How to Teach Threading to a Dolphin

Misuse of Home IoT Networks

András Tevesz

Nullcon Goa
2025

Who am I?

Current

Senior Vulnerability Researcher

CUJO AI

Former

Morgan Stanley

Cloudera

BDO

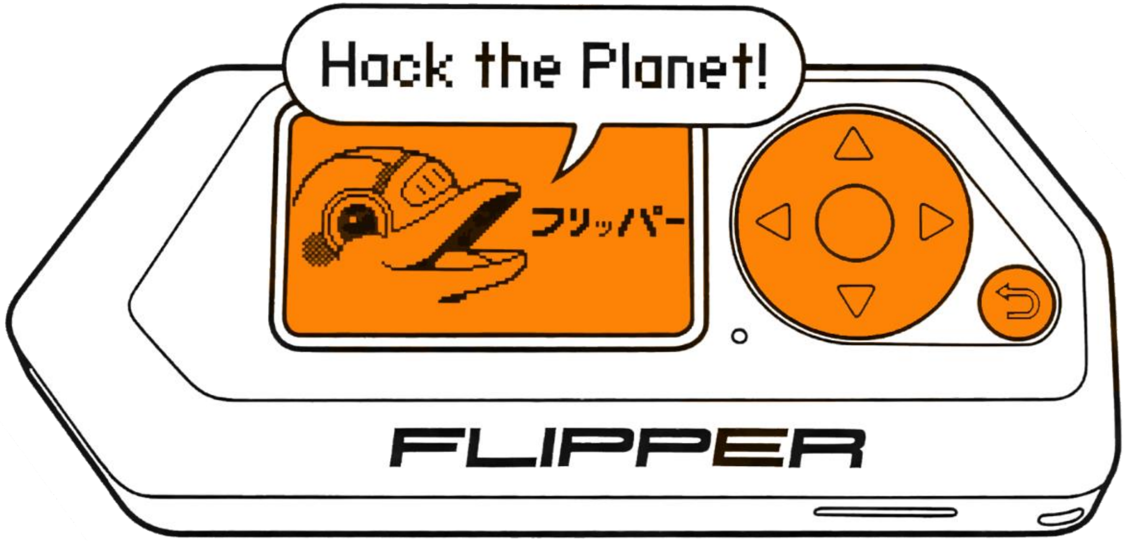
What I do

Reverse engineering

IoT Vulnerability research

Coding / Design

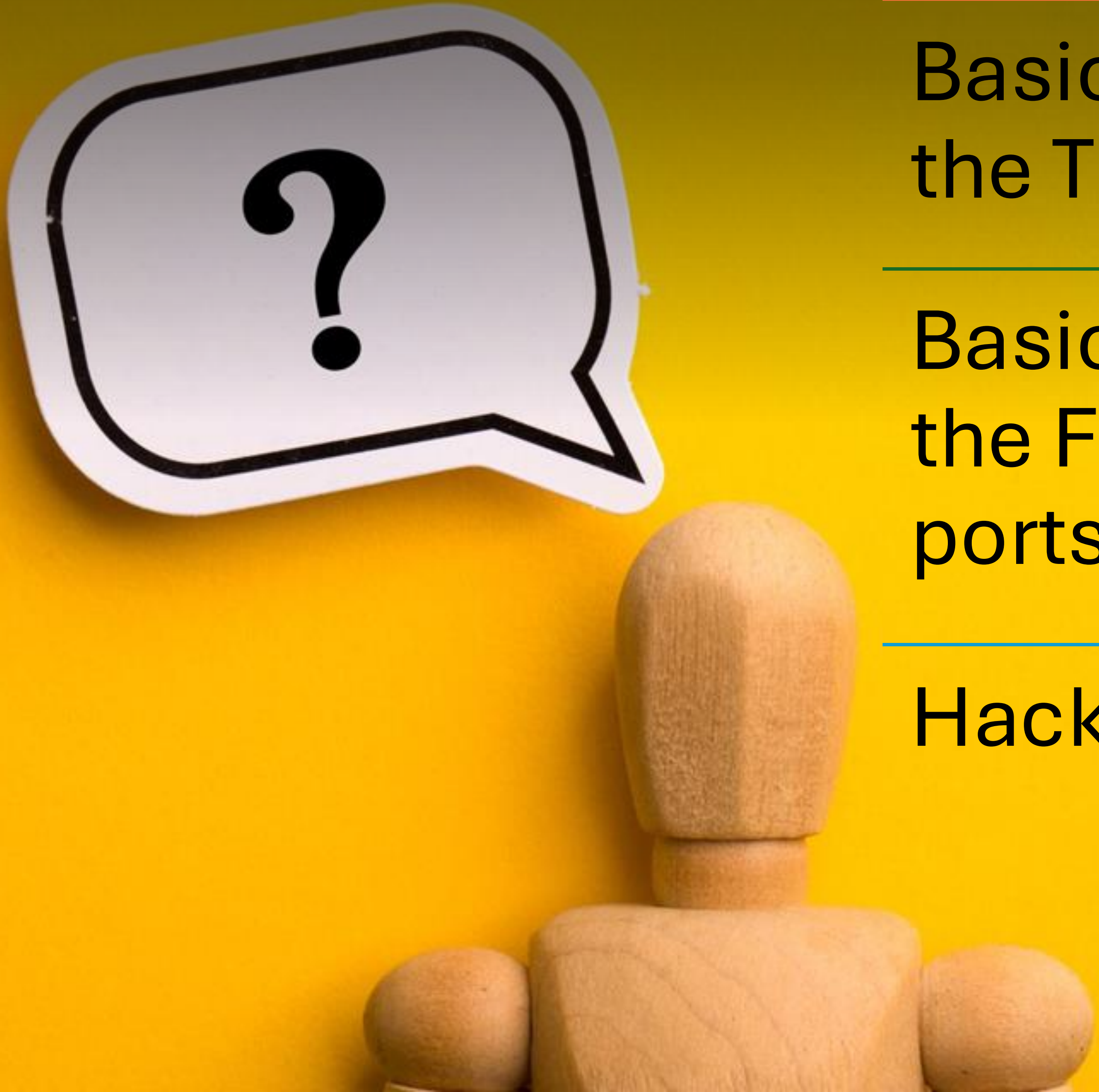
What is this presentation about?
Agenda



THREAD

matter

What's in it for you?



Basic understanding of the Thread and Matter

Basic understanding of the FlipperZero’s GPIO ports

Hacking

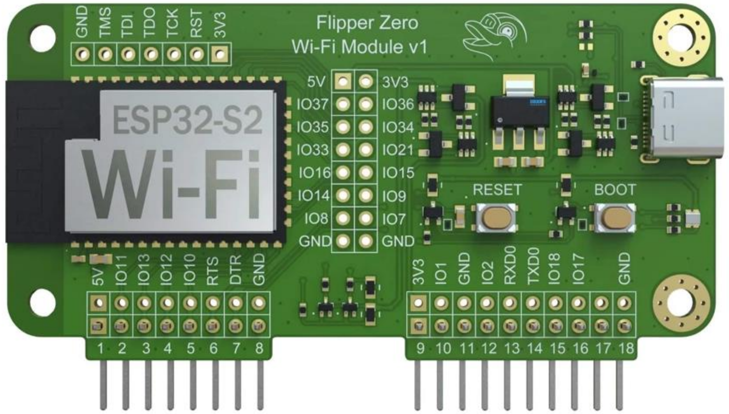
Where did this come from?

I conducted a research project on the Thread protocol.

I wanted to understand how the TCP connection could be monitored and, if necessary, blocked.

I found no device on the market to easily interact with Thread.

During the research, I encountered challenges with devices, SDKs, and changing codebases.



Banned in Brazil...

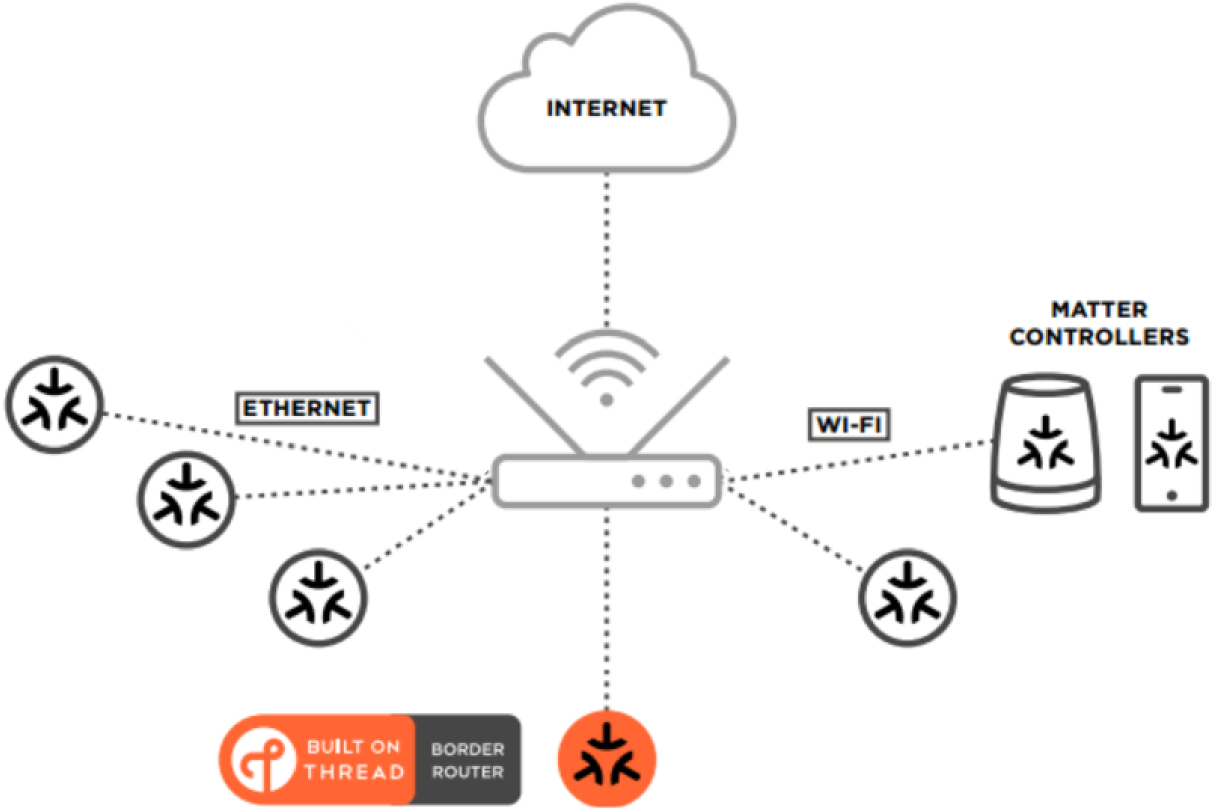
Flipper Zero Multi-tool device for geeks

- 125 kHz **RFID**
- Sub 1 GHz transceiver
- **NFC** proximity cards
- **Bluetooth**
- **Infrared** transceiver
- MicroSD card reader
- **USB-C, GPIO**
- SPI, **UART**, I2C

