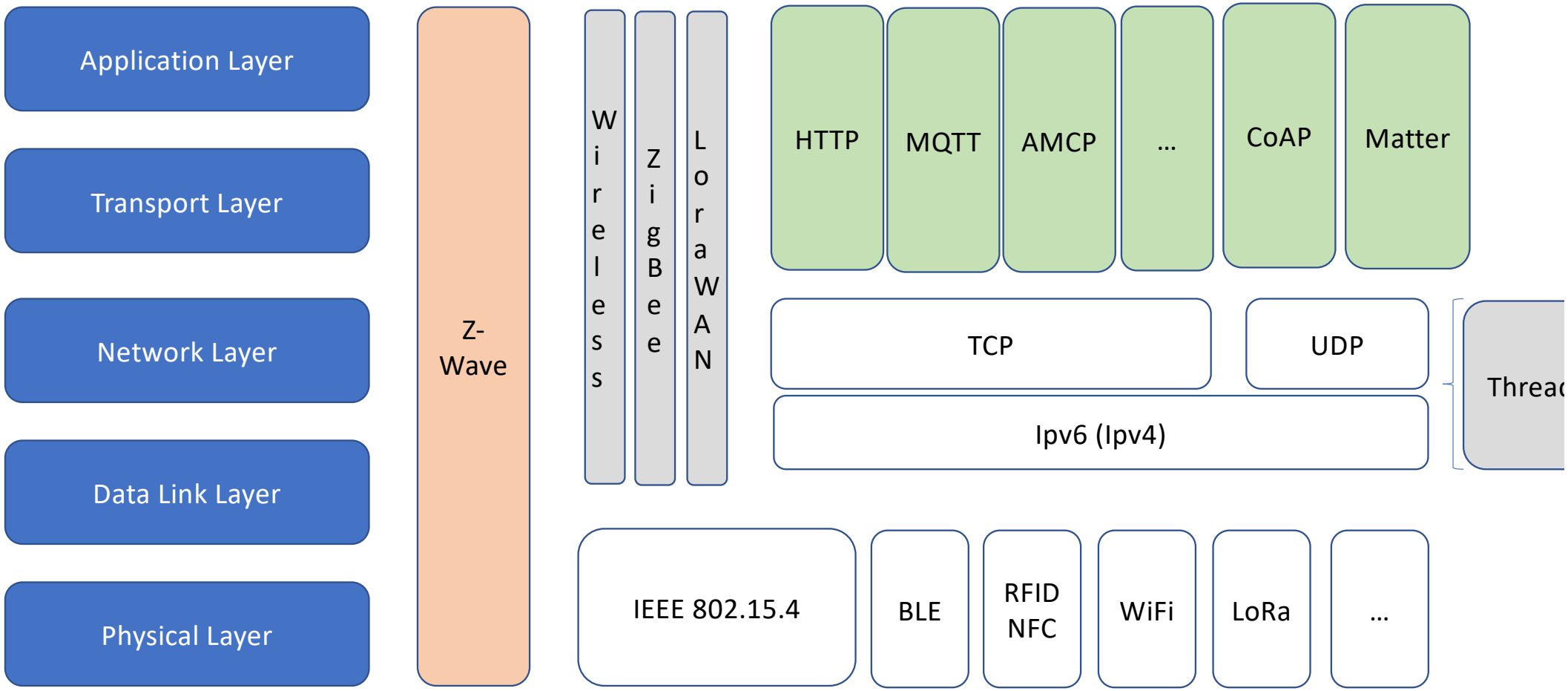
Agenda – Topics which should be covered in this course

- Start with IoT Lab
- **Connectivity and Networking**
- IoT Networks and Communication frameworks
  - MQTT
  - LoRa**WAN**
  - **CoAp**
  - **ZigBee**
  - **Thread, Matter**
- **Embedded power supplies, energy harvesting and constrains**
- **Internet principals, routing for IoT**
- **Information security and privacy concepts**
- **Big Data and Machine Learning**

IoT Protocols

- **Remember in IoT we have**
  - **Sensors which has an interface to the microcontroller**
    - **Analog signal, digital signal, I2C bus, onewire bus, BLE and other …**
  - **Microcontroller which communicate with a gateway or direct to a server using**
    - **MQTT (over TCP), Wifi, LTE, ZigBee, LoRaWAN, other**
    - **Most times they communicate wireless, however communication via Ethernet is also used depending on use cases**
  - **Gateway which receive the data from the microcontroller and forward it to a Server Application – most time TCP/IP based**
  - **Server application which receive the data for pre-/post-processing**

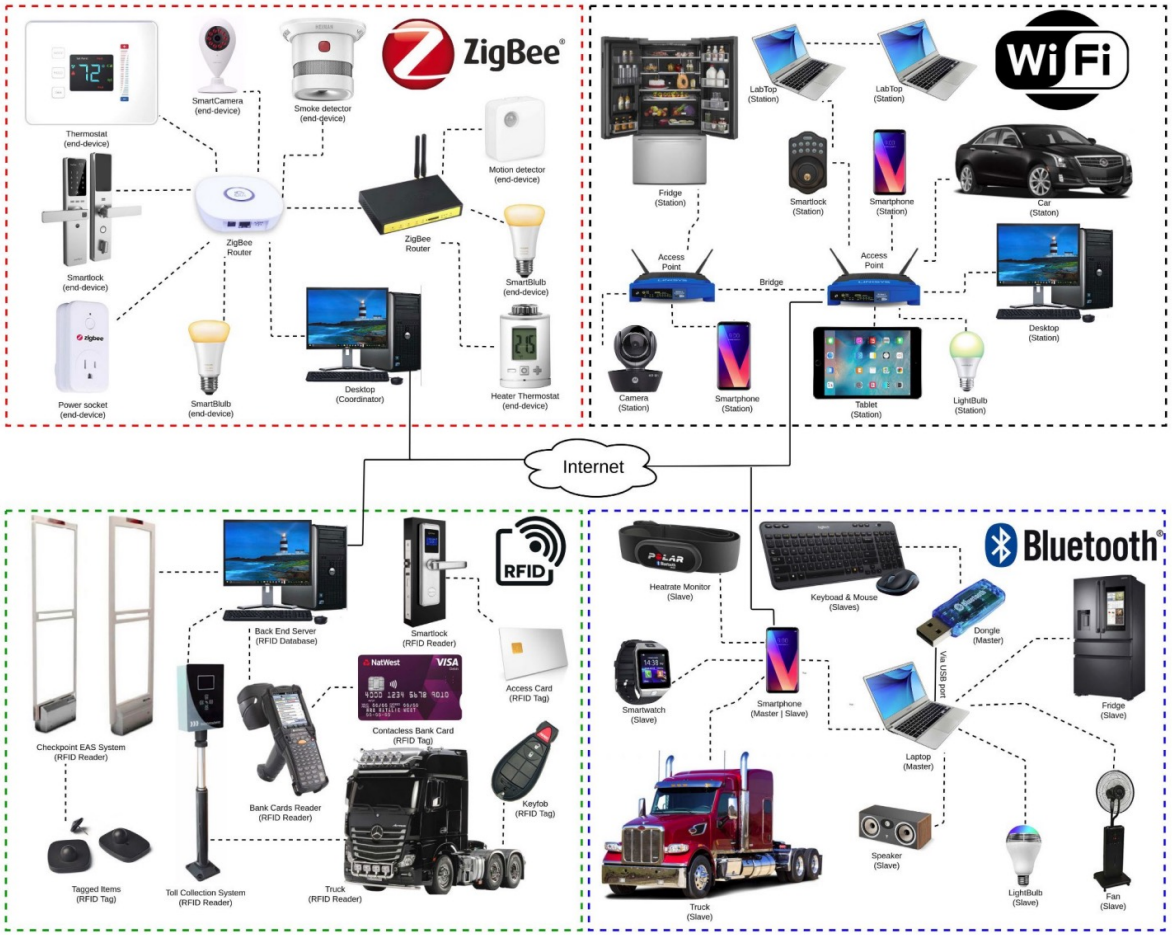
The IoT Network Stack



Source: https://www.mdpi.com/2079-9292/9/1/111/htm

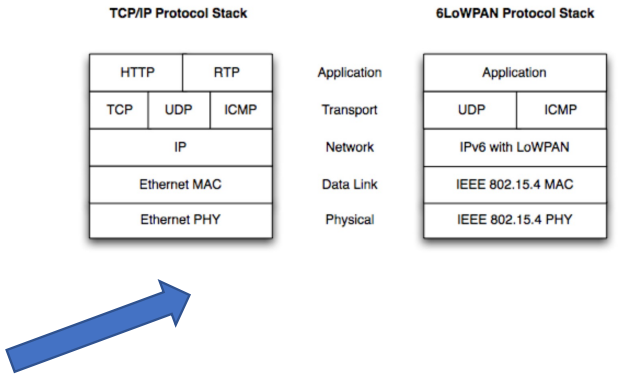
With my own modification

The Physical / Data Link Layer



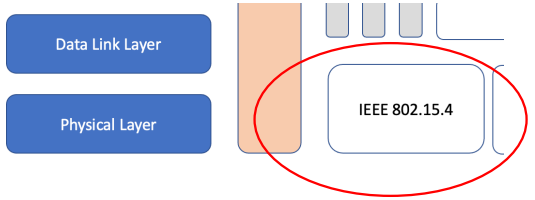
Source: K. Lounis, M. Zulkernine: Attacks and Defenses in Short-Range Wireless Technologies for IoT

Physical layer IEEE 802.15.4



- Standard for low power IoT networks
- Physical & data link layers
- Base for ZigBee, Thread, WirelessHART
- 6LoWPAN relies on it (*IPv6 over Low Power Wireless Personal Network*)
- Support for multiple topologies (star, mesh)
- 64-bit MAC addresses, 16-bit short addresses

