

Developing Matter 1.0 products with nRF Connect SDK

Nordic Tech Webinar

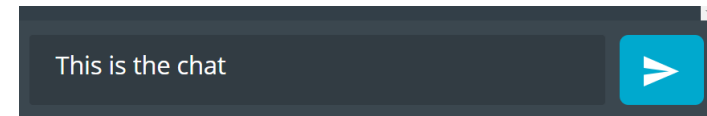
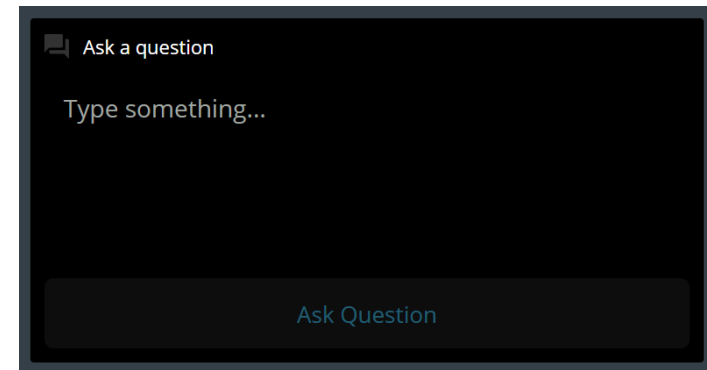
Kamil Kasperczyk / Firmware Engineer

Marcin Kajor / Firmware Engineer

November 2022

Practicalities

- Duration: about 60 minutes
- Questions are encouraged!
 - Please type questions in the top of the right sidebar
 - All questions are anonymous
 - Try to keep them relevant to the topic
 - We will answer towards the end
- The chat is not anonymous, and do not use for questions
- Go to DevZone if you have more questions
- A recording of the webinar will be available together with the presentation at webinars.nordicsemi.com



Today's Host

Finn Boetius



Product Marketing Engineer

Today's Speakers

Kamil Kasperczyk



Firmware Engineer

R&D

Marcin Kajor



Firmware Engineer

R&D

Previous Nordic Matter webinars

- Introduction to Matter – Sep 2021

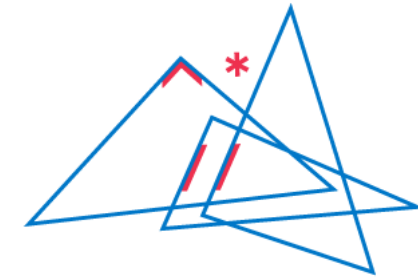
Krzysztof Loska

https://youtu.be/v_285vCHifw

- Developing Matter Products with nRF Connect SDK – Nov 2021

Łukasz Duda & Damian Królik

<https://youtu.be/kdMJQFDRoss>



NORDICTECH
WEBINARS

Missed this webinar?
Sign up: nordicsemi.com/webinars

Agenda

- **Introduction**
 - Matter 1.0 is here!
 - Support for Matter in nRF Connect SDK
 - Development tools
- **Matter over Thread and Matter binding**
 - Setting up the environment
 - Building Matter over Thread applications in nRF Connect SDK
 - Controlling accessories from PC
 - Controlling Matter over Thread light bulb from Matter over Thread light switch
- **Matter multi-fabric scenario**
 - Controlling accessory from Android smartphone
 - Adding accessory to the new Matter fabric
- **Creating Matter accessory**
 - Configuring data model
 - Network topology
 - Developing application logic and testing of the setup
- **Q&A session**

Introduction

Matter 1.0 is here!

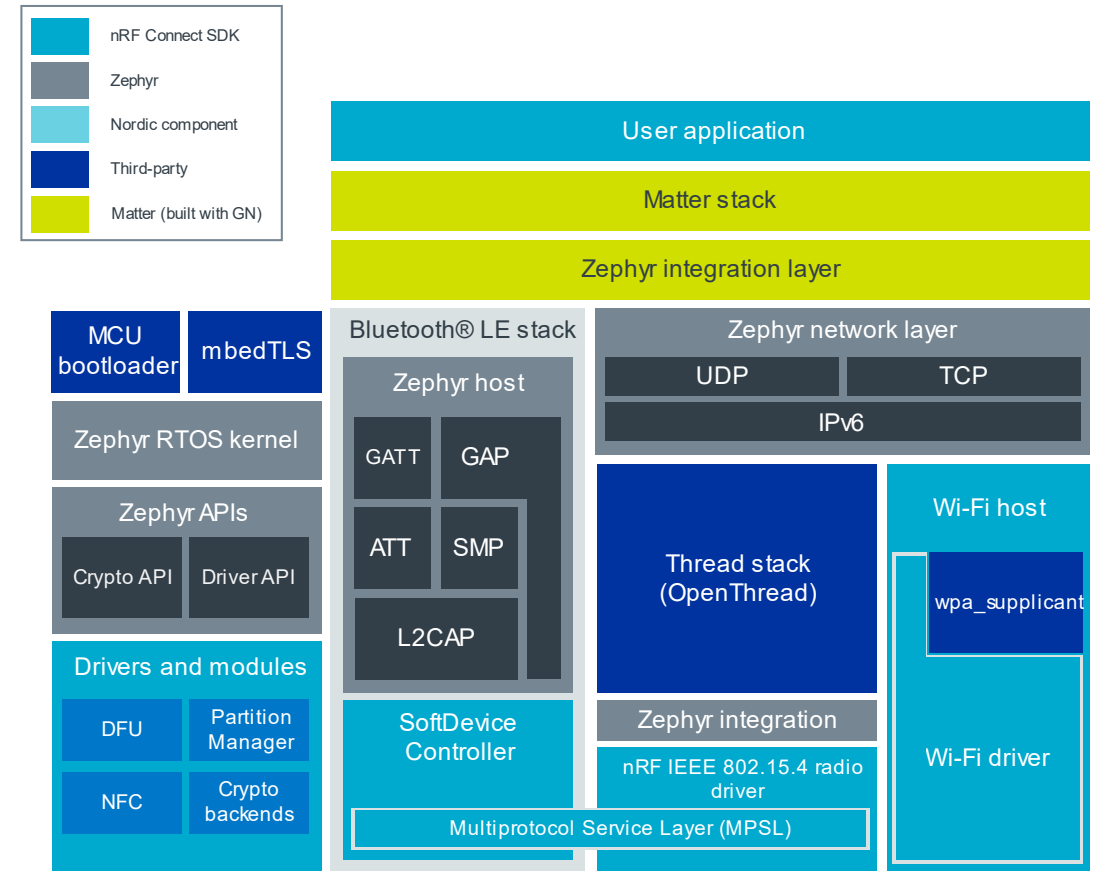
- Open source reference implementation which conforms to the Matter 1.0 specification
- <https://github.com/project-chip/connectedhomeip/releases/tag/v1.0.0>
- <https://csa-iot.org/all-solutions/matter/>



A screenshot of the GitHub release page for the V1.0.0 release of the project-chip/connectedhomeip repository. The page shows the release title "V1.0.0 Release" with a "Latest" badge and a "Compare" dropdown. Below the title, it indicates that user andy31415 released this version 27 days ago, with 19 commits to the v1.0-branch since this release. The release description states "V1.0.0 - the first release on the v1.0-branch." Under the "Assets" section, there are two source code files: "Source code (zip)" and "Source code (tar.gz)", both uploaded 27 days ago. At the bottom of the release card, there are reaction icons for thumbs up (33), thumbs down (4), clap (96), heart (19), and rocket (30), along with a note that 134 people reacted.

Support for Matter in nRF Connect SDK

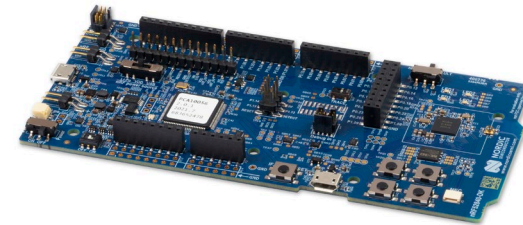
- Matter is deployed in nRF Connect SDK as a one of the submodule repositories (dedicated Matter fork)
- Powered by Zephyr OS
- Certified support for Thread 1.3 and Bluetooth Low Energy stacks
- Matter sample apps included
- Release 1.0.0 already supported in nRF Connect SDK v.2.1.1!
- Nordic Weather Station has successfully passed the Matter Specification Validation Event!
- https://developer.nordicsemi.com/nRF_Connect_SDK/doc/2.1.1/nrf/ug_matter.html



Development tools

- Hardware Development Kits (based on nRF52, nRF53 and nRF70 families)
- nRF Connect for VS Studio
- J-Link GDB debugging tool
- Nordic Power Profiler Kit II
- nRF Thread Topology Monitor
- nRF Sniffer for 802.15.4 and Bluetooth LE
- Pre-compiled Matter controllers for Linux and Android
- Matter specific tools:

https://developer.nordicsemi.com/nRF_Connect_SDK/doc/2.1.1/nrf/ug_matter_gs_tools.html#id11



