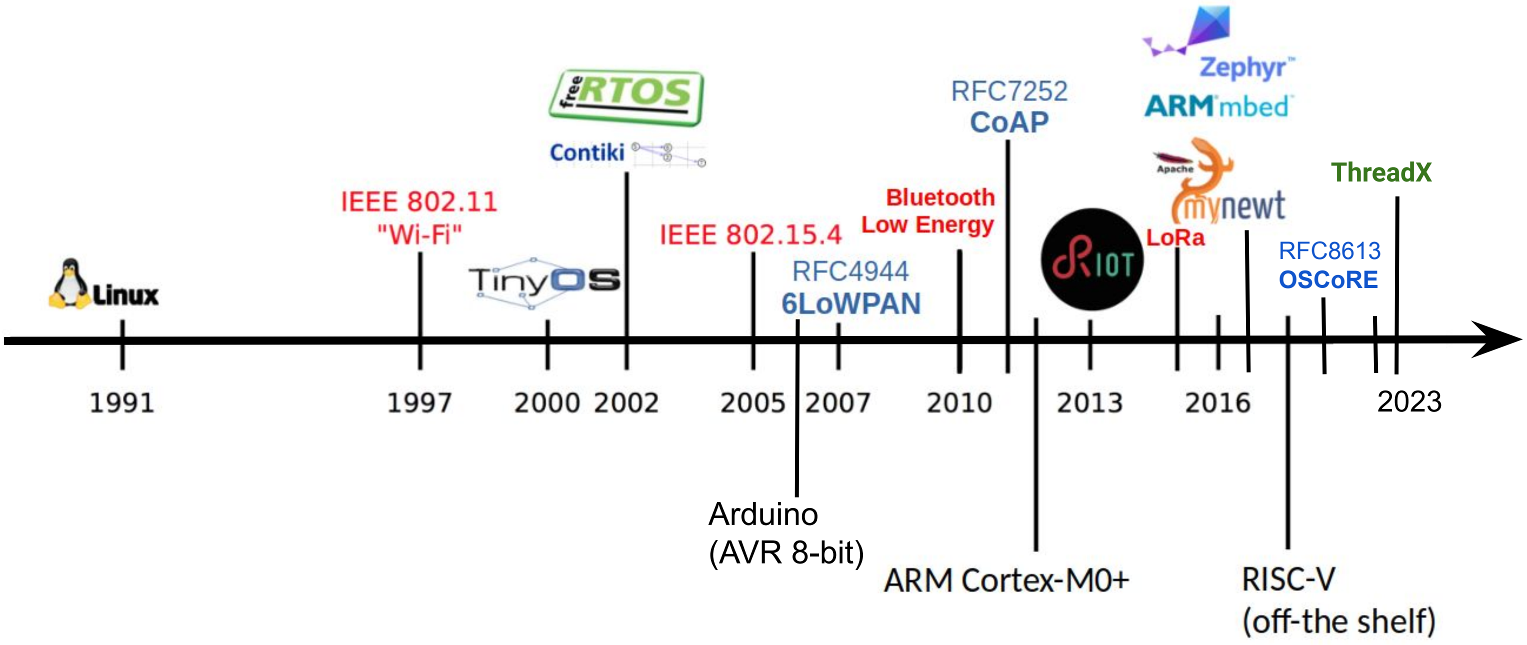
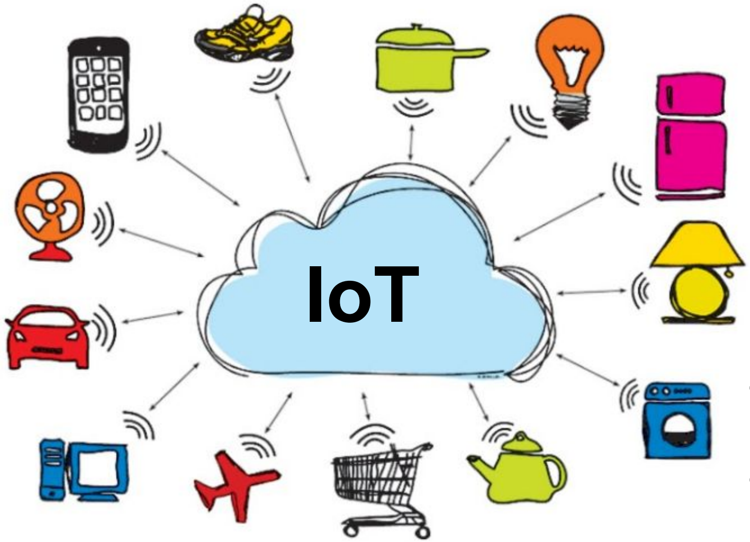
Longer-Term Security for Low-Power IoT Software



Emmanuel Baccelli
Berlin, June 2024

Inria
FREIE UNIVERSITÄT BERLIN
EINSTEIN CENTER Digital Future

IoT is inserted everywhere...
… predictive maintenance, Industry 4.0, smart health, building automation, precision agriculture, AI ...



Clipart: Opentechdiary

Agenda

1. German-French collaboration on RIOT
2. Low-power IoT?
3. Security for Low-power IoT Software Updates
4. Safety for Embedded IoT Software
5. Reformable TinyML

A good example of potential kicking in via German-French collaboration:

What is RIOT?

✓ A general-purpose OS for low-power, microcontroller-based IoT devices
✓ A free embedded software platform & ecosystem
✓ A large community of open source software developers

A good example of potential kicking in via German-French collaboration:

➔ 2013: a research project involving **FU Berlin**, **Inria** and funded by ANR/BMBF

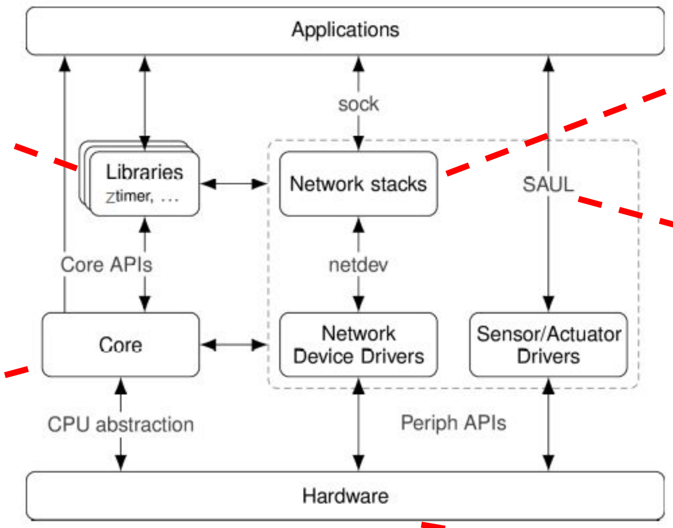
➔ More recent facts & numbers
- 45,000 commits to the master branch, from 350+ developers worldwide
- products shipping RIOT since 2017 (e.g. from Continental)
- 5% market share reached in 2019*

* see Eclipse Foundation, IoT Developer Survey 2019

Longer-Term Security for Low-Power IoT Software

Modular, integrates many libs such as:

- nimBLE
- LVGL
- littleFS
- …



Open network standards: 6LoWPAN, IPv6, RPL, UDP, TCP, CoAP, SUIT...

Unified APIs, across all hardware (vendor & techno. independence)

Micro-kernel

- Multi-threading
- Real-time capable & priority-based scheduling

Minimal config.
2,6kB RAM
3,2kB Flash
(on Cortex-M)

Supported hardware:

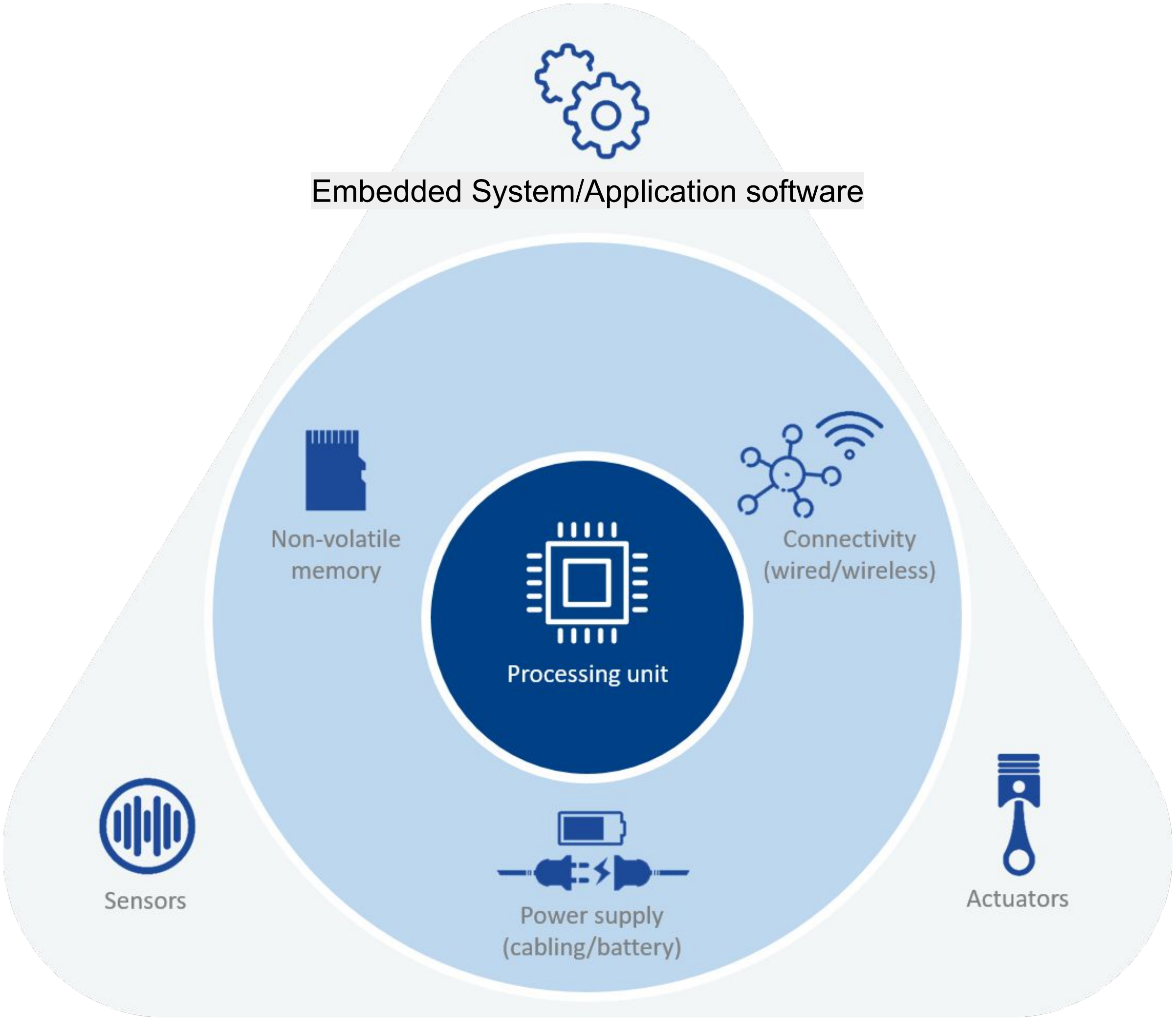
- all types of microcontrollers
- tons of sensors/actuators
- 300+ boards & devkits

Open source code & community:
see github.com/RIOT-OS/RIOT and forum.riot-os.org
Academic publication:
E. Baccelli, et al. *'RIOT: an Open Source Operating System for Low-end Embedded Devices in the IoT,'* IEEE Internet of Things Journal, 2018.

