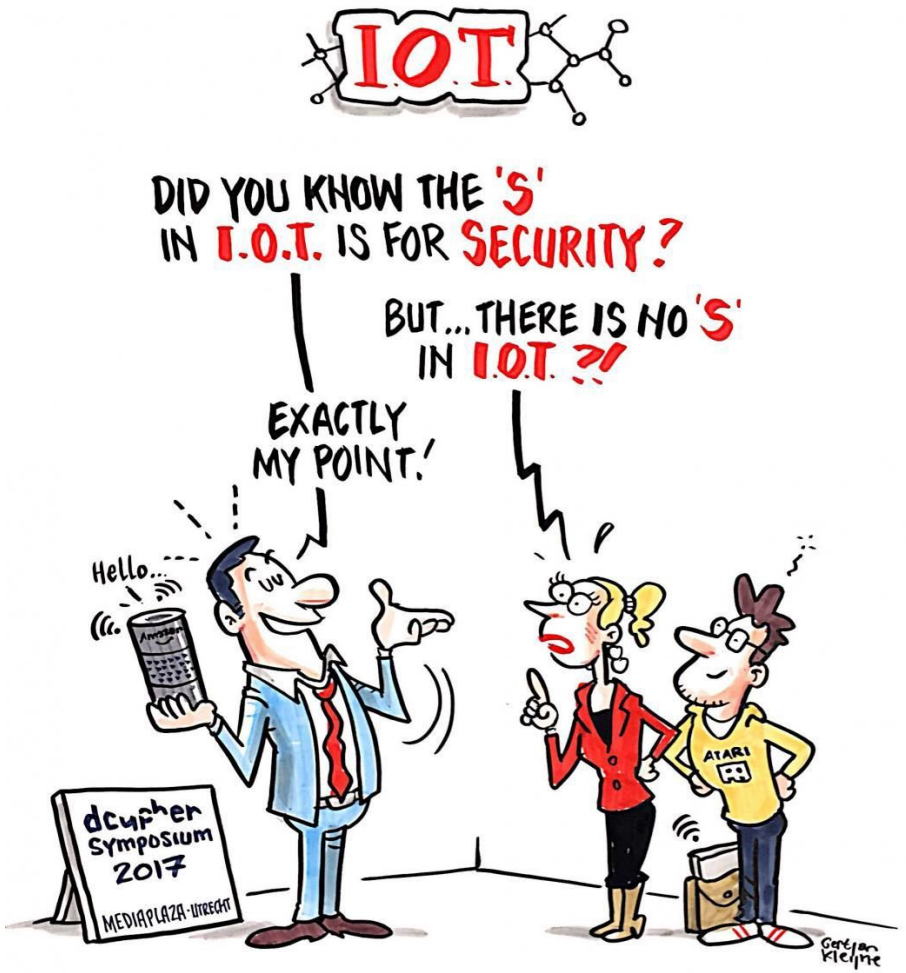
Longer-Term Security for Low-Power IoT Software

Eclipse IoT Day 2023

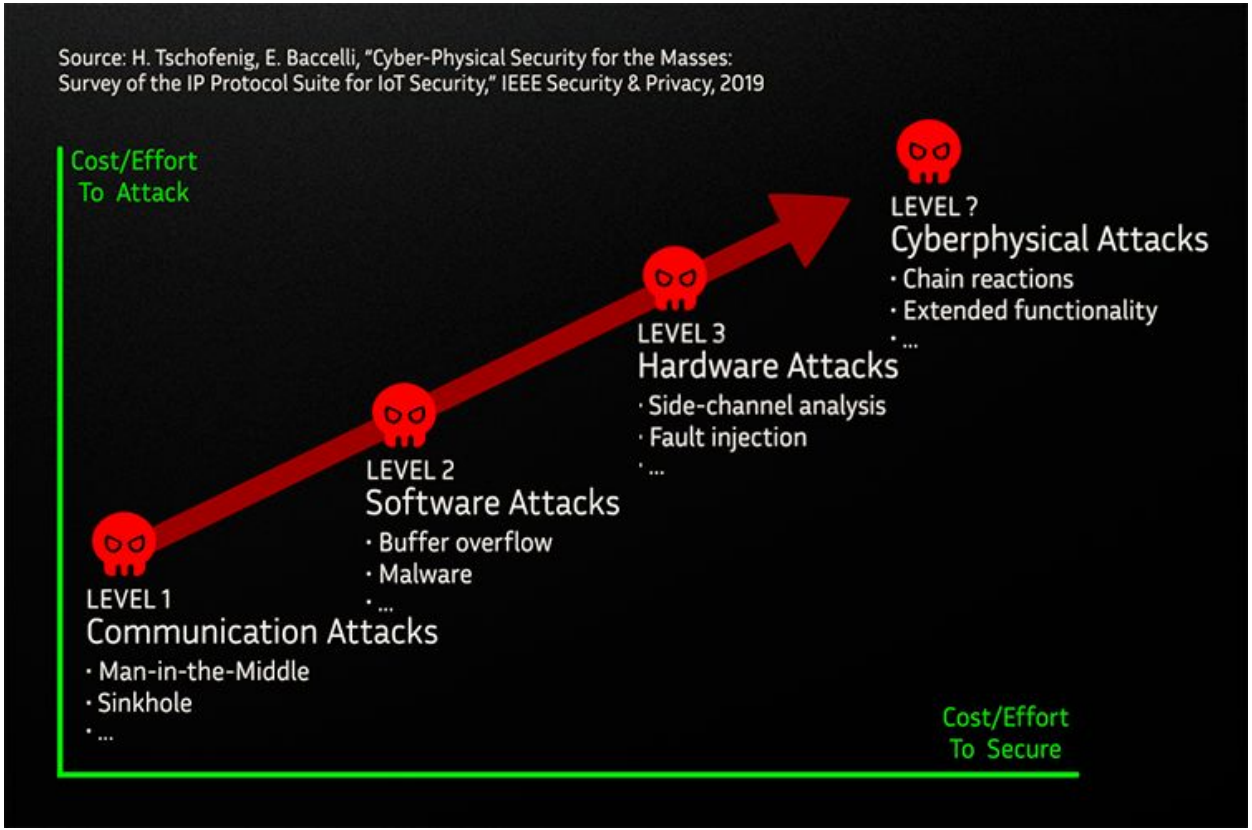
Emmanuel Baccelli

Inria & FU Berlin

The Internet of Things (IoT)