Matter in nRF Connect SDK

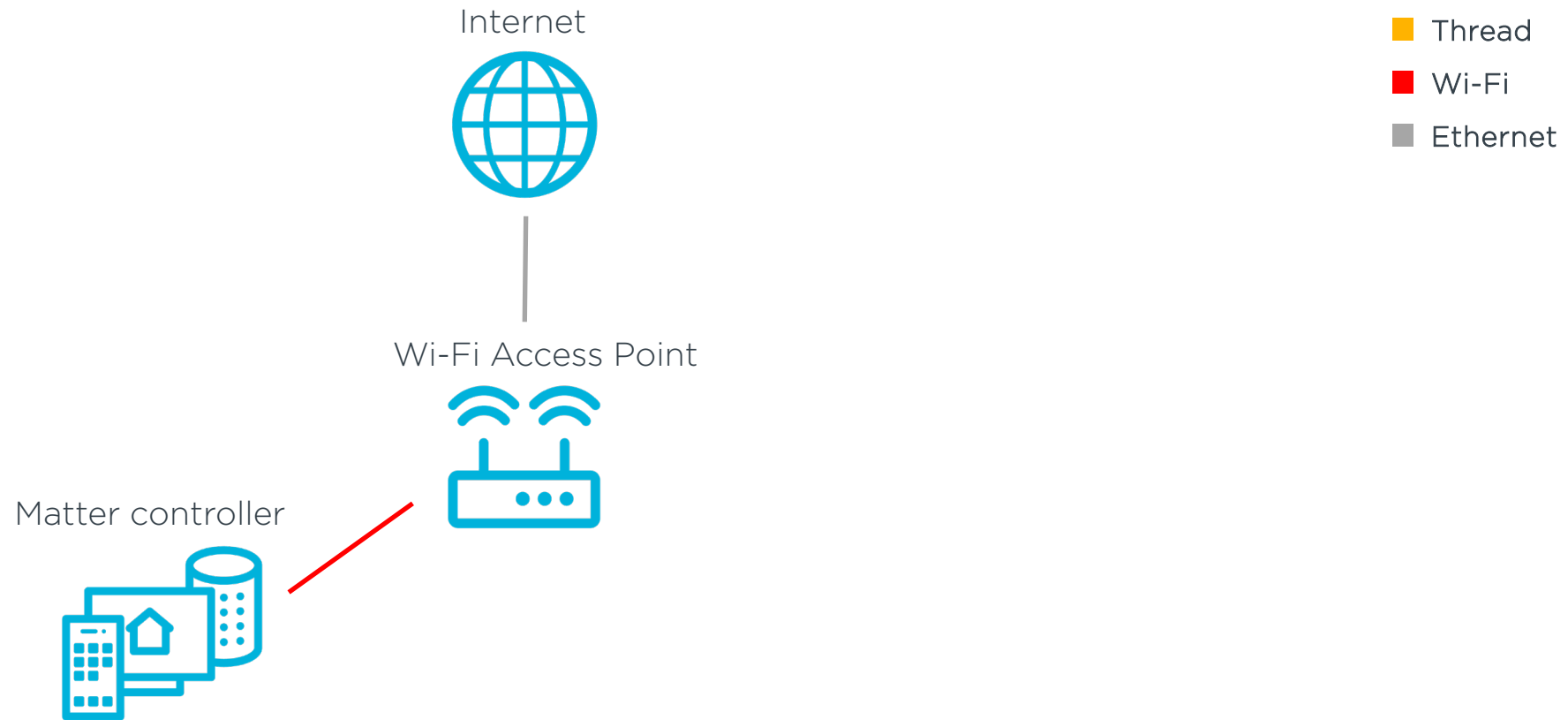
Matter over Thread and Wi-Fi

Matter network topology

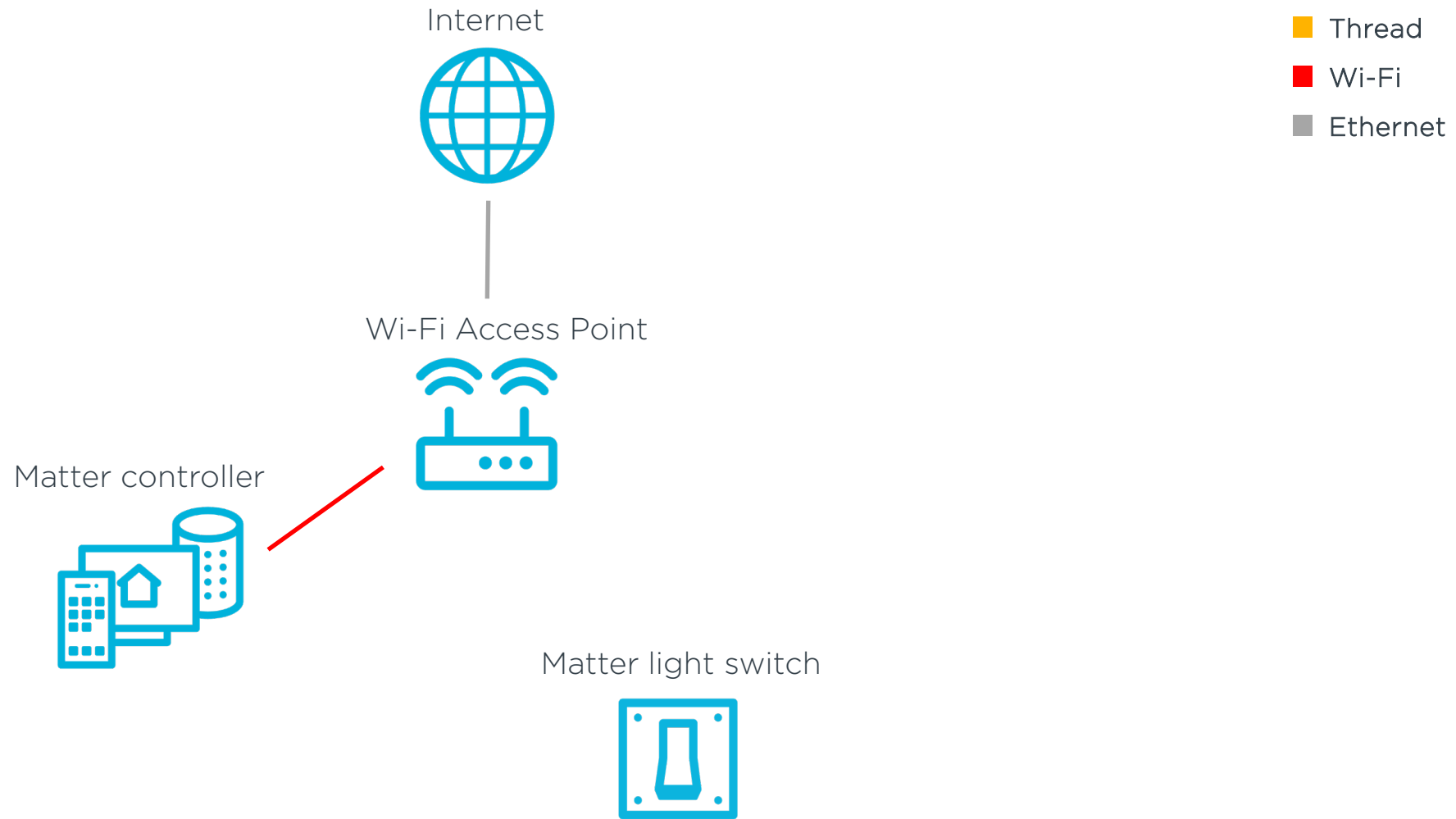


- Thread
- Wi-Fi
- Ethernet

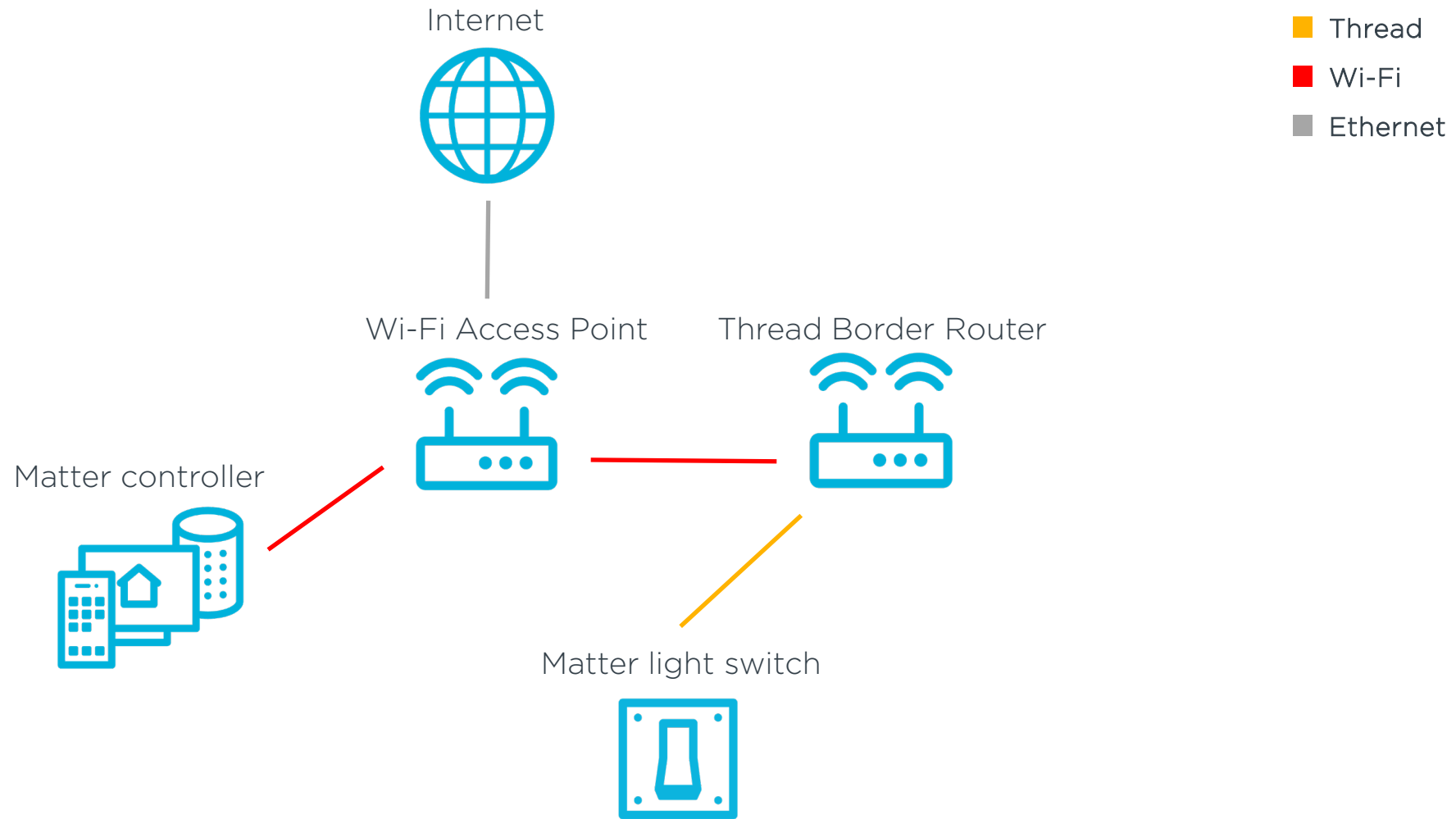
Matter network topology



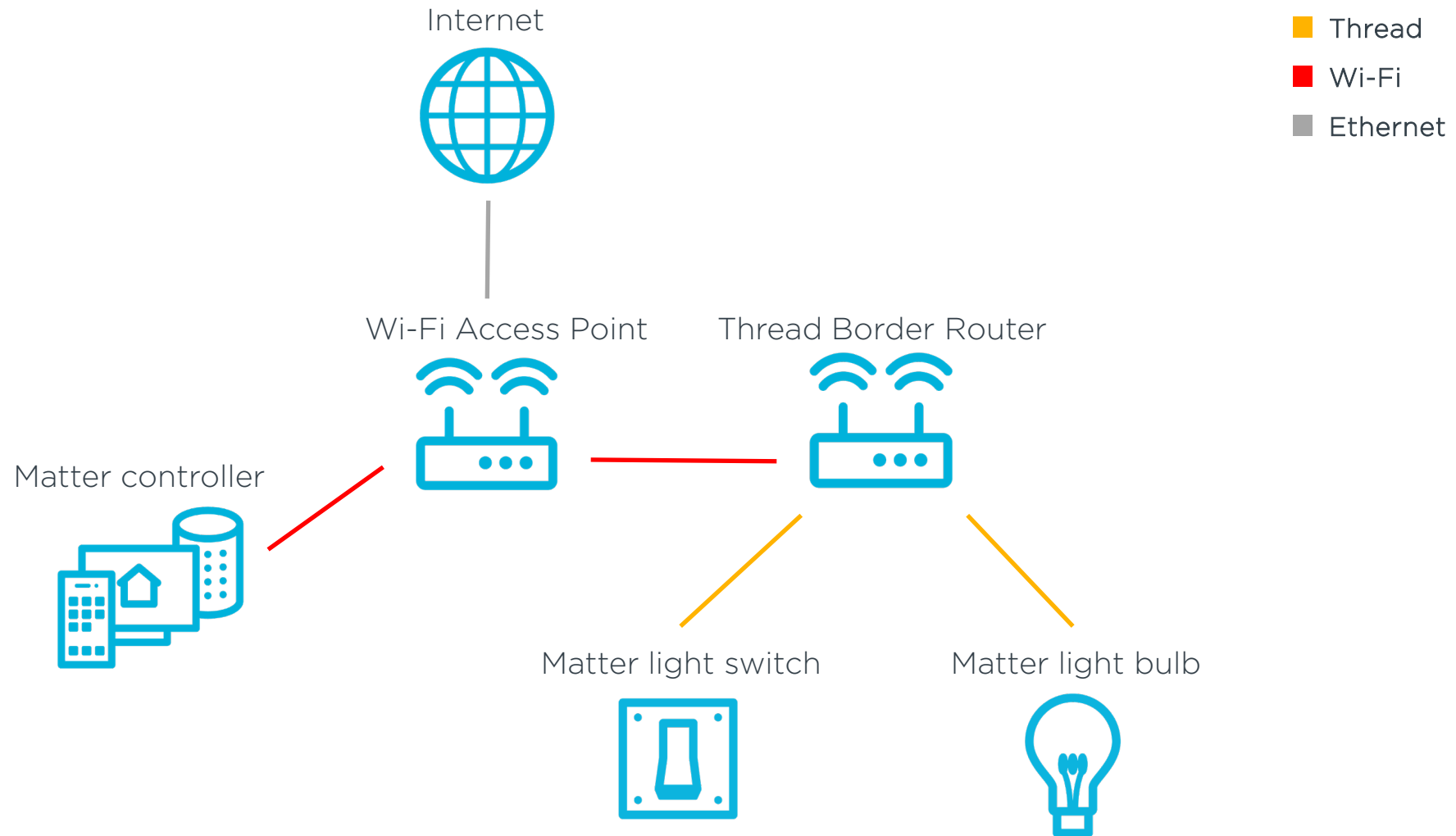
Matter network topology



