Towards Secure IoT with Open Source and RIOT

Emmanuel Baccelli

emmanuel.baccelli@inria.fr

Agenda

- Context
- Improving IoT functionalities
- Mitigating risks: areas of work
  - (Trusting IoT Hardware)
  - Crypto Primitives for IoT
  - Secure IoT Software
  - Secure IoT Networking
  - IoT Software Updates
- RIOT-FP

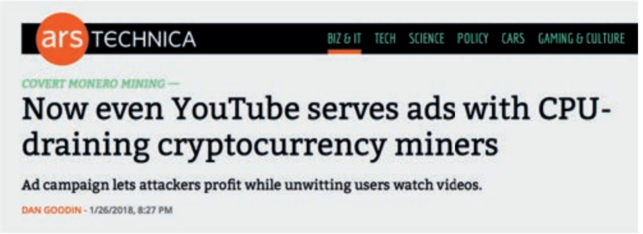
Context

- World War III is online
  - State-driven (geopolitics), or profit-driven (pirates)
- Personal data-hungry Behemoths
  - Fighting back: GDPR at EU level, BCP 188 from IETF
- Extreme computing power becomes… average
  - Not just NSA, but also botnets, crypto-miners…

  (… and soon quantum computing?)

Anonymous

reality.ai


ars TECHNICA | BIZ & IT | TECH | SCIENCE | POLICY | CARS | GAMING & CULTURE

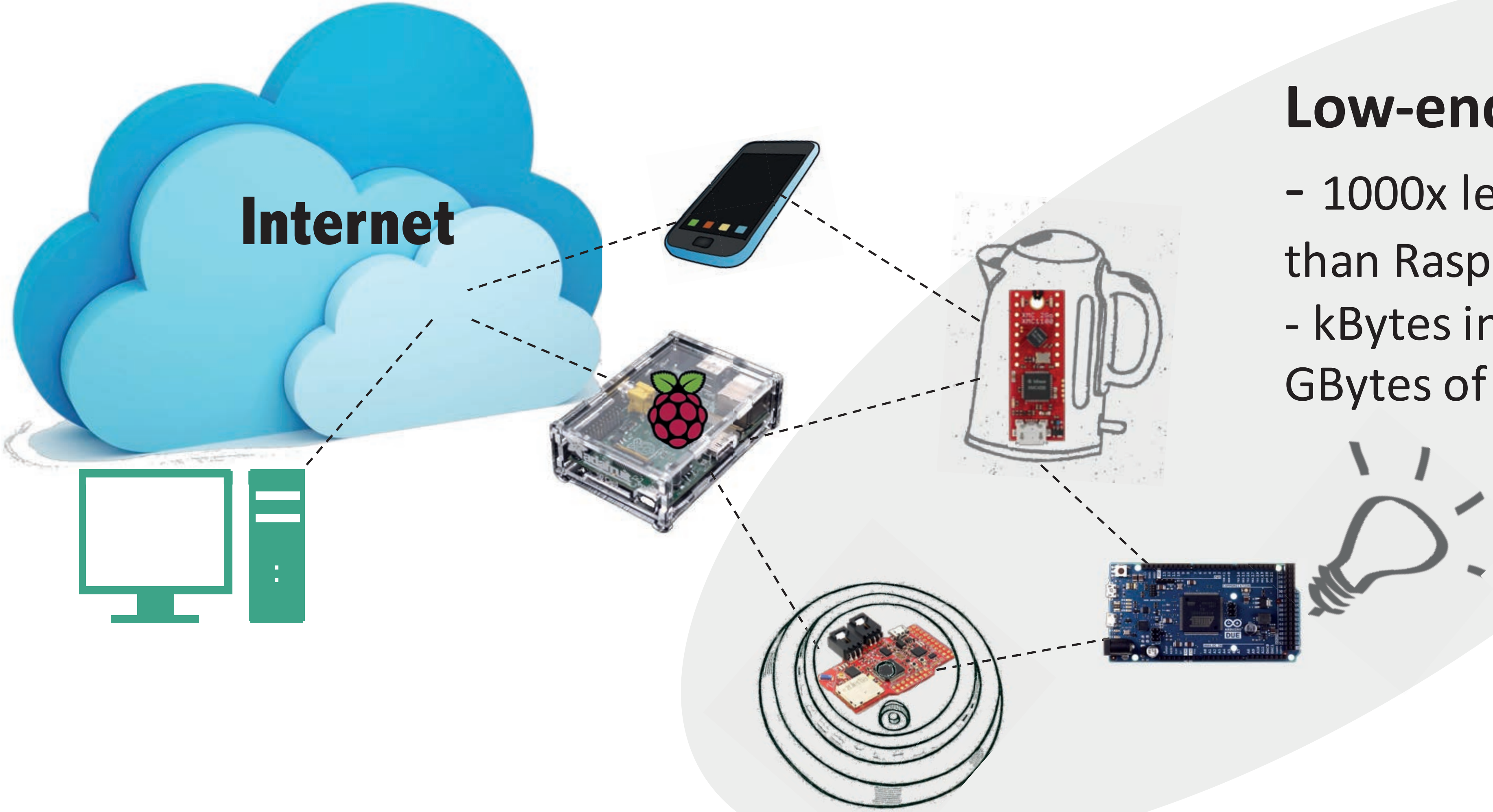
COVERT MONERO MINING —

Now even YouTube serves ads with CPU-draining cryptocurrency miners

Ad campaign lets attackers profit while unwitting users watch videos.

DAN GOODIN - 1/26/2018, 8:27 PM

IoT: Deploying a Giant Cyber-Physical Robot



Low-end IoT devices

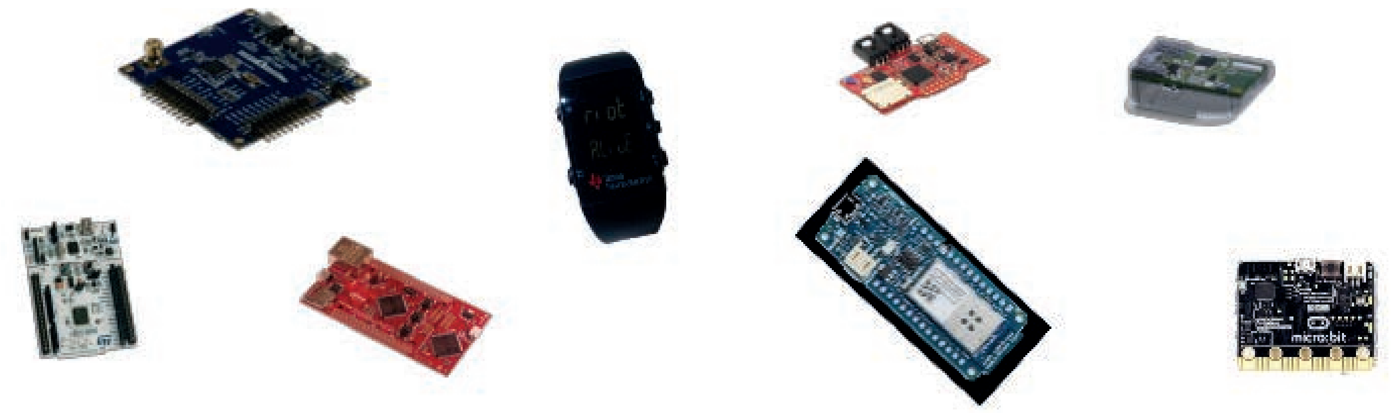
- 1000x less energy than RaspberryPi
- kBytes instead of GBytes of memory