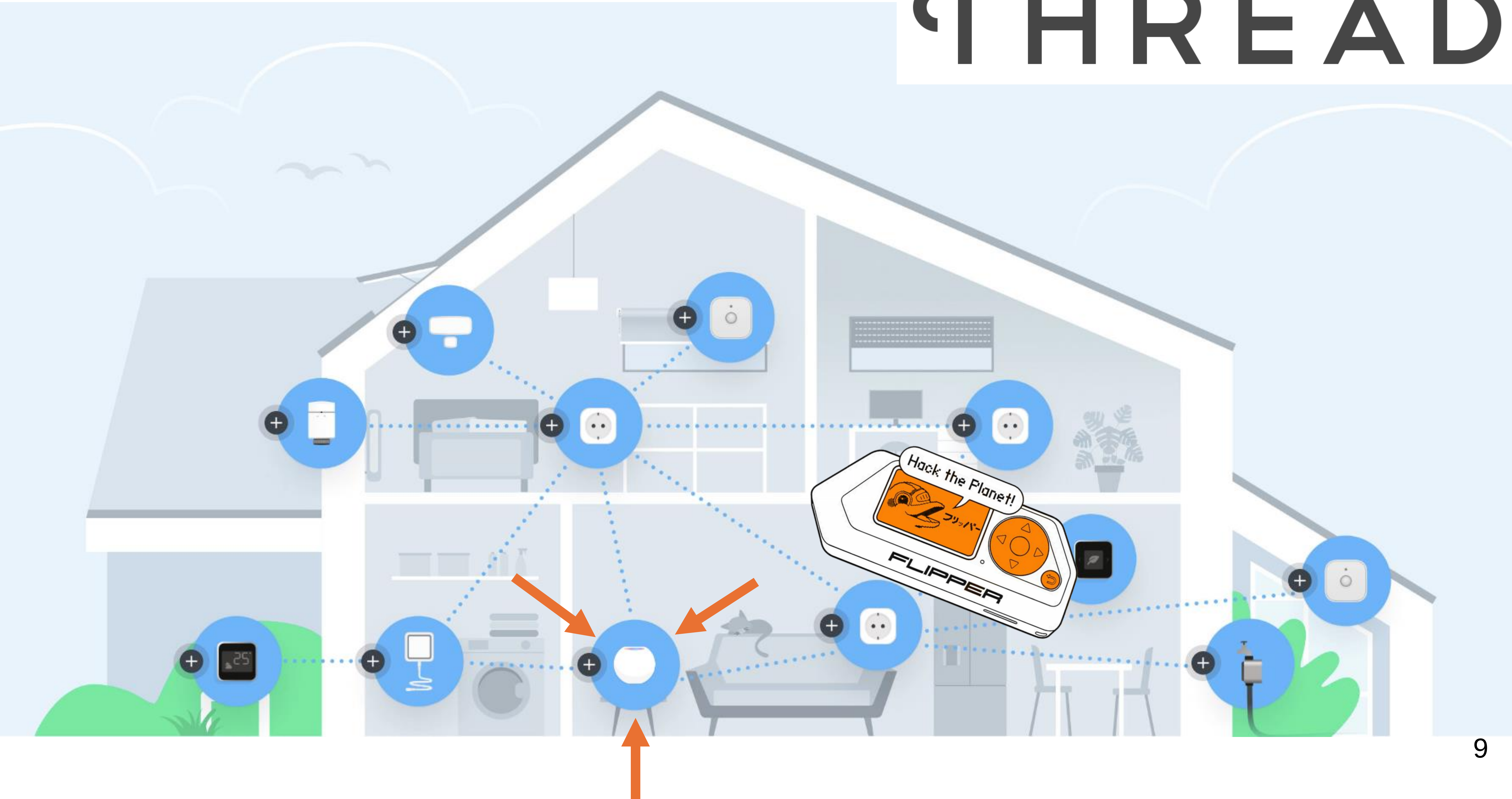
The Protocols

matter

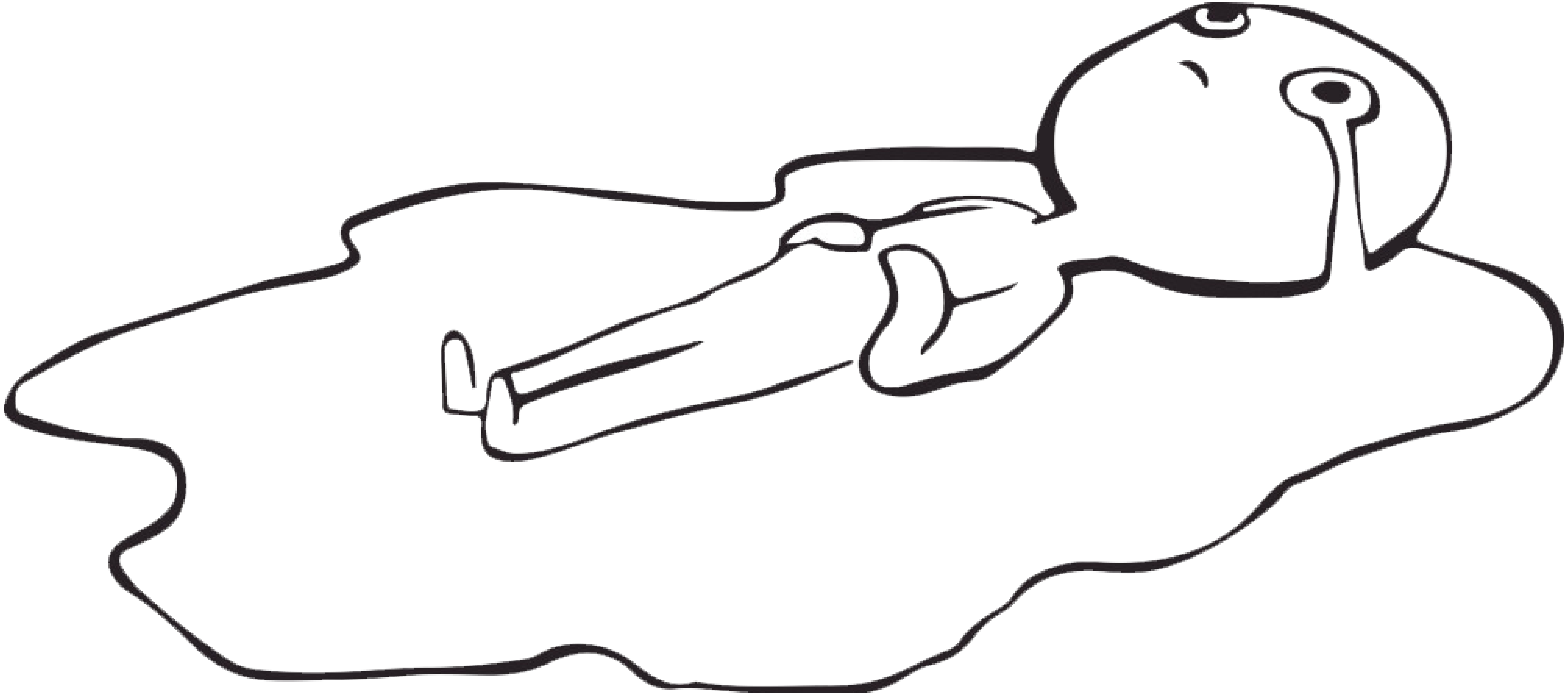
alexa
eve.


INTERNET
ETHERNET
WI-FI
MATTER CONTROLLERS
BUILT ON THREAD
BORDER ROUTER

THREAD

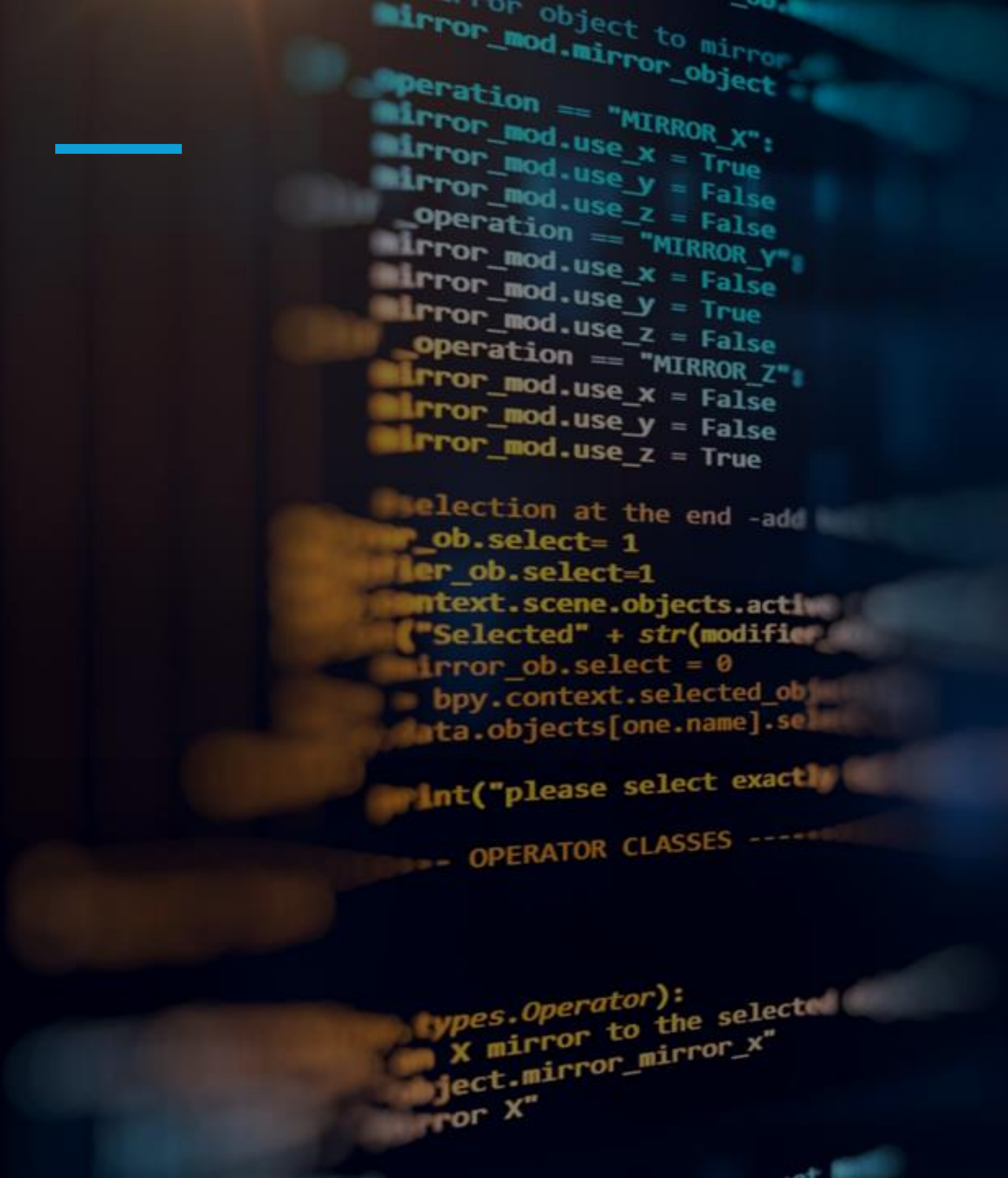


Hackers before Aircrack-ng and packet sniffing



The devices

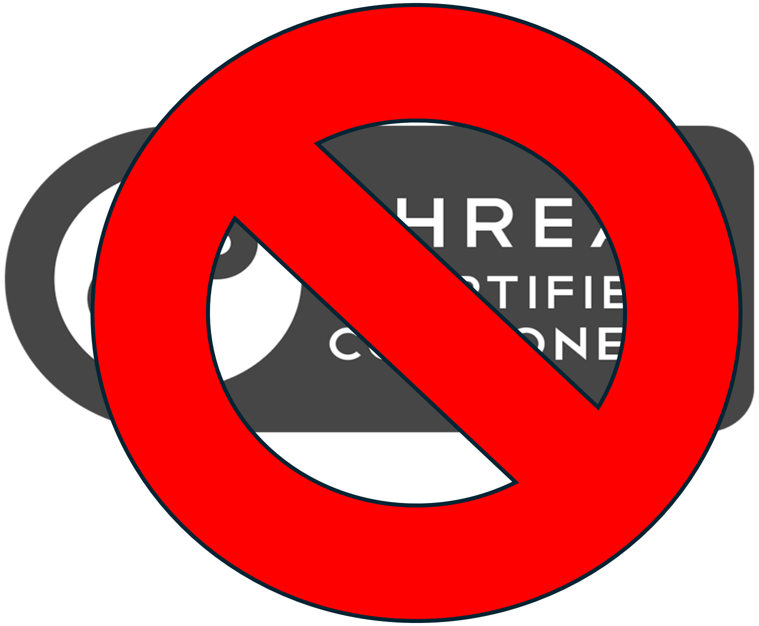
Do you have them?



The Software

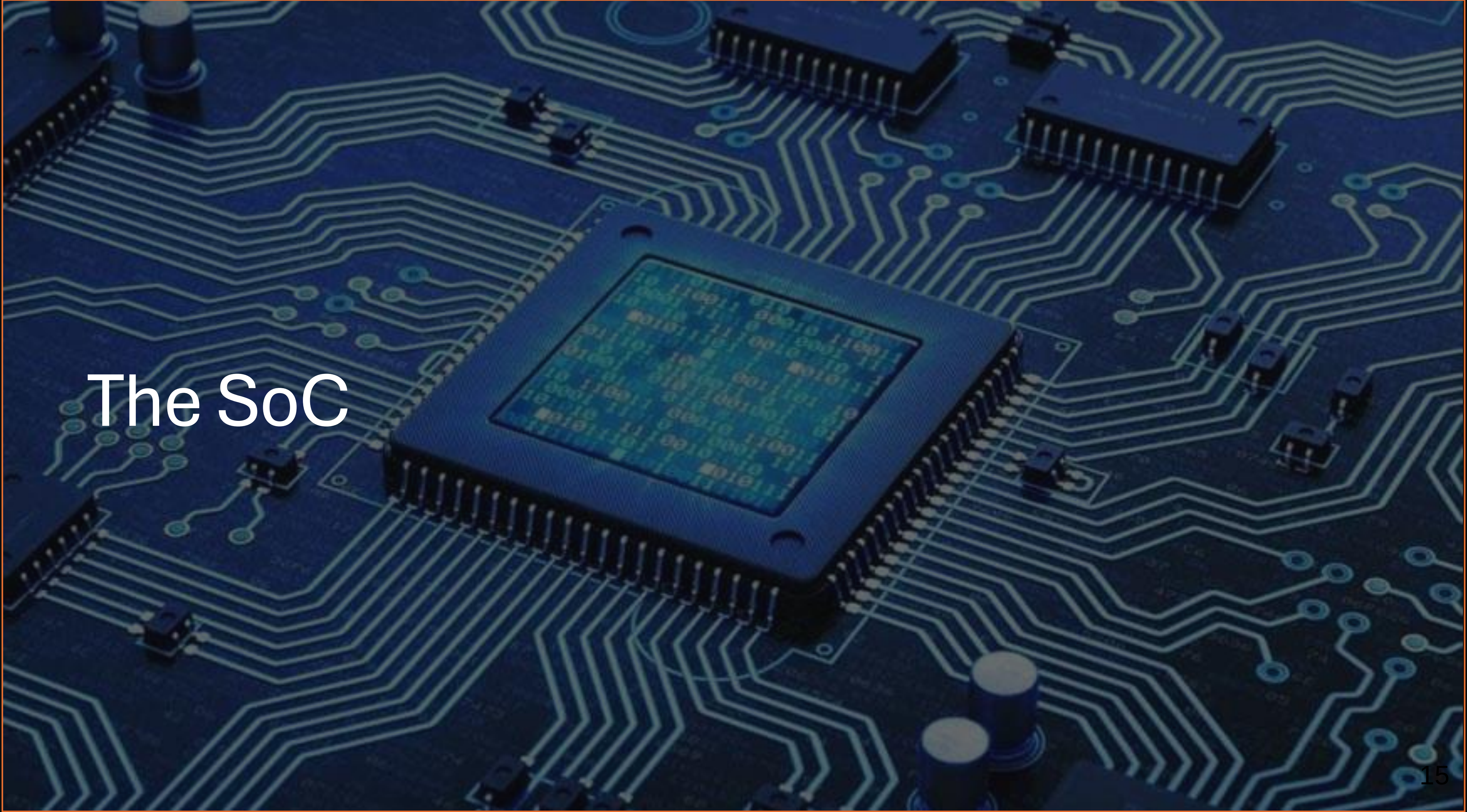
THREAD

CONNECTS
WITH
THREAD



Thread 1.1	• Maybe only Amazon is still using it
Thread 1.2	• You might use it (most devices are using it)
Thread 1.3	• It's almost there (some devices are supporting it)
Thread 1.4	• It's released but (not there yet)