Inria | FREIE UNIVERSITÄT BERLIN | EINSTEIN CENTER Digital Future

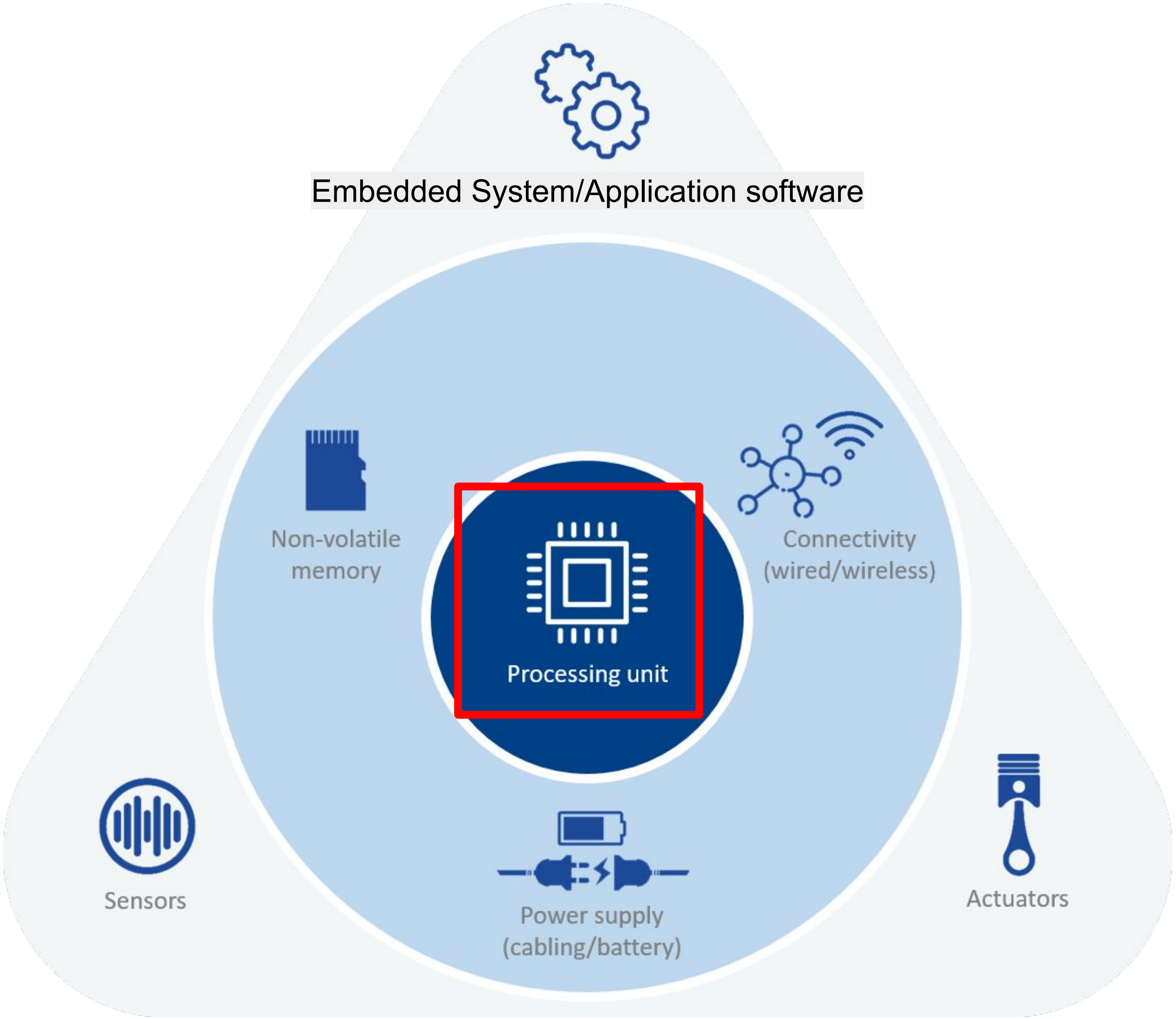
Agenda

Anatomy of a Low-Power IoT Device



Source: 2018 Enisa Summer school

Anatomy of a Low-Power IoT Device



Source: 2018 Enisa Summer school

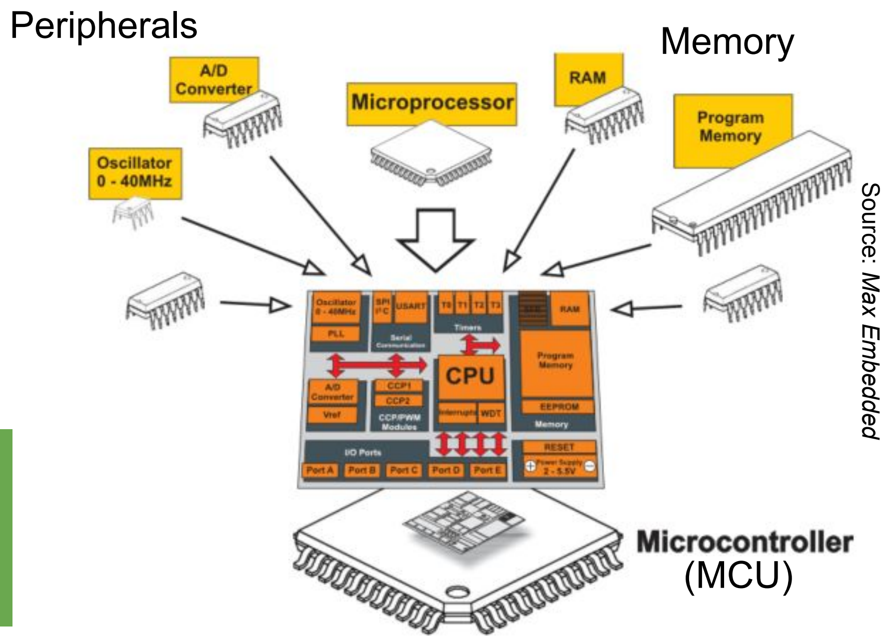
Low-Power ? Microcontroller !

Microcontrollers:

- ➔ milliWatt
- ➔ kiloBytes
- ➔ megaHertz

Compared to processors in “high-end” IoT (phone, RasPi...):

- ➔ much less capacity in computing, networking, memory;
- ➔ much smaller energy consumption & tiny price tag (<1$).

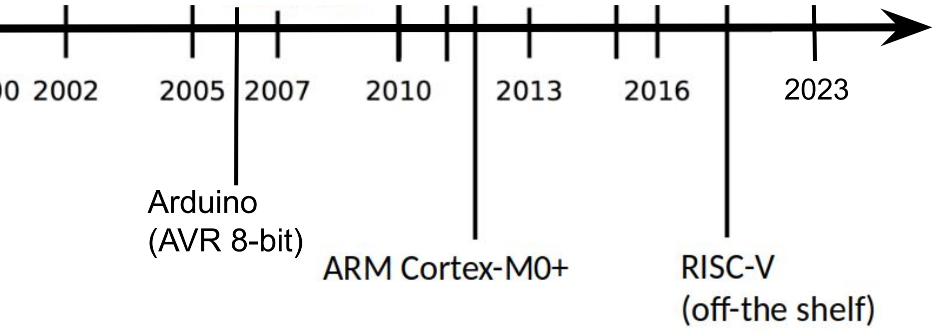


Some stats:

- 28 billion MCU shipped in 2018
- 250 billion microcontrollers used worldwide in 2020

Source: venturebeat.com/2020/01/11/why-tinyml-is-a-giant-opportunity/

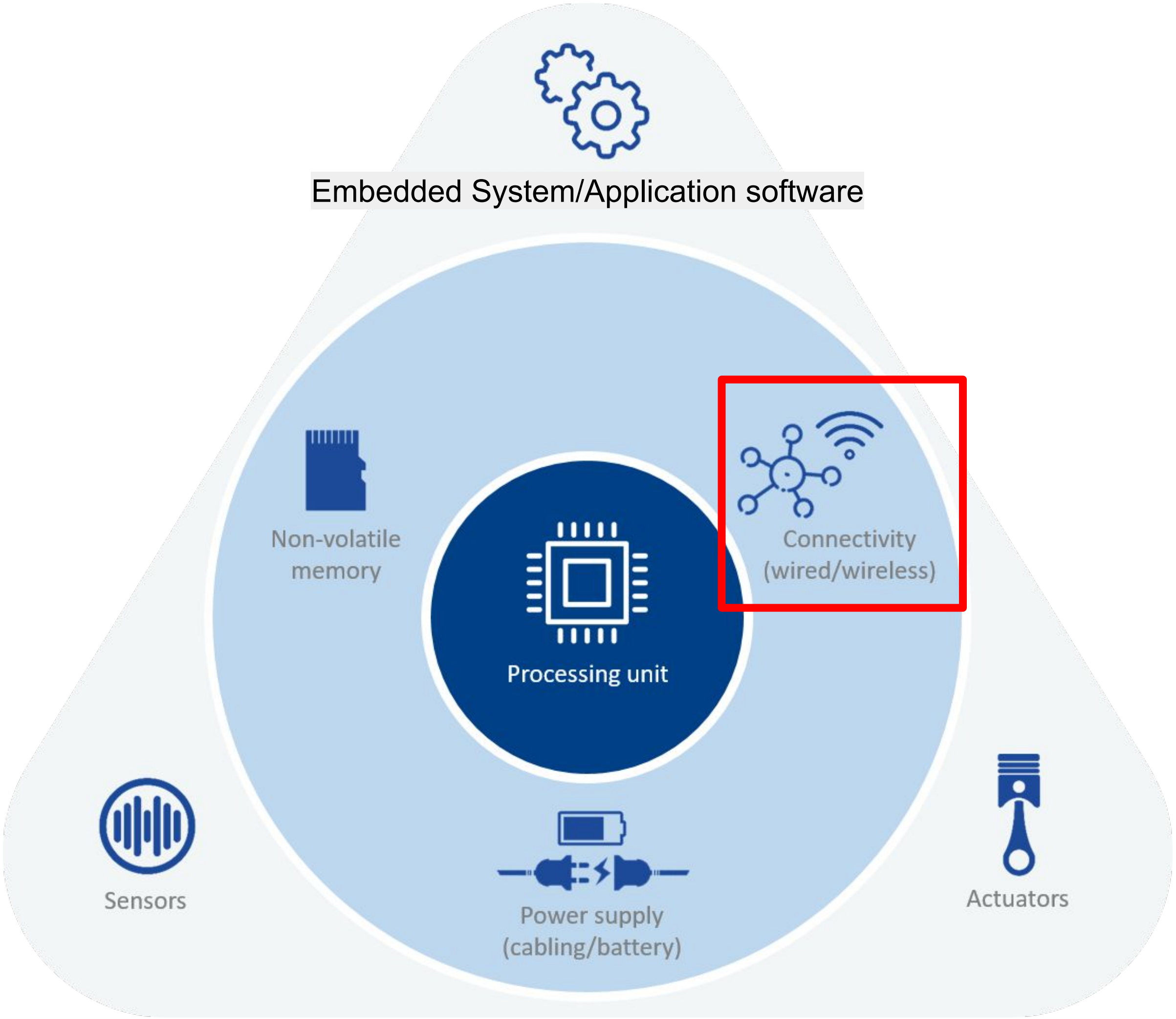
Low-Power Landscape



Low-power Hardware

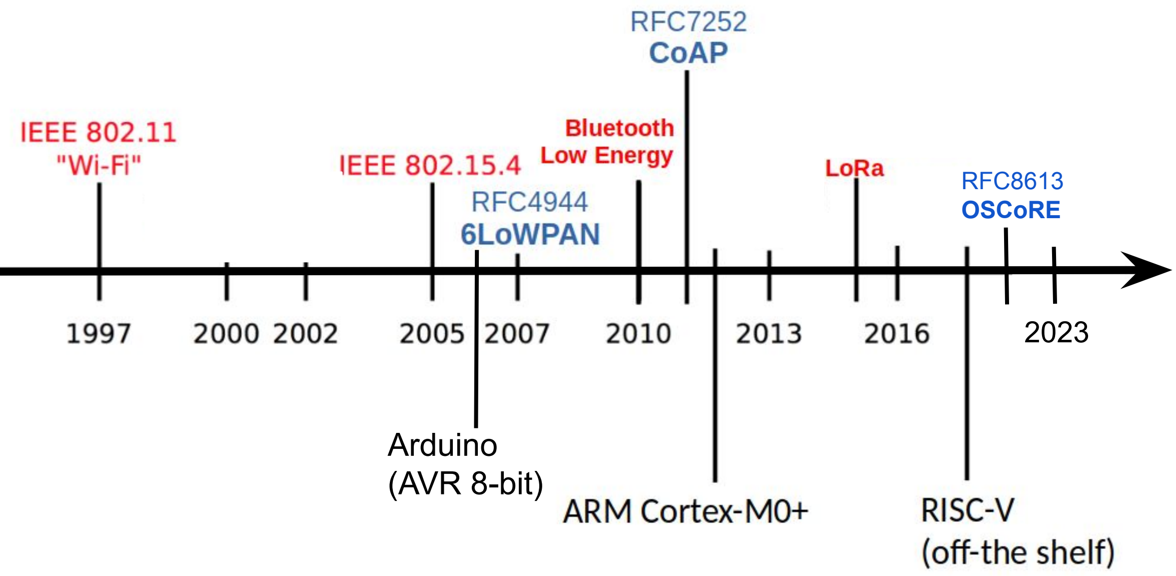
★ Modern 32-bit MCUs: **Arm Cortex-M**, ESP, **RISC-V** (open source HW)...