A multitude of smaller, cheaper microcontroller-based devices

IoT: Low-end Device Polymorphism

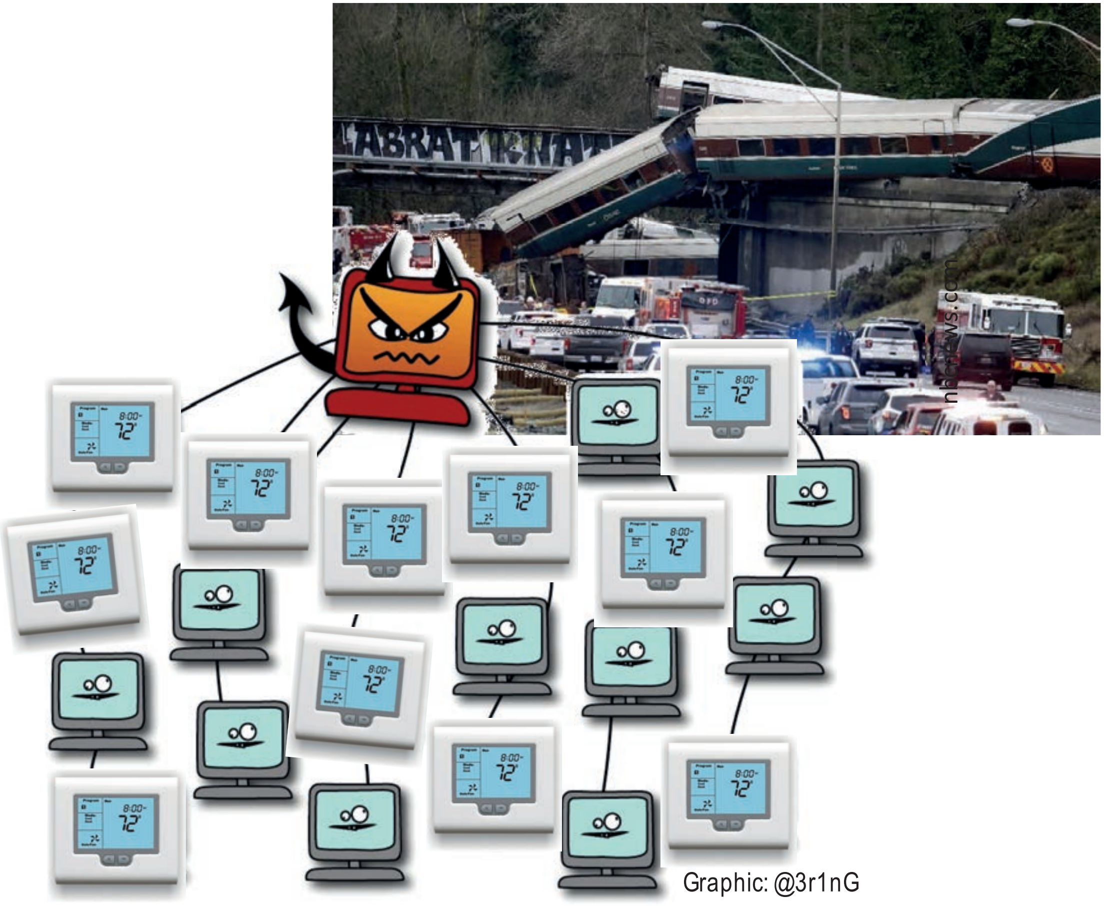
iotworm.com

- Extremy varied use-cases
- Various vendors
- Various architectures (8-bit, 16-bit, 32-bit)
- Various low-power communication technologies (BLE, 802.15.4, DECT…)



IoT: Bad Risk *vs* Functionality Tradeoff (for now)

- Hacked system (cyber-physical robot) can cause direct physical harm
  - ⇒ **acceptable risks are changed**
- Sensors everywhere, all the time
  - ⇒ **scope of privacy breaches are changed**
- Extended functionality attacks*
  - new types of attacks based on **chain reactions** **
- Low-end IoT devices are the new **weakest link**



* E. Ronen, A. Shamir " *Extended Functionality Attacks on IoT Devices: The Case of Smart Lights*," 2016.
** S. Soltan et al. " *BlackIoT: IoT Botnet of High Wattage Devices Can Disrupt the Power Grid*," in UsenixSec, 2018.

IoT: Bad Risk *vs* Functionality Tradeoff (for now)

- Work to improve this tradeoff ?