IoT: Cyber-Attack Surface

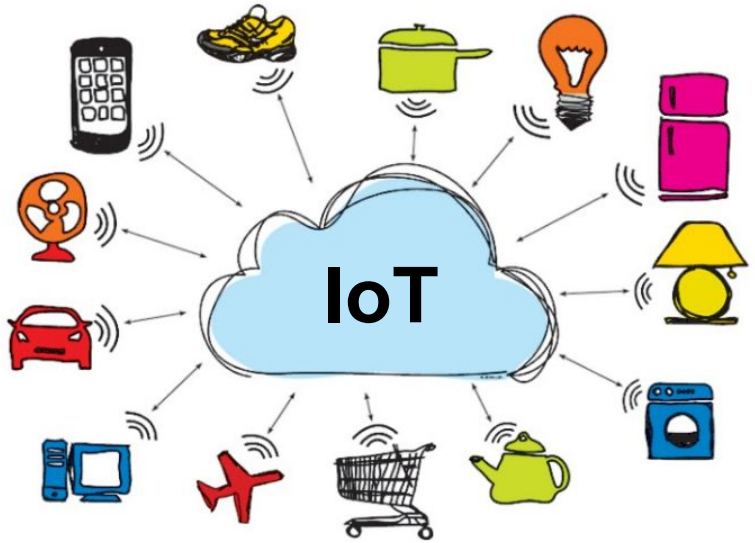


Main focus in this talk: defend against
- ★ communication attacks;
- ★ software attacks;

The Internet of Things (IoT)

IoT is everywhere...
In virtually all verticals: predictive maintenance (Industry 4.0), smart health, (token) contact tracing, connected vehicles ECUs, smart home/building, precision agriculture, TinyML etc.

… and IoT depends upon low-power devices.
Low-power devices are used in more varied use cases. They run increasingly complex software.



Clipart: Opentechdiary

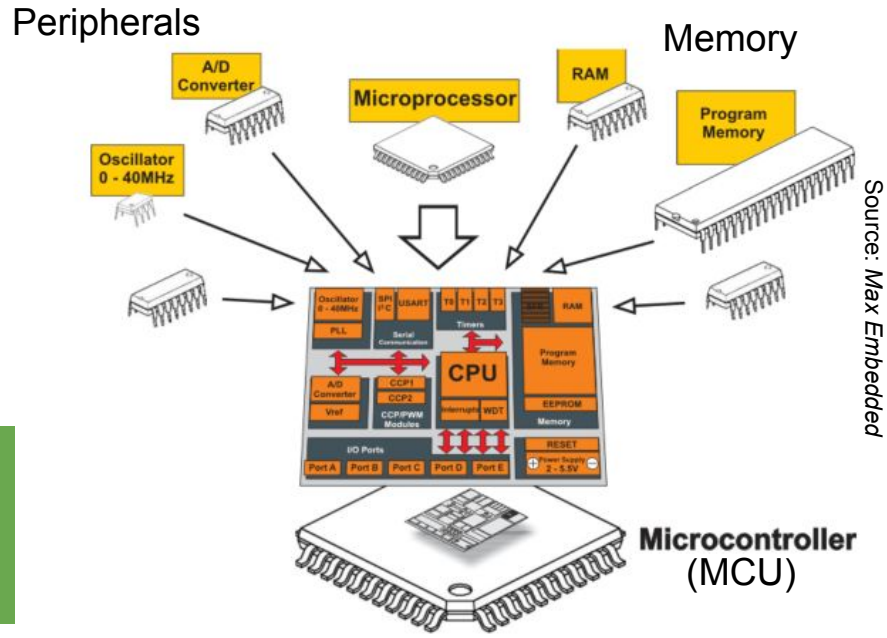
Agenda

1. Context
2. Anatomy of Low-Power IoT
3. Firmware Update Security for Low-Power IoT
4. Function-as-a-Service Primitives for Low-Power IoT

Low-Power IoT ?

What's low-power? Microcontrollers.

- ➔ milliWatt
- ➔ kiloBytes
- ➔ megaHertz



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

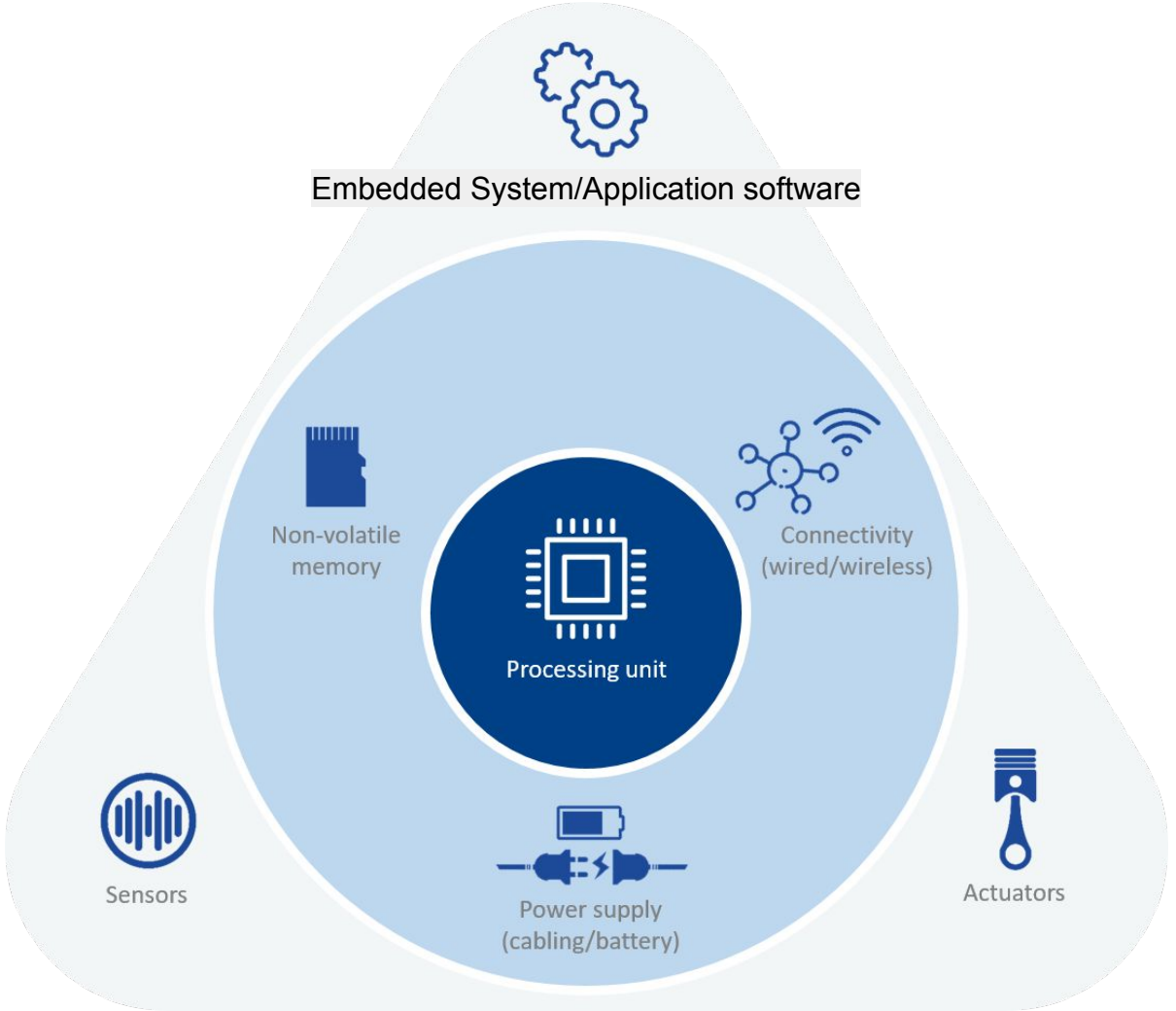
- ➔ much less capacity in computing, networking, memory;
- ➔ much smaller energy consumption & tiny price tag (<1$).