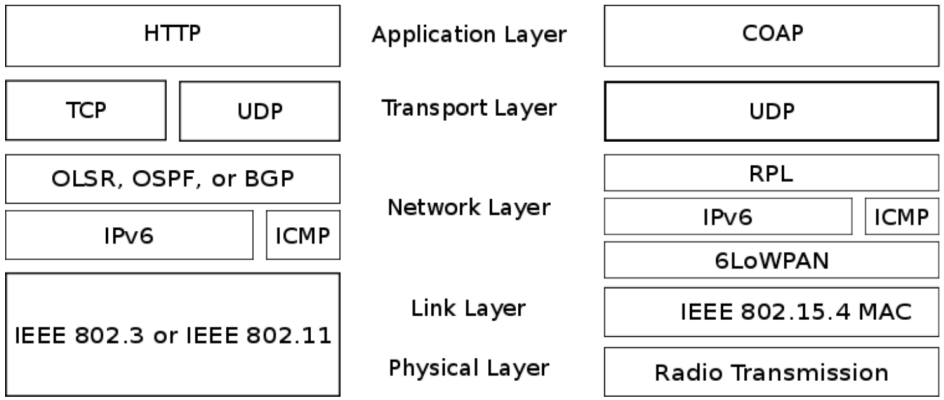
- Small packet size – 128 bytes including MAC, 103 bytes payload
- Data rates between 20kbps and 250kbps
- Range: – Usually between 10m and 30m – Some strong transceivers
- Hundreds of meters / km
- Line of sight
- ZigBee Pro



Application Layer Protocol

- HTTP – HyperText Transfer Protocol
- CoAP – Constrained Application Protocol
- WebSocket
- MQTT – Message Queue Telemetry Transport
- XMPP – eXtensible Messaging and Presence Protocol
- DDS – Data Distribution Service
- AMQP – Advanced Message Queuing Protocol
- Matter

Different protocol stack models or the future

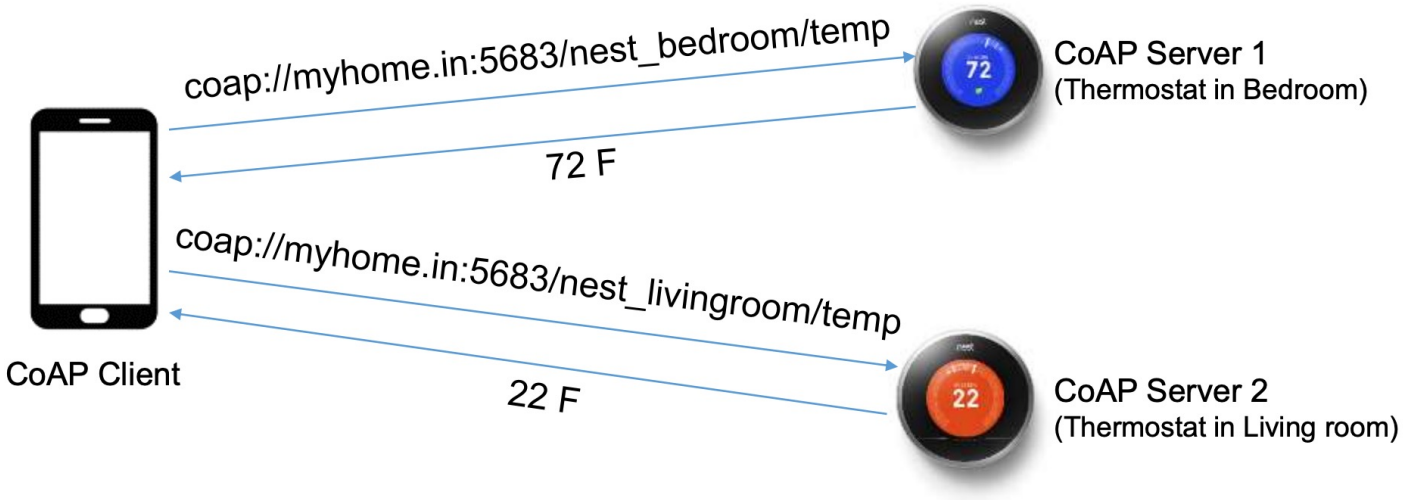


And another IoT application protocol - CoAP

- **CoAP -> Constrained Application Protocol**
  - **It is a specialized web transfer protocol for use with constrained nodes and constrained networks in IoT**
  - **The protocol is designed for machine-to-machine communications and IoT applications such as smart energy and building automation**

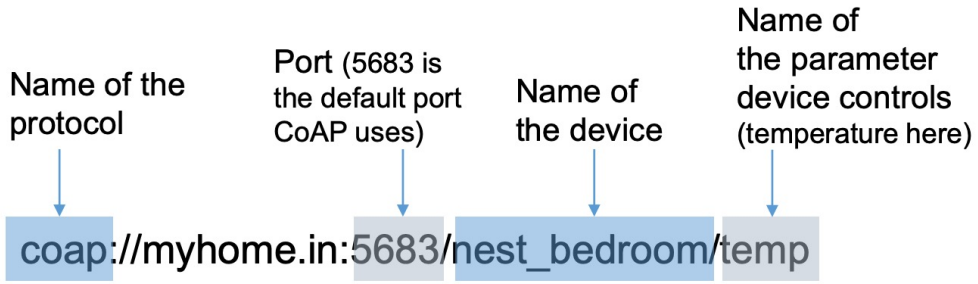
  - Inspired from HTTP – similar to HTTP. CoAP also uses a request response model.
  - Can be transparently mapped to HTTP - However, CoAP also provides features that go beyond HTTP such as native push notifications and group communication.
  - From a developer point of view, CoAP feels very much like HTTP. Obtaining a value from a sensor is not much different from obtaining a value from a Web API.
  - REST model for small devices - It implements the REST architectural style
  - Made for billions of nodes with very less memory - The Internet of Things will need billions of nodes, many of which will need to be inexpensive. CoAP has been designed to work on microcontrollers with as low as 10 KiB of RAM and 100 KiB of code space.
  - Designed to use minimal resources, both on the device and on the network. Instead of a complex transport stack, it gets by with UDP on IP. A 4-byte fixed header and a compact encoding of options enables small messages that cause no or little fragmentation on the link layer.
  - Like HTTP, CoAP can carry different types of payloads, and can identify which payload type is being used. CoAP integrates with XML, JSON, CBOR, or any data format of your choice.
  - Discovery integrated - CoAP resource directory provides a way to discover the properties of the nodes (devices/things in IoT) on your network

CoAP in 2 foils – just to be aware it exist and it is used in IoT



CoAP Methods

Get
Post
Put
Delete



MQTT vs CoAP

The following distinctions should be made between the MQTT and CoAP protocols:

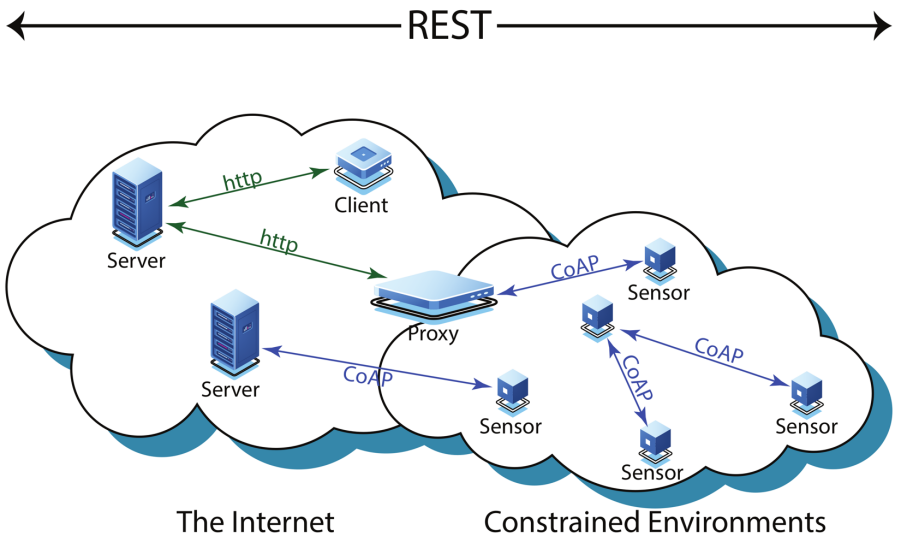
	CoAP	MQTT
Communication Model	Request/Response	Publish/Subscribe
Transport Protocol	User Datagram Protocol (UDP)	Transmission Control Protocol (TCP)
Security	DTLS over UDP	SSL/TLS over TCP
Bandwidth Usage	Very minimal, ideal for very small networks	Efficient, but requires additional data
Reliability	Built-in acknowledgments possible but optional	High reliability through TCP
Architecture	Direct communication between devices	Communication via a central broker

CoAP strengths and weaknesses

- The Constrained Application Protocol offers numerous advantages for resource-constrained and connected devices but also comes with certain technical challenges. Below is a summary of the main strengths and weaknesses of the CoAP standard.
- **Strengths of CoAP:**
- Lightweight and requires minimal memory and processing power
- Ideal for battery-powered devices
- Low overhead saves bandwidth and energy
- Easy integration into existing IoT systems thanks to REST-based architecture
- Supports multicast for addressing multiple devices simultaneously
- Asynchronous communication capability increases efficiency in connected systems
-
- **Weaknesses of CoAP:**
- Uses UDP without built-in error correction, which can lead to packet loss
- Requires its own mechanisms for reliability, such as acknowledgment messages
- Not directly compatible with traditional web protocols like HTTP
- Data integration requires proxies, increasing system complexity

ESP32 - Arduino Framework and CoAP

- For CoAP on ESP32, you can use the ESP-IDF CoAP client/server examples to connect to a server, send GET/PUT/POST/DELETE requests, and handle secure DTLS connections.