The SoC

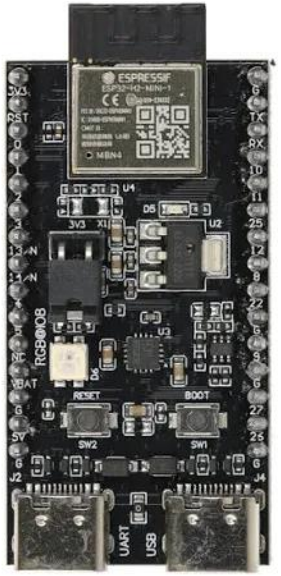


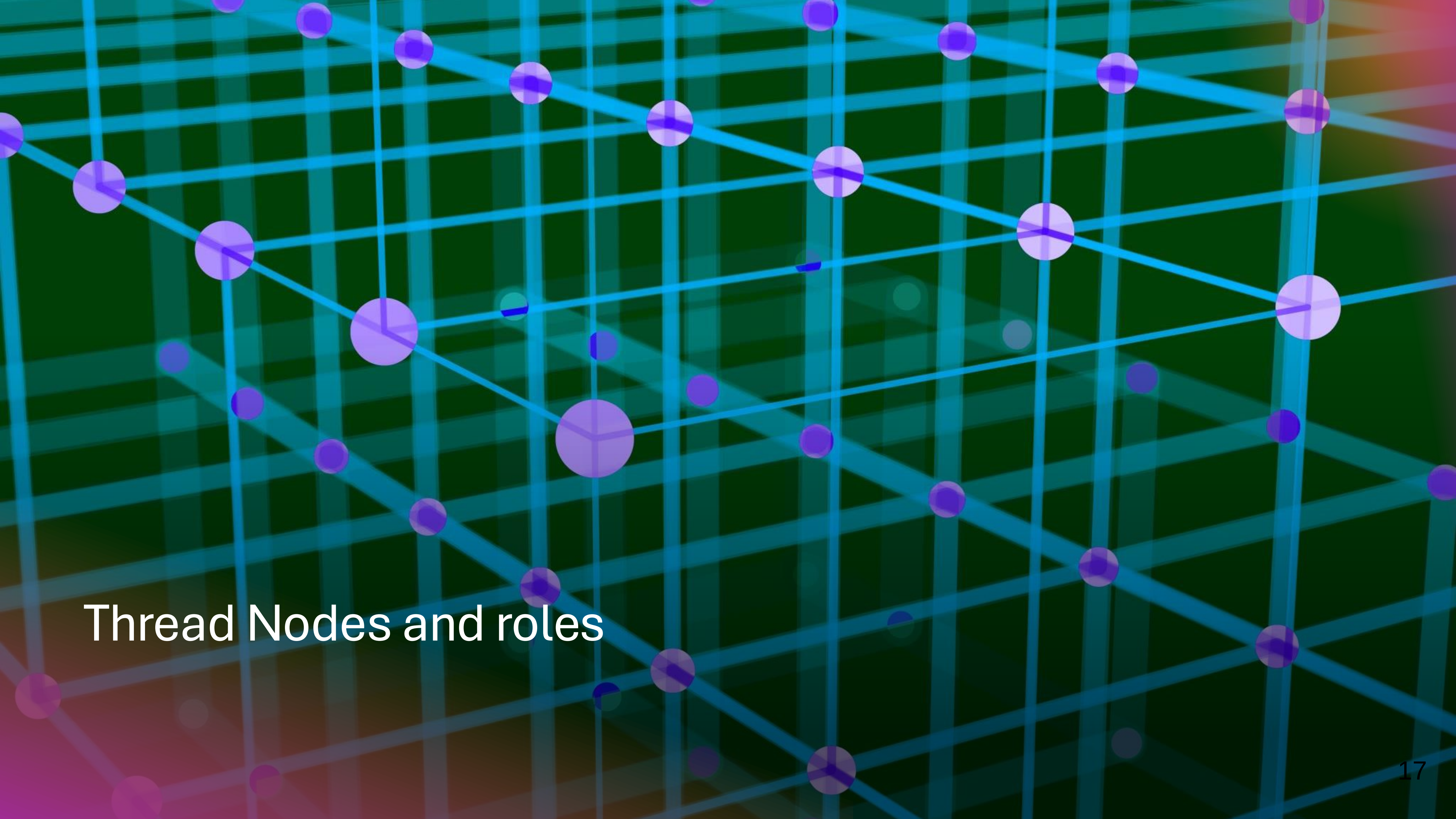
Radio & SOC

Nordic Nrf 52840

SiliconLabs MG26

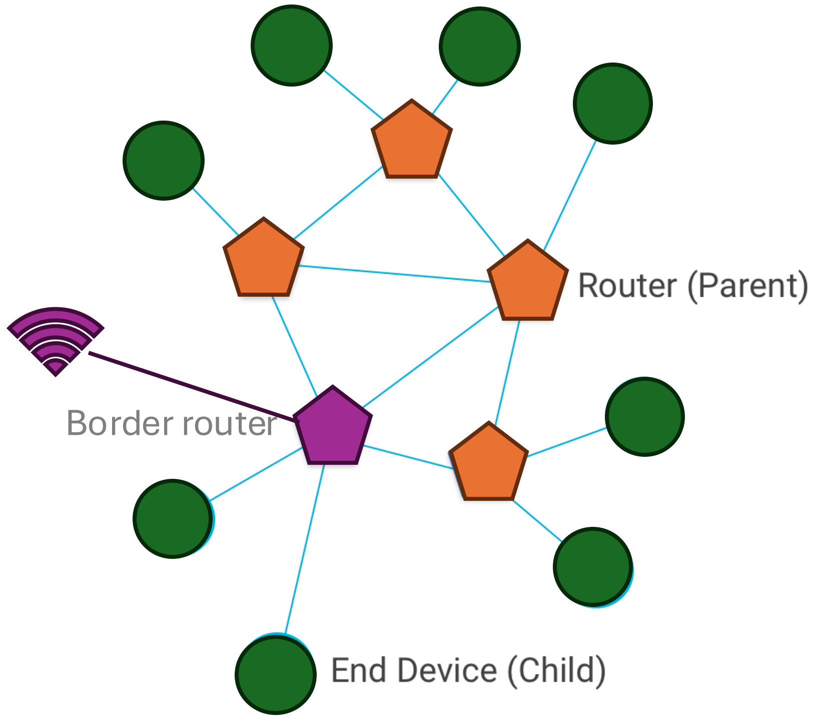
Esp32-H2





Thread Nodes and roles

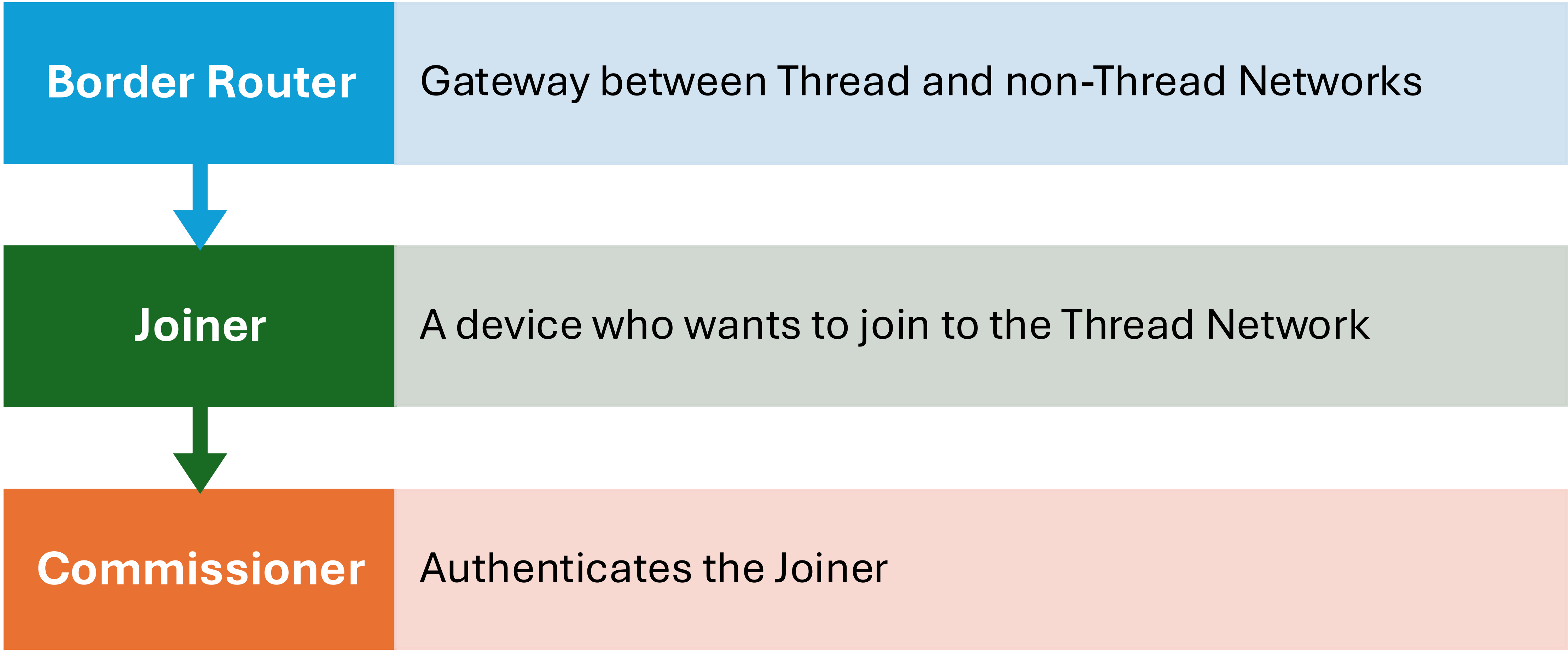
Node Roles



Thread Roles	Functions
Router	- **forward** packets **for other** devices - **accepts** joiners - keeps radio **on**
End Devices ED	- communicates with a **single router** - does NOT forward packets - **can disable** radio
Border Router	- **relays** between **Thread** and **non-Thread** - act as a **gateway** for others

The Commissioning

Who is who in commissioning

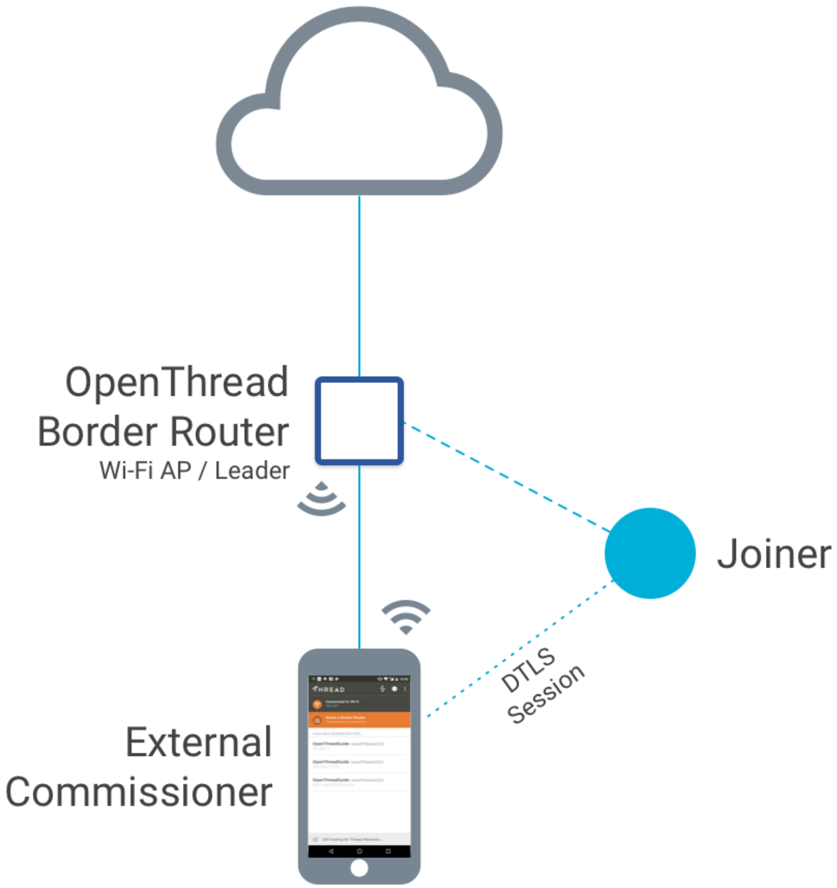


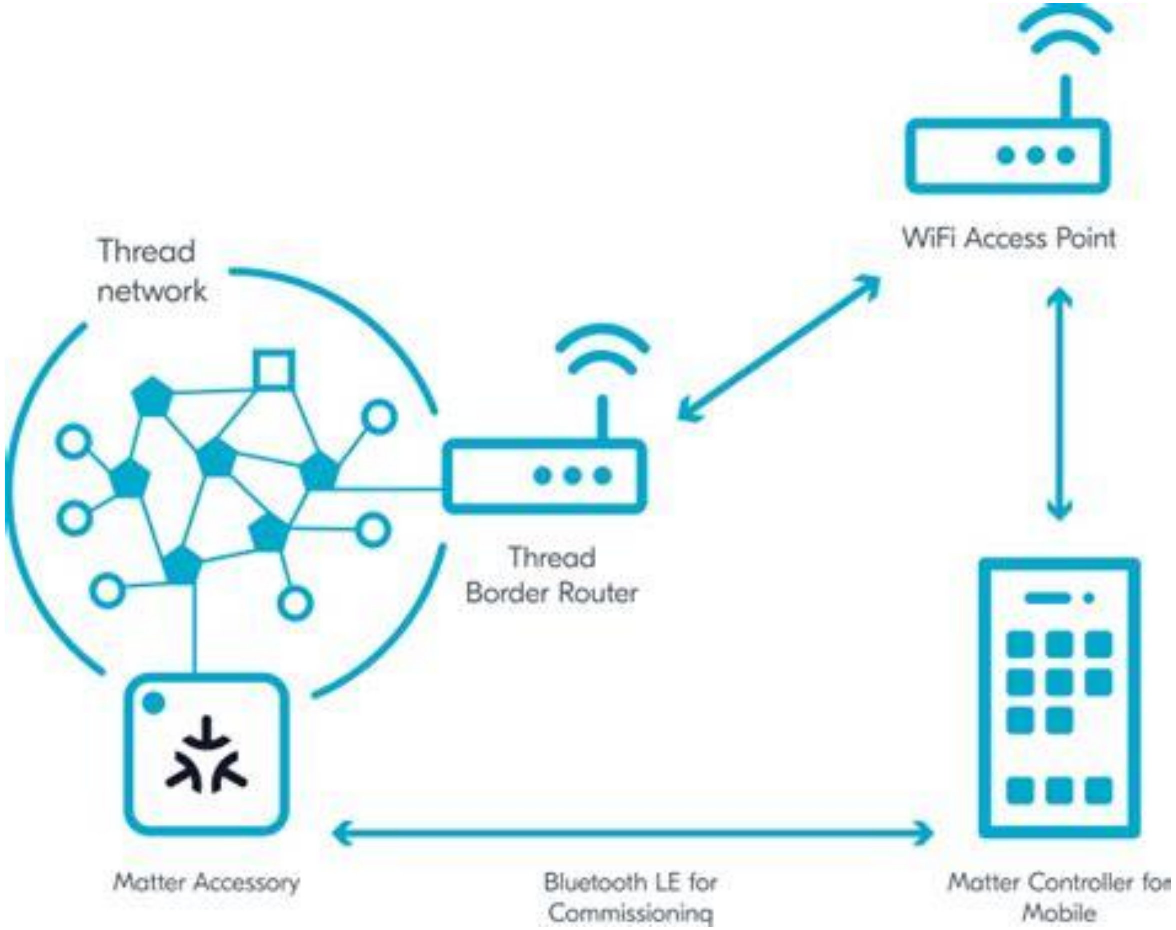
Border Router Gateway between Thread and non-Thread Networks