Anatomy of a Low-Power IoT Device



Source: 2018 Enisa Summer school

Low-Power Landscape



Low-power Hardware

★ Modern 32-bit MCUs: Arm Cortex-M, ESP, RISC-V (open source HW)...

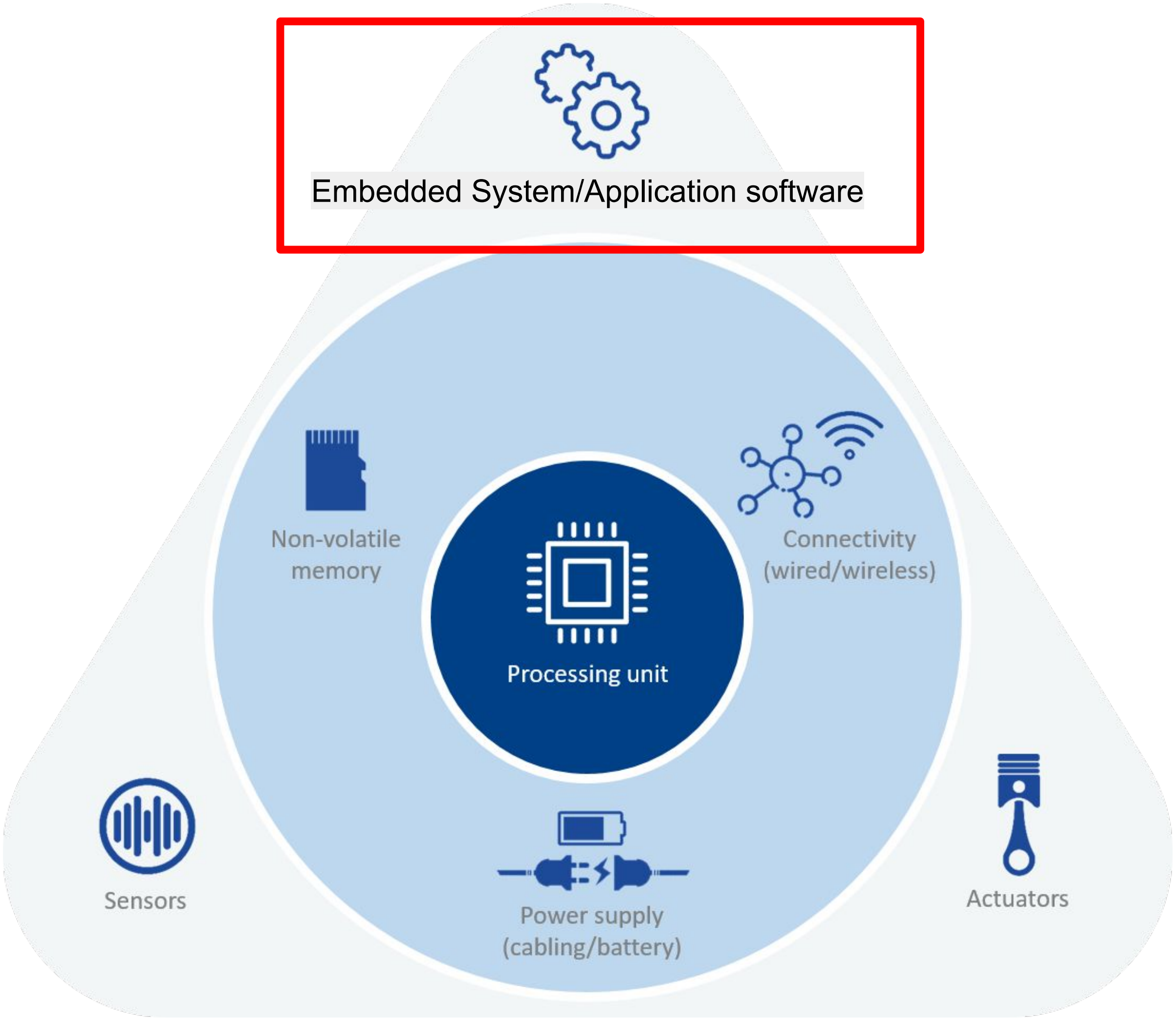
Low-power Wireless Networking

★ Hardware PHY / MAC based on BLE, 802.15.4, LoRa, NB-IoT, (EnOcean)...

★ Internet-compliant protocol stack: **6LoWPAN**, CoAP…

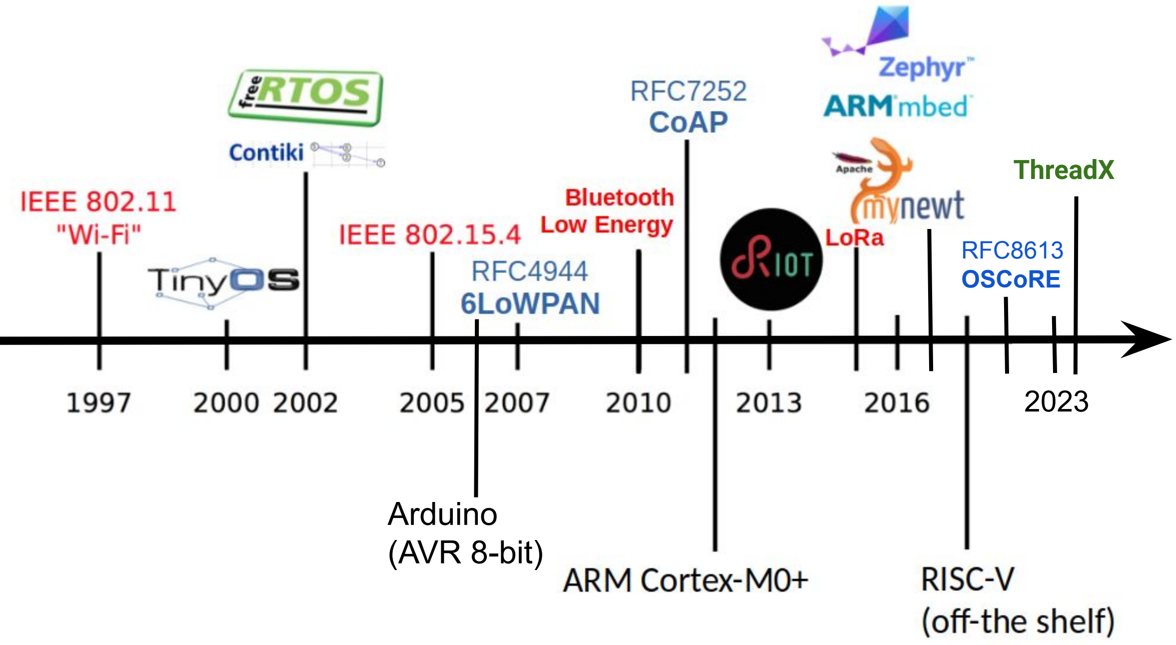
★ Interact with cloud/edge, or local devices

Anatomy of a Low-Power IoT Device



Source: 2018 Enisa Summer school

Low-Power Landscape



Low-power Hardware

★ Modern 32-bit MCUs: Arm Cortex-M, ESP, RISC-V (open source HW)...

Low-power Wireless Networking

★ Hardware PHY / MAC based on BLE, 802.15.4, LoRa, NB-IoT, (EnOcean)...
★ Internet-compliant protocol stack: 6LoWPAN, CoAP…
★ Interact with cloud/edge, or local devices

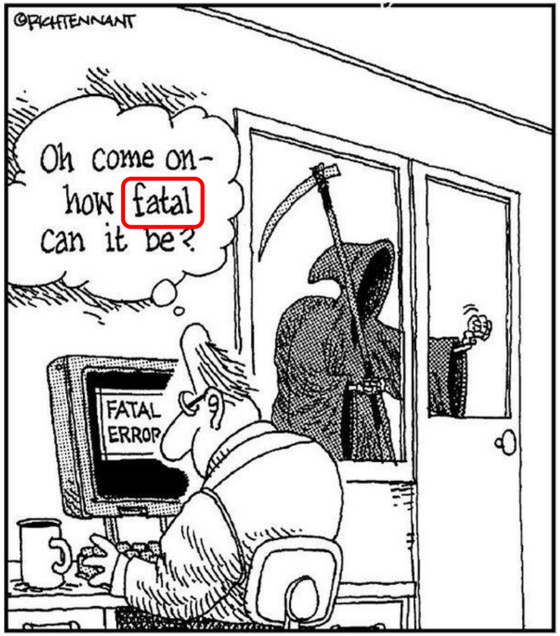
Embedded Software (more & more open source)

★ *Libraries & network stacks*: Eclipse projects, mbedTLS, LVGL, openThread, uTensor…
★ *Operating systems*: **RIOT**, Contiki, mbedOS (Arm), Zephyr (Intel), FreeRTOS (Amazon), ThreadX (Microsoft)

Now… What's the Bottleneck?

Security?