Some stats:

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

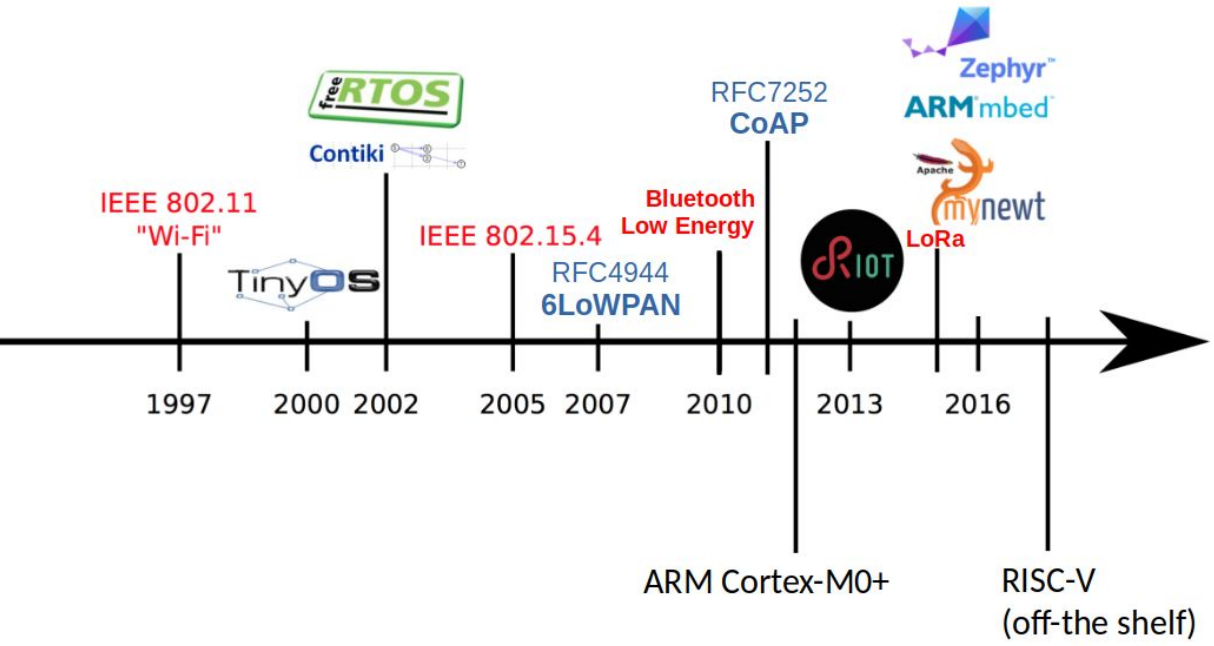
Source: venturebeat.com/2020/01/11/why-tinyml-is-a-giant-opportunity/

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)...

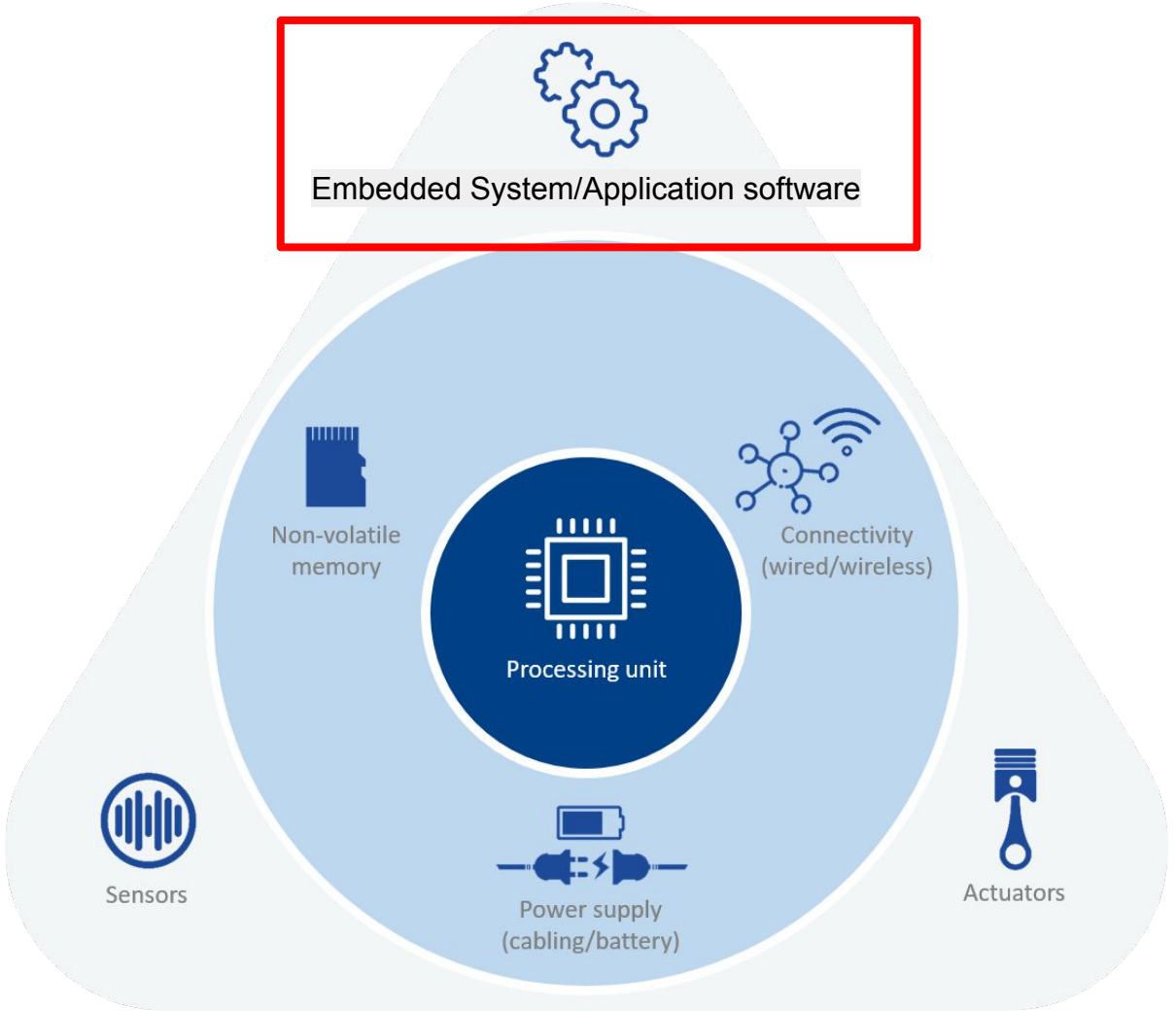
Low-power Wireless Networking

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

Embedded Software

- ★ Ecosystem of “plug-in” libs & network stacks: **Eclipse projects,** mbedTLS, LVGL, libCOSE, openThread, littleFS, uTensor…
- ★ Open source operating systems: **RIOT**, Contiki, mbedOS (Arm), Zephyr (Intel), FreeRTOS (Amazon), LiteOS (Huawei)

Anatomy of a Low-Power IoT Device



Source: 2018 Enisa Summer school

IoT Software Long-Term Cybersecurity?