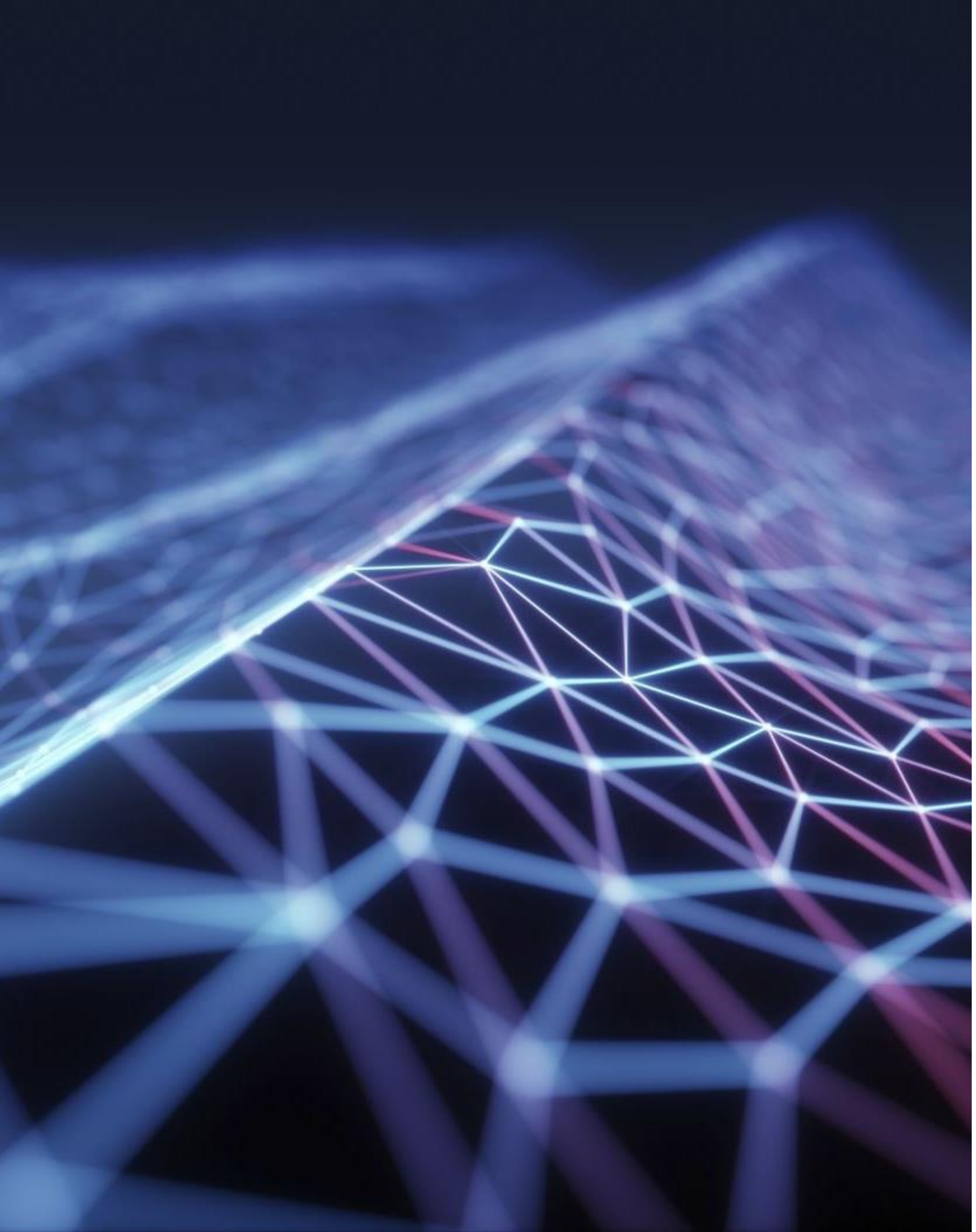
Joiner A device who wants to join to the Thread Network

Commissioner Authenticates the Joiner

How does connection in Thread work?







Is there another way to connect?

Use a joiner password (it needs a joiner window to be usable)