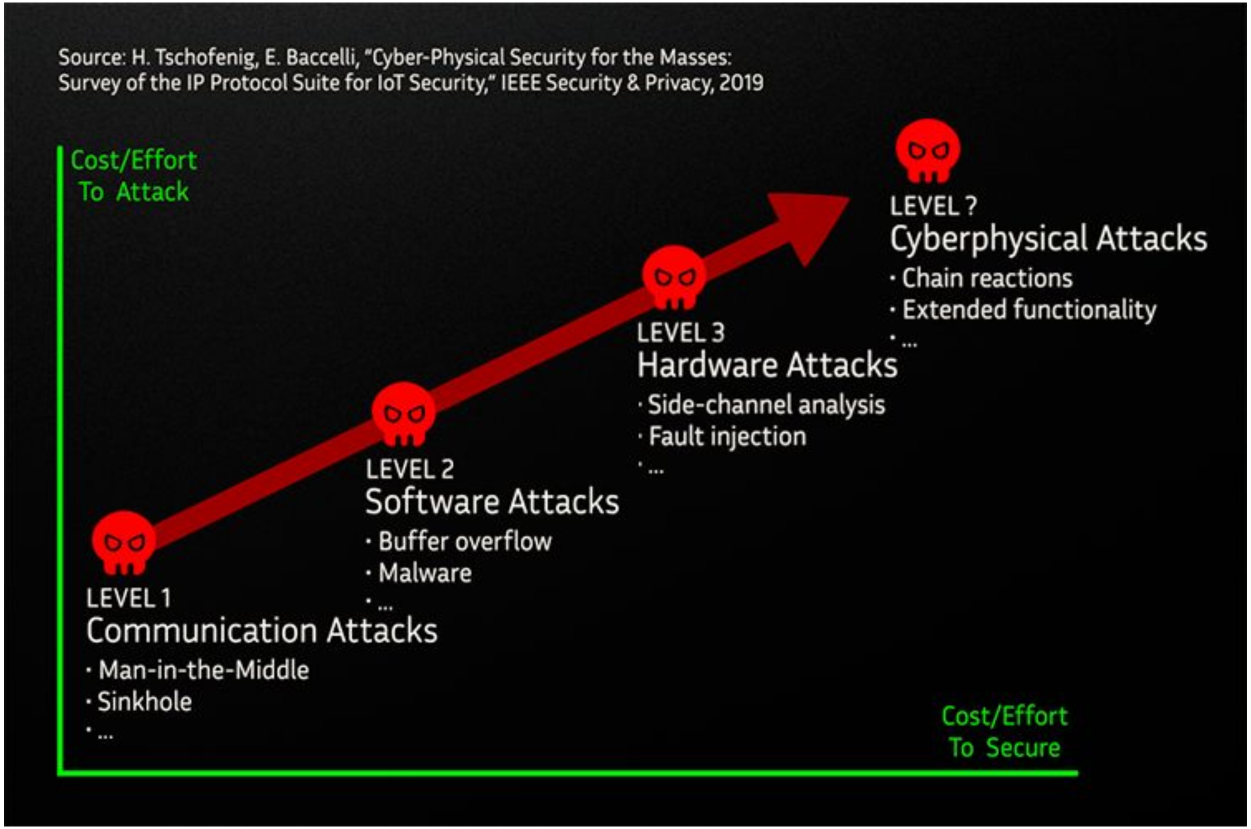
Safety?



Agenda

1. German-French collaboration on RIOT
2. Low-power IoT?
3. Security for Low-power IoT Software Updates
4. Safety for Embedded IoT Software
5. Reformable TinyML

IoT: Cyber-Attack Surface



Our main focus: defend against
- ★ communication attacks;
- ★ software attacks;

Predicates?

1. **You can't secure what you can't update** - but updates are also attack vectors;
2. Software updates happen through the network - else they tend to not happen at all;
3. **Complex software becomes composite**, (tele)maintenance must be distributed.
4. **Formal verification should complement updates**, on critical parts of the software.

Constraints from IoT

- Ultra-small storage on device
- Weak CPU
- Ultra-constrained network transport
- … and more (memory protection, secured boot...)

xda-developers.com

Minimum guarantees on updates

- Authentication
- Integrity
- Authorization
- … and more? (roll-back, pre-conditions...)

Low-Power IoT: The New *Weakest Link*?

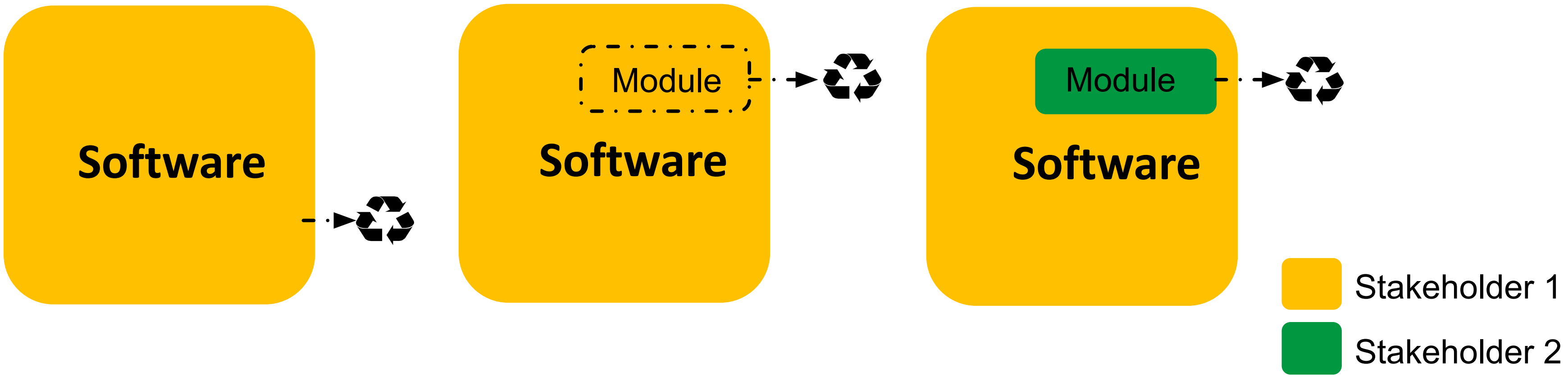
General strategy:

1. Facilitate long-term interoperability? Use (open) standards;
2. Facilitate long-term maintenance? Use open source, collaborative software;
3. Future-proof security level? Post-quantum authentication/authorization (for software updates).

Pain points for low-power IoT:

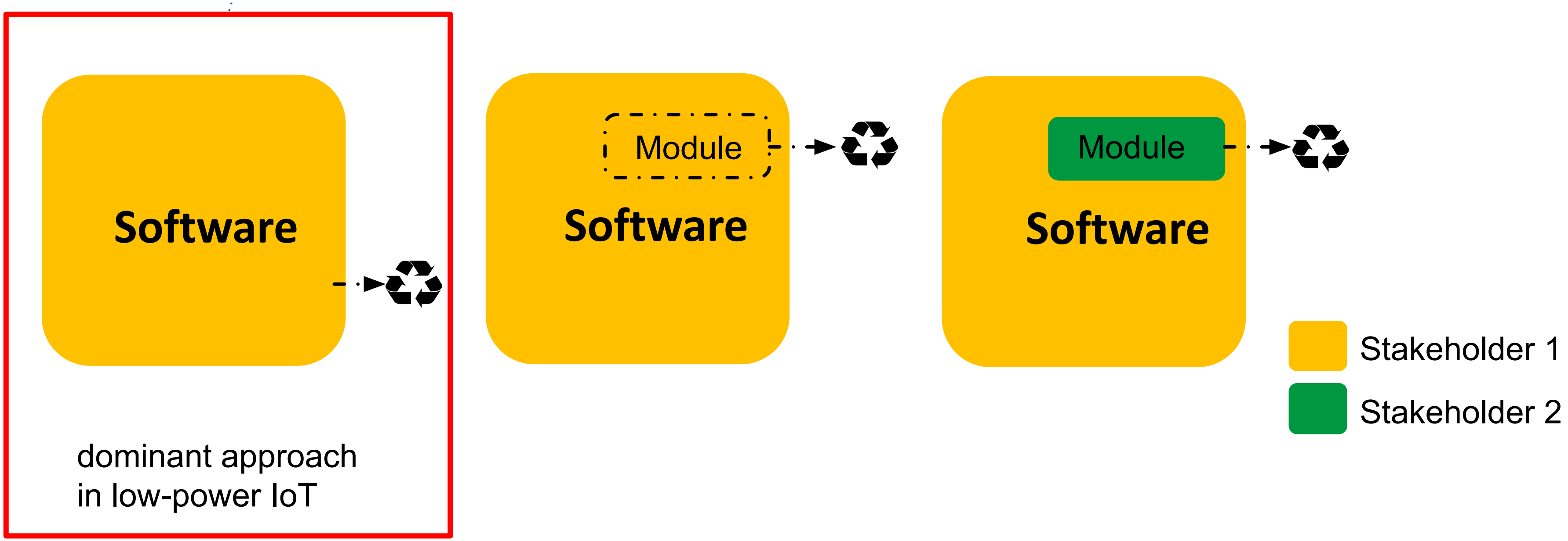
- Challenge 1: Democratizing IoT software updates;
- Challenge 2: Securing modular/multiparty software on low-power devices;
- Challenge 3**:** Safer low-level IoT software.

- Case 1 : monolithic software update, single stakeholder
- Case 2 : modular software updates, single stakeholder
- Case 3 : modular software updates, multiple stakeholders



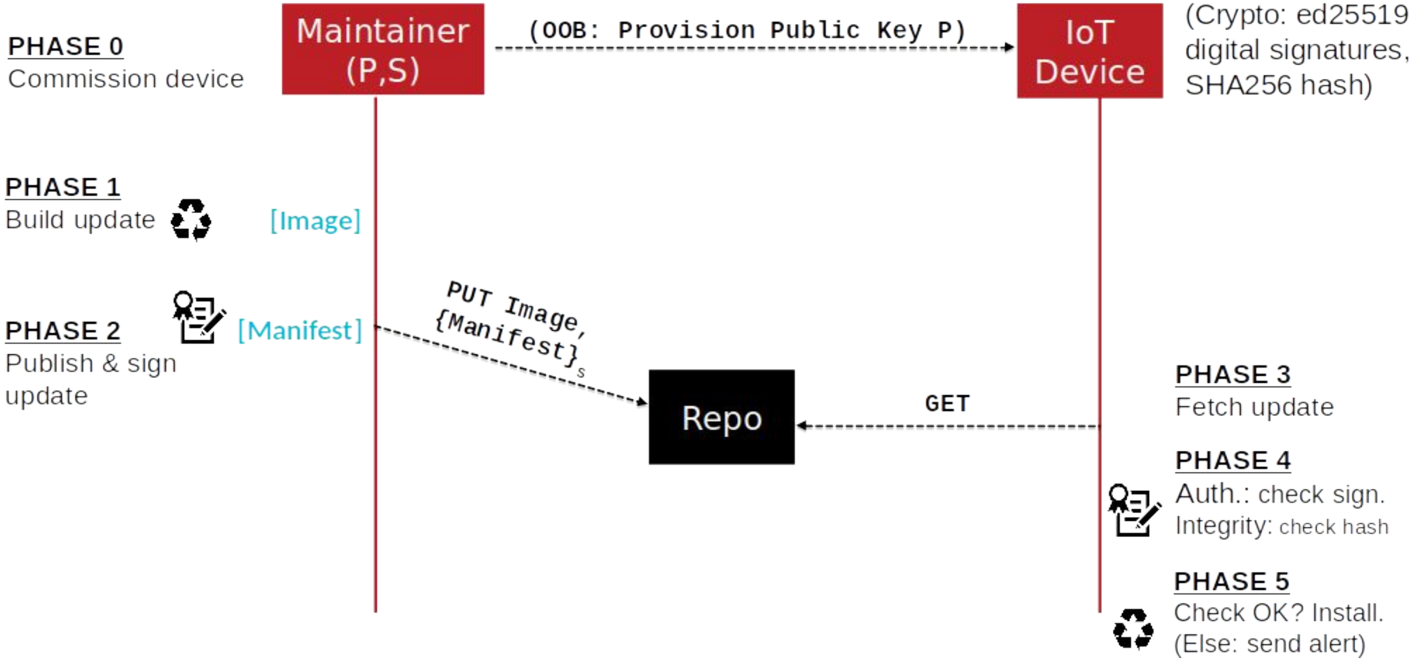
Challenge 1: **Democratize IoT Software Update**

- Case 1 : monolithic software update, single stakeholder
- Case 2 : modular software updates, single stakeholder
- Case 3 : modular software updates, multiple stakeholders



SUIT working group at the IETF

SUIT = new architecture, metadata & serialization for lightweight IoT firmware update security : authentication, integrity checks (and more) specified at IETF, currently in the final stages of standardization: see https://datatracker.ietf.org/wg/suit/about/