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.

Constraints from IoT?

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

Minimum guarantees?

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

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

What's a reasonable general strategy?

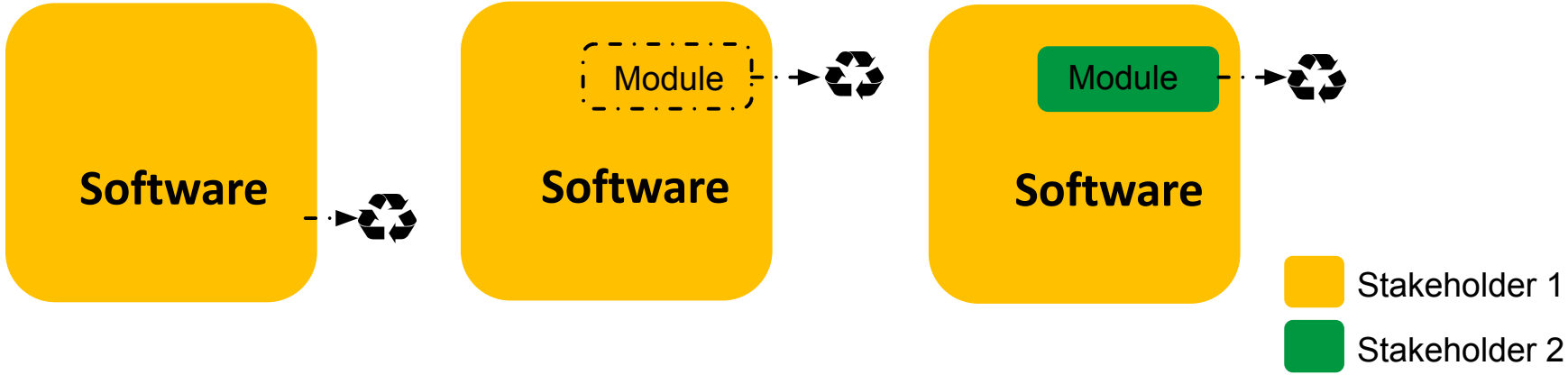
1. Facilitate long-term interoperability? Use (open) standards;
2. Facilitate long-term maintenance? Use open source collaborative software;
3. Quantum-level security? Minimum: software update authentication/authorization.

Pain points & Challenges for low-power IoT:

- Securing modular/multiparty software on low-power devices;
- Quantum-resistance adds to an already-tall order…
- **Democratizing software updates**, over low-power networks.

Scenarios?

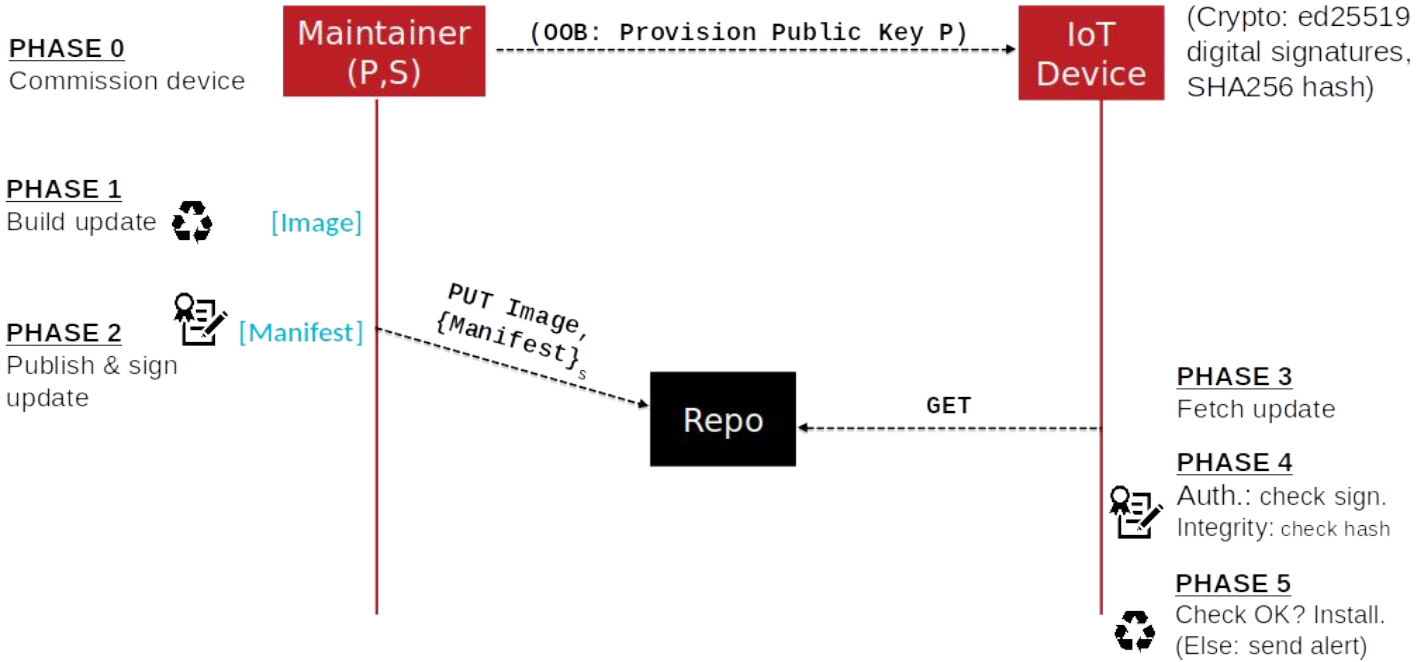
- Case 1 : monolithic software update, single stakeholder
- Case 2 : modular software updates, single stakeholder
- Case 3 : modular software updates, multiple stakeholders



Agenda

1. Context
2. Anatomy of Low-Power IoT
3. Firmware Update Security for Low-Power IoT
4. Function-as-a-Service Primitives for Low-Power IoT