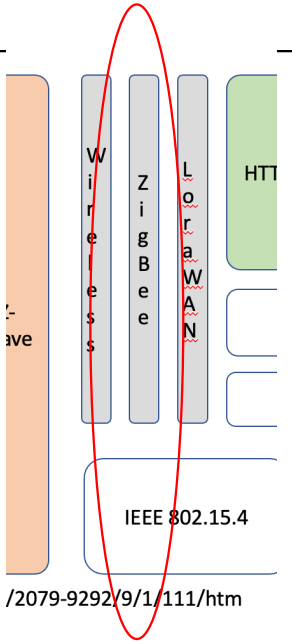


A few words about ZigBee

- **Good to know ☺**
  - **The technique that honey bees uses to communicate new-found food sources to other members of the colony is referred as the ZigBee principle.**

- **The ZigBee standard is currently an ‘open’ standard only to those that are part of the ZigBee Alliance. For this reason, the ZigBee standard was not used to implement the application layer.**

- **ZigBee** is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used to create personal area networks with small, low-power digital radios, such as for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small scale projects which need wireless connection. Hence, ZigBee is a low-power, low data rate, and close proximity (i.e., personal area) wireless ad hoc network.



ZigBee general characteristics

- Data rate of 20kbps up to 250kbps
- Intended for 2,45GHz, 868MHz and 915Mhz band
- Data rates
  - 250kbps on 2,45Ghz
  - 40kbps on 915 Mhz
  - 20kbps on 868Mhz

- Star or Peer to Peer network topologies
- Support for low latency devices
- Handshaking
- Low power consumption
- Extremely low duty cycle (<0,1%)

ZigBee what does it do?

- Designed for wireless controls and sensors
- Operates in Personal Area Networks (PAN's) and device to device networks
- Connectivity between small packet devices
- Control of lights, switches, thermostats, appliances, ….

- On devices there are three different nodes available
  - ZigBee coordinator node
  - ZigBee full function node
  - ZigBee reduced function node

- The mode of operation is
  - Beacon and
  - Non-beacon

ZigBee Coordinator node

- Root of the network tree
- Acts as a bridge to other networks
- Stores info about the network
- There is only one ZigBee coordinator node for the complete network
- Acts as the trust centre and is the repository for security keys.

- Set the permissions
- Allow user access
- Coordinate area network

- It starts the network with predefined parameter
- works in the same way as ZigBee router after start

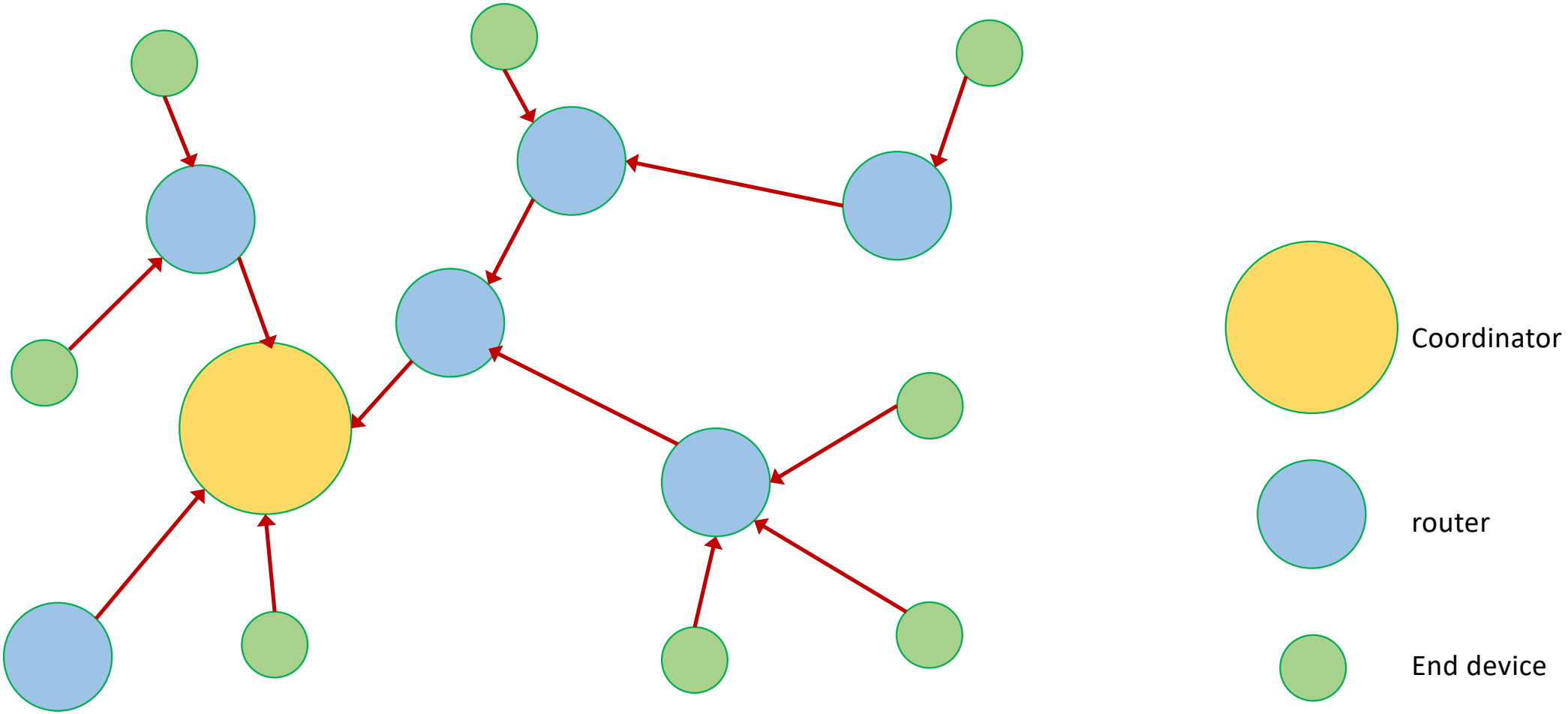
ZigBee full function node

- An intermediate router in the network
- Transmitting and Receiving data from other devices
- Needs less memory than ZigBee coordinator node
- Cheaper
- Can operate on all topologies

ZigBee reduced function node

- Also called End Device
- Device capable of talking in the network
- Can’t relay data from other devices
- Cheaper than full function node

Device types and roles

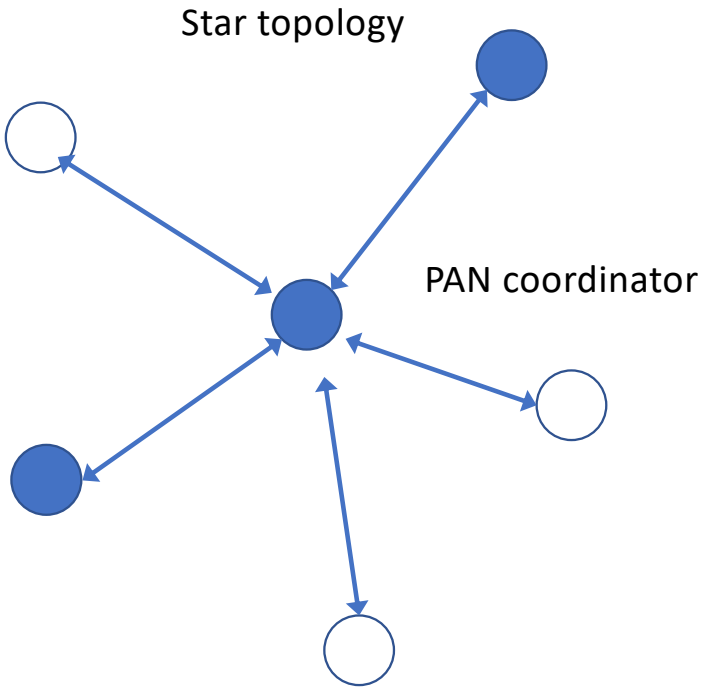


ZigBee mode of operations

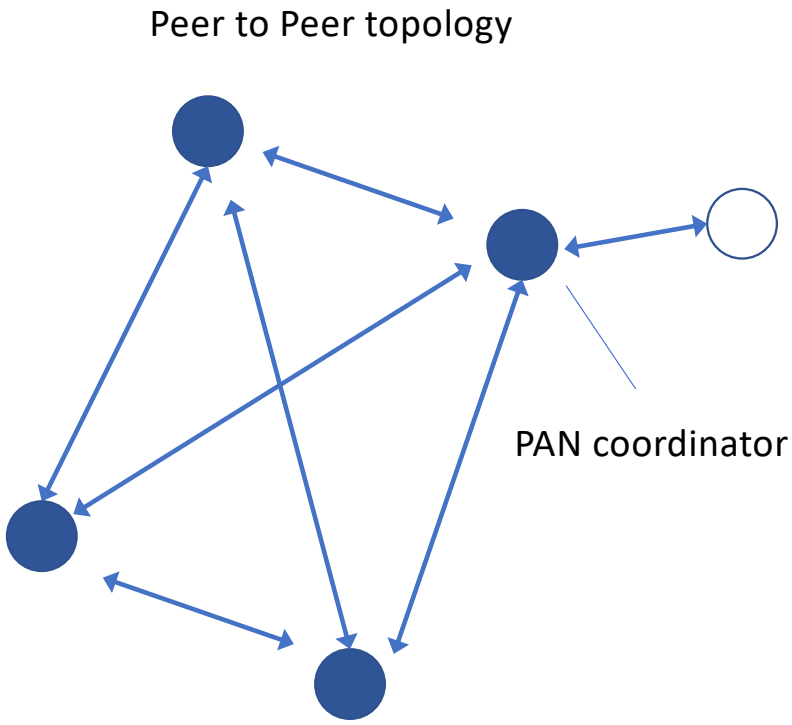
- **Beacon mode:**
  - **In beacon enabled networks, the special network node call ZigBee Router transmit periodically beacons to confirm their presence to other network nodes. Nodes may sleep between beacons, thus lowering their duty cycle and extending their battery life**
- **Non beacon mode**
  - **In non-beacon enabled networks, an CSMA/CA (Carrier-sense multiple access with collision avoidance) channel access mechanism is used.**

    ZigBee routers typical have their receivers continuedly active to monitor this CSMA/CA channel