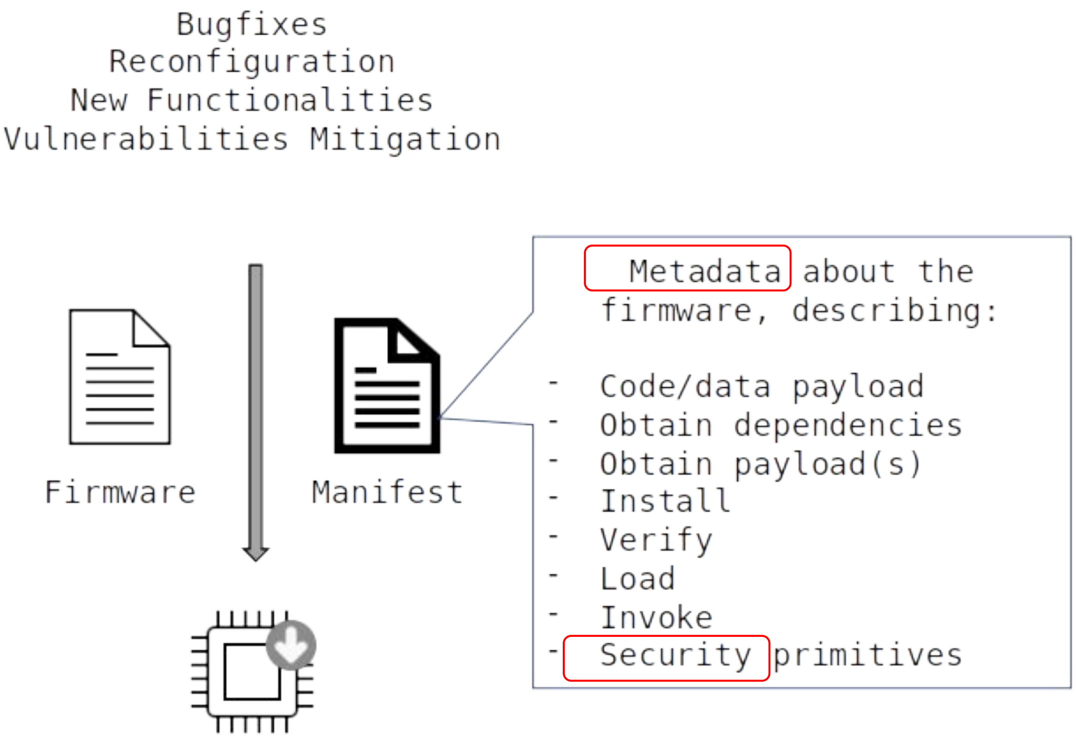


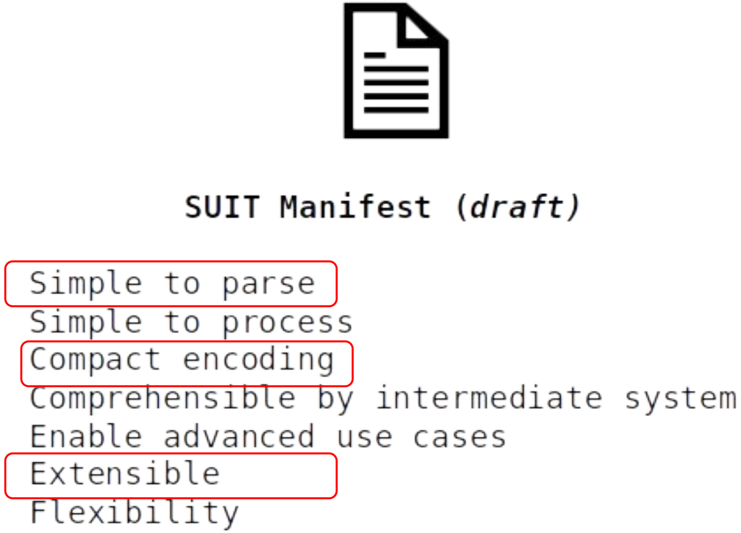
Other primitives can be used too!

SUIT Manifest Specifications

- Latest specs for the SUIT manifest see: B. Moran et al., "*CBOR-based Serialization Format for the SUIT Manifest,*" IETF draft *draft-ietf-suit-manifest-25,* Feb. 2024.

IETF

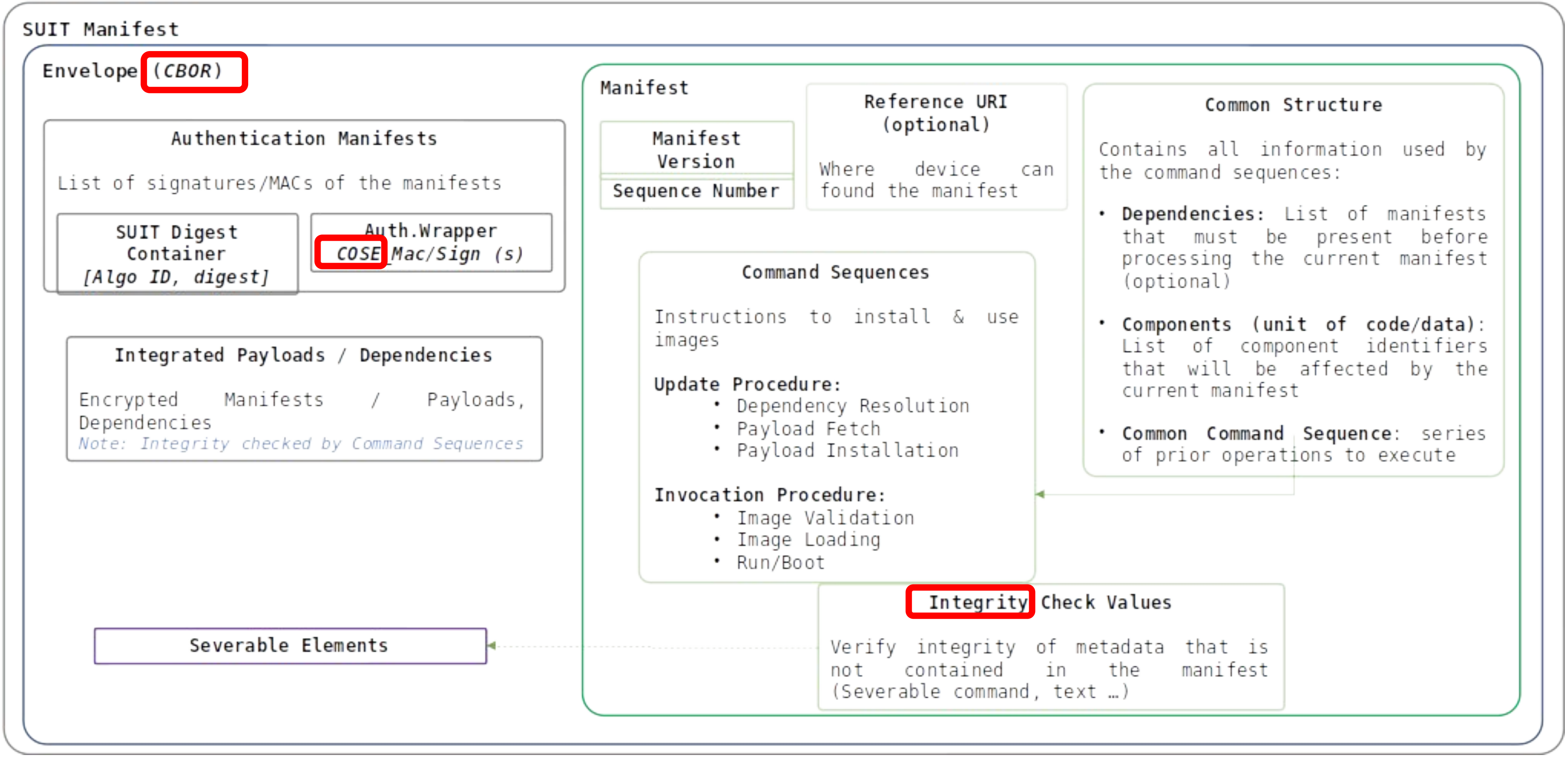




Inria FREIE UNIVERSITÄT BERLIN EINSTEIN CENTER Digital Future

Source: 2022 slide from Interop / Loïc Dalmasso

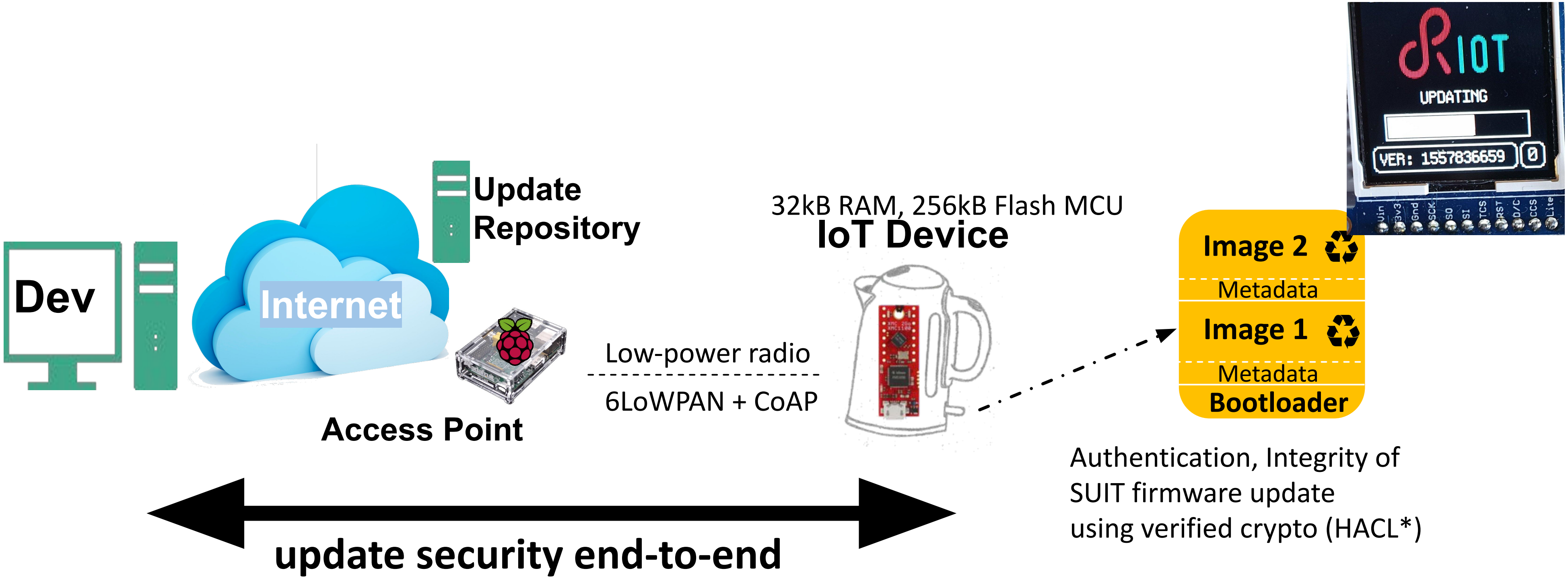
SUIT Metadata Structure (Sketch)



Source: 2022 slide from Interop / Loïc Dalmasso

Open Source Implementation of SUIT

★ Integration in RIOT, see https://github.com/RIOT-OS/RIOT/tree/master/examples/suit_update
★ Support out-of-the-box for ~150 boards (and ~10^5 software configs)



Experimental Evaluations of SUIT

Studies of SUIT performance for pre-quantum [1] and post-quantum [2]

★ in [2] evaluation of cost of security level upgrade
  ○ from pre-quantum 128-bit security (with ed25519 or p-256)
  ○ to NIST Level 1 post-quantum security (with Falcon, Dilithium or HSS-LMS)

Benchmarks:

★ using different 32-bit microcontrollers: ARM Cortex-M, RISC-V, ESP32
★ using different families of PQ crypto (lattice- and hash-based)
★ software update workflow => focus is *not* signature generation

Table 7: Relative cost increase for SUIT with quantum resistance (on ARM Cortex M-4).