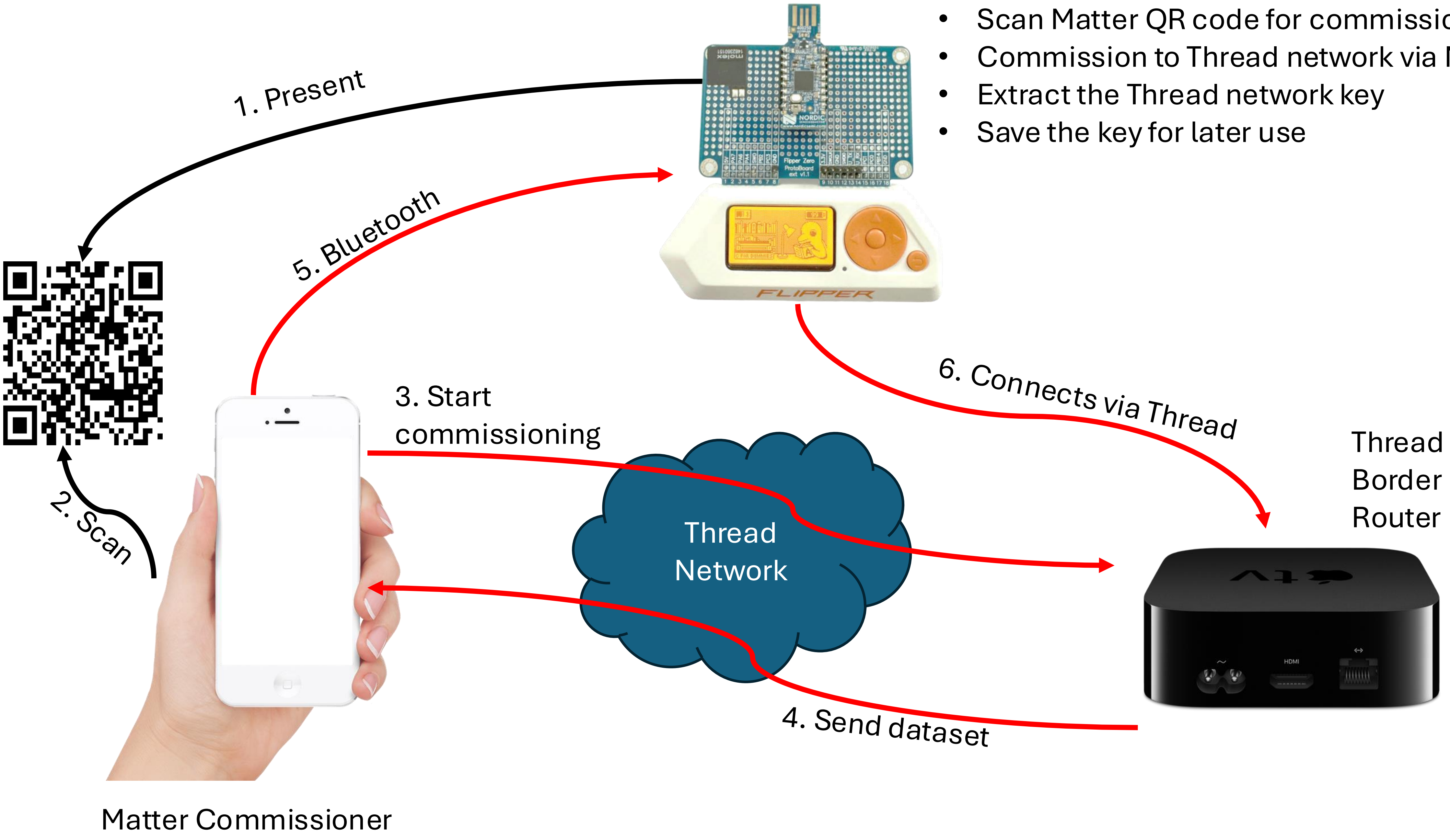
Use a leaked dataset

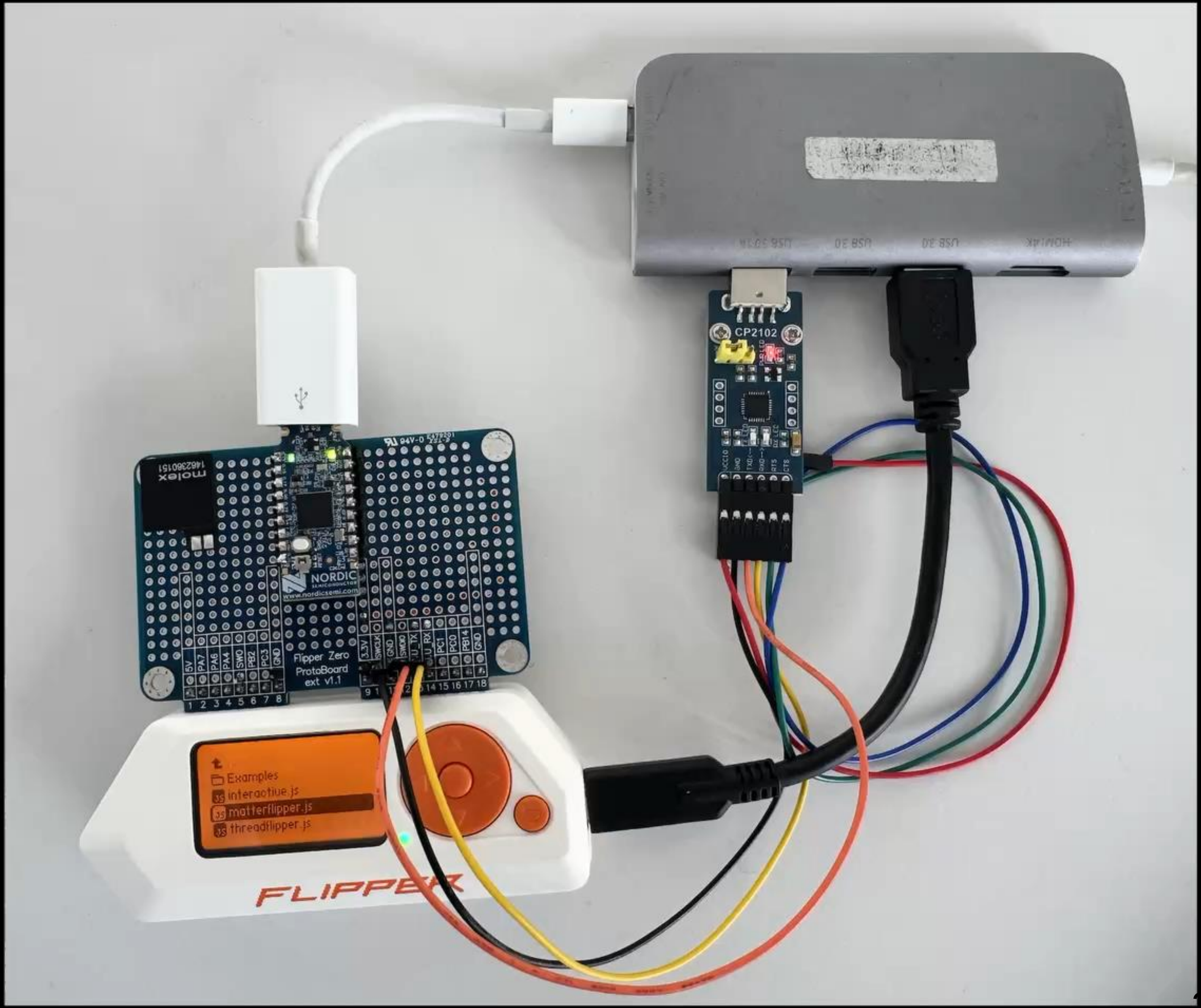
Use a known network key

Demo

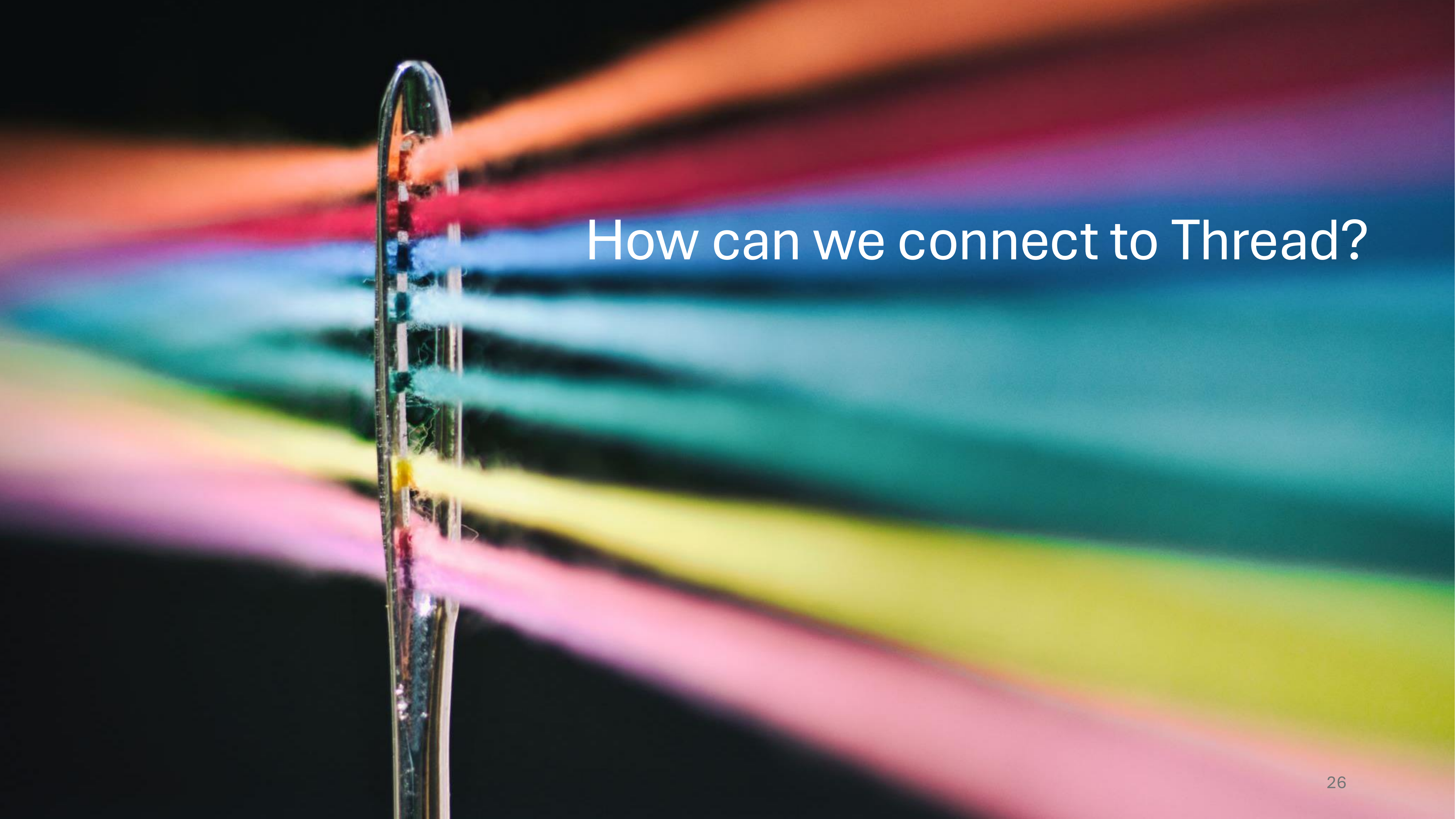
Commission Flipper’s evil led

- Scan Matter QR code for commissioning
- Commission to Thread network via Matter
- Extract the Thread network key
- Save the key for later use



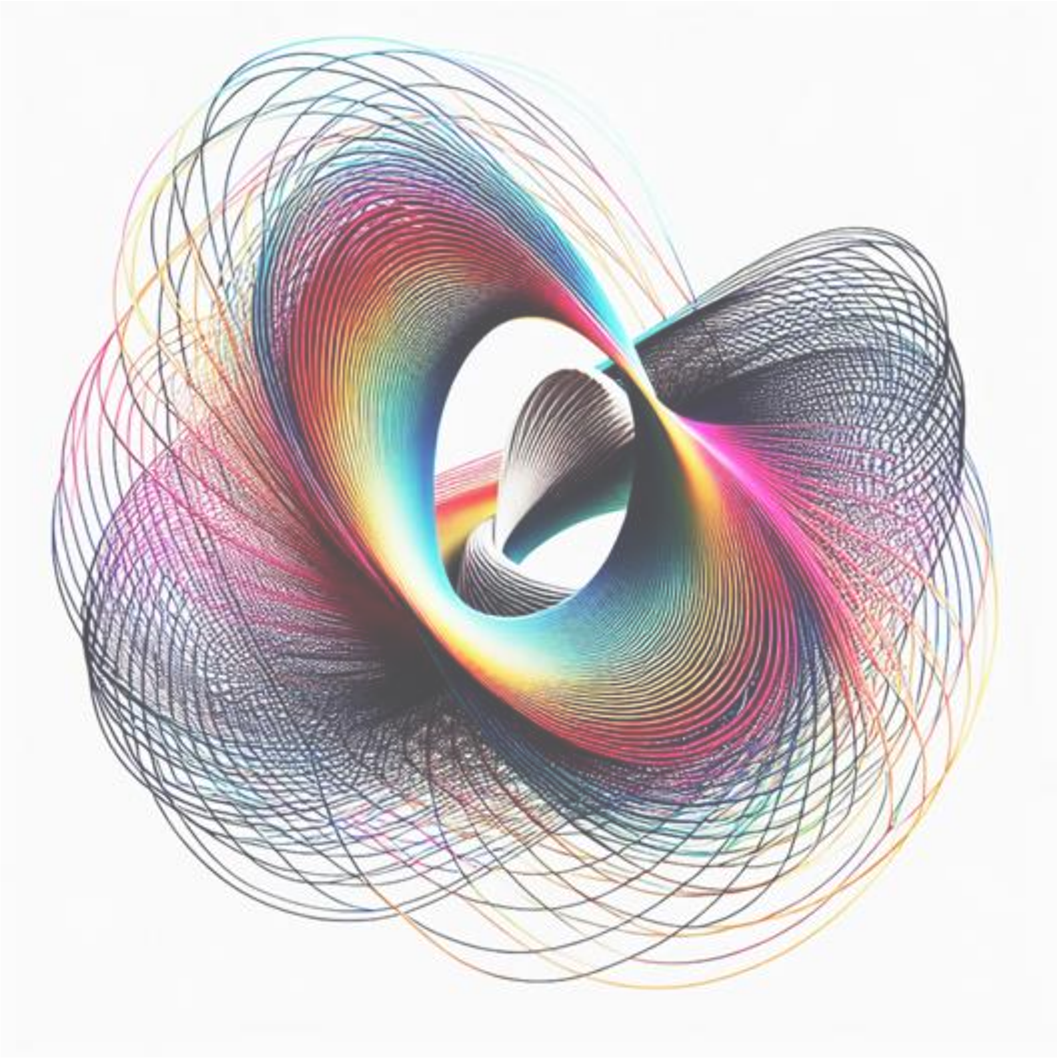


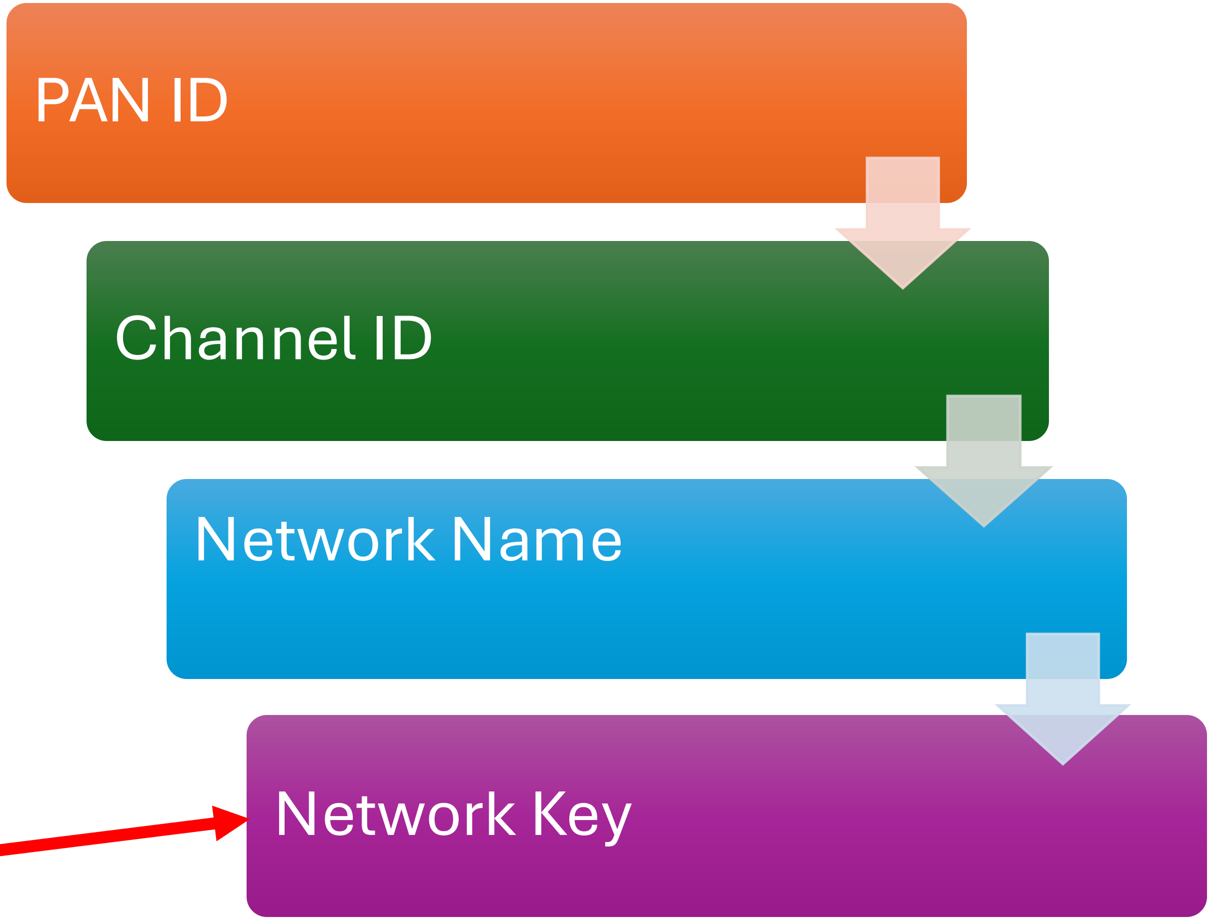
MatterFlipper
Demo



How can we connect to Thread?

Mandatory for a connection





PAN ID

Channel ID

Network Name

Network Key

What's not advertised for anyone?

How to get PAN, Channel, and Network name

uart# ot scan -> otLinkActiveScan

Send 802.15.4 Beacon (Layer 1.)

```
| PAN  | MAC Address          | Ch | dBm | LQI |
| 9749 | ee9afe59d77e515e  | 11 | -60 | 128   |
| e948 | 9273124c7a125bc8 | 25 | -61 | 128   |
| e948 | 866d554cead1f46f  | 25 | -57 | 152   |
```

uart# ot discover -> otThreadDiscover

Send MeshCoP Discovery (Layer 3.)

```
| Network Name       | Extended PAN         | PAN  | MAC Address          |
| AMZN-Thread-9749 | f23dd4876455b41f   | 9749 | ee9afe59d77e515e |
| MyHome44015048  | 555c7d90aea746ca | e948 | 767d9c53c6dfb1bd |
| MyHome44015048  | 555c7d90aea746ca | e948 | 866d554cead1f46f  |
| MyHome44015048  | 555c7d90aea746ca | e948 | 9273124c7a125bc8 |
```