ZigBee network topology

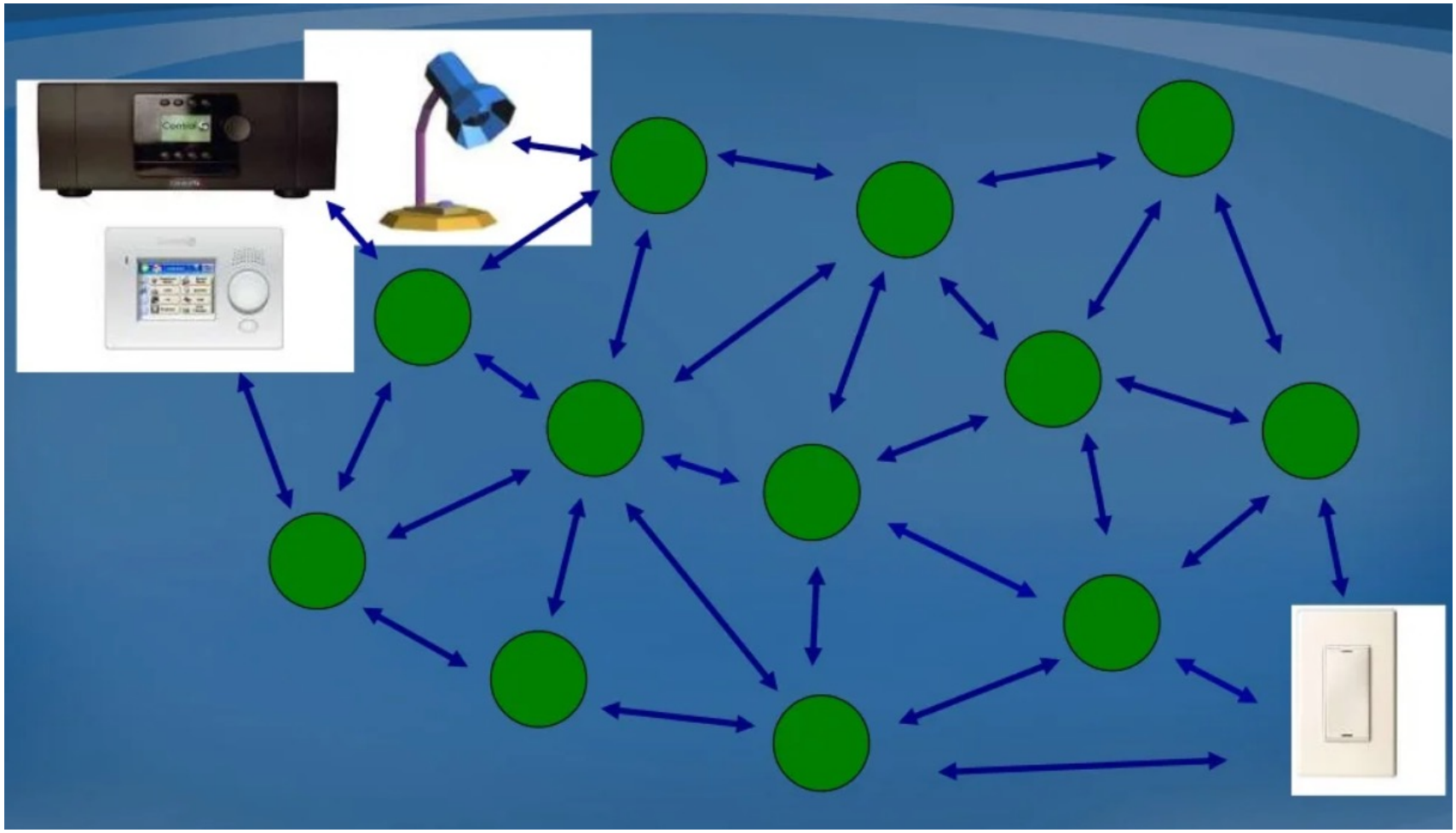


Reduced function device

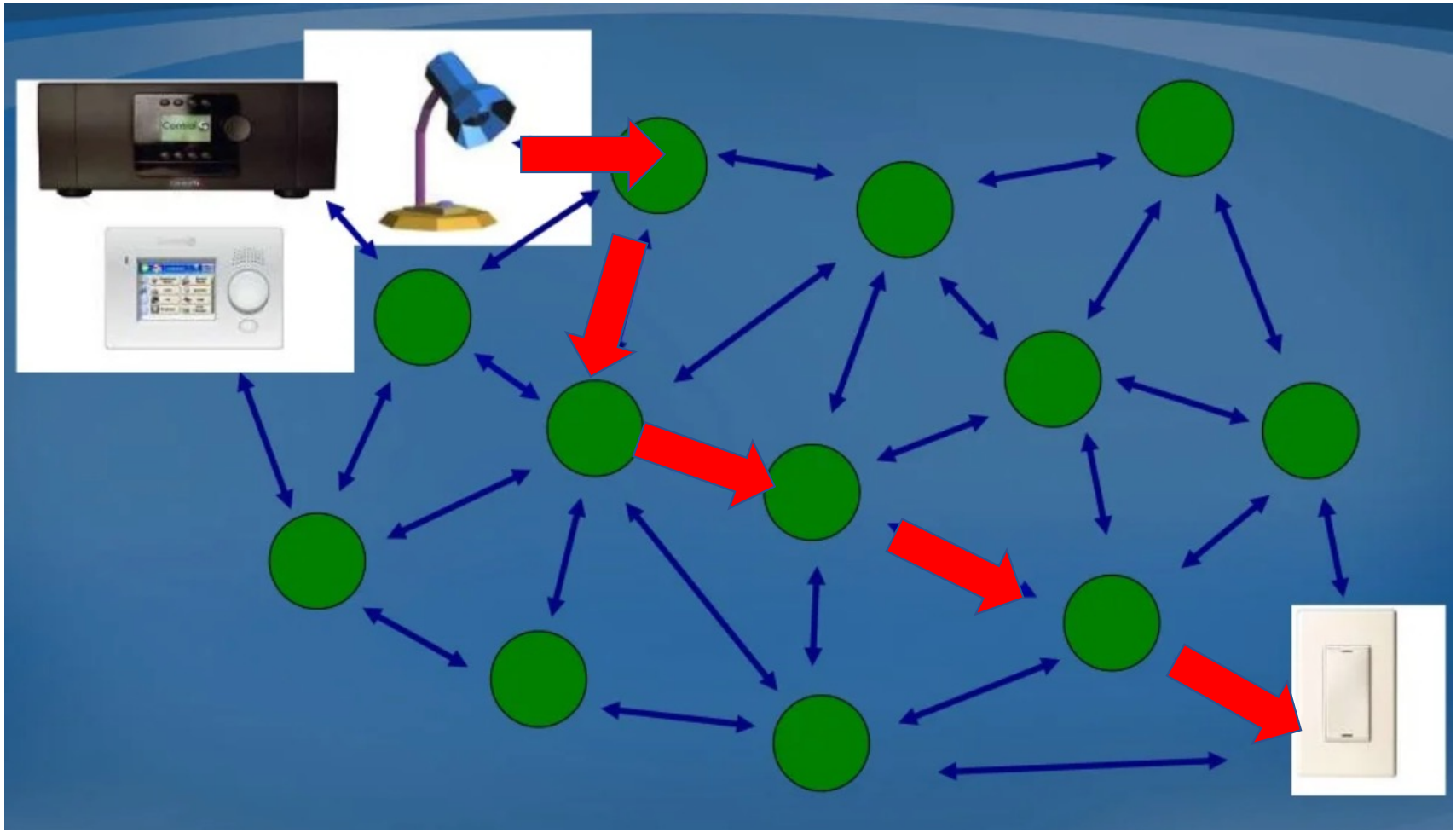


Full function device

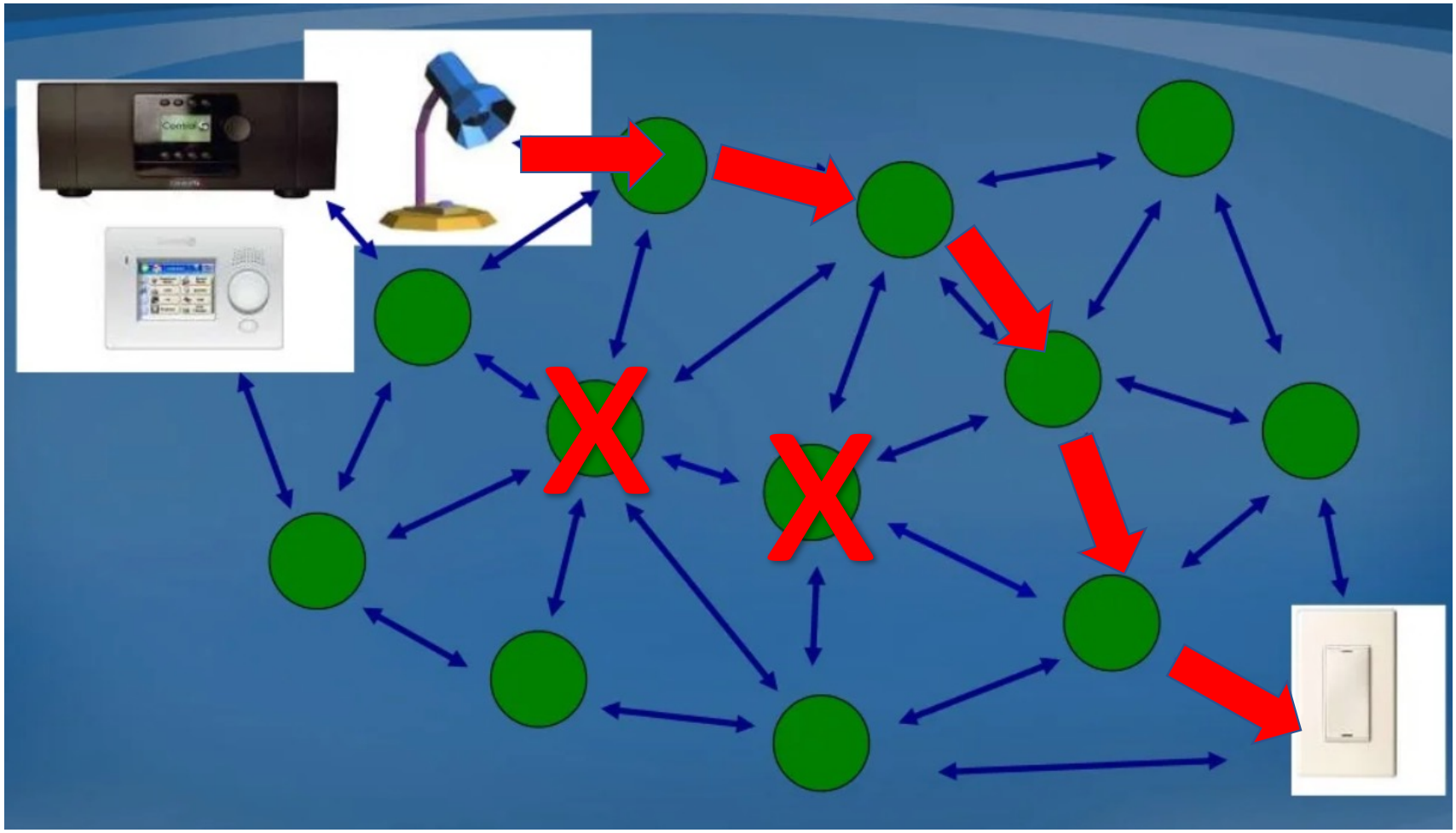
ZigBee mash network



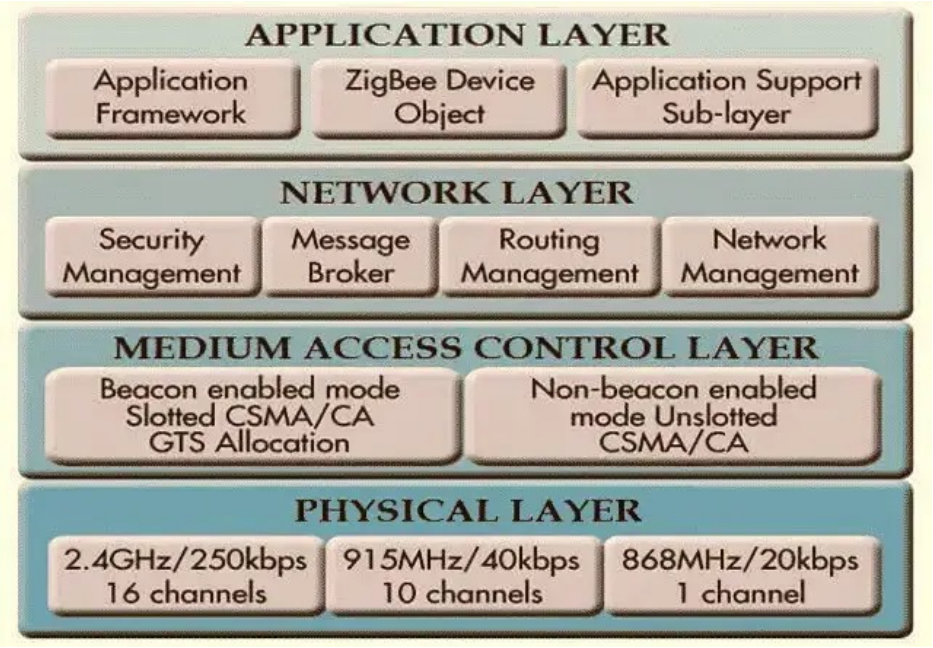
ZigBee mash network

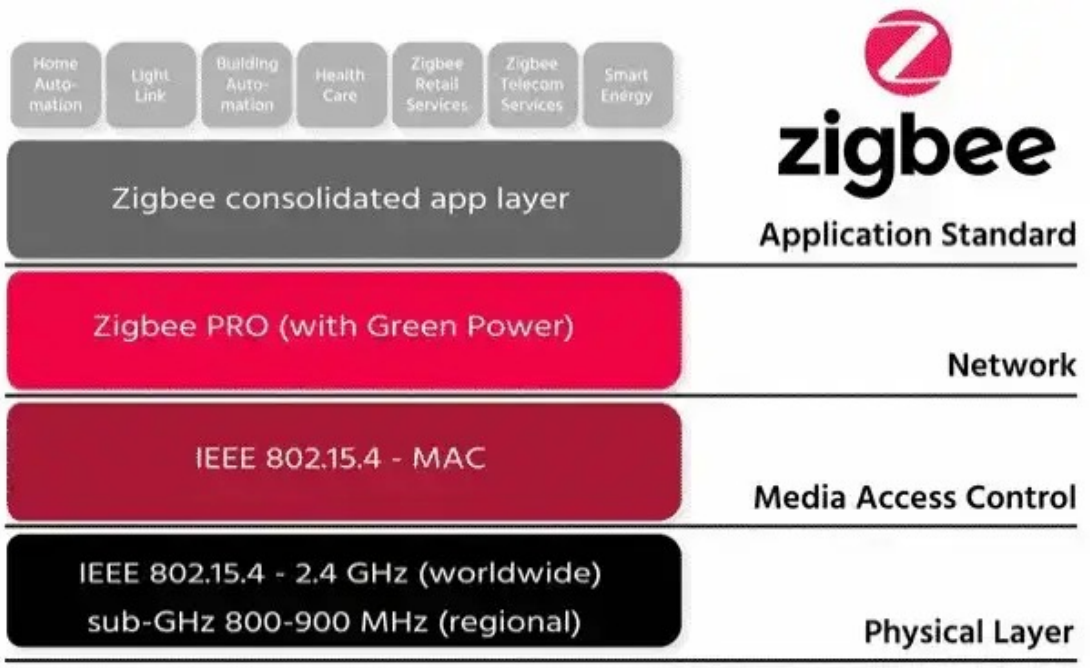


ZigBee mash network



ZigBee architecture





Device addressing

- All devices have a IEEE address
- Short addresses can be allocated
- Addressing modes:

  - Network + device identifier
  - Source / destination identifier
  - Source / destination cluster tree + device identifier

  - Every device has a 64 bit mac address
  - 16 bit network address is used to route packets within the network

ZigBee

- **Approximately 400 companies are in the ZigBee alliance**
- **More than 2500 devices available which support the ZigBee protocol**
  - **E.g. Philips hue**
  - **Sonoff**
  - **Ikea**
  - **….**

ZigBee and Bluetooth comparison

Features	BLE	ZigBee
Power	Very low	low
Complexity	Complex	Simple
Nodes/master	It depends (no limit) however smartphone up to 10	64000
Latency	10 …30 msec	30msec … 1sec
Range	…60m	70m … 300m
Extendibility	No	Yes