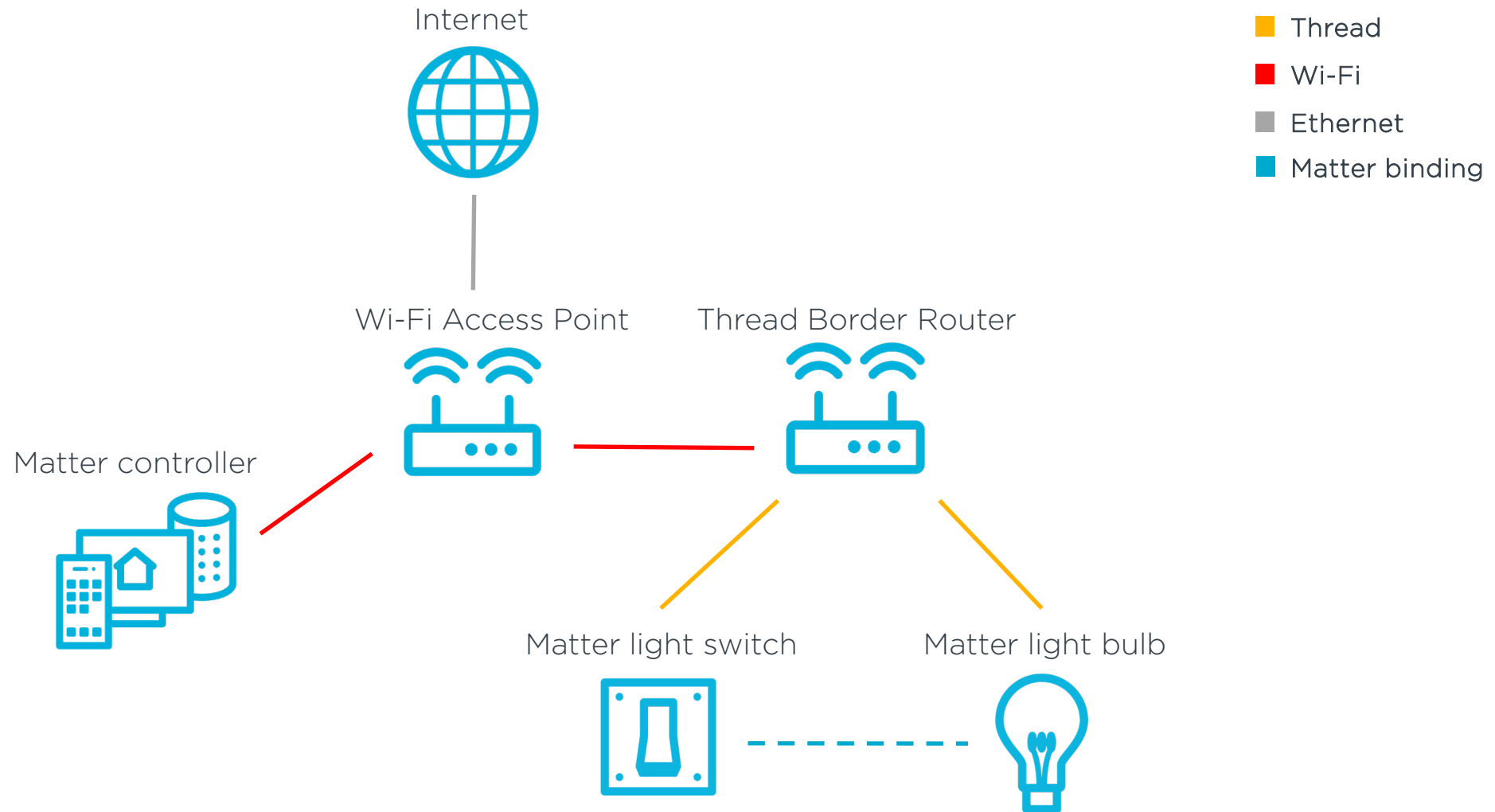
Matter network topology



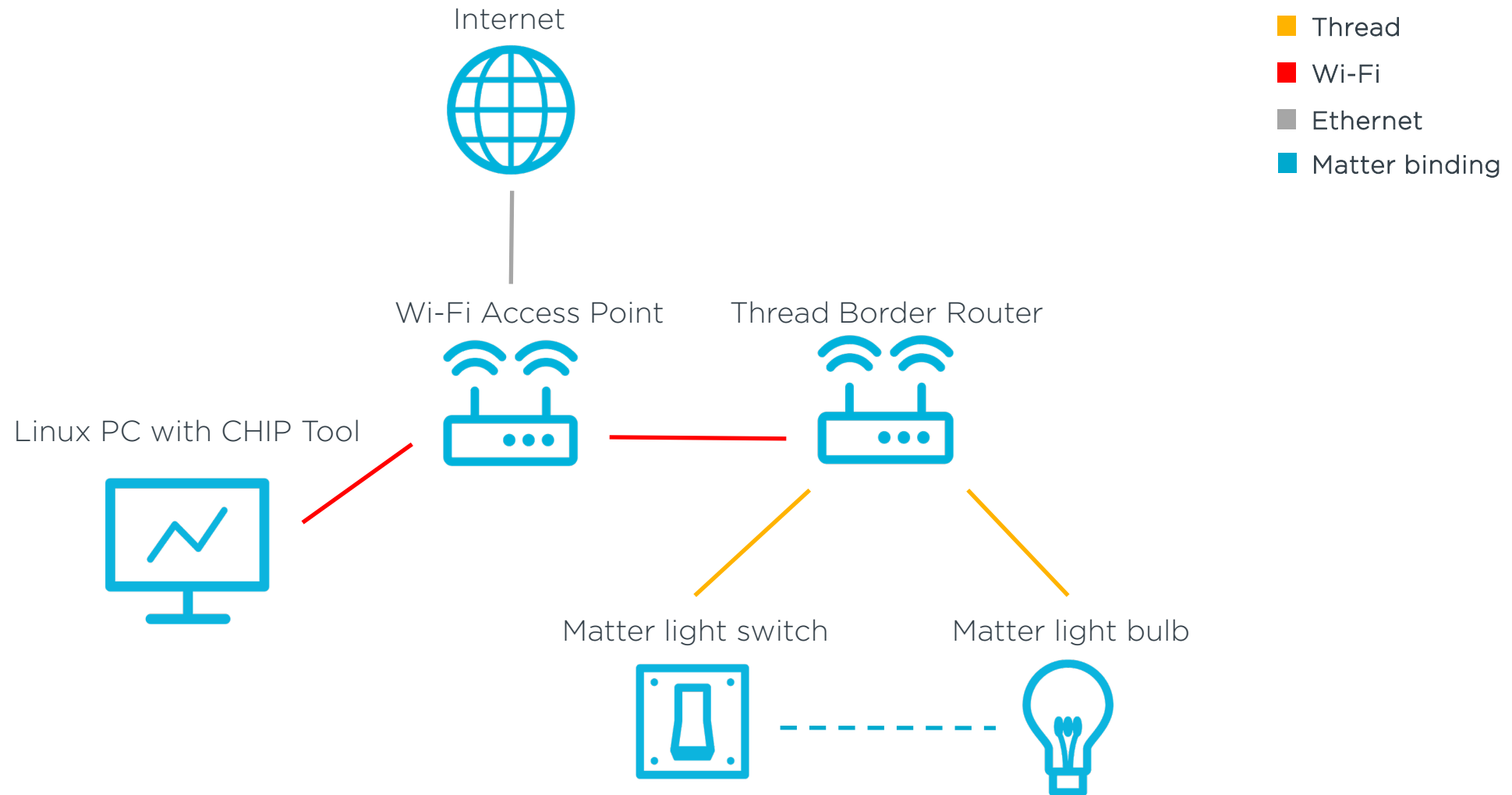
Matter network topology



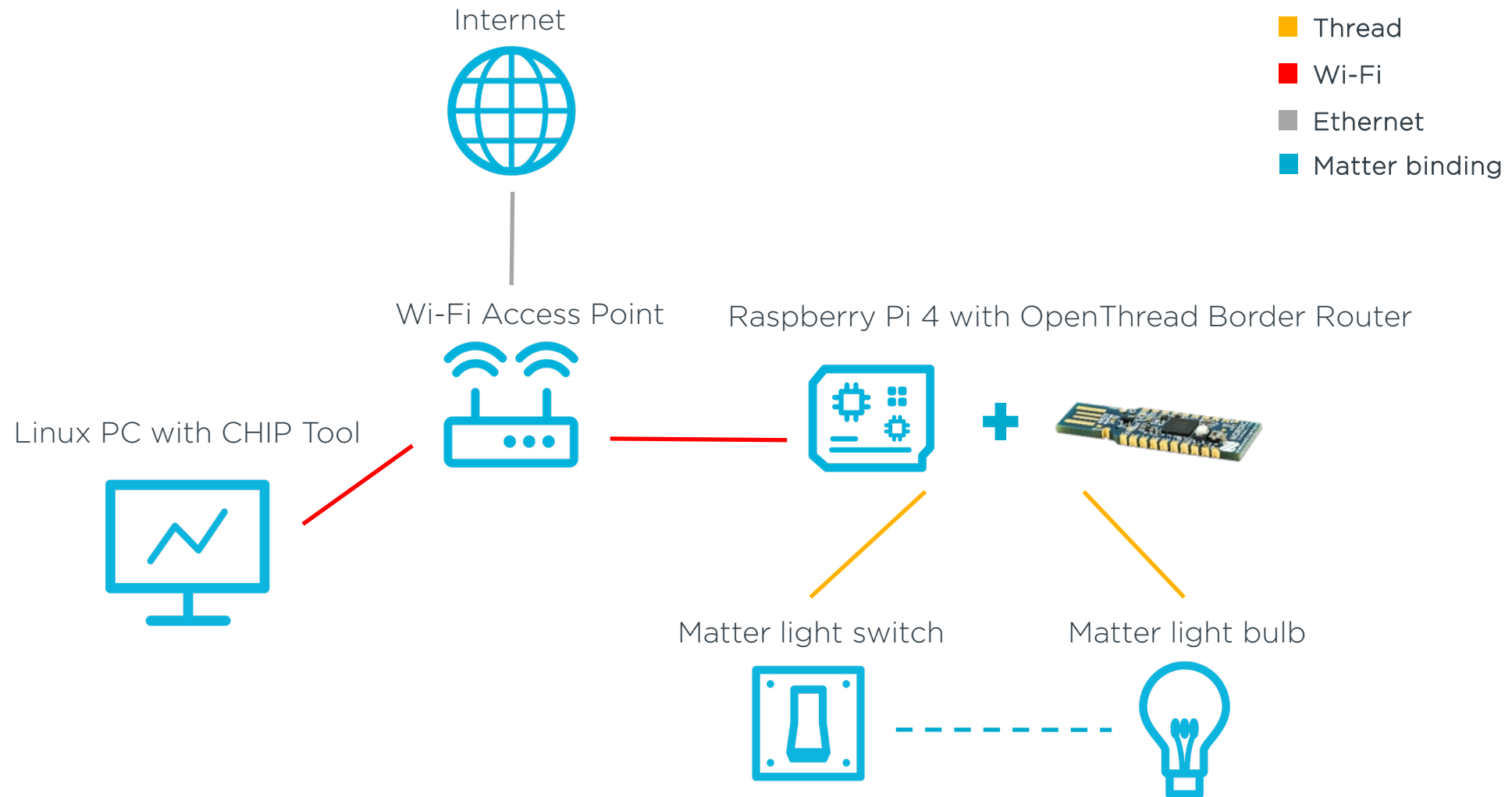
Matter network topology



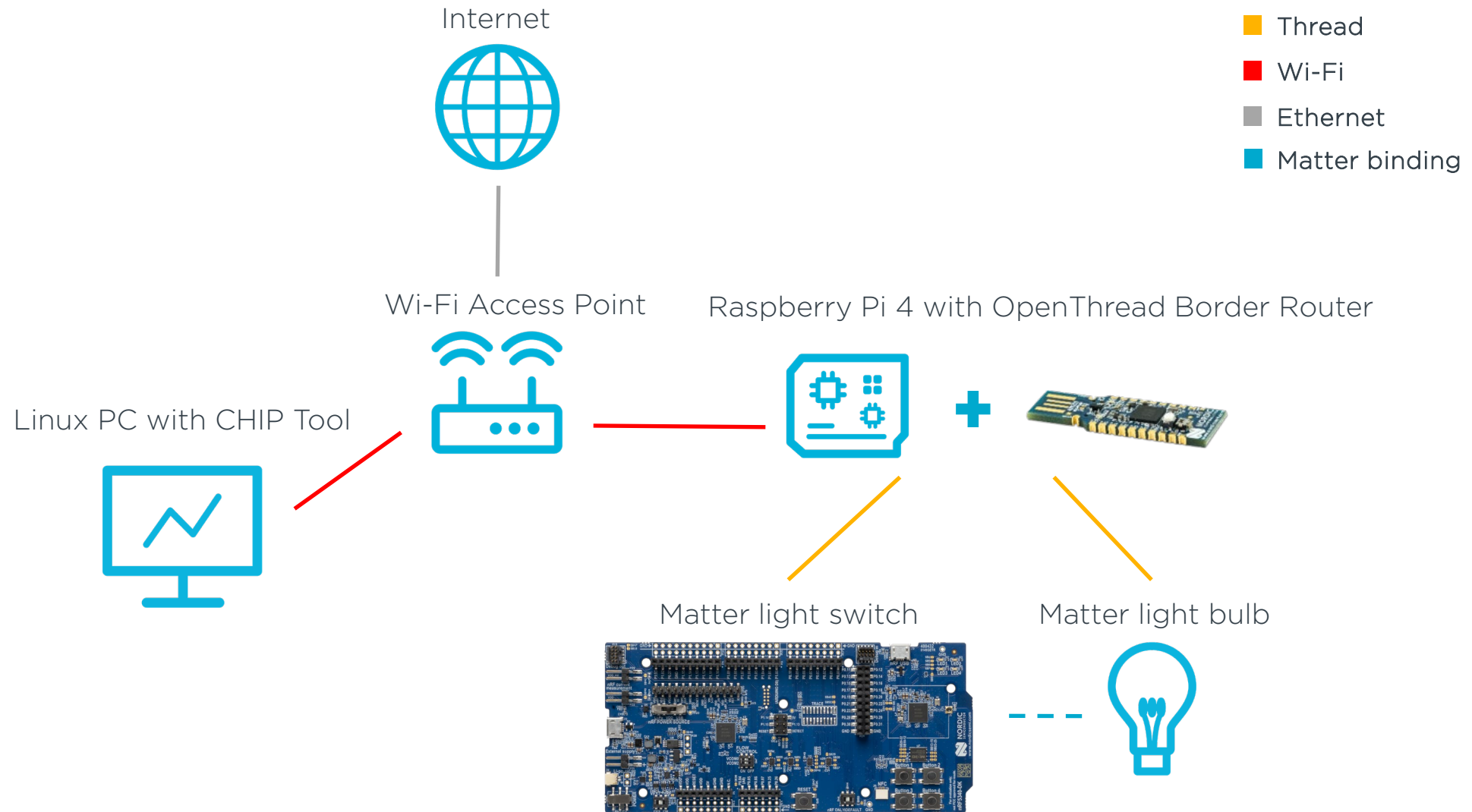
Matter network topology



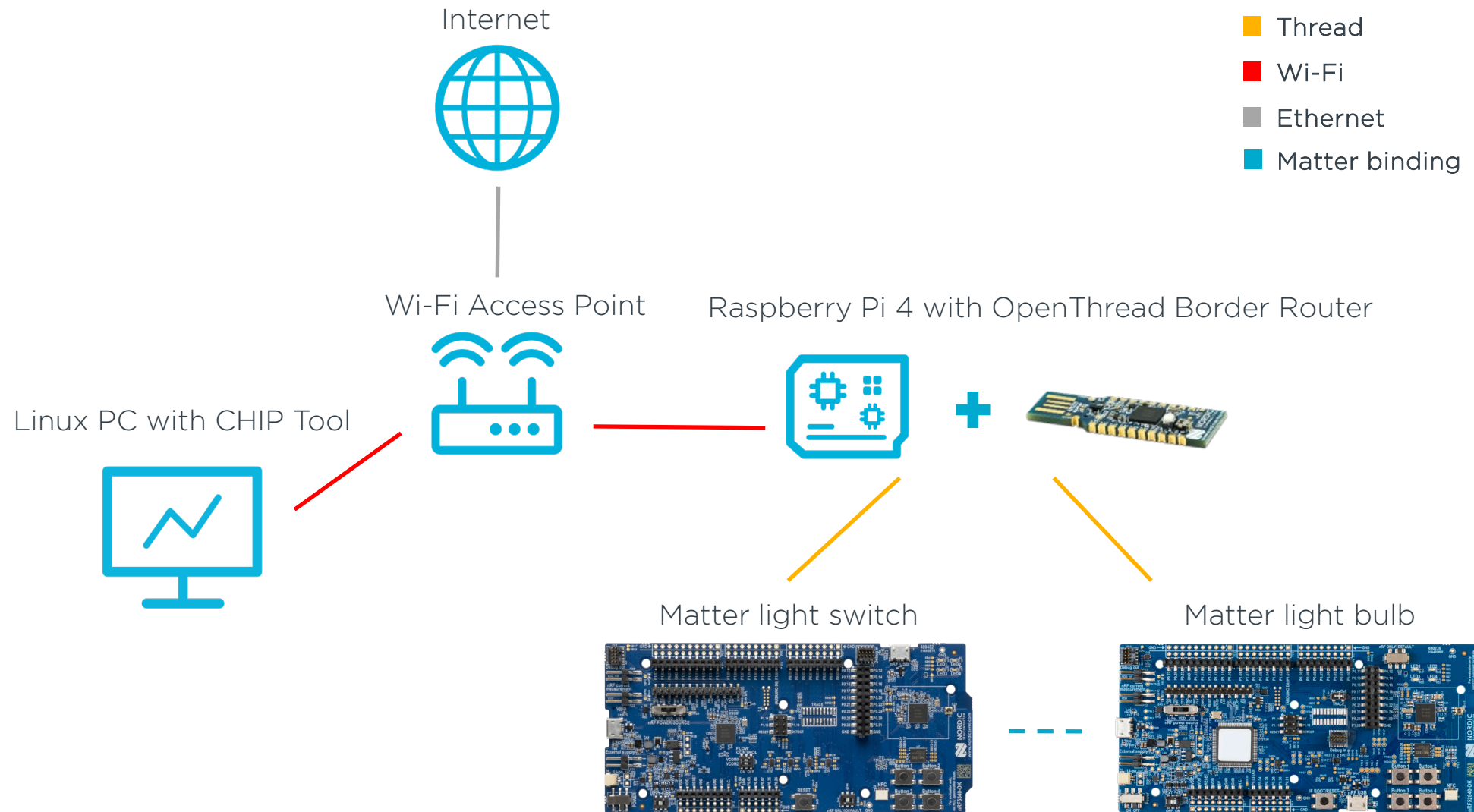
Matter network topology