SUIT	Flash	Stack	Transfer	Transfer w. crypto
base w. Ed25519 / SHA256	52.4kB	16.3kB	47kB	53kB
with Falcon / SHA3-256	+120%	+18%	+1.1%	+120%
with LMS / SHA3-256	+34%	+1.2%	+9%	+43%
with Dilithium / SHA3-256	+30%	+210%	+4.3%	+34%

[1] K. Zandberg et al. Secure firmware updates for constrained IoT devices using open standards: A reality check, in IEEE Access, Sept. 2019.
[2] G. Banegas et al. Quantum-Resistant Security for Software Updates on Low-power Networked Embedded Devices, in ACNS, June 2022.

Next Steps with SUIT Post-quantum

PQ-OTA is an R&D project about to start on this topic

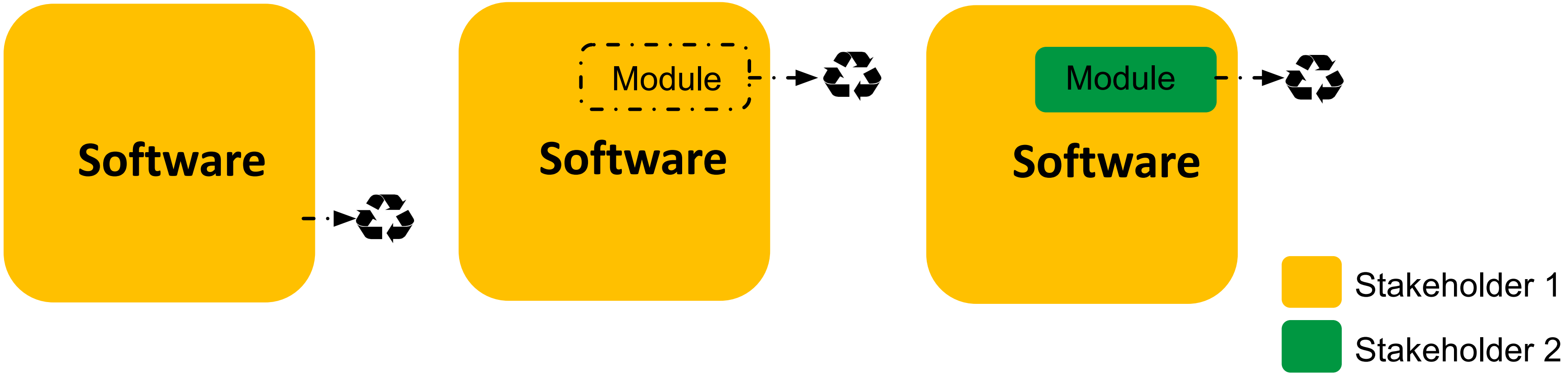
- ✓ **Collaboration between Inria, FU Berlin and Continental**
- ✓ Continental telematics products use RIOT + software updates
- ✓ Use of SUIT + optimizations for post-quantum + multi-core MCUs
- ✓ Hosted at CampusCyber, financed by PTCC / ANR

CAMPUS CYBER

Continental

Challenge 2: **Multiparty Software Updates**

- Case 1 : monolithic software update, single stakeholder
- Case 2 : modular software updates, single stakeholder
- Case 3 : modular software updates, multiple stakeholders



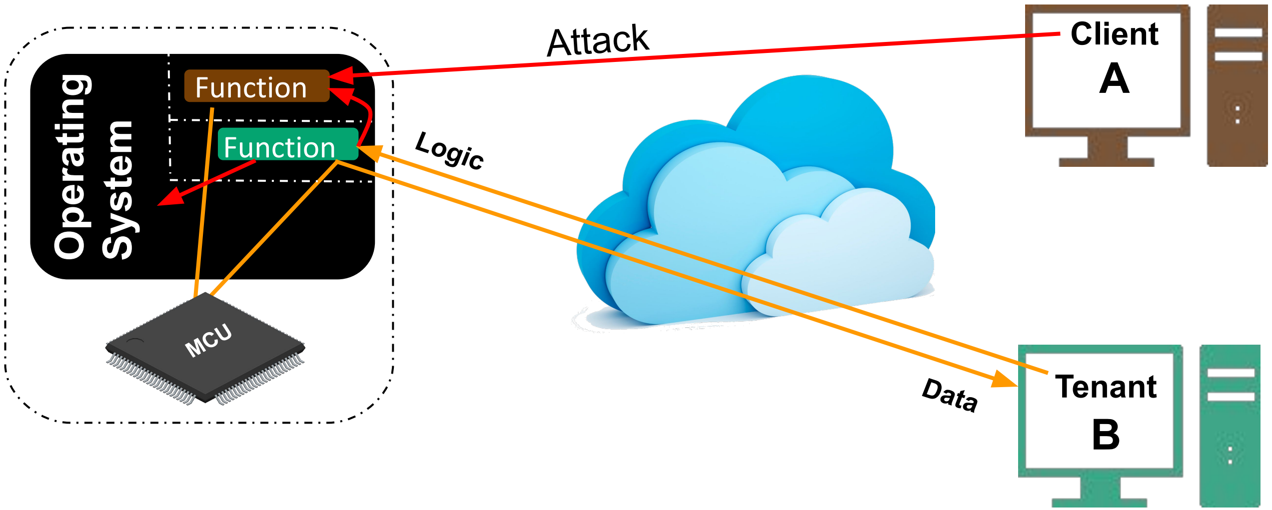
Challenge 2: **Multiparty Software Updates**

Goal: **we want to modify deployed software,** on-the-fly

- Hosting additional functions
- Hosting debug/monitoring snippets

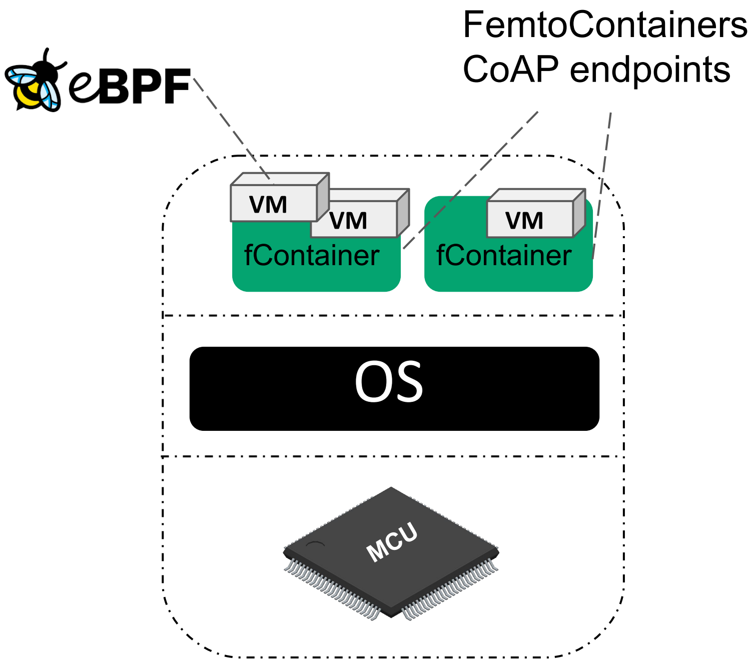
Threat model: **we want function fault-isolation**, to protect against

- Malicious tenants: Escape the sandbox?
- Malicious clients: Install-time attacks?



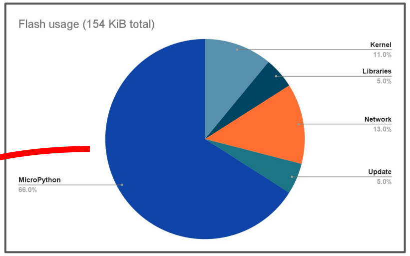
Femto-Containers

- **Ultra-lightweight virtualization: rBPF** [4]
  - register-based VMs with eBPF instruction set to microcontrollers
- Real-Time OS (RTOS) syscalls
  - Allows & controls sensor interaction, network services
  - Reference implementation in RIOT
- Remote over-the-air (OTA) management
  - Femto-container(s) exposed as CoAP resources
  - SUIT-compliant software updates of containerized microservices
- Femto-Container **hosting engine = only 1000 LoC (!)**
  - allowed **formal verification** [5] for fault-isolation



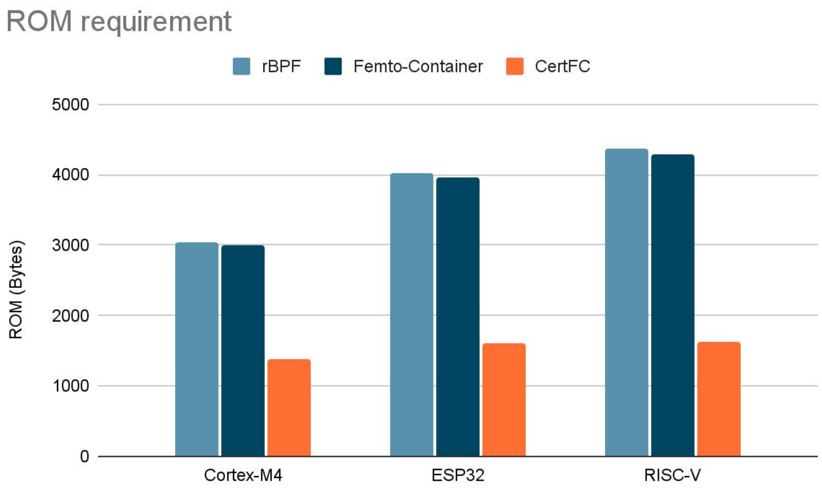
[4] K. Zandberg et al. Minimal Virtual Machines on IoT Microcontrollers: The case of Berkeley Packet Filters with rBPF, in PEMWN, 2020.
[5] S. Yuan et al End-to-end Mechanized Proof of an eBPF Virtual Machine for Microcontrollers, in CAV, Aug. 2022

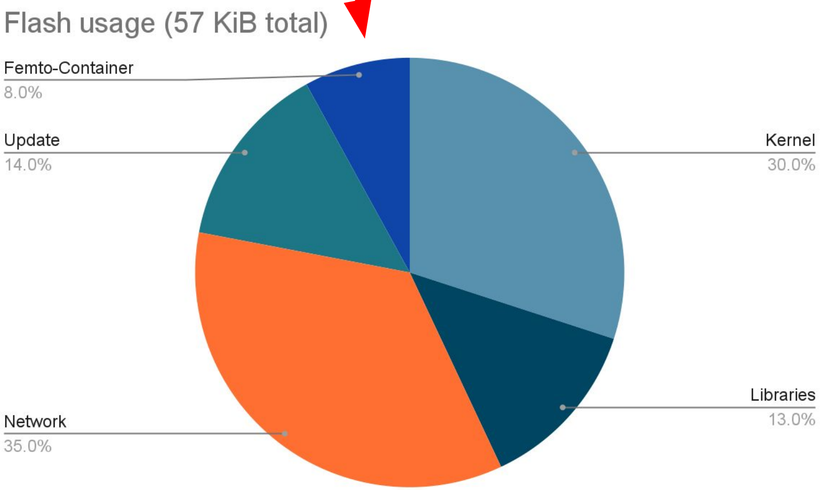
Experiments with Femto-Container



Performance study [6] on Cortex-M, ESP32, RISC-V

=> Compared to native exec., memory overhead is 10% or less!