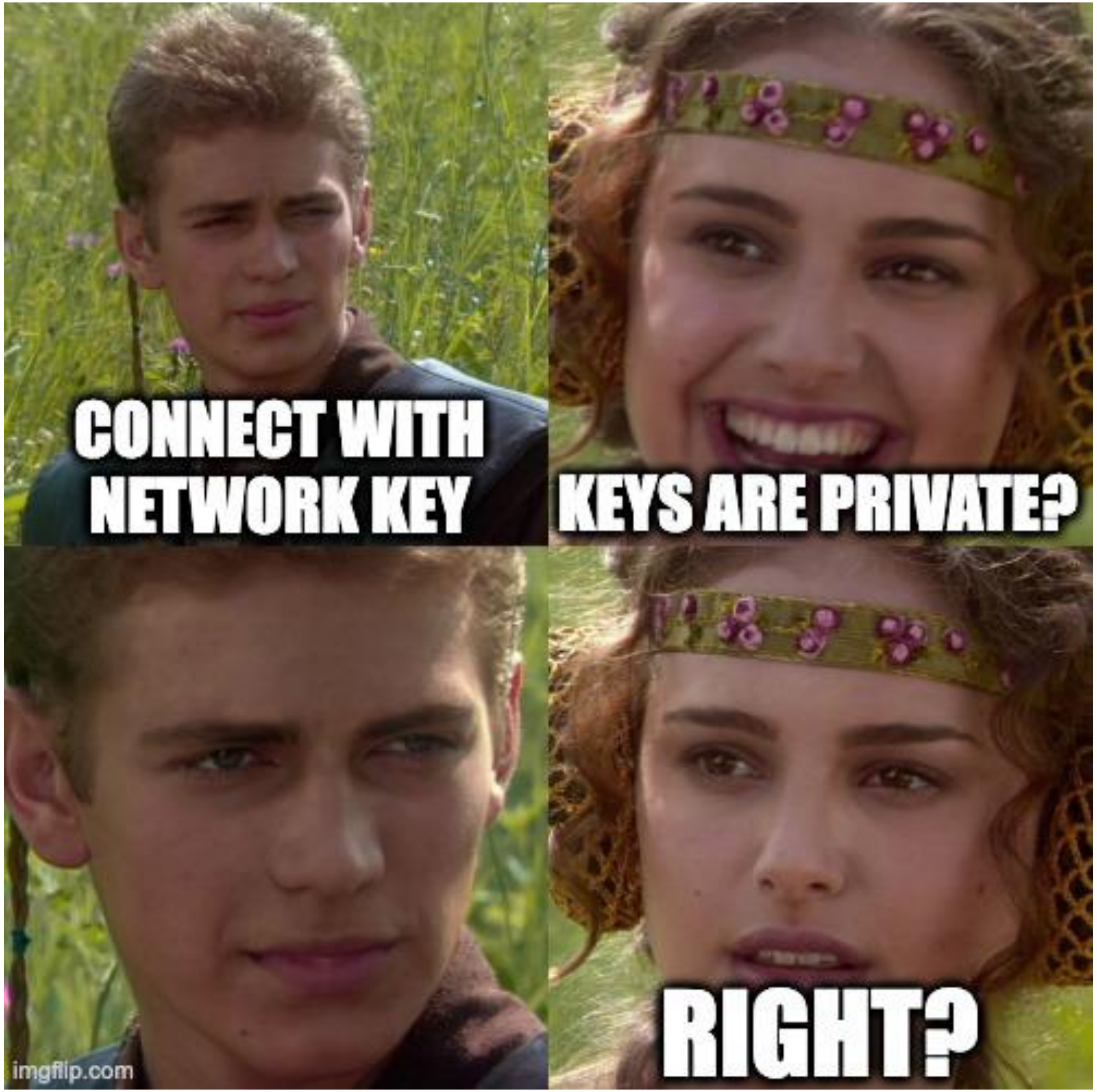
What’s in your Thread dataset

TLV Tag Length VALUE encoding

$ python3 tlv-parser.py

```
0e 08 0000000000010000
00 03 000012
35 06 0004001fffe0
02 08 a1fce8946f2f9b1d
07 08 fd505ff6fd1b325b
05 10 e67446d4e450ad76cd3ad5472530d410
03 0f 4f70656e5468726561642d65653937
01 02 ee97
04 10 42743e8b67c06353cd038520a0ab8b7f
0c 04 02a0f7f8
```

```
t: 14 (ACTIVETIMESTAMP), l: 8, v: 0x0000000000010000
t:  0 (CHANNEL), l: 3, v: 0x000012
t: 53 (CHANNELMASK), l: 6, v: 0x0004001fffe0
t:  2 (EXTPANID), l: 8, v: 0xa1fce8946f2f9b1d
t:  7 (MESHLOCALPREFIX), l: 8, v: 0xfd505ff6fd1b325b
t:  5 (NETWORKKEY), l: 16, v: 0xe67446d4e450ad76cd3ad5472530d410
t:  3 (NETWORKNAME), l: 15, v: b'OpenThread-ee97'
t:  1 (PANID), l: 2, v: 0xee97
t:  4 (PSKC), l: 16, v: 0x42743e8b67c06353cd038520a0ab8b7f
t: 12 (SECURITYPOLICY), l: 4, v: 0x02a0f7f8
```



default Open Thread

- 11112233445566778899DEAD1111DEAD
- 1234c0de7ab51234c0de7ab51234c0de
- 00112233445566778899aabbccddeeff

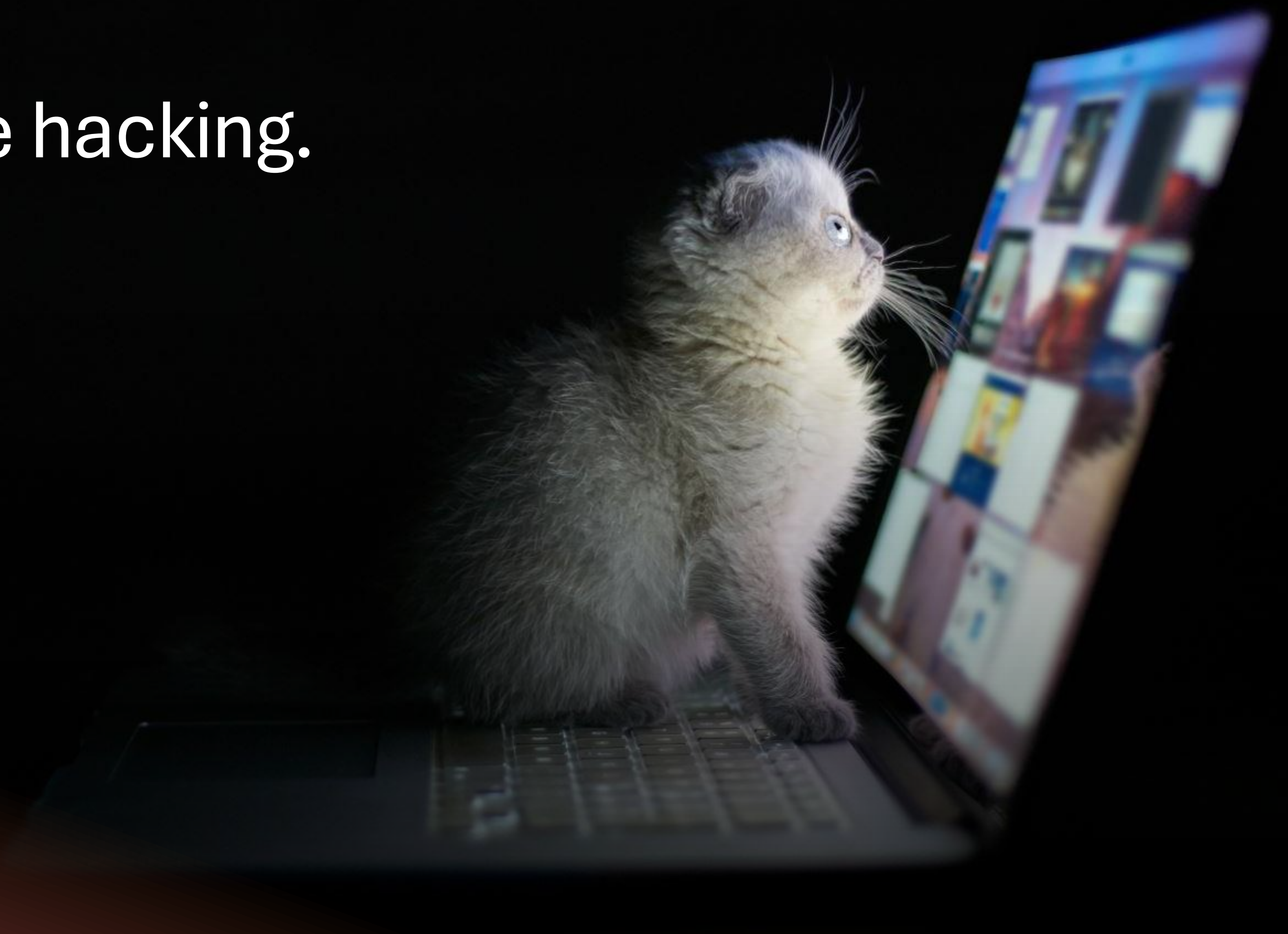
https://github.com/simenkid/ot-ctl/blob/main/index.js

- e947a2e6b08b8cfefa6961b5c3943928
- 89722adb7ef02054ec73111c337ec6a9

https://docs.gl-inet.com/iot/en/thread_board_router/gl-s200/openthread_border_router_codelabs/

- e67446d4e450ad76cd3ad5472530d410

Let’s dive into the hacking.