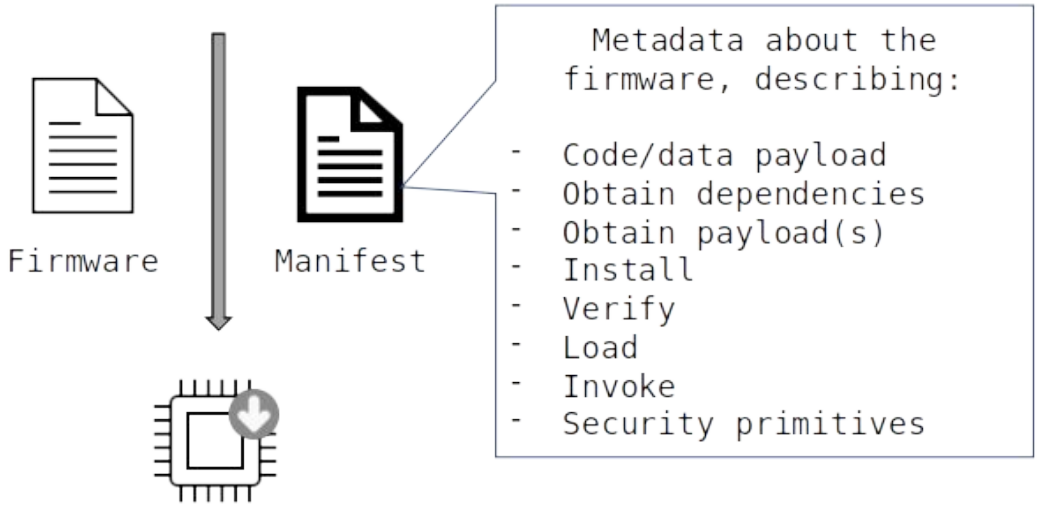
SUIT = new architecture, metadata & serialization for lightweight IoT firmware update security : authentication, integrity checks (and more) specified at IETF, currently in the final stages of standardization: see https://datatracker.ietf.org/wg/suit/about/



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-21,* Nov. 2022.

Bugfixes
Reconfiguration
New Functionalities
Vulnerabilities Mitigation

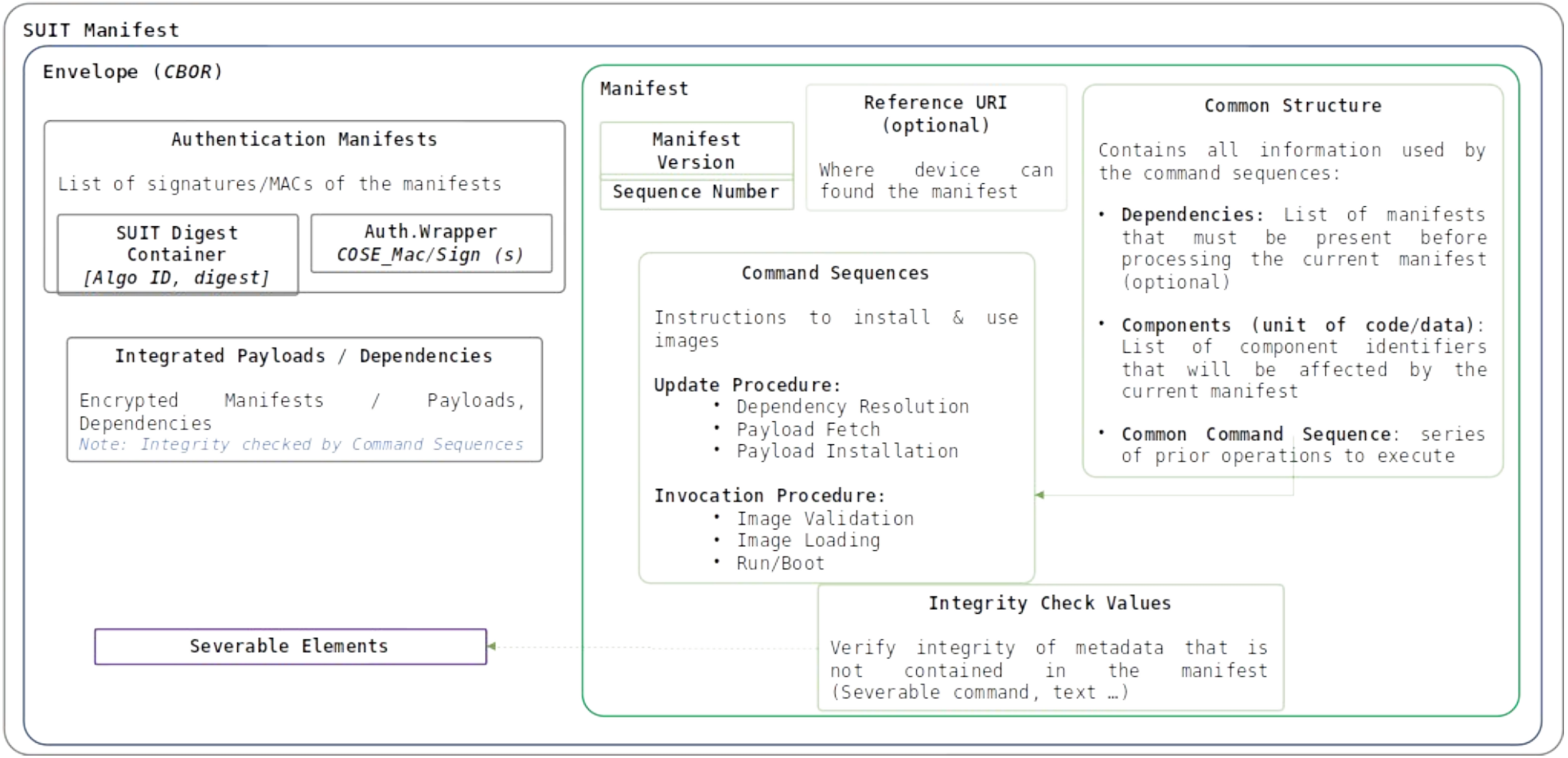


SUIT Manifest (*draft*)

Simple to parse
Simple to process
Compact encoding
Comprehensible by intermediate system
Enable advanced use cases
Extensible
Flexibility