Datarate	1 mbps	250kbps
Security	64bit, 128bit	128 bit AES and app. layer

ZigBee device compatibility repository

- https://ZigBee.blakadder.com/
- **A lot of devices are available which support ZigBee**
- **There are also adapter available to bridge the ZigBee network to other networks**
- **e.g. the ZigBee2MQTT**

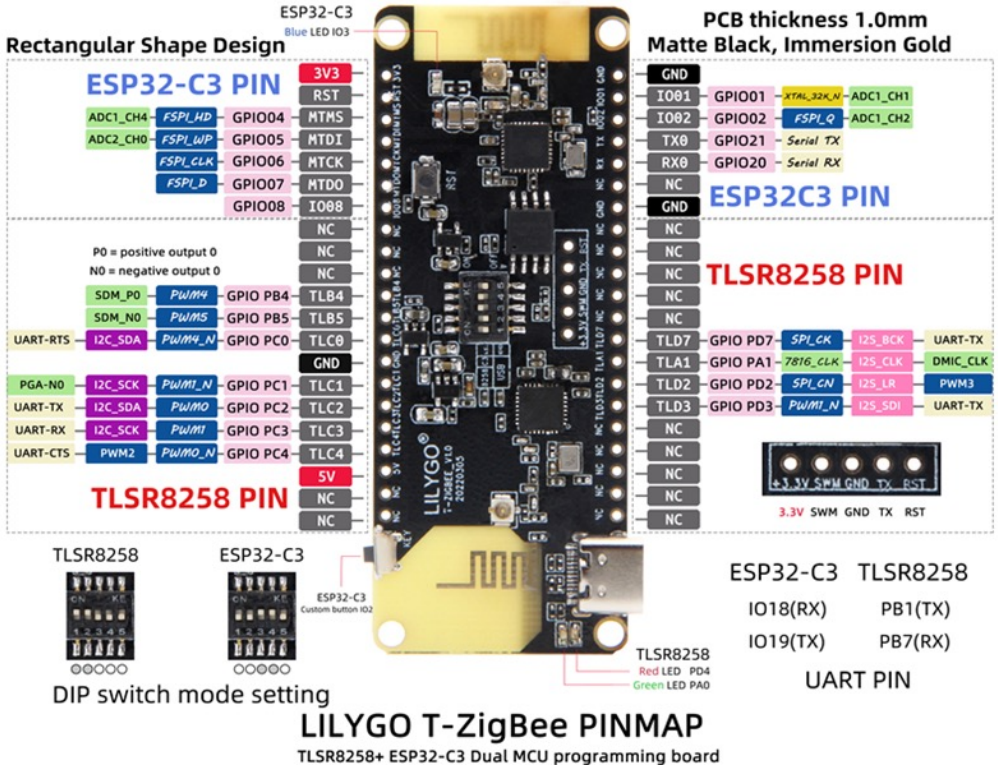
In order to use Zigbee2MQTT we need the following hardware:

1.  **A Zigbee Adapter** which is the interface between the Computer (or Server) where you run Zigbee2MQTT and the Zigbee radio communication. Zigbee2MQTT supports a variety of adapters with different kind of connections like USB, GPIO or remote via WIFI or Ethernet. Recommended adapters have a chip starting with CC2652 or CC1352. See supported Adapters.
2.  **A Server** where you would run Zigbee2MQTT. Most Raspberry-Pi models are known to work but you can run it on many computers and platforms including Linux, Windows an MacOS. It should have an MQTT broker installed. Mosquitto (Tutorial for Raspberry-Pi) is the recommended MQTT broker but others should also work fine.
3.  One or more **Zigbee Devices** which will be paired with Zigbee2MQTT.