$\Rightarrow$ Improving functionality

Hardware 

Software

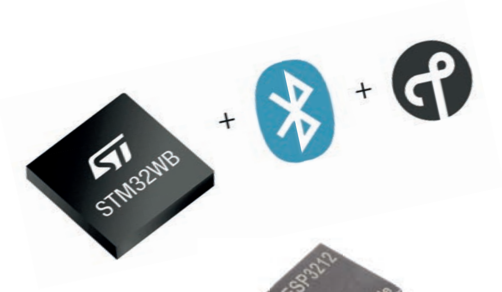
$\Rightarrow$ Reducing risk 

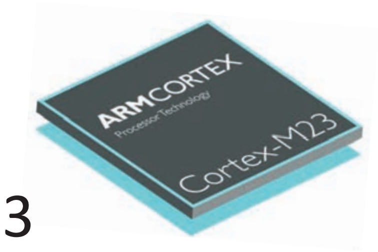
Agenda

- Context
- Improving IoT functionalities
- Mitigating risks: areas of work
  - (Trusting IoT Hardware)
  - Crypto Primitives for IoT
  - Secure IoT Software
  - Secure IoT Networking
  - IoT Software Updates
- RIOT-FP

IoT: Improving Functionalities (Hardware)

- Trends:
  - multi-radio: Nordic nrf52, STM32WB, ESP32
  - multi-core: ESP32, STM32WB
  - nvram + energy-harvesting: MSP430FR59XX
  - trusted execution environment: ARM TrustZone Cortex-M23
  - …

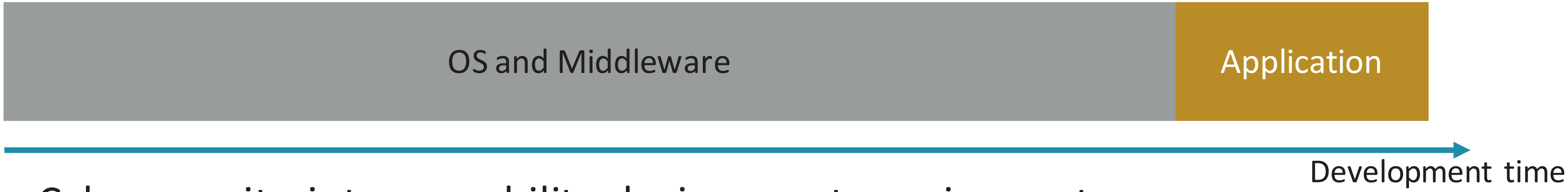




IoT: Improving Functionalities (Software)

- IoT software before
  - rudimentary embedded software, vendor-specific (vendor-locked?)

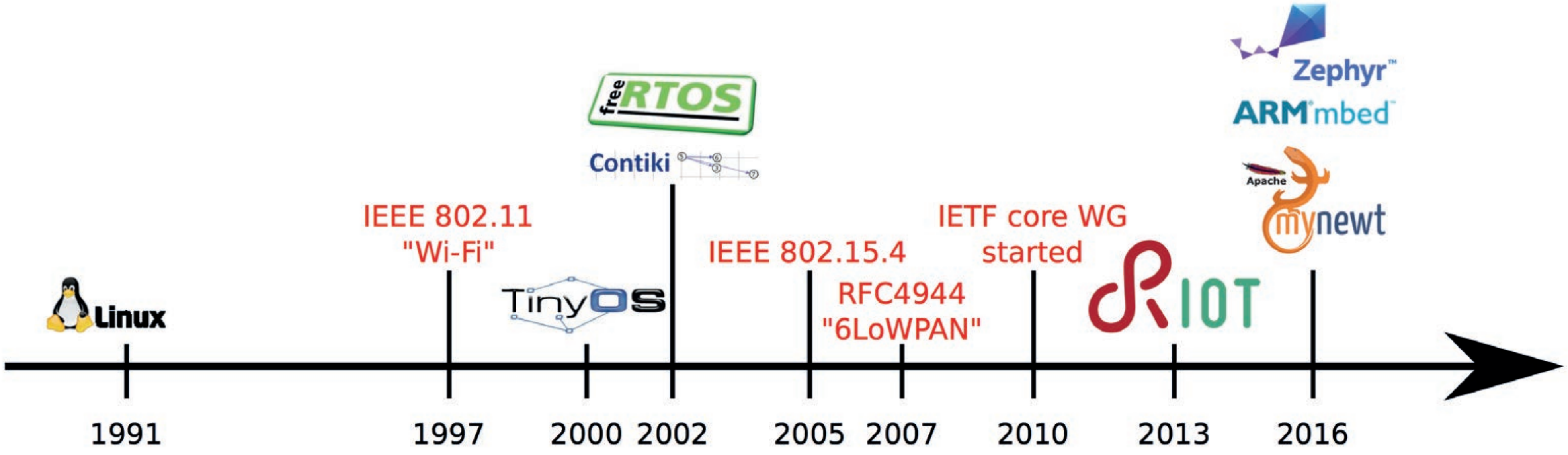
- IoT software now



  - Cybersecurity, interoperability, device mgmt requirements… increase complexity + drive the need for a real OS