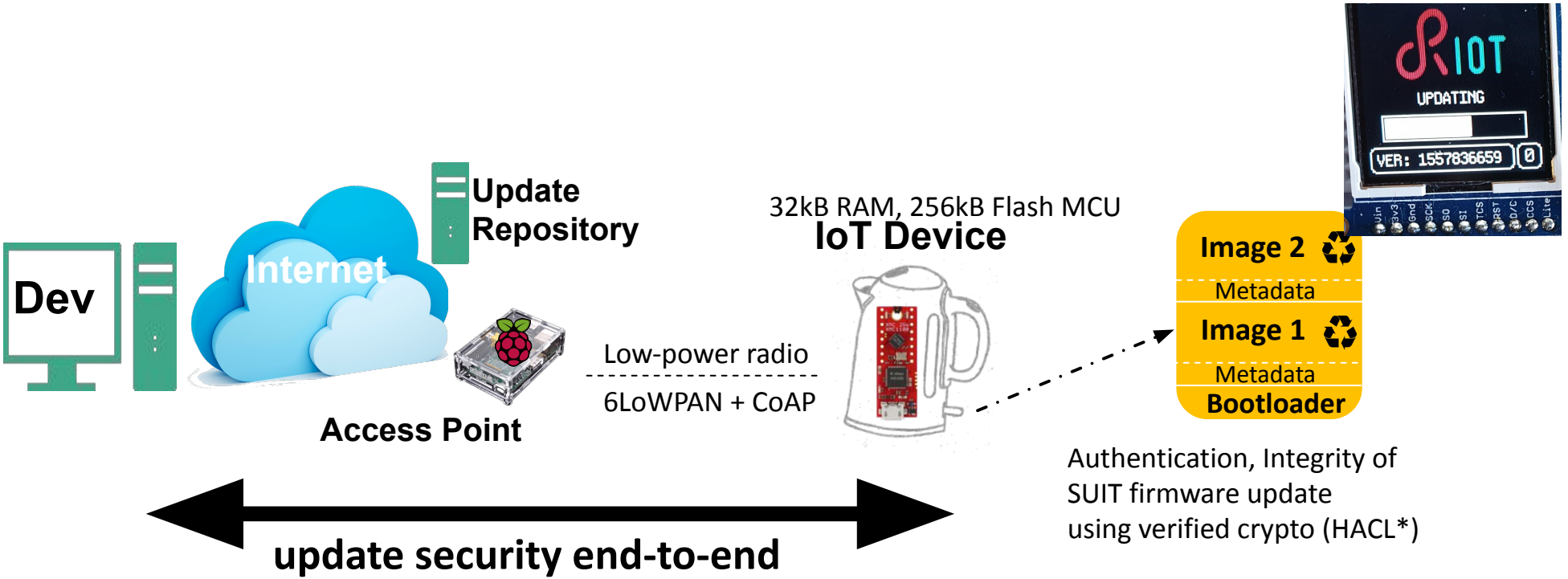
Source: 2022 slide from Interop / Loïc Dalmasso

SUIT Metadata Structure (Sketch)



Source: 2022 slide from Interop / Loïc Dalmasso

★ 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)



Authentication, Integrity of SUIT firmware update using verified crypto (HACL*)

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

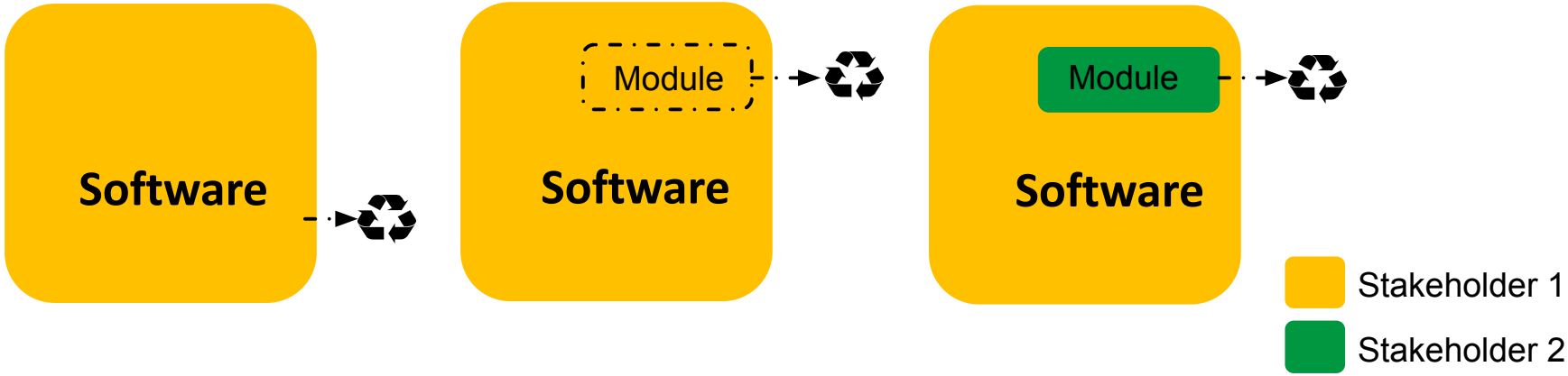
Table 7: Relative cost increase for SUIT with quantum resistance (on ARM Cortex M-4).

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.

Scenarios?

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



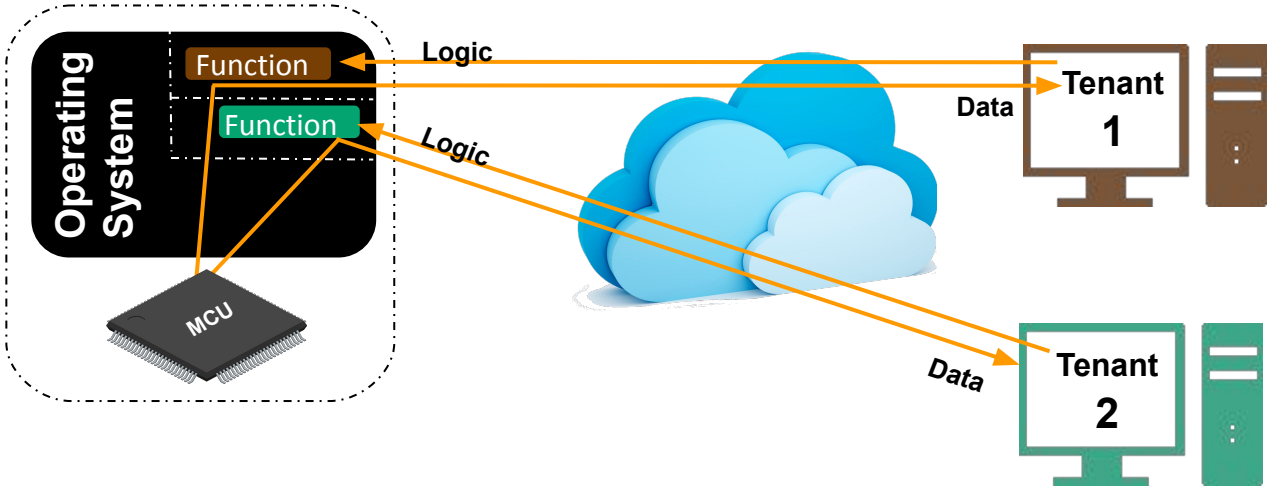
Agenda

1. Context
2. Anatomy of Low-Power IoT
3. Firmware Update Security for Low-Power IoT
4. Function-as-a-Service Primitives for Low-Power IoT

Micro-Services on Microcontrollers?