ZigBee ESP32 development board

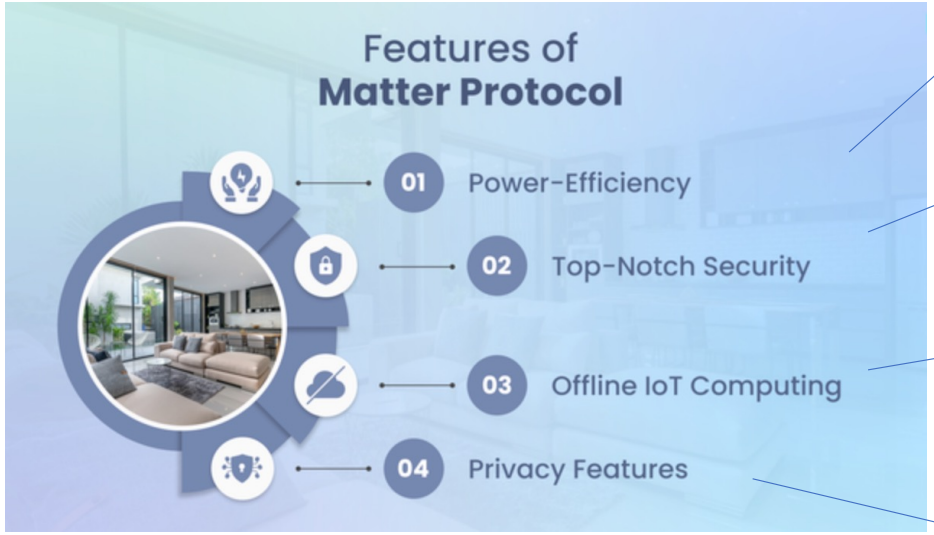
- **e.g.** LILYGO T-Zigbee Dual MCU Development Board ESP32-C3 TLSR8258 Ultra Low Power IOT WiFi BLE Smart Control Module (T-Zigbee T- U2T Kit)

- TLSR8258
  - Low power consumption: □
    - Whole Chip RX mode: 5.3mA
    - Whole Chip TX mode: 4.8mA @ 0dBm with DCDC
    - Deep sleep with external wakeup (without SRAM retention): 0.4uA



A view words about the Matter protocol

- **Launched November 2022**
- **Matter is an open-source standard**
- **More than 170 companies will make their devices ‘matter friendly’**



Matter uses the thread protocol and thread devices are low power devices that uses a minimum power to transmit data.

The Matter protocol takes a layered approach to ensure security.

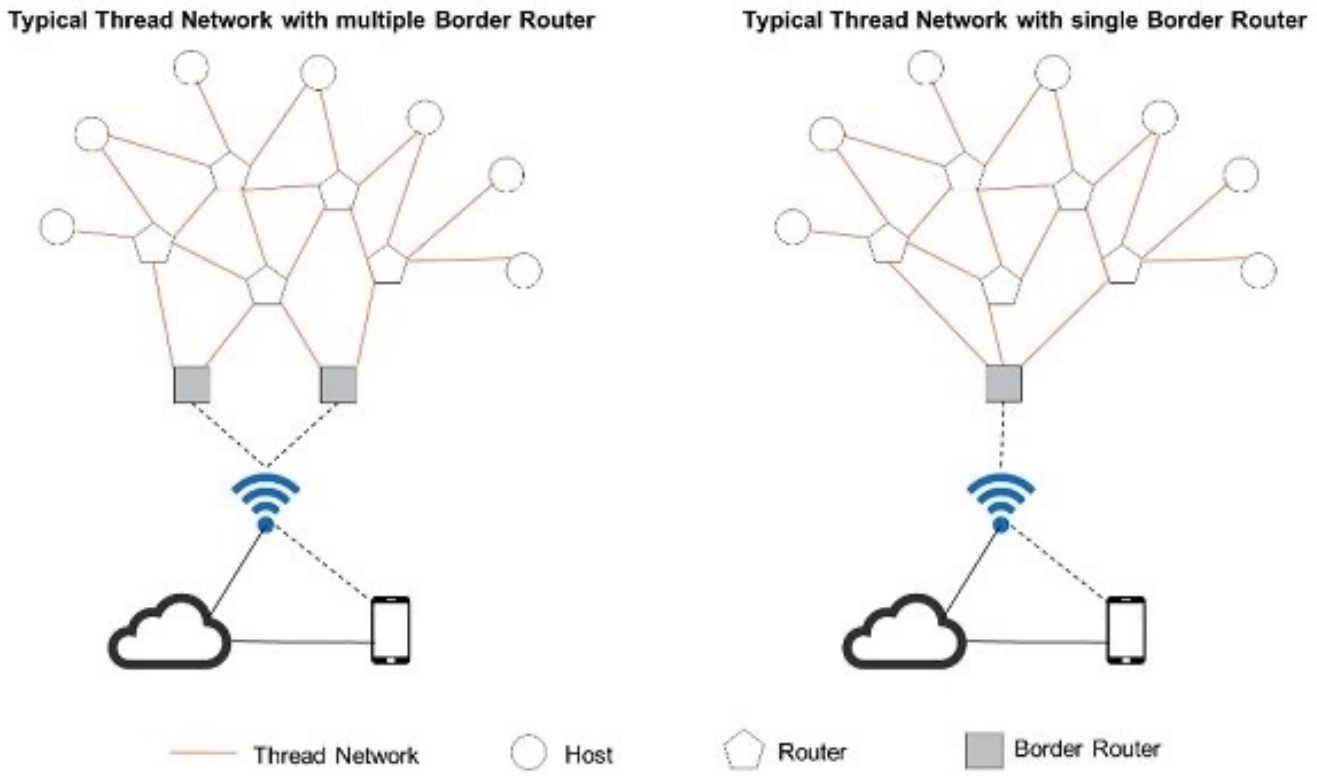
With the ability to communicate with each other directly, devices do not need access to the internet. This feature of Matter makes it more secure, especially for devices like security cameras and smart locks.

All the devices are required to provide proof of identiy. It can be an attestation keypair coupled with an X.509 certification provided by a trusted certificate authority.

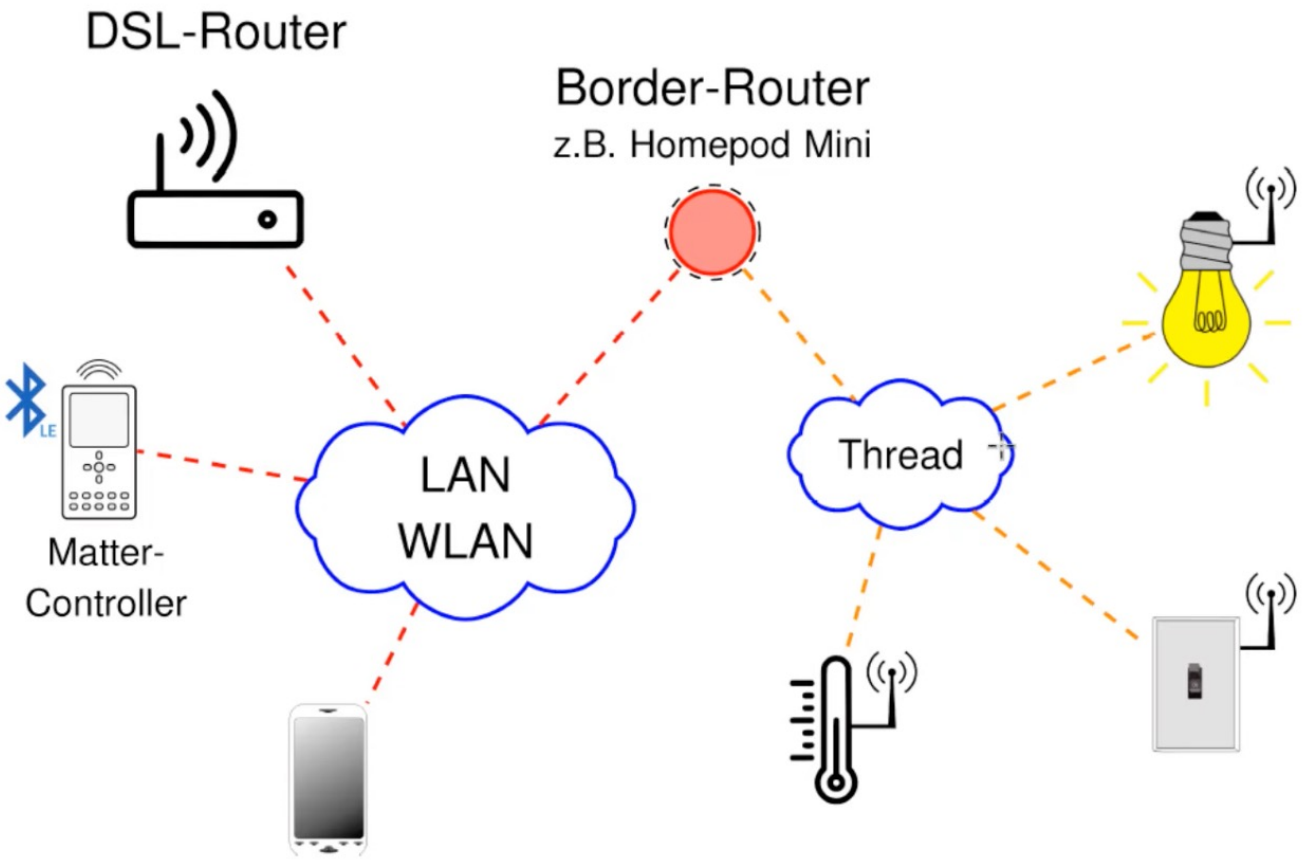
Matter and Thread

- **Matter Protocol:**
  - **Interoperability**: Matter is an application layer protocol that focuses on device communication and interoperability across different ecosystems (like Apple HomeKit, Google Assistant, Amazon Alexa).
  - **Standardization**: It standardizes how devices communicate, ensuring they can work together seamlessly regardless of the manufacturer.
  - **Device Types and Services**: Matter defines specific types of devices (like lights, locks, sensors) and services, allowing developers to implement them consistently across platforms.
- **Thread Protocol:**
  - **Networking Technology**: Thread is a low-power, wireless mesh networking protocol specifically designed for connecting low-power devices in IoT applications.
  - **IPv6 Support**: Thread is built on IPv6, allowing devices to communicate directly over the Internet without the need for a central hub.
  - **Reliability and Scalability**: Its mesh networking capability means devices can communicate with each other directly, enhancing reliability and extending the range by allowing devices to pass messages to one another.
  - **Low Power Consumption**: Thread is designed for battery-operated devices, enabling long battery life while maintaining secure and reliable communication.
- **Relationship between Matter and Thread:**
  - **Matter over Thread**: Matter can be implemented over the Thread networking protocol, allowing Matter-enabled devices to benefit from the low-power and reliable communication characteristics provided by Thread.
  - **Interoperability in a Mesh Network**: When powered by Thread, Matter devices can easily form a mesh network, allowing them to communicate efficiently with each other, even if they are not directly connected to a central hub.
  - **Enhanced Connectivity**: The combination of Matter for device interoperability and Thread for efficient networking allows for a comprehensive and future-proof smart home ecosystem.