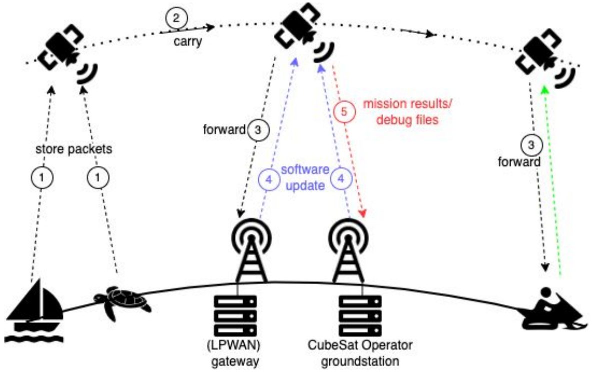


[6] K. Zandberg et al. [Femto-Containers: Lightweight Virtualization and Fault Isolation For Small Software Functions on Low-Power IoT Microcontrollers](), in ACM MIDDLEWARE, Nov. 2022

Next steps with Femto-Containers

ThingSat an R&D project we joined recently

✓ Collaboration between Université Grenoble-Alpes, Inria, FU Berlin
✓ Development of Femto-Containers applications for nanosatellites
✓ Extending recent work described in [7] using SUIT and RIOT



[7] F. Molina et al. Cubedate: Securing Software Updates in Orbit for Low-Power Payloads Hosted on CubeSats, in IEEE PEMWN, Sept. 2023

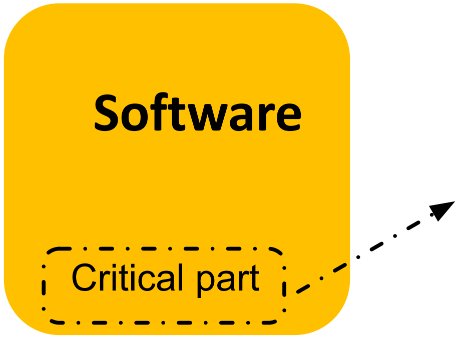
Agenda

1. German-French collaboration on RIOT
2. Low-power IoT?
3. Security for Low-power IoT Software Updates
4. Safety for Embedded IoT Software
5. Reformable TinyML

Challenge 3: **Safety for Critical Perimeter**

Updates can be delivered, authenticity/integrity/authorization can be checked. Now what?

➔ **Still require more safety on selected (critical!) parts of the embedded software**



Guarantees within a critical perimeter? e.g.

★ does not panic?
★ fault-isolated?
★ deeper (functional) guarantees?

Limits with C

So far embedded software (incl. RIOT) is written in C…

… but we were hitting limits w.r.t. safety with C

- Making mem protection + MPU first class citizens
- Providing configuration(s) with "defensive" code
- Catching errors: Graceful shutdown / restart of threads

…

… as well as other *limits with C* abstractions and tooling

Enter Rust

The “new” kid on the block, challenging C…

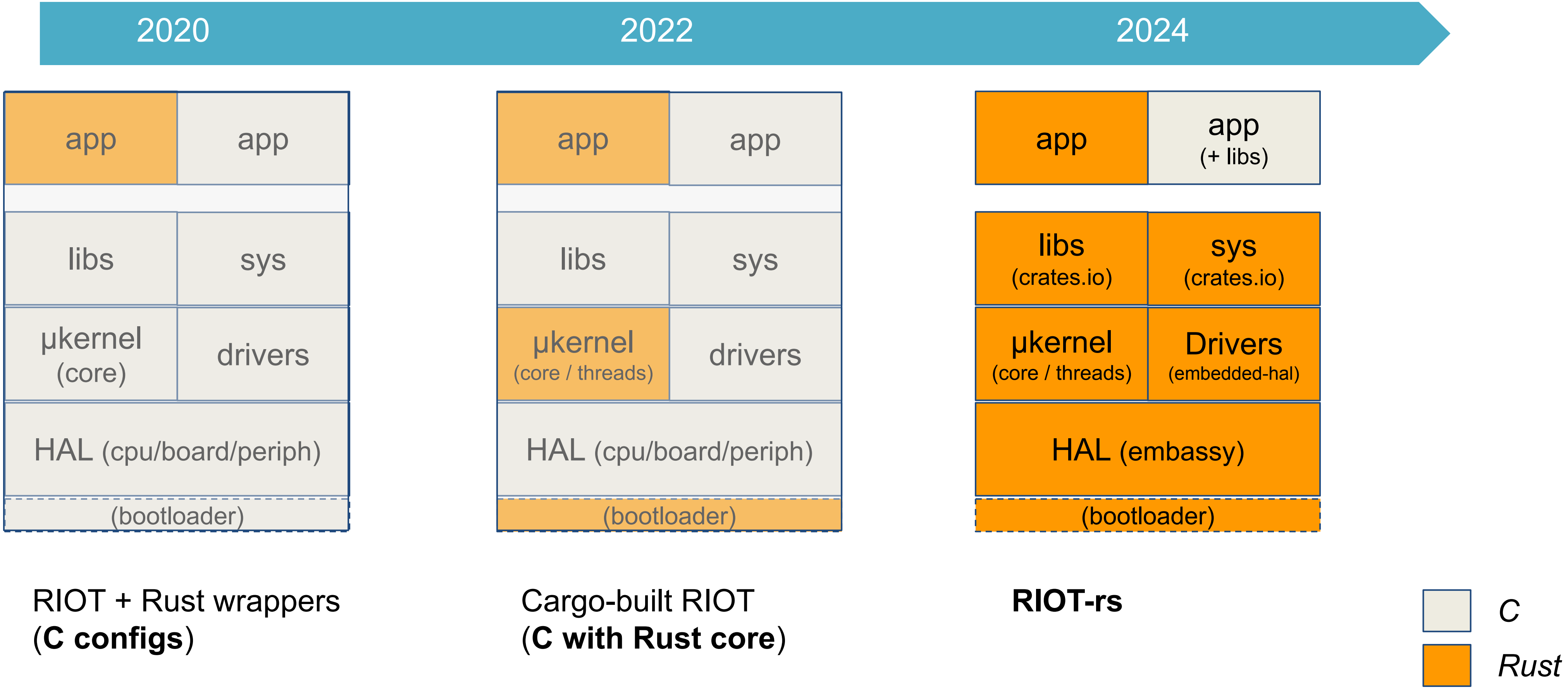
… with a different trade-off combining:

- High-level ergonomics;
- **Built-in memory safety;**
- Low-level control;

With modern tooling (build with *cargo*, import *crates*)...

Recent Rust rant: see this post on Google Open Source Blog

Next steps with Rust in RIOT



Rust + Formal Verification with RIOT

RIOT-rs is a full transition from a primarily C-base to primarily Rust-base [10]

- ★ blueprint for smooth ride, with compatibility for high-level RIOT APIs
- ★ leveraging the best of RIOT, async Rust + embedded Rust prior work
- ★ a lot of work to be done still to match richer RIOT-C features

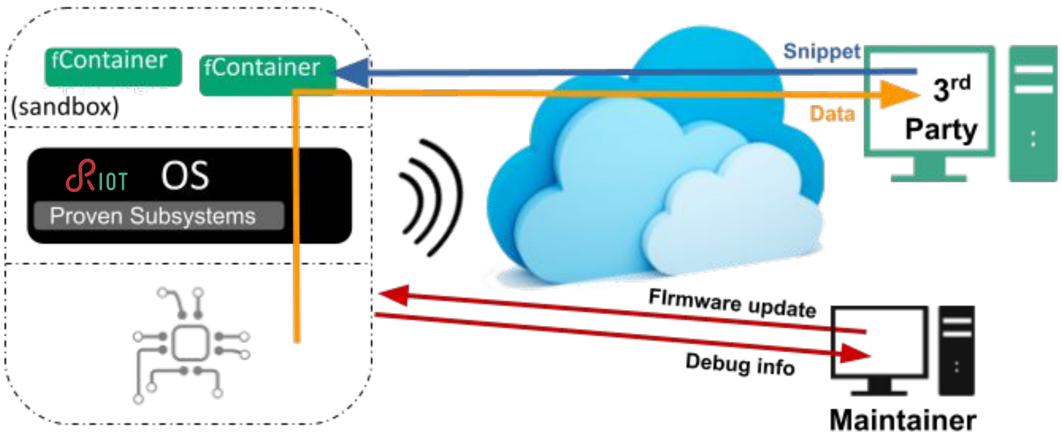
PTCC project collab. with Cryspen & PROSECCO to refine+integrate the new formal verification tool **hax** [8] [9] in RIOT-rs continuous integration workflow:

- ★ hax takes as input (functional) embedded Rust, and outputs Coq, F*...
- ★ automated proofs (panic-freedom) or deeper proofs (functional correctness)
- ★ started applying hax to RIOT-rs modules (e.g. runQueue, EDHOC-rs)

[8] **hax** open source repository https://github.com/hacspec/hax
[9] D. Merigoux et al. Hacspec, Tech. Report, 2021
[10] **RIOT-rs** open source repository https://github.com/future-proof-iot/RIOT-rs

Research

Larger project on next-level **cybersecurity for IoT** software on ultra-low power devices.



Objectives

1. **Open ecosystem+platform**, roughly equivalent to the Linux ecosystem;
2. **Small+safe OS perimeter**, roughly equivalent to the seL4 kernel;
3. **Quantum-resistant** cybersecurity;
4. **Modern+secure DevOps**, as “easy as Amazon Lambda” over low-power networks.

Output

Software: upstream/maintenance of 15+ open source repositories, including RIOT & RIOT-rs;
Standards: 50+ standardization docs at IETF;
Publications: 30+ articles in journals, conferences and preprints;

RIOT-fp Useful Info

RIOT-fp participants include Shenghao Yuan, Gustavo Banegas, Koen Zandberg, Timothy Claeys, Malisa Vucinic, Frederic Besson, JP Talpin, **Benjamin Smith**, Emmanuel Baccelli, Kaspar Schleiser, Francisco Molina, Alexandre Abadie, **Karthik Bhargavan**, Denis Merigoux, Geovane Fedrecheski, Thomas Watteyne

Teams involved: TRiBE, AIO, GRACE, TEA, EPICURE, PROSECCO **and FU Berlin**

Website :
https://future-proof-iot.github.io/RIOT-fp/
including full publication list at https://future-proof-iot.github.io/RIOT-fp/publications

Code :
https://github.com/future-proof-iot
including also contribs to the RIOT code base at https://github.com/RIOT-OS/RIOT

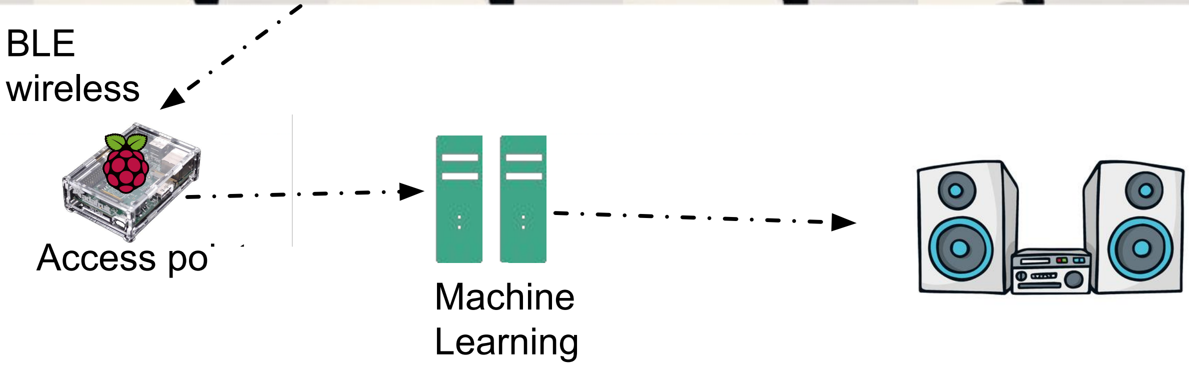
Agenda

1. German-French collaboration on RIOT
2. Low-power IoT?
3. Security for Low-power IoT Software Updates
4. Safety for Embedded IoT Software
5. Reformable TinyML

AI + IoT with RIOT

RIOT wearables with Bluetooth Low-Energy used for the VKKO project, a smart conductor vest [11] collab. with Berit Greinke (UdK) & Felix Biessmann (BHT)