Matter network topology



Matter network topology



Prerequisites

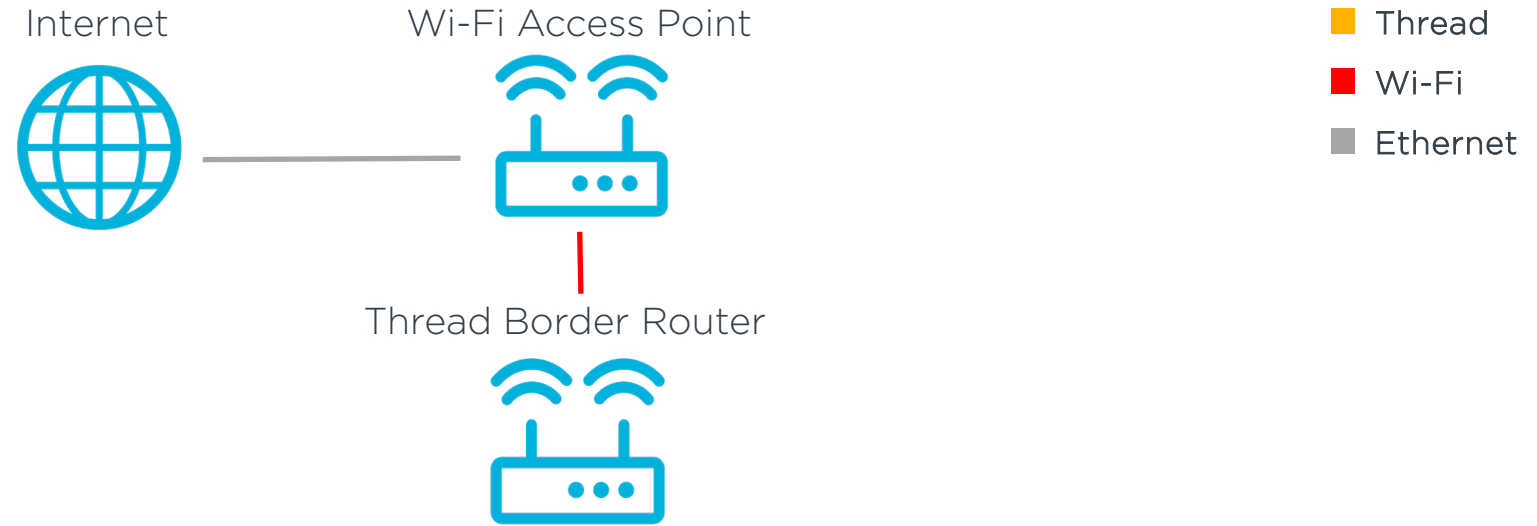
- Linux PC with software installed:
 - nRF Connect SDK v2.1.1
https://developer.nordicsemi.com/nRF_Connect_SDK/doc/2.1.1/nrf/getting_started.html
 - nRF Command-line tools
<https://www.nordicsemi.com/Products/Development-tools/nrf-command-line-tools/download>
 - Visual Studio Code with nRF Connect Extension Pack for VS Code
<https://nrfconnect.github.io/vscode-nrf-connect/>
- Raspberry Pi 4 running OpenThread Border Router
https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/ug_thread_tools.html#installing-otbr-manually-raspberry-pi