Typical Thread Network Topology



Sample topology

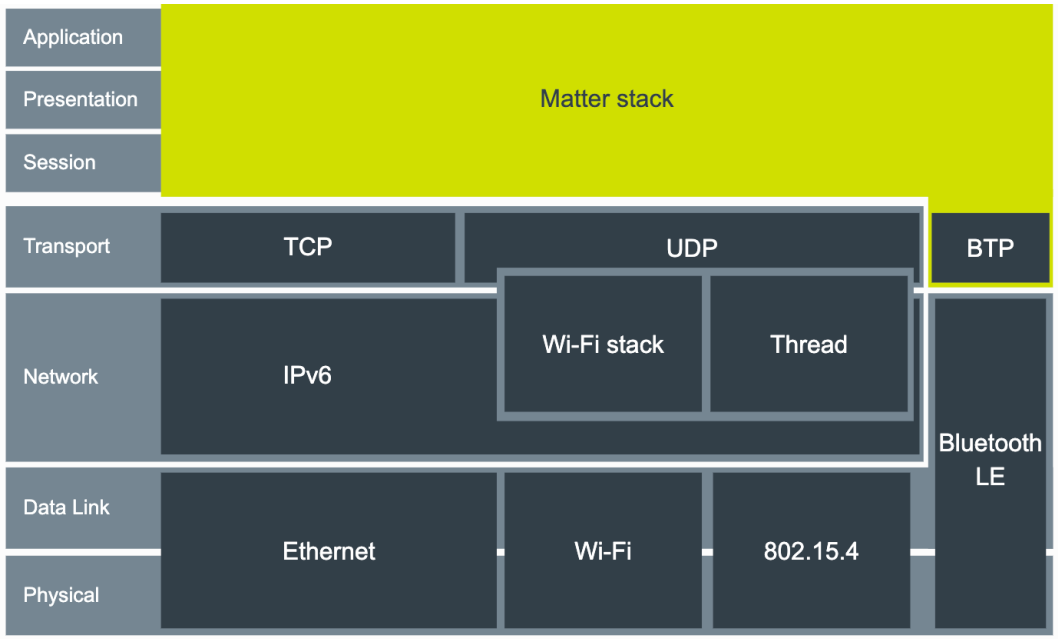


Overview: Devices compatible with Matter

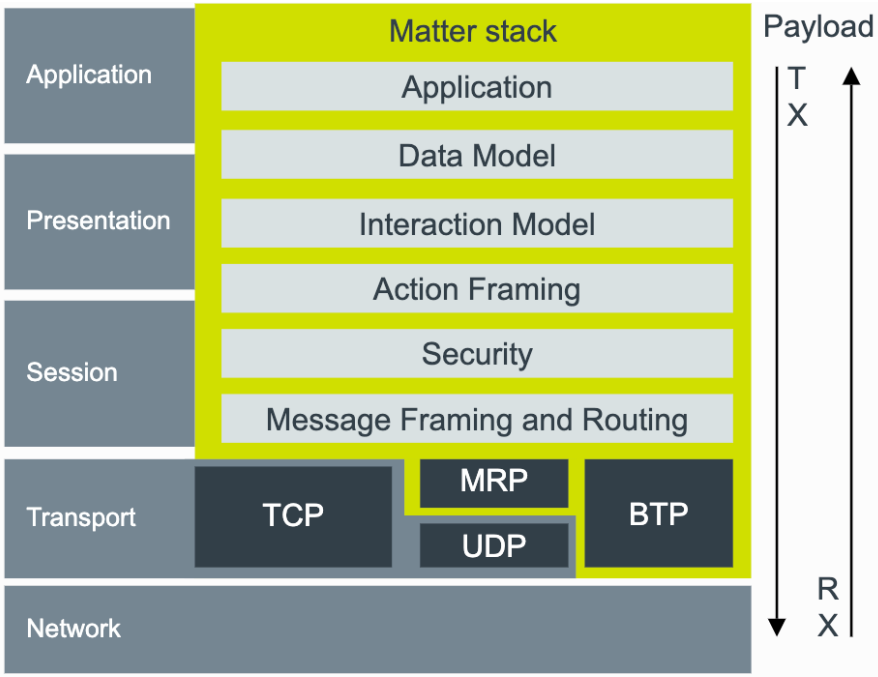
Appliances				
Samsung	Family Hub Fridge	W	C	announced
Apps + Software				
Amazon	Alexa-App (Android)		C	ready
Amazon	Alexa-App (iOS)		C	ready
Apple	Home-App[6]		C	ready
Google	Home-App (Android)		C	ready
Google	Home-App (iOS)		C	ready
Open Source	Home Assistant		C	beta
Samsung	SmartThings-App (Android)		C	ready
Samsung	SmartThings-App (iOS)		C	ready
Blinds				
Eltako	Beschattungsaktor 62-IP	W	B2B	announced
Eve Systems	Eve MotionBlinds	T	B2B	announced
Eve Systems	Eve ShutterSwitch	T	C	announced
Eve Systems	Eve MotionBlinds Upgrade Kit	T	C	announced
SmartWings	Matter Shades	T	C	ready
SwitchBot	SwitchBot Curtain (with Hub)	Br.	C	ready
Zemismart	Matter MT01 Curtain Slide	T	C	announced
Zemismart	Matter Roller Shade Motor	W	C	announced
Control Units + Buttons				
Brilliant	All-in-One Switch Panel[4]	W	B2B, C	announced
Mui Lab	Mui Board 2.0	W	B2B, C	announced
Shortcut Labs	Flic Buttons / Twist (with Hub)	W,E	C	announced
Tuo	Tuo Smart Button	T	C	ready
Decorative lights				
Eve Systems	Eve Flare[3]	T	C	announced
Nanoleaf	Skylight	W	C	announced
Nanoleaf	Matter Smart Holiday String Lights	W	C	announced
Umbra	Cono – powered by Nanoleaf	T	C	announced
Umbra	Cup – powered by Nanoleaf	T	C	announced
Door locks				

- **https://matter-smarthome.de/en/overview-products-compatible-with-matter/**

The matter stack

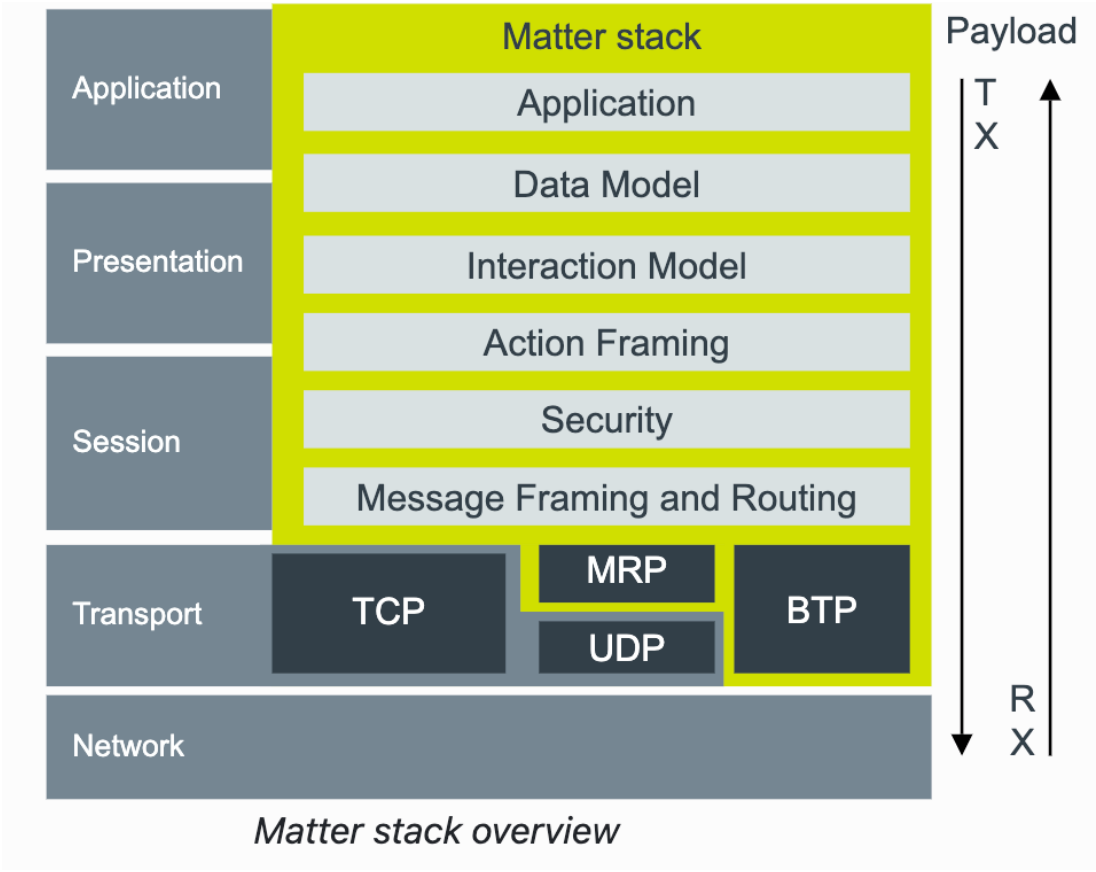


Matter architecture overview



Matter stack overview

The matter stack



Matter stack overview

https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/protocols/matter/overview/architecture.html