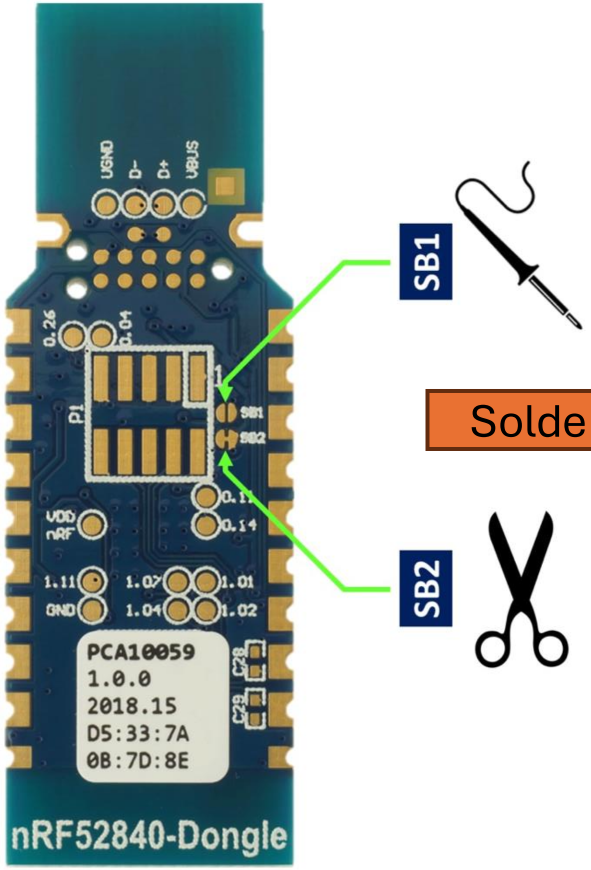


How to create an addon board

Write
Firmware

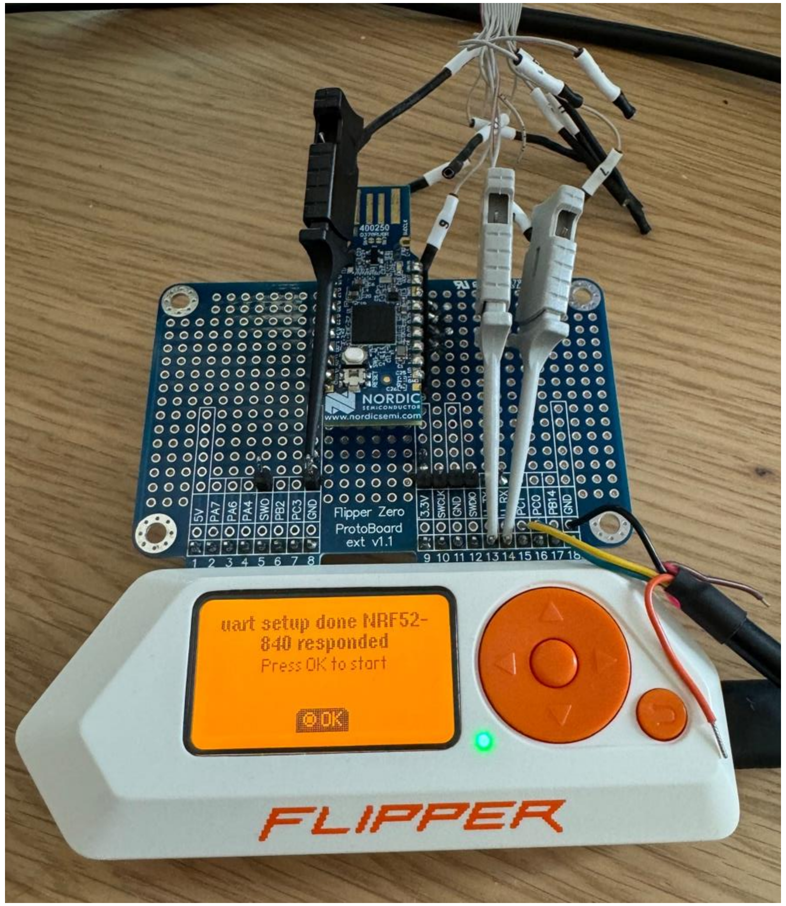
Zephyr FW

Enable
VDDOUT PIN



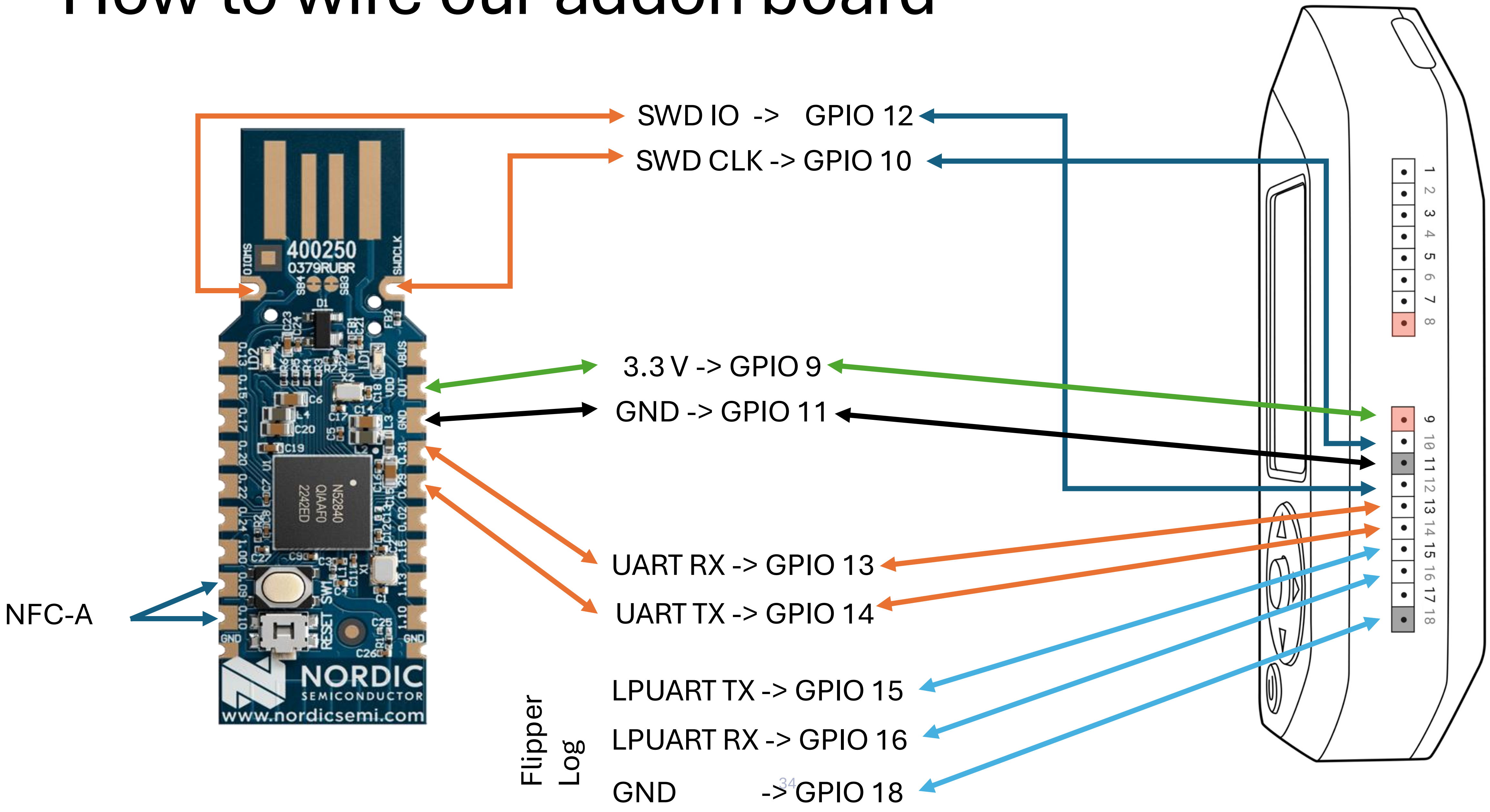
Soldering

ThreadFlipper



uart setup done NRF52-840 responded
Press OK to start

How to wire our addon board



SWD IO -> GPIO 12

SWD CLK -> GPIO 10

3.3 V -> GPIO 9

GND -> GPIO 11

UART RX -> GPIO 13

UART TX -> GPIO 14

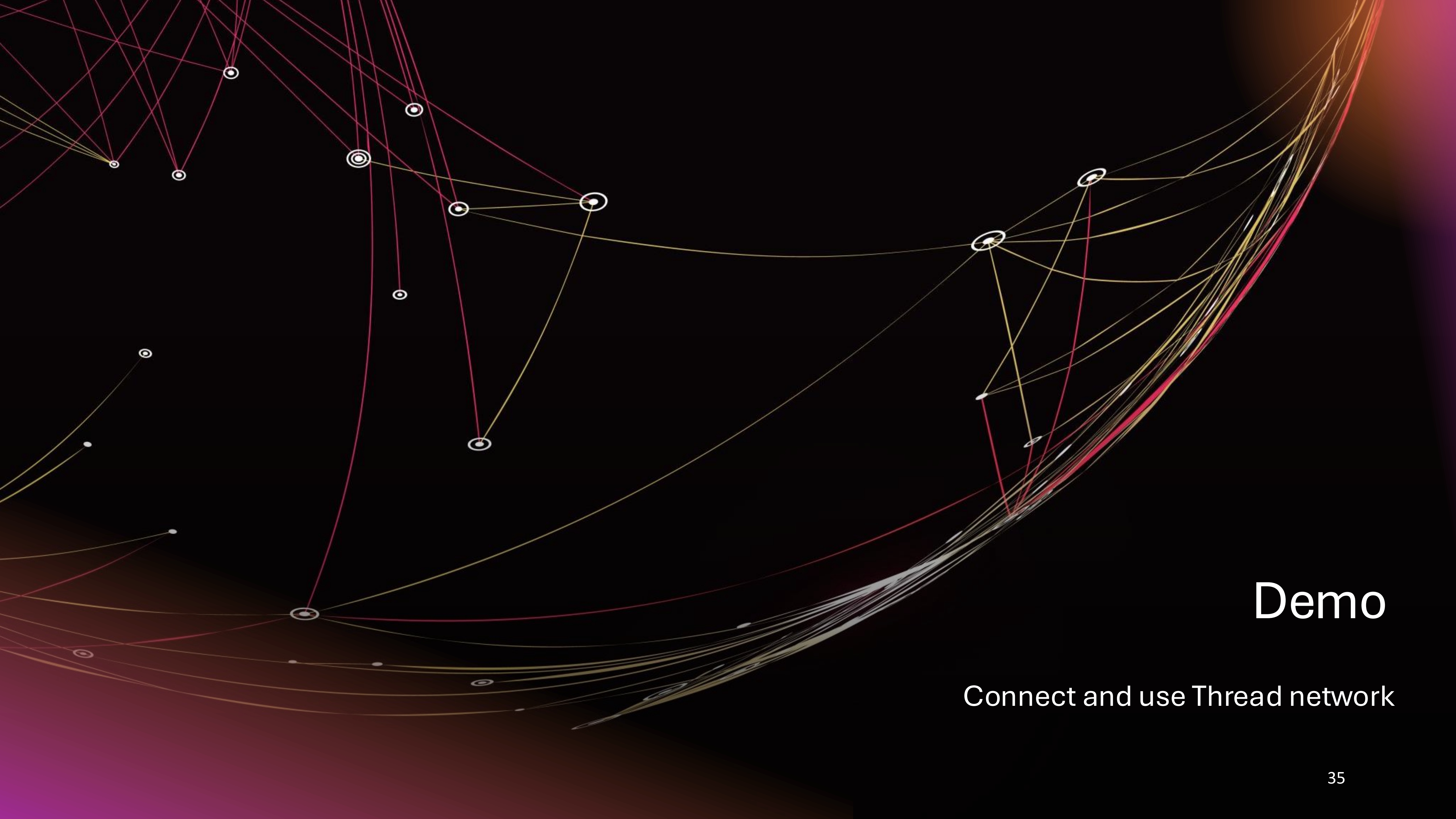
Flipper Log

LPUART TX -> GPIO 15

LPUART RX -> GPIO 16

GND -> GPIO 18

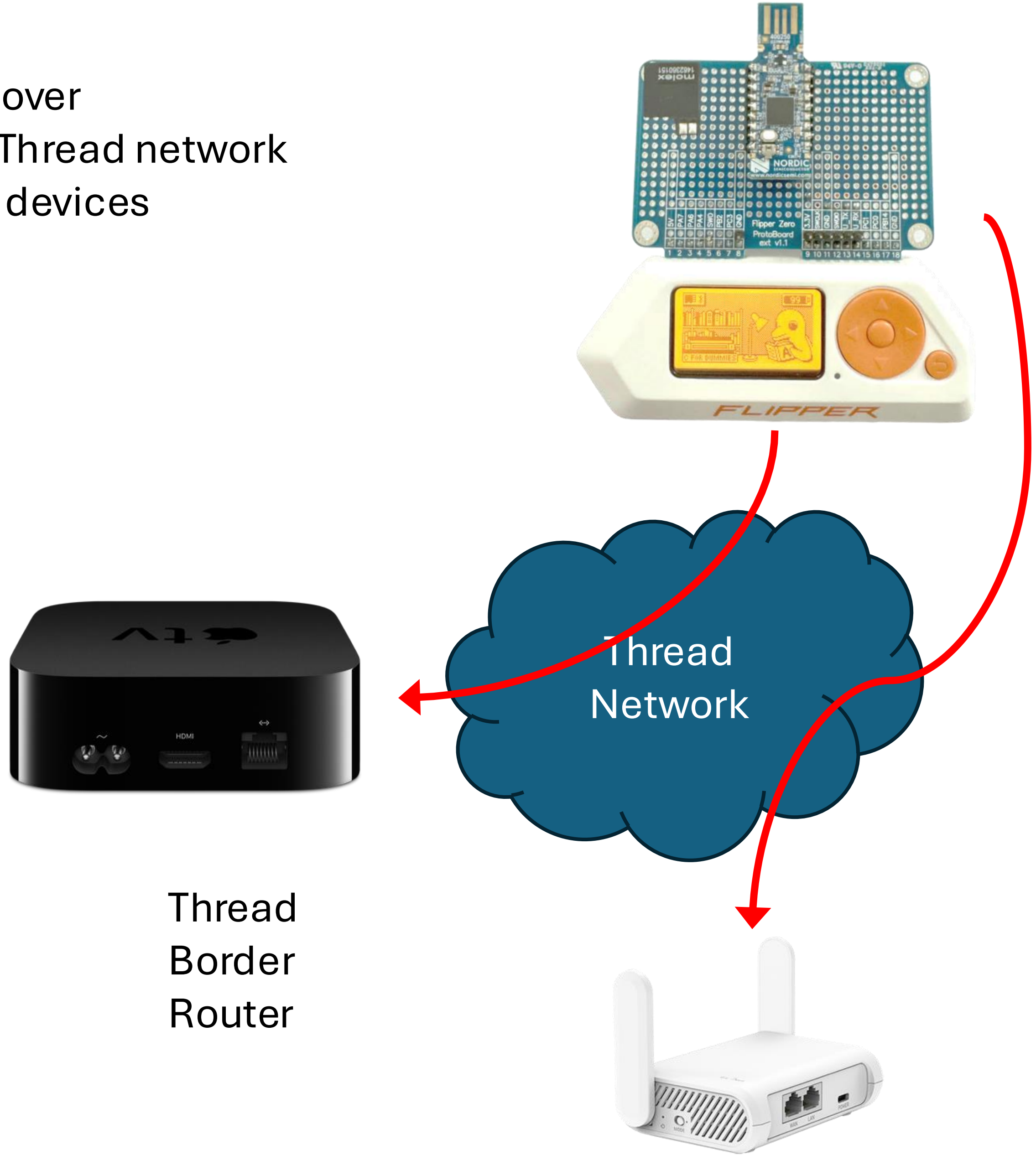
NFC-A



Demo

Connect and use Thread network

- Thread Discover
- Connect to Thread network
- Ping Thread devices

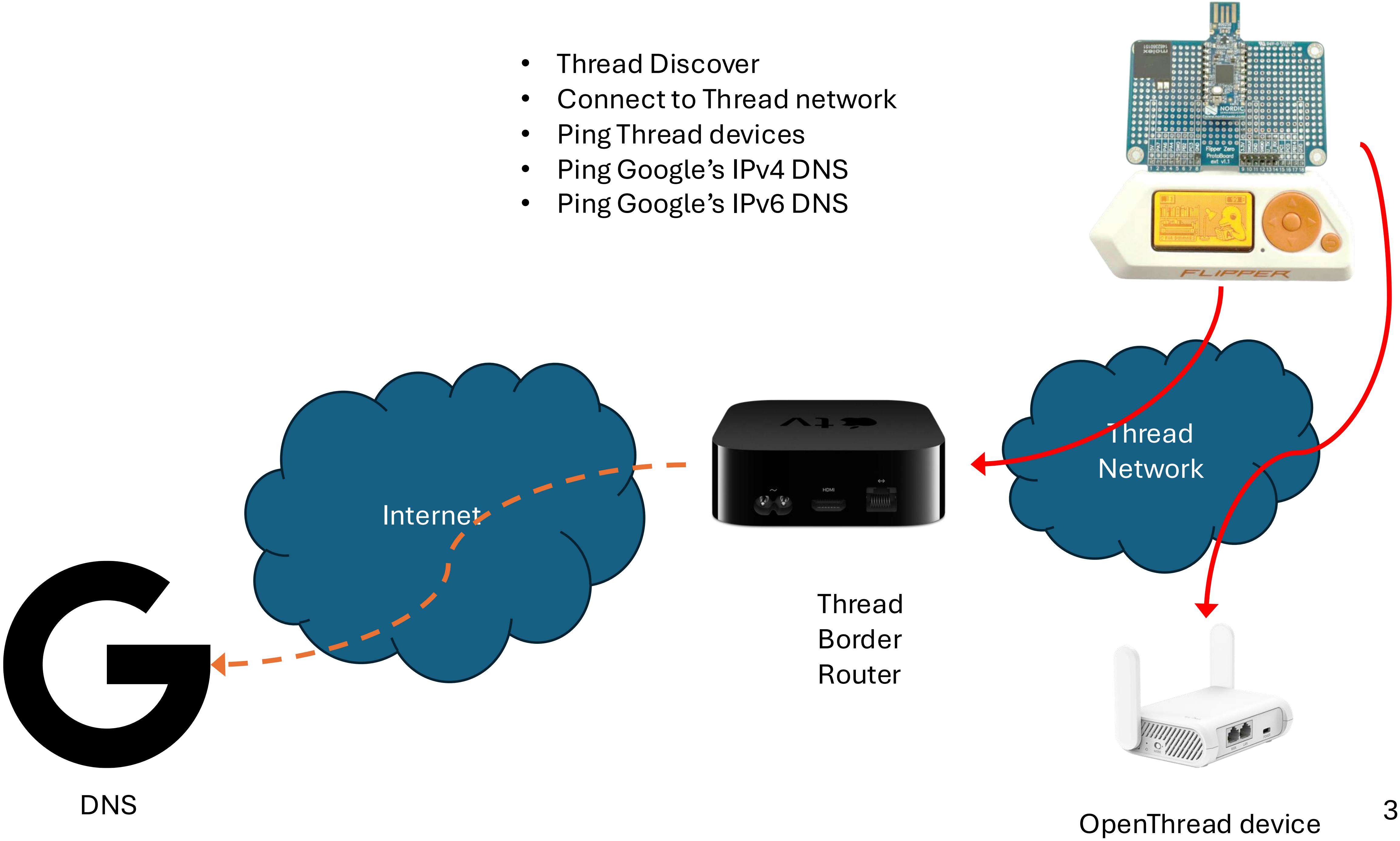


Thread Network

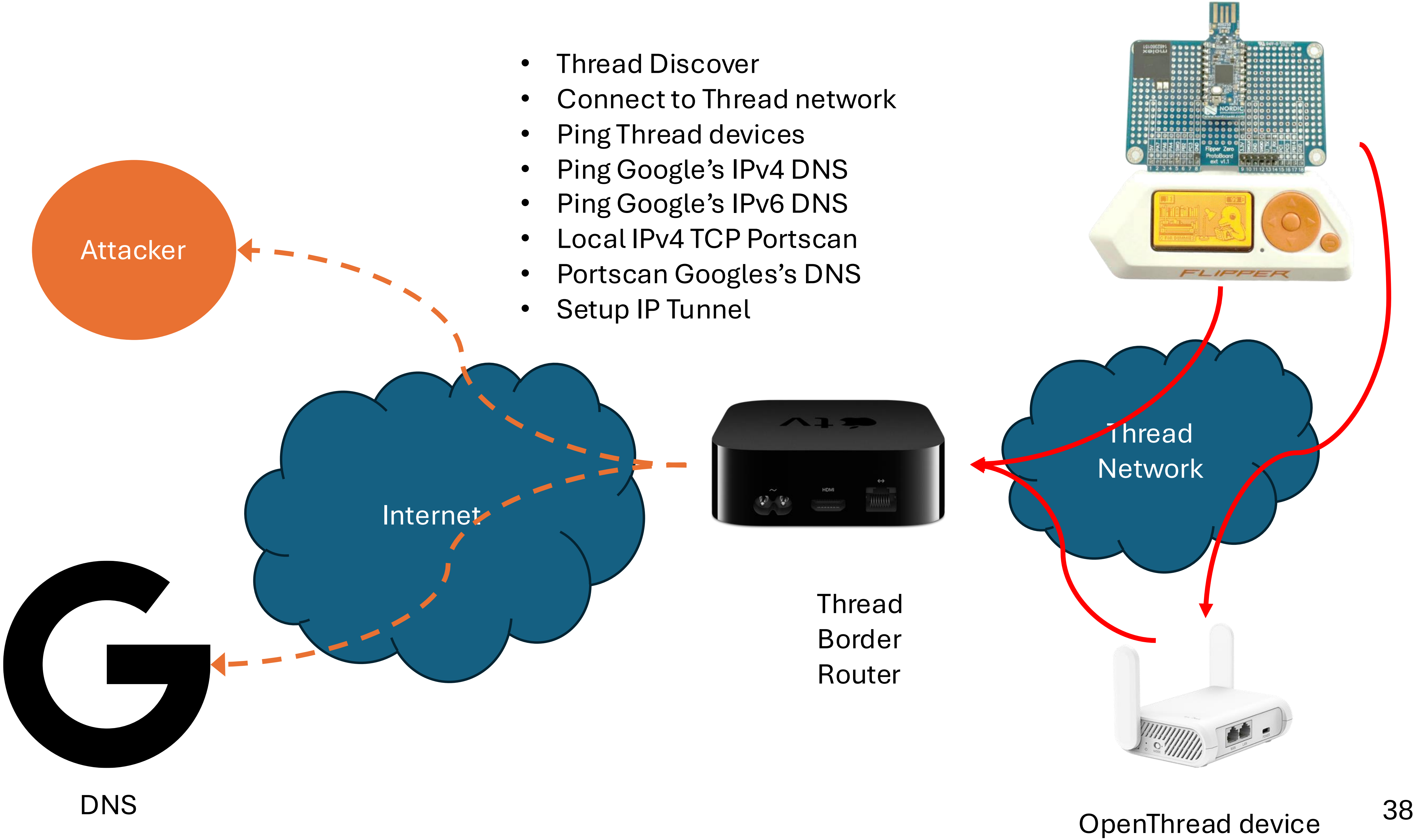
Thread Border Router

OpenThread device

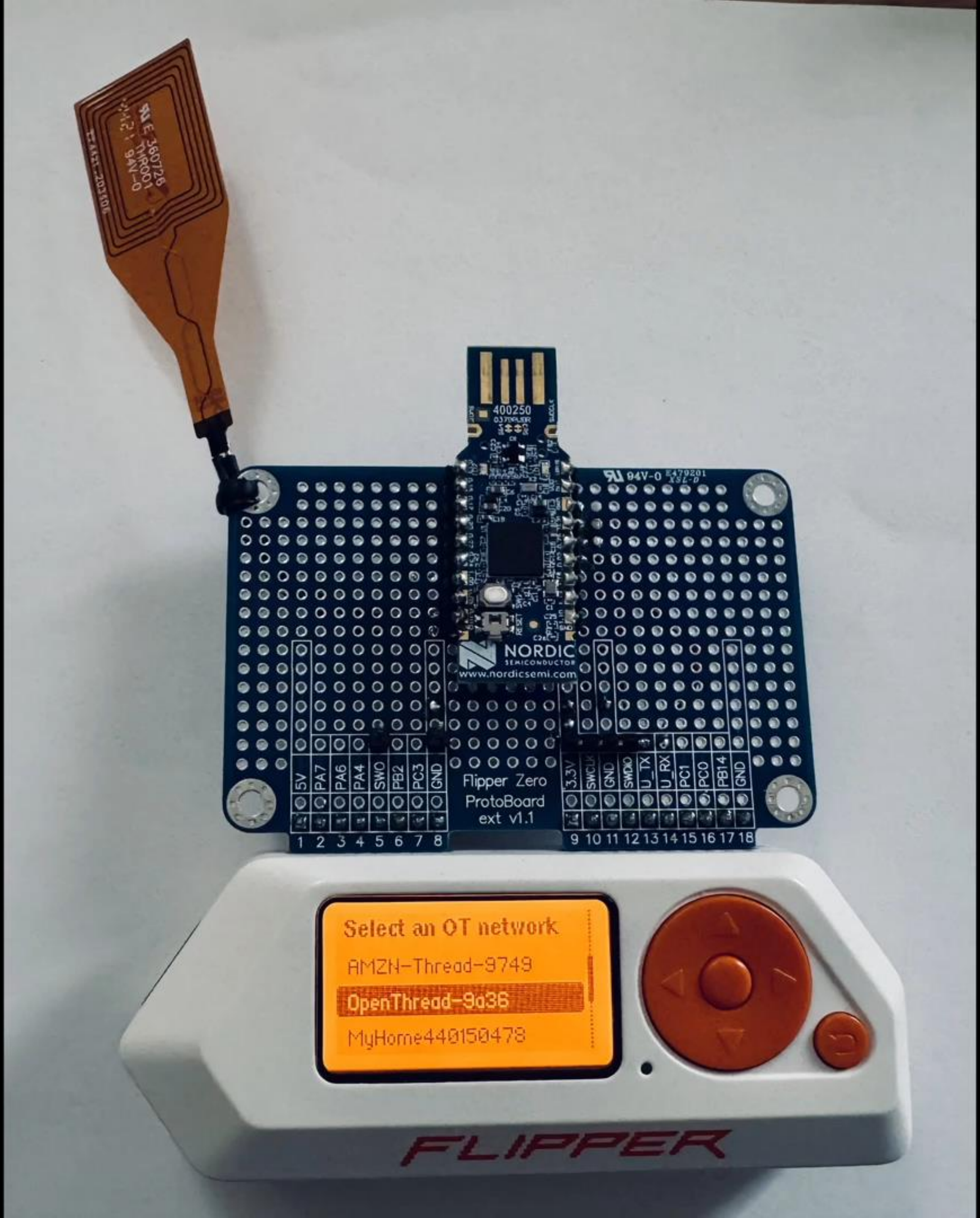
- Thread Discover
- Connect to Thread network
- Ping Thread devices
- Ping Google’s IPv4 DNS
- Ping Google’s IPv6 DNS

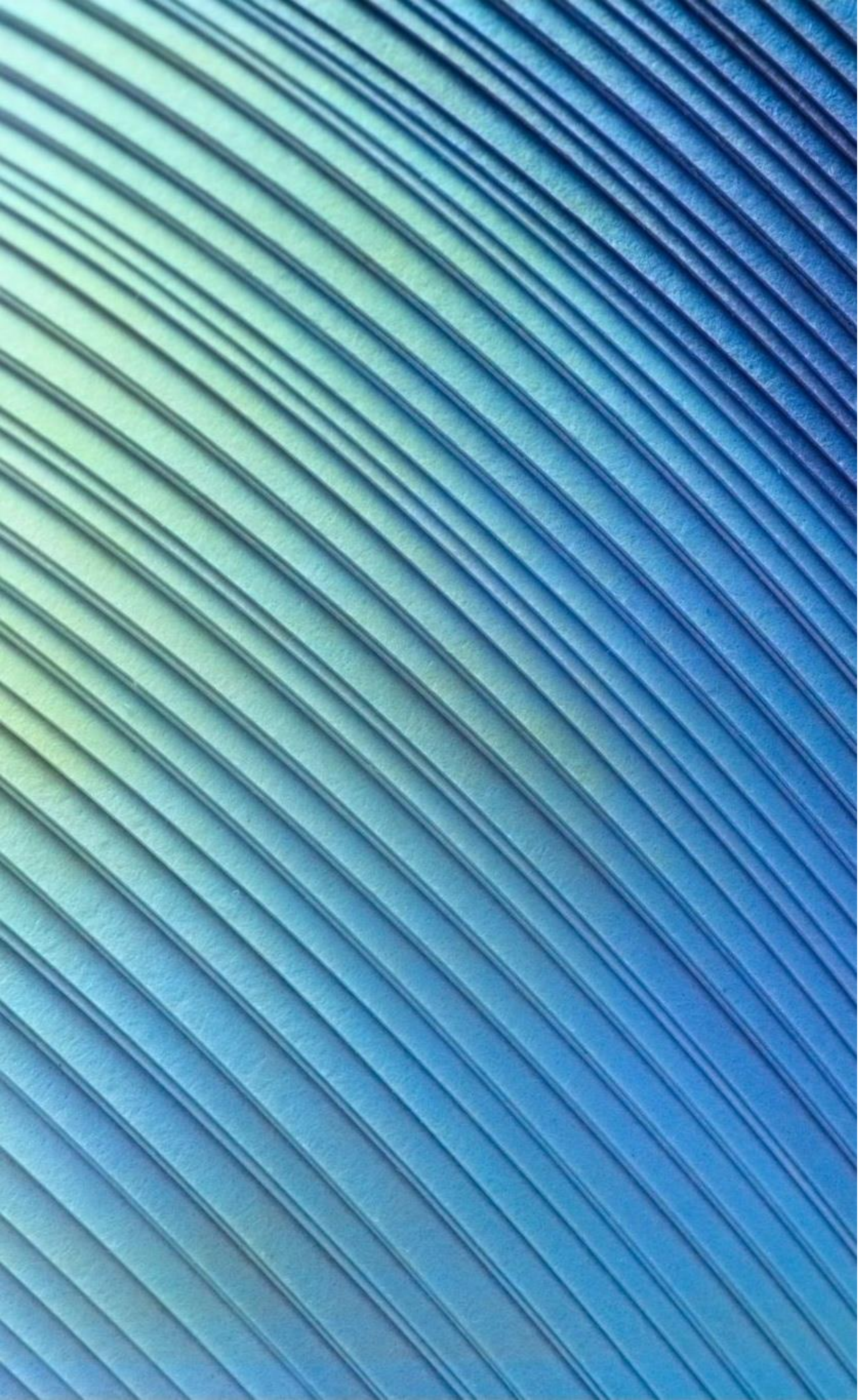


- Thread Discover
- Connect to Thread network
- Ping Thread devices
- Ping Google’s IPv4 DNS
- Ping Google’s IPv6 DNS
- Local IPv4 TCP Portscan
- Portscan Googles’s DNS
- Setup IP Tunnel



ThreadFlipper
Demo

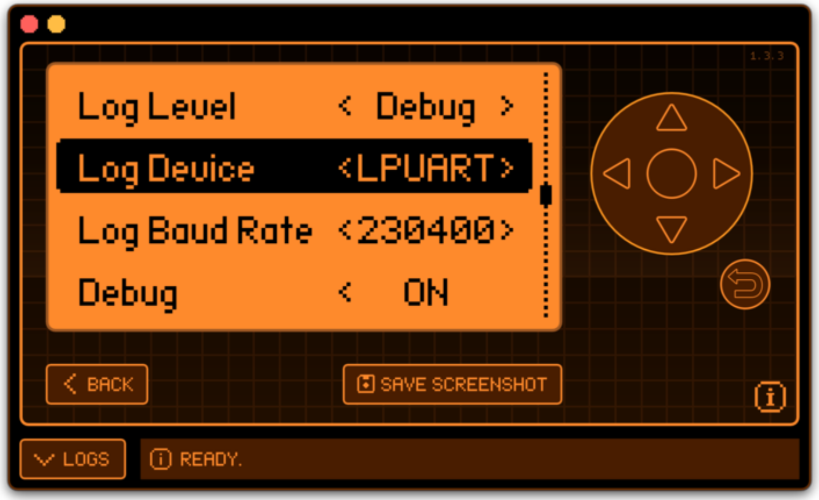




Future work

Native APP	Finish native Flipper Zero app, instead the mJS scripts
SWD	Integrate SWD and support automated flashing of firmware images
NFC	Integrate an NFC antenna .
Protection	Add some protection to the PCB (reverse polarity, voltage regulator, hotplug support)
5V	Use the 5v power from Flipper Zero with a voltage regulator to provide more juice for thread

Challenges



Log Level < Debug >

Log Device <LPUART>

Log Baud Rate <230400>

Debug < ON