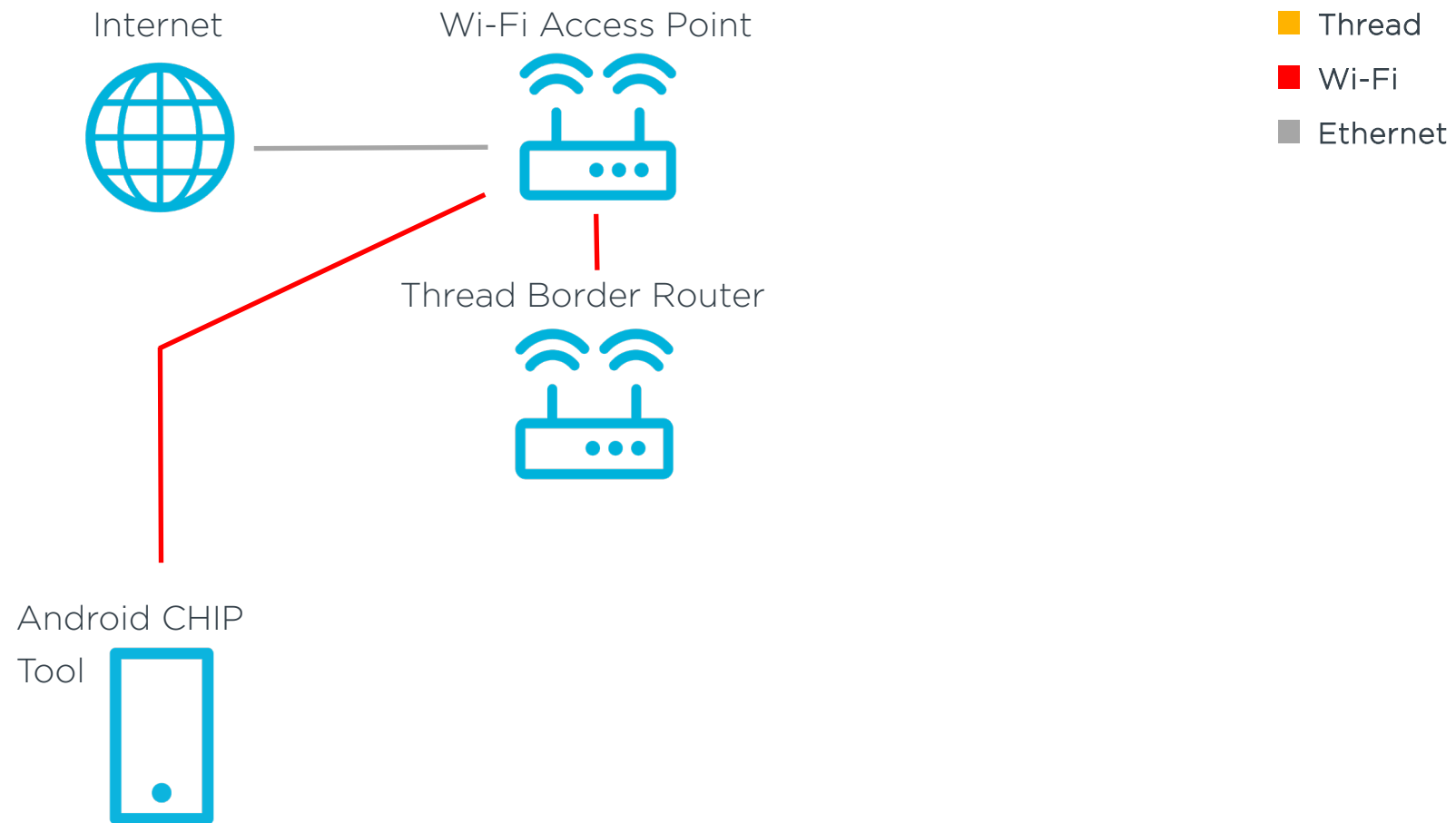
Matter in nRF Connect SDK

Matter multi-fabric scenario

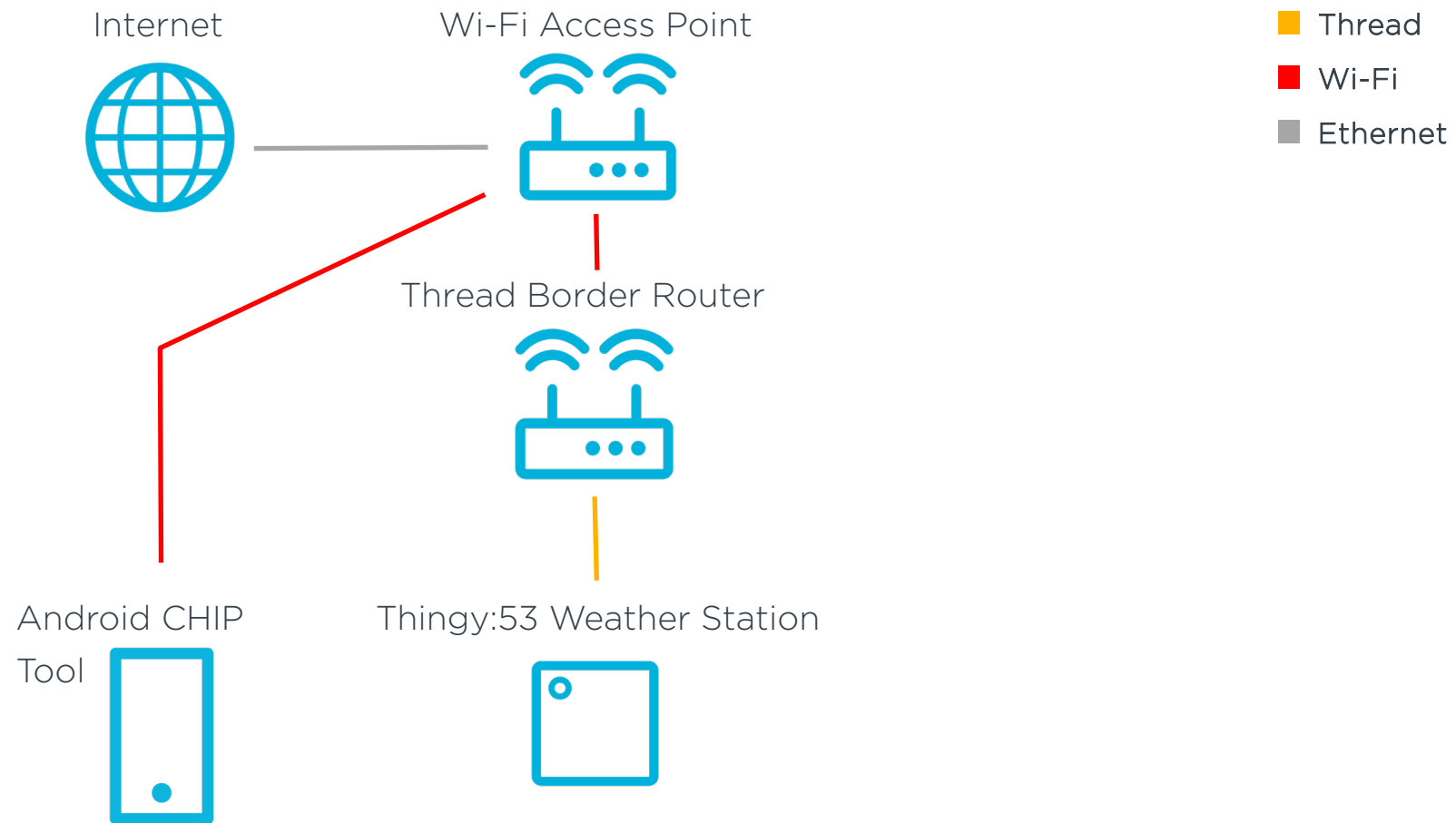
Matter multi-fabric topology



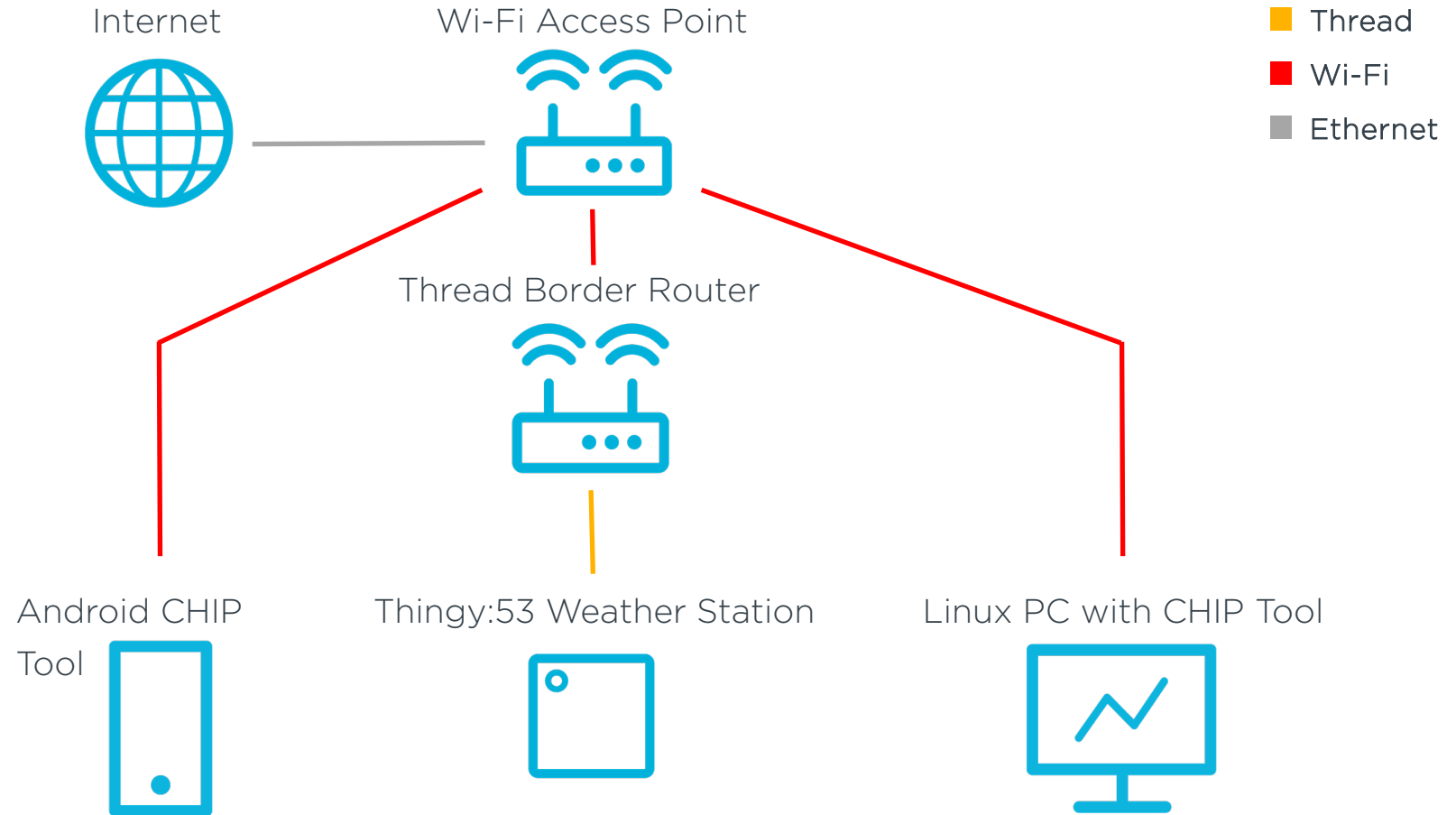
Matter multi-fabric topology



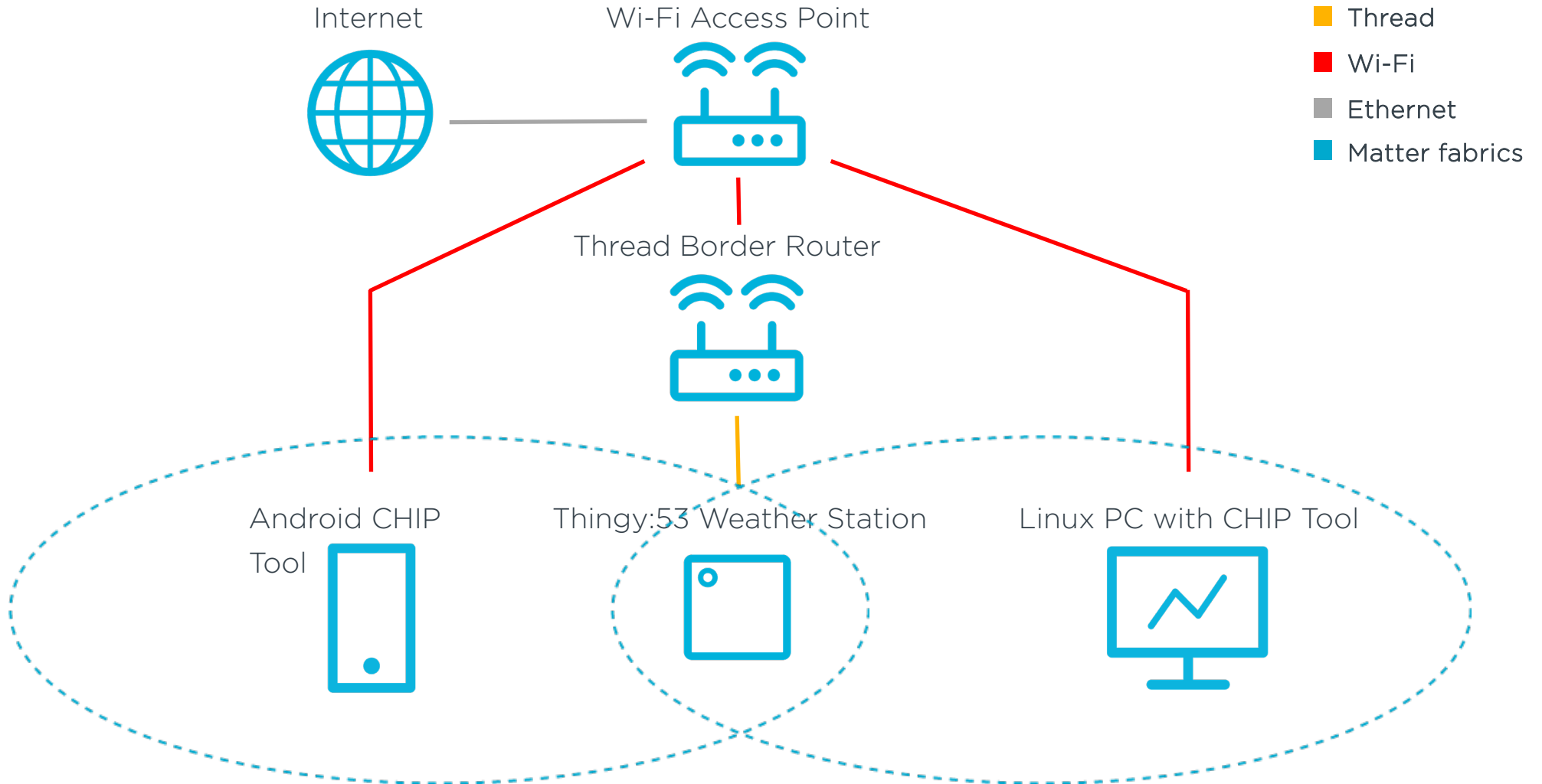
Matter multi-fabric topology



Matter multi-fabric topology



Matter multi-fabric topology

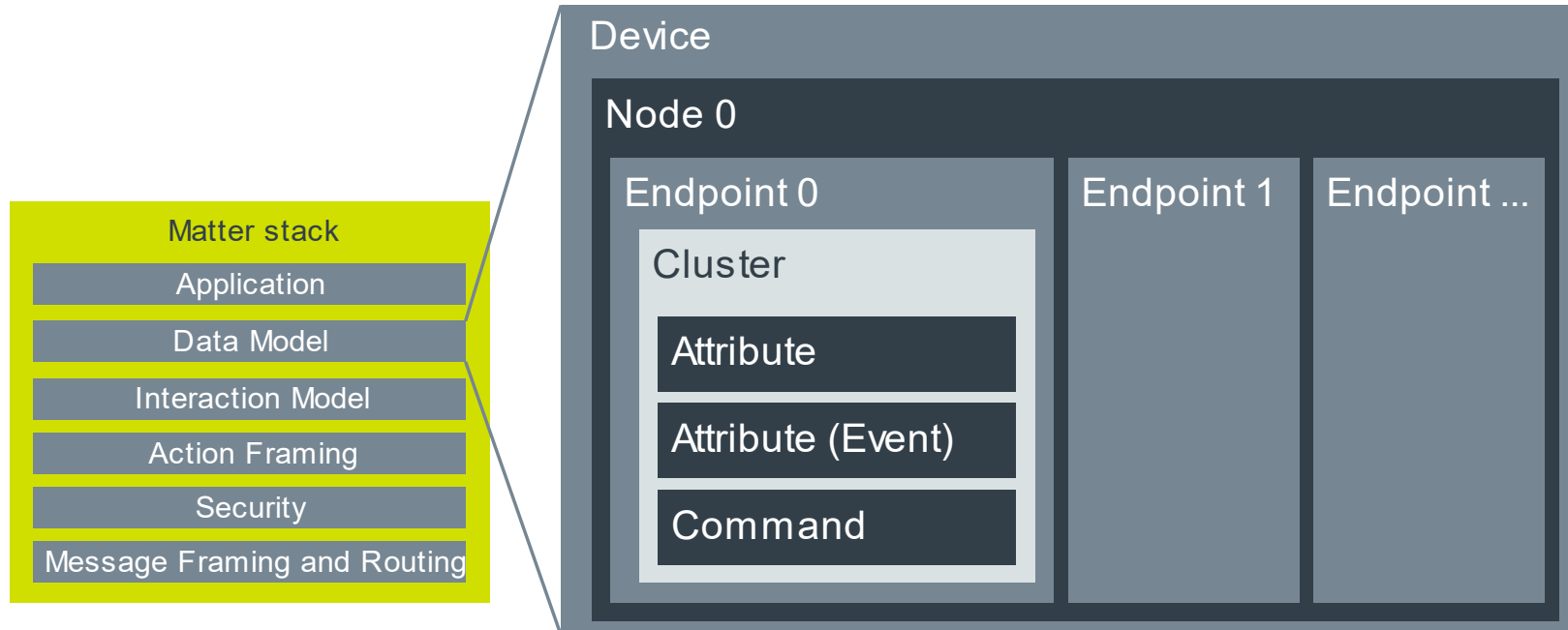


Matter in nRF Connect SDK

Creating Matter accessory

Configuring data model

- Understanding Matter data and interaction model



Configuring data model

- Using ZAP Tool to configure specific clusters and attributes:

<https://github.com/project-chip/zap>

The screenshot shows the ZCL Cluster Configurator interface. The main content area displays the 'ZCL Advanced Platform' page, which includes a progress bar for various tests (all passing) and a 'What is ZAP?' section. The 'What is ZAP?' section explains that ZAP is a generic generation engine and user interface for applications and libraries based on the Zigbee Cluster Library specification. It lists three main functions: performing SDK-specific customized generation of all global artifacts, providing a UI for selecting specific application configuration, and performing SDK-specific customized generation of all user selected configuration artifacts. Below this, there is a 'Quick instructions' section with a brief overview of how to run the application and the commands to install dependencies and run the tool.

ZCL Advanced Platform

Generation and back-end tests: passing Cypress UI tests: passing SonarCloud: passing Build & Release: passing