Customize on-the-fly **deployed IoT software** with additional/modifiable functions:

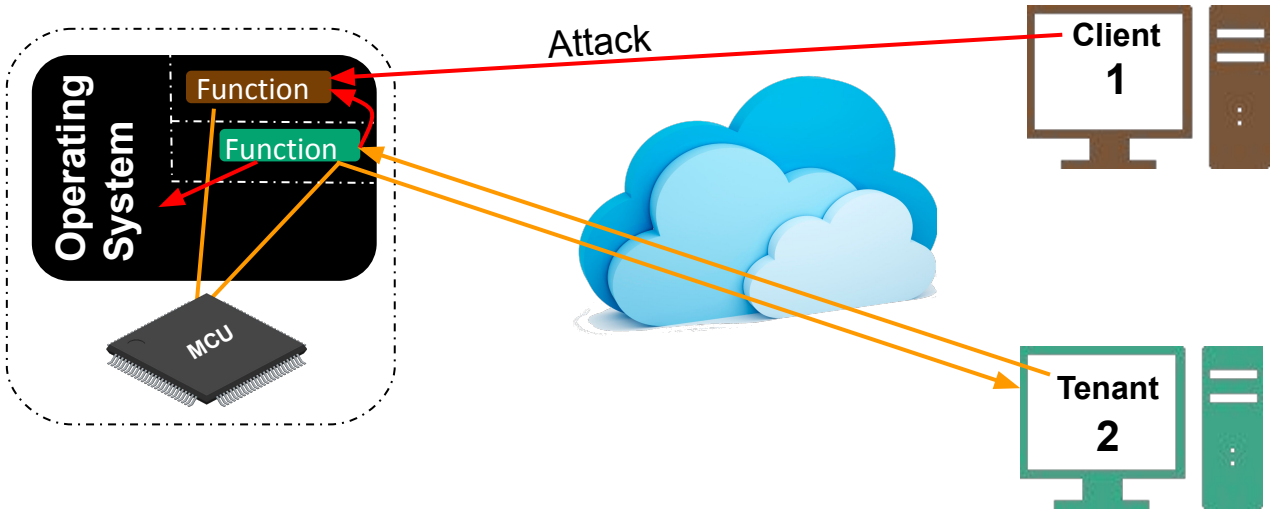
- Host business logic applications
- Host debug/monitoring snippets
- Host multiple functions, by different tenants



Micro-Services on Microcontrollers?

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

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



Micro-Services on Microcontrollers?

Closest related works: **NanoLambda**, a FaaS-like embedded engine [1]

Main limitations:

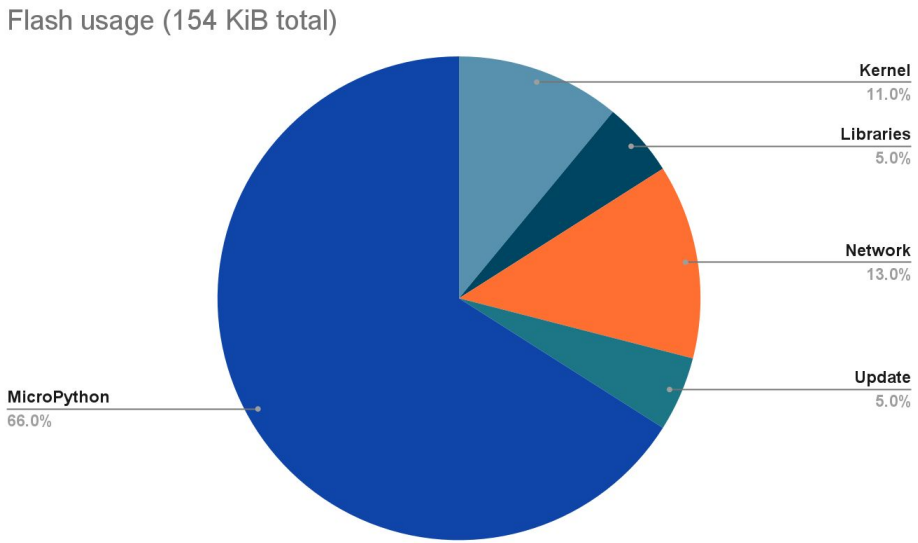
- Flash memory budget explosion
- 1000x slower than native code.

Other engine: **WebAssembly (WASM)**

- Promises nicer isolation…
- But similar flash budget explosion

Also: **MicroPython**

- Performance similar to Nanolambda



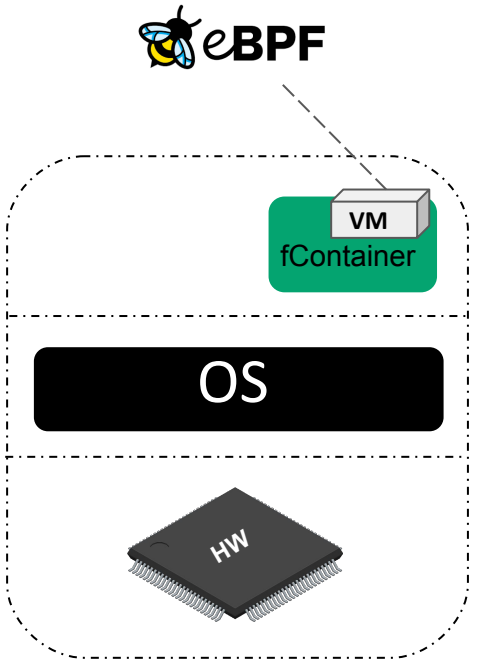
[3]: G. George et al. "*NanoLambda: Implementing Functions-as-a-Service at All Resource Scales for the Internet of Things*," IEEE/ACM Symposium on Edge Computing , 2020

FemtoContainer virtualization for MCUs

FemtoContainers use ultra-lightweight virtualization:
rBPF [4], the eBPF VM ported to microcontrollers

- Register-based VM
- Super small memory requirement
- Limited instruction set
- Designed as sandbox

(Allows for usage of existing compiler toolchains,
supports C, C++, Rust, any LLVM-compiled language)



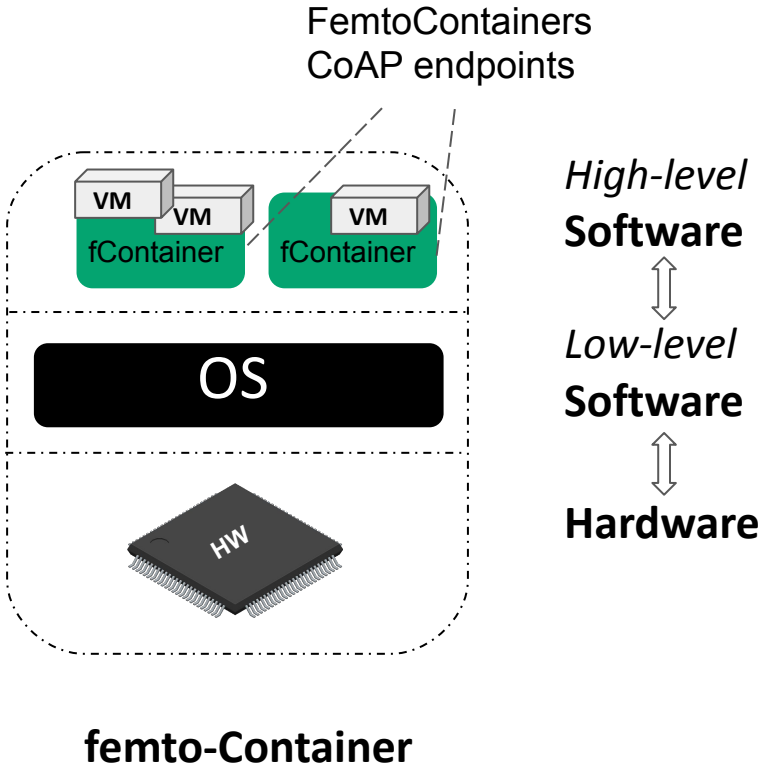
High-level
Software
Low-level
Software
Hardware

femto-Container

[4] K. Zandberg et al. Minimal Virtual Machines on IoT Microcontrollers: The case of Berkeley Packet Filters with rBPF, in PEMWN, Dec. 2020