- Debugging a Flipper App with a connected Thread board via a WI-Fi extension board is impossible as they use the same UART IO ports. Moving to LPUART will not help, as you will lose the Flipper Logs.
- Jumper Wires can be used to connect just the SWD pins for the WI-FI extension debugger.
- No documentation explains how the esp32 Blackmagic debugger uses the SWD pins.
- Flipper with debug mode enabled is prone to get stuck in a pre-boot breakpoint without a screen.
- Flipper JS uses a lib called mJS (50k JS with 1k RAM); the version I started lacks useful JS functions. The stock firmware did not support features like storage in JS, so we had to use Momentum
- Firmware development with Zephyr is hard, with all possible and conflicting CONFIG parameters.
- Manually set the SEGGER JLink Voltage detection to 3.3V; otherwise, the SWD will fail.
- SEGGER JLink might help to recover from a seemingly bricked flipper (it helped me more then 10x times)
- Adding pins for the SWD port supports JLink SWD debug

So Long, and Thanks for All the Fish!

- András Tevesz
- Linkedin

Appendix

https://docs-be.nordicsemi.com/bundle/ug_nrf52840_dongle/attach/nRF52840_Dongle_User_Guide_v2.1.1.pdf?_LANG=enus https://docs-be.nordicsemi.com/bundle/ps_nrf52840/attach/nRF52840_PS_v1.11.pdf?_LANG=enus	nRF 52840 Dongle guide, leds, pins
https://www.nordicsemi.com/Products/Development-hardware/nrf52840-dongle https://www.nordicsemi.com/Products/Development-hardware/nRF52840-DK	nRF 52840 Dongle and Development Kit sites
https://threadgroup.org https://github.com/openthread/openthread	OpenThread reference
https://flipperzero.one https://docs.flipper.net/development/hardware/modules-blueprints	Flipper Zero development
https://momentum-fw.dev/	Flipper Zero firmware with proper JS support