What is ZAP?

ZAP is a generic generation engine and user interface for applications and libraries based on [Zigbee Cluster Library](#), the specification developed by the [Zigbee Alliance](#).

ZAP allows you to perform the following:

- perform SDK-specific customized generation of all global artifacts (constants, types, IDs, etc) based on the ZCL specification
- provide UI for the end-user to select specific application configuration (clusters, attributes, commands, etc.)
- perform SDK-specific customized generation of all user selected configuration artifacts (application configuration, endpoint configuration, etc) based on ZCL specification and customer-provided application configuration.

ZAP is a generic templating engine. Examples are provided for how to generate artifacts for the C language environment, but one could easily add new templates for other language environments, such as C++, java, node.js, python or any other.

Quick instructions

This is a node.js application. In order to run it, you need to have [npm](#) installed. The best way is to simply download latest install of [node](#) and you will get npm. If you have an older version of node installed on your workstation, it may give you trouble, particularly if it's very old. So make sure you have decently recent (v12.x or v14.x should work as of 2021) version of node available. Run `node --version` to check what version is picked up.

Once you have a desired version of node, you can run:

```
npm ci
```

which will download install all the project dependencies. It is not uncommon to run into native library compilation problems at this point. There are various `src-script/install.*` scripts for different platforms. Please refer to [FAQ](#) for additional details of which script to run on different platforms and then rerun `npm ci`.

Then run:

```
npm run zap
```

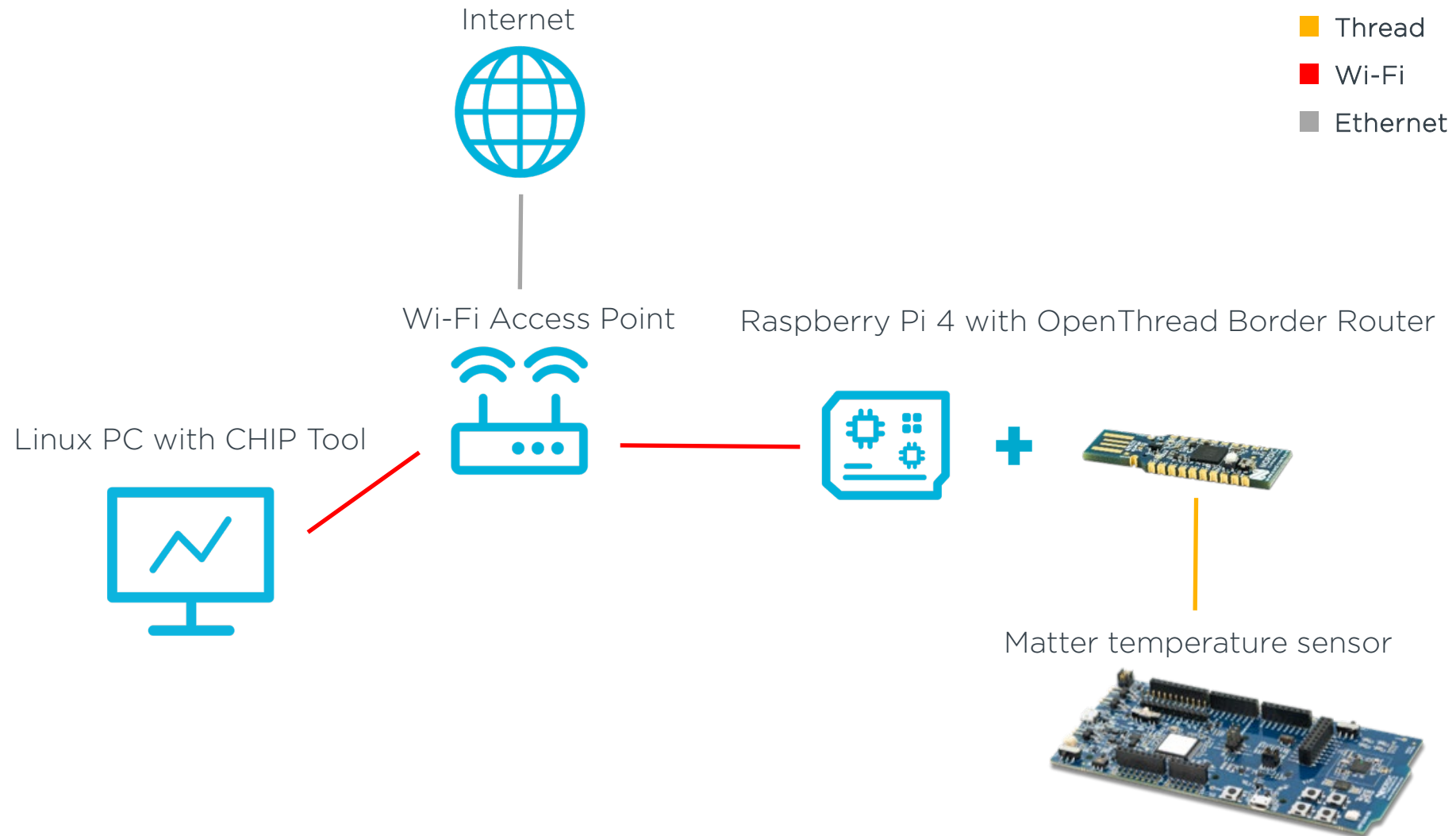
The screenshot shows the ZCL Cluster Configurator interface with the 'Endpoint 0 Clusters' configuration page. A sidebar on the left displays the device information for 'Endpoint - 0'. The main content area shows a table of clusters with columns for Cluster, Required Cluster, Cluster ID, Manufacturer Code, Enable, and Configure. The 'Enable' column contains dropdown menus, and the 'Configure' column contains gear icons. The 'Descriptor' and 'Access Control' clusters are highlighted in blue.

Endpoint 0 Clusters

Device: Matter Root Node (0x0016)
 Network: 0
 Profile ID: 0x0103
 Version: 1
 Enabled Clusters: 14
 Enabled Attributes: 256
 Enabled Reporting: 284

Cluster	Required Cluster	Cluster ID	Manufacturer Code	Enable	Configure
Identify		0x0003	---	Not Enabled	⚙️
Groups		0x0004	---	Not Enabled	⚙️
Scenes		0x0005	---	Not Enabled	⚙️
On/Off		0x0006	---	Not Enabled	⚙️
On/Off Switch Configuration		0x0007	---	Not Enabled	⚙️
Level Control		0x0008	---	Not Enabled	⚙️
Binary Input (Basic)		0x000F	---	Not Enabled	⚙️
Pulse Width Modulation		0x001C	---	Not Enabled	⚙️
Descriptor	Server	0x001D	---	Server	⚙️
Binding		0x001E	---	Not Enabled	⚙️
Access Control	Server	0x001F	---	Server	⚙️

Network topology



Developing application logic and testing of the setup

- Coding demo