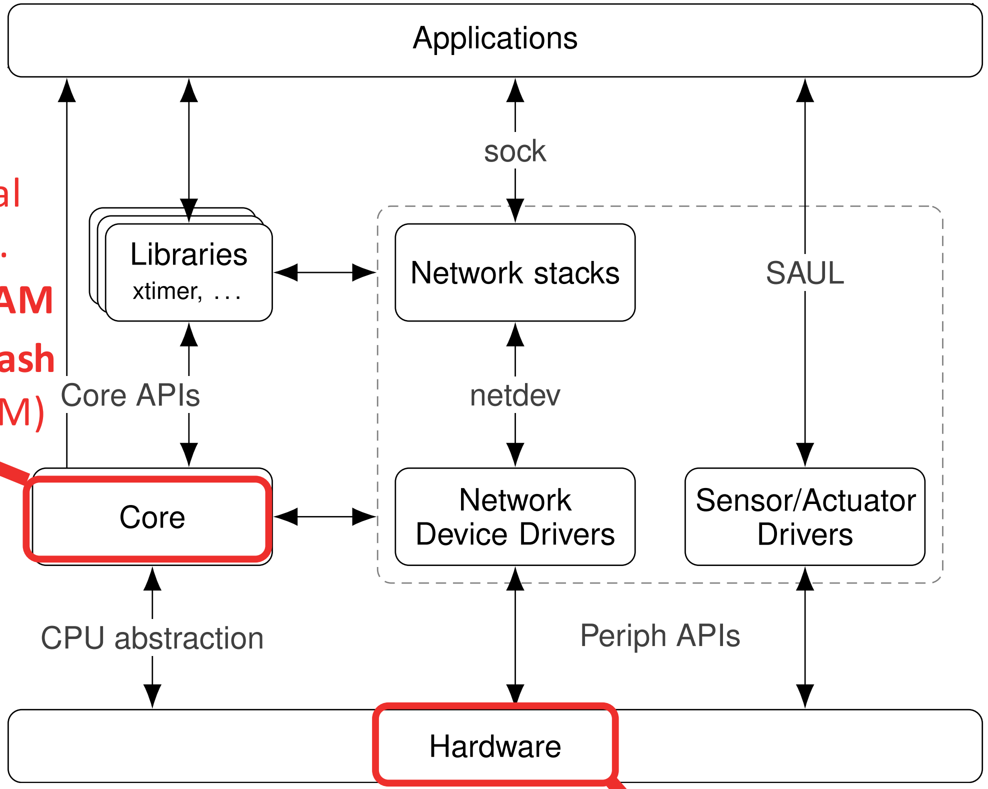
This slide is partly borrowed from Hannes Tschofenig, ARM

IoT: Improving Functionalities (Software)



O. Hahm et al. "*Operating Systems for Low-End Devices in the Internet of Things: a Survey*," IEEE Internet of Things Journal, 2016.