Femto-Containers: Implementation

- Real-Time OS (RTOS) syscalls
  - Allows & controls sensor interaction, network services
  - Reference implementation in RIOT, available at: https://github.com/future-proof-iot/Femto-Container_tutorials
- Femto-container(s) exposed as CoAP resources
  - Trigger container applications via networked endpoints
- SUIT-compliant software updates
  - OTA updates of containerized microservices over CoAP
- Femto-Container hosting engine = only 1000 LoC (!)
  - allowed formal verification [5] for fault-isolation

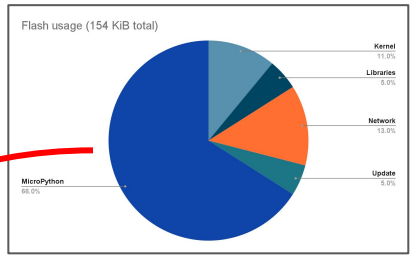


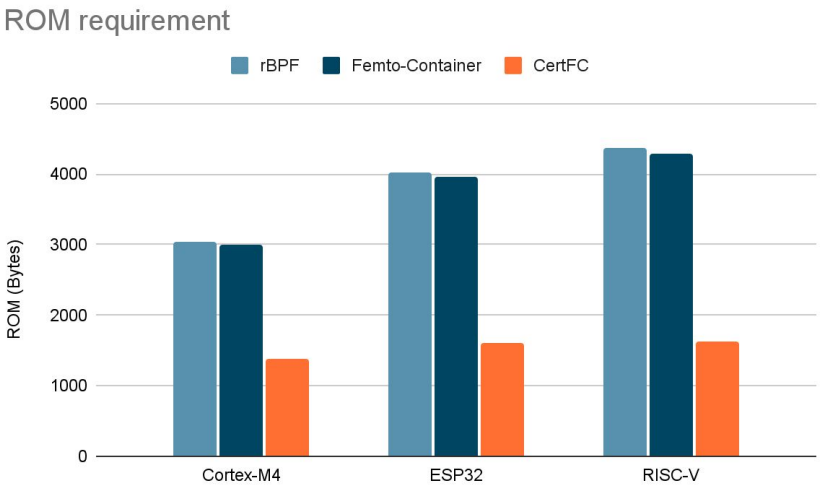
[5] S. Yuan et al End-to-end Mechanized Proof of an eBPF Virtual Machine for Microcontrollers, in CAV, Aug. 2022

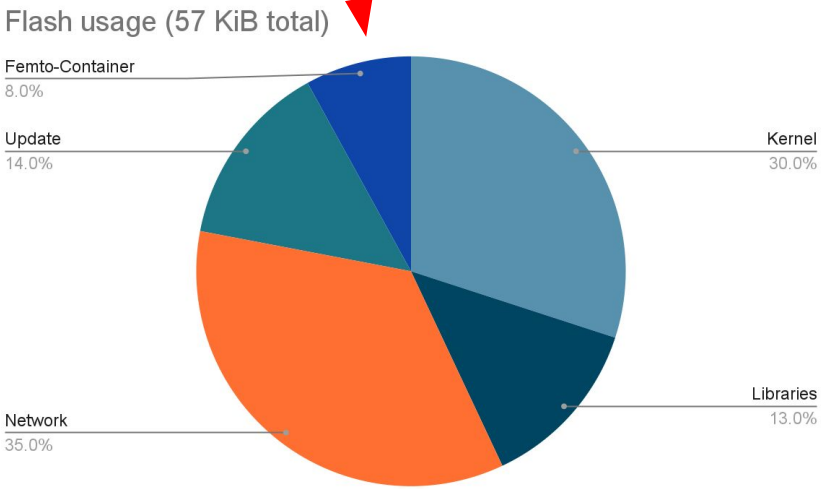
Experimental Evaluation

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

Take-away

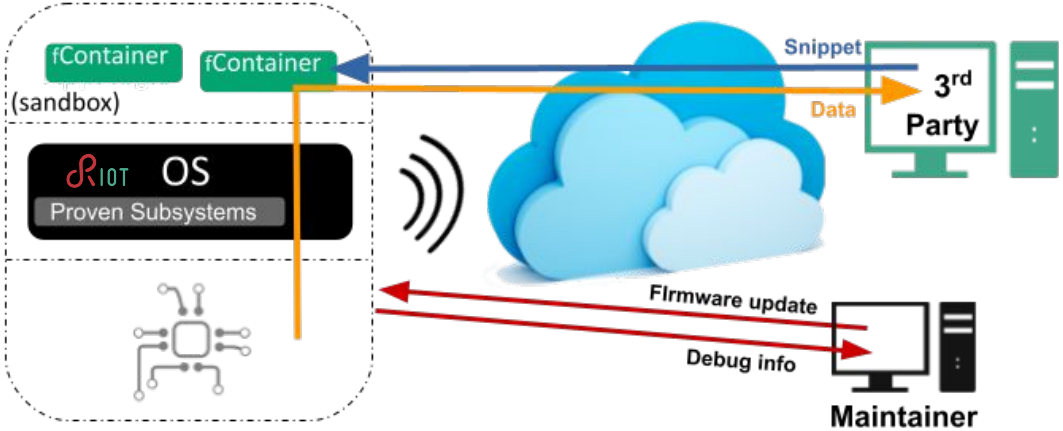
- ★ Long-term low-power IoT cybersecurity requires secure software updates
- ★ SUIT
  - ○ is a good standard option for the security of low-power IoT updates
  - ○ has open source implementation available
  - ○ was shown to as doable even with post-quantum 128-bit security (e.g. on RIOT devices)
  - ○ can secure modular update of IoT software other than firmware (e.g. eBPF VMs)
- ★ Femto-Containers and eBPF are an option for DevOps on low-power IoT

Défi Inria **RIOT-fp**

Online: https://future-proof-iot.github.io/RIOT-fp

Context

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

Publications: at many academic journals & conferences;
Software: jumpstart/maintenance of 10+ open source repositories (including RIOT);
Standards: several standardization docs at IETF (including one RFC already).

Teams involved: TRiBE, EVA, GRACE, TEA, CELTIQUE, PROSECCO (+ FU Berlin)

Inria

THANKS! QUESTIONS? SHOOT!

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

Email :
emmanuel.baccelli@inria.fr

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