No.	Location	Type
1 (ch0)	Right elbow (out)	Stretch sensor strap
2 (ch1)	Back (horizontal)	
3 (ch2)	Right shoulder	Resistive knitted tube
4 (ch3)	Left armpit	Pressure sensor foam
5 (ch4)	Left shoulder	
6 (ch5)	Right elbow (in)	Resistive knitted tube



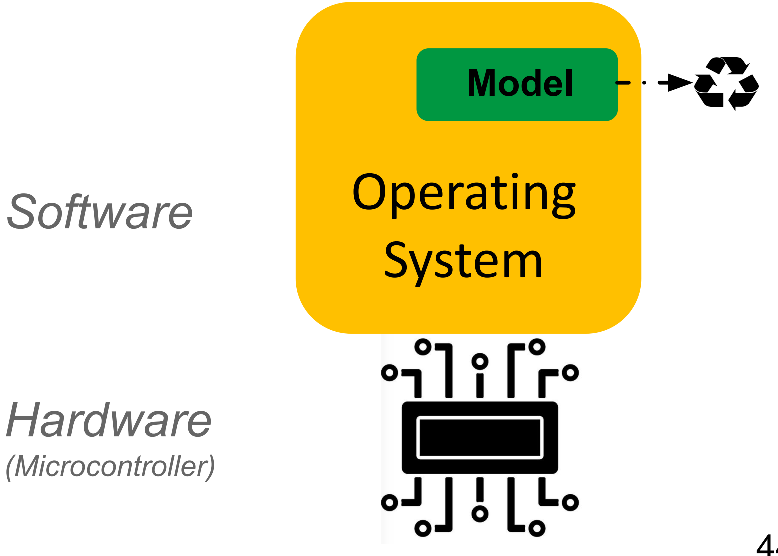
[11] B. Greinke, Berit et al. "An Interactive Garment for Orchestra Conducting: IoT-enabled Textile & Machine Learning to Direct Musical Performance." in ACM TEI 2021

Some software module embedded on microcontrollers can implement machine learning!

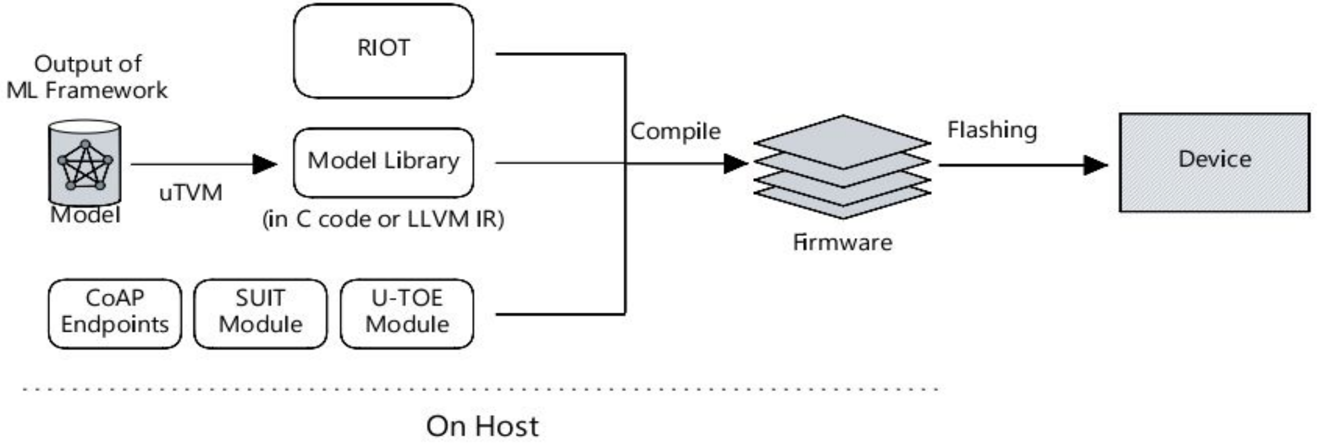
TinyML: intersection of AI and Internet of Things (AIoT)

- Various machine learning models on diverse microcontrollers!
- Need for experimental toolkits for evaluation

Reformable TinyML: Continuous deployment + TinyML

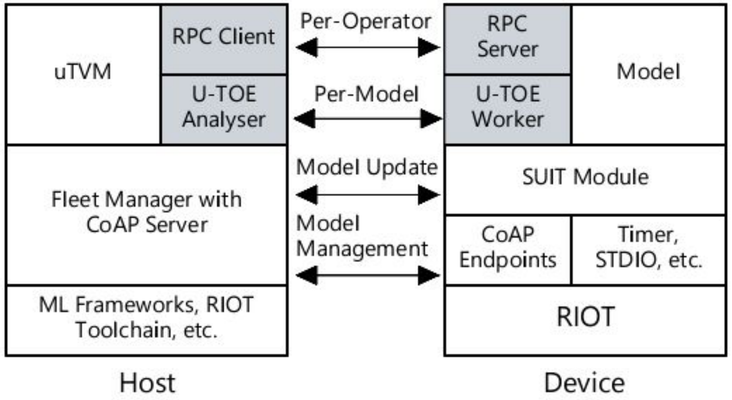


Experimental Toolkit for TinyML



RIOT-ML: toolkit for model benchmark at design stage and secure update workflow at maintenance stage [12].

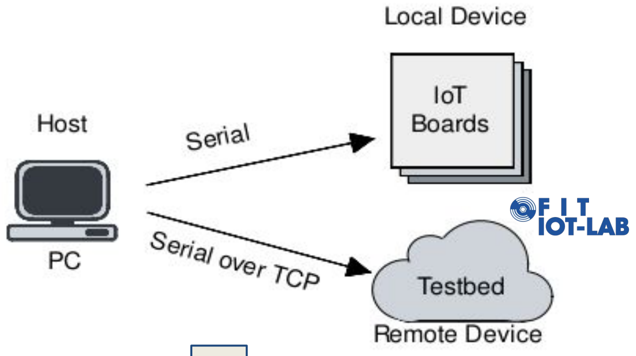
➔ Support out of the box for most RIOT hardware!
➔ Various granularity: Model >> Operator



[12] Z. Huang, et al. "RIOT-ML: toolkit for over-the-air secure updates and performance evaluation of TinyML models." *Annals of Telecommunications* (2024): 1-15.

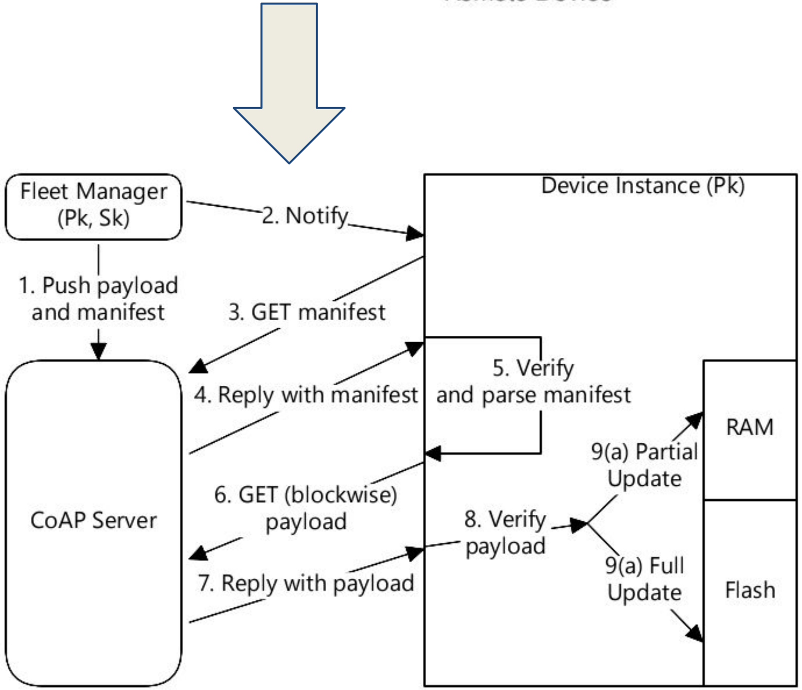
RIOT-ML Toolkit Architecture

Bench. & Management: Local >> Remote
(FIT IoT-LAB testbed)



Model Continuous Deployment

- Secure over-the-air CI/CD via SUIT
- Transport using CoAP/6LoWPAN
- Firmware or partial model update



Benchmarking results with RIOT-ML

Benchmarks uncover outliers!

Evaluation of various models on stm32f746-disco board.

Model	Task	Memory	Storage	Latency
DS-CNN Small	Keyword Spotting	68.992	71.796	461.396
MobileNetV1-0.25x	Visual Wake Words	185.352	491.668	1435.938
LeNet-5	Image Classification	12.068	65.851	39.601
Deep AutoEncoder	Anomaly Detection	6.532	292.696	35.638
RNNoise	Noise Suppression	4.688	119.652	12.154

Memory and storage consumption in KB, computational latency in ms.

 # Parameters: DS-CNN ~22K, MobileNet ~500K, LeNet-5 ~40K, Deep AutoEnc. ~264K, RNN. ~87K

Evaluation results of LeNet5 on various IoT boards.

Board	MCU Core	Memory	Storage	Latency
arduino-zero	M0+ @ 48 MHz	11.292	64.940	182.068
rpi-pico	M0+ @ 125 MHz	28.704	65.172	70.117
openmote-b	M3 @ 32 MHz	11.100	66.080	200.367
IoT-LAB M3	M3 @ 72 MHz	11.296	62.260	97.751
nucleo-wl55jc	M4 @ 48 MHz	11.288	63.180	98.661
nrf52840dk	M4 @ 64 MHz	11.348	61.332	66.088
b-l475e-iot01a	M4 @ 80 MHz	11.288	61.604	52.901
stm32f746g-disco	M7 @ 216 MHz	11.076	64.712	39.601
esp32c3-devkit	RISC-V @ 80 MHz	258.874	222.272	54.953
sipeed-longan-nano	RISC-V @ 108 MHz	103.108	106.422	37.789
hifive1b	RISC-V @ 320 MHz	60.884	66.492	153.747

Memory and storage consumption in KB, computational latency in ms.

Per-Operator Evaluation Output of TFlite sinus model.

Ops	Latency	Latency (%)	Asso. Params	Memory	Storage
add_nn_relu	8.856	15.22%	p0, p1	0.128	0.128
add_nn_relu_1	46.682	80.23%	p2, p3	0.128	1.088
add	2.646	4.54%	p4, p5	0.068	0.068

Memory and storage consumption in KB, computational latency in us.

Future work with RIOT-ML

- Support On-device Learning
- Optimization for multi-core MCU scheduling
- Generalize to non-neural network model
- Support federated-learning scenarios

Paper

RIOT-ML preprint: https://bit.ly/3UgZ4JT
RIOT-ML Code: https://github.com/TinyPART/RIOT-ML

Code

Q&A

Contact: `emmanuel.baccelli@inria.fr`
`emmanuel.baccelli@fu-berlin.de`

Wrapping up kick-off June 5th-7th

- Our D day on June 5th : we had one of the better ones!
- Some of the detected potential new projects / complementarities
    - Digital Health:
        - potential collab. on crano-facial image analysis Charité/ECDF/Inria
        - collab. with PTB / TU Berlin on standardization…
    - Digital Humanities:
        - sharing open science policies across institutions and infrastructure (e.g. DFKI - Inria);
        - using language data to create forward-looking indicator
    - Applied Mathematics:
        - expand WIAS-Inria collab. on photovoltaic.
        - Facilitate Rennes/Potsdam internships. Helmholtz use of Inria experts for reviews
    - IoT & Security:
        - expand Cryspen / FU Berlin collaboration on RIOT-rs and crypto
- Potential beyond: maybe other AI topics (e.g. TinyML with DFKI?),