General-Purpose OS for low-end IoT



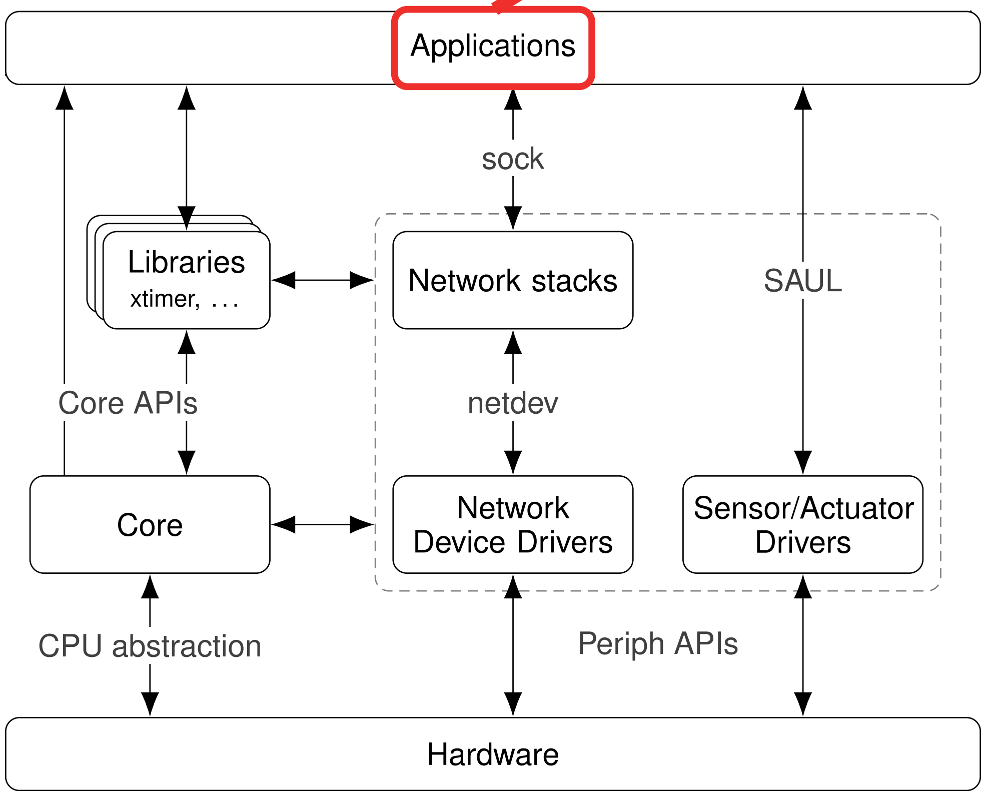
Minimal config.
2,6kB RAM
3,2kB Flash
(Cortex-M)

120+ boards supported
8-bit, 16-bit, 32bit MCUs

- ✓ **Unified APIs** – across all hardware, even for hardware-accessing APIs; Enables code reuse and minimizes code duplication;
- ✓ **Vendor & techno. independence** – Vendor libraries are avoided; Design decisions don't tie RIOT to a particular technology;
- ✓ **Modularity** – building blocks, to be combined in all thinkable ways; Caters for versatile use cases & memory constraints;