App: Business Logic, e.g. lamp on / off

Data Model: describes supported operations

Interaction Model: describes what interactions can be performed by a client or server (read, write, invoke, subscribe)

Action Framing: Serializes the interaction

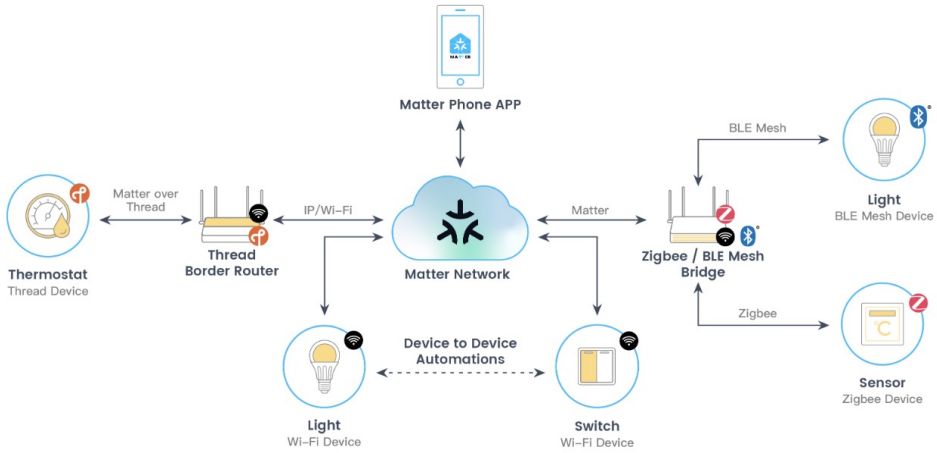
Security: encrypts the frame and append a MIC

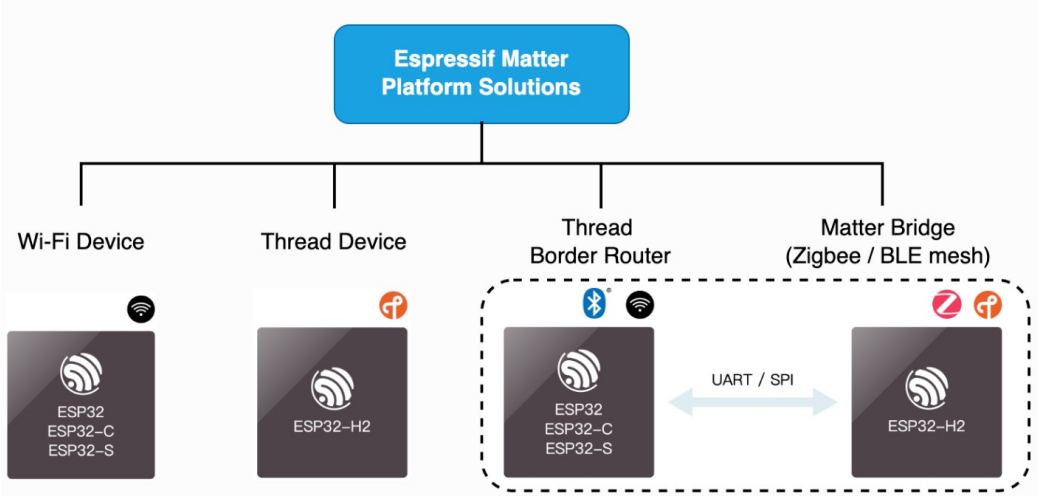
Msg Framing and Routing: generates the generic Matter frame

Msg relialabilty protocol: guaranties reliable delivery of UDP pkg

(BLE – commissioning via BLE)

Espressif Matter Solution- Espressif SDK for Matter (ESP32 Arduino Matter is also available)

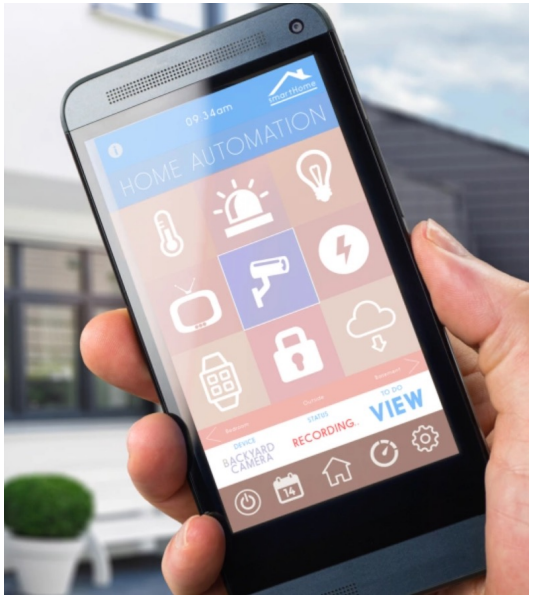


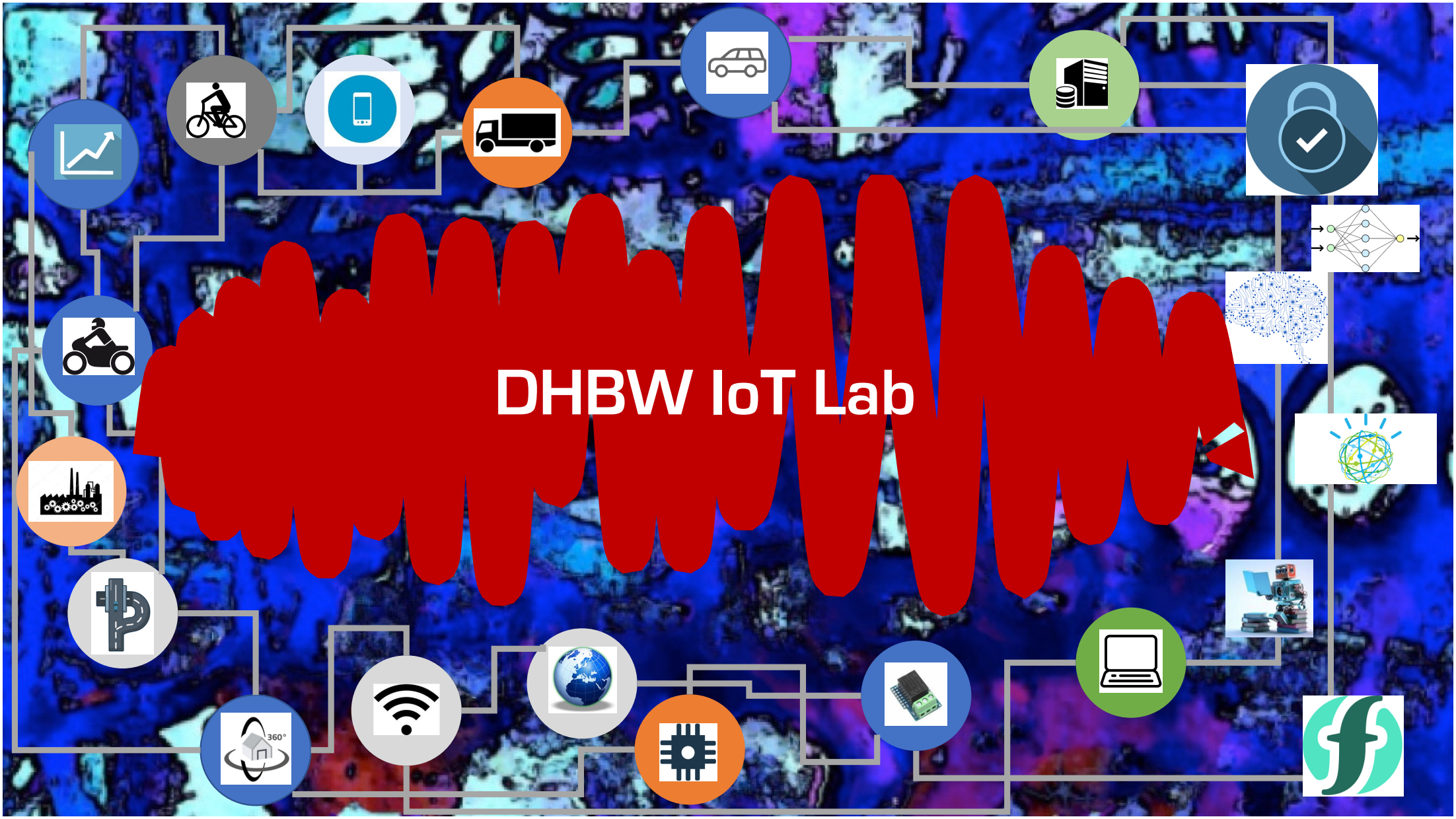


ESP32-H SoCs and modules with 802.15.4 can be used to build **Matter Thread devices**.

Something to try

Matter communication




DHBW IoT Lab

Goal of the IoT lab

- **Build a real IoT solution**
  - **Select Hardware and put it together**
  - **Develop a program which reads sensor data, do some pre-calculation, transmit data to a IoT gateway, forward it to Server app.**

- **Get familiar with IoT sensors (from a HW point of view)**
- **Practice some microcontroller programming (which is slightly different from other IT programming)**
- **Get familiar with IoT connectivity topics**
- **Learn how to use different communication protocols (BLE, LoRaWAN, WiFi, MQTT)**
- **Analyse the collected data using analytics functions**

Block diagram IoT Lab

Accelerator sensor

Gyroscope sensor

Moisture sensor

Ultrasonic sensor

Temperature sensor

Micro-controller

LoRaWAN

Connectivity Hub LoraWan e.-g. TTN

WiFi

WiFi

BLE

Connectivity Hub e.g. smartphone

WiFi

LTE

IoT server

Analytics on sensor data

MQTT will also be used