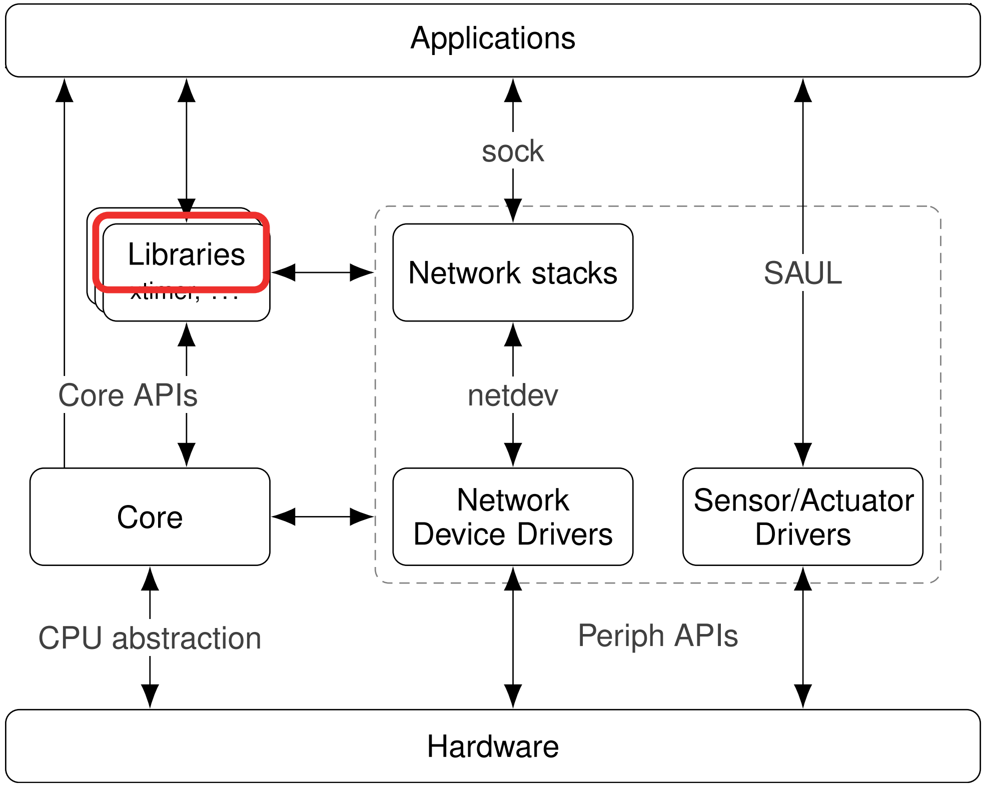
System-level Interoperability

C, C++ (JavaScript, Rust)



- Drawbacks:
  - some memory overhead, but still fits low-end IoT devices memory budget
  - some more work because re-implement from scratch (behind vendor header files)
- Advantages:
  - **Efficient & highly reusable code** across all supported hardware
  - **Emulation** of RIOT as a Linux process
  - Reusability of well-known **3rd-party tools** such as `gdb`, `valgrind`, `gprof`…

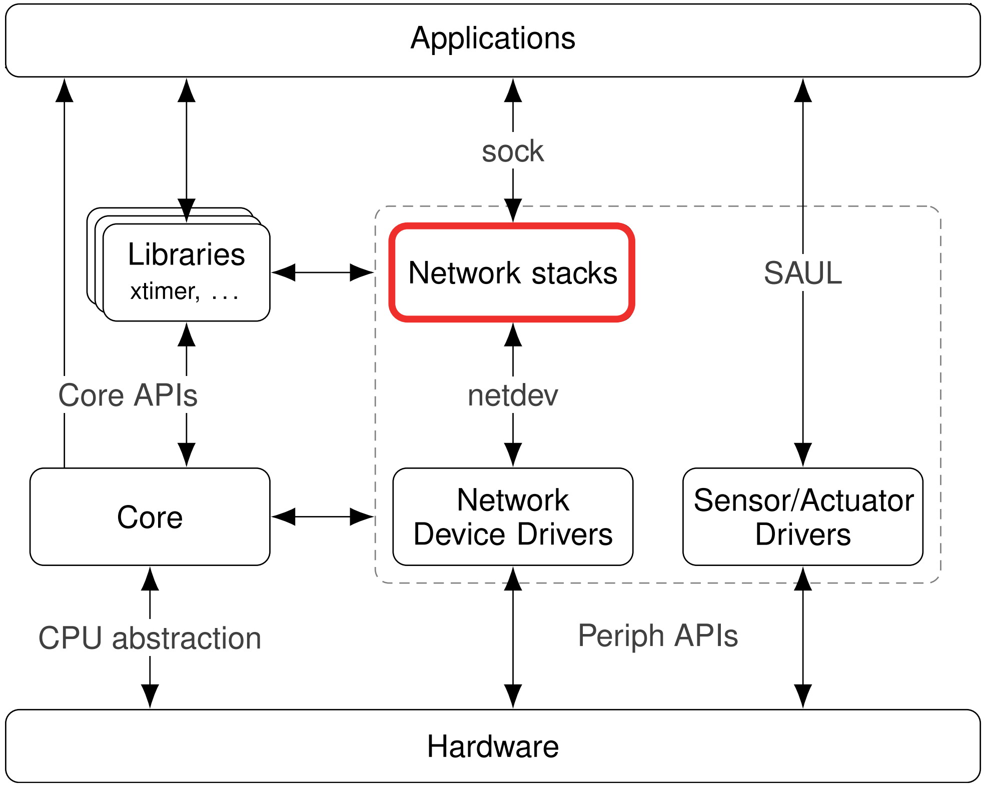
Numerous Libraries



- **Packages**: bundling 3rd-party libraries
  - Integrated on-the-fly at build–time
  - Easy to add: just requires 2 Makefiles
  - Patches (if needed) are typically minimal

Package	Overall Diff Size	Relative Diff Size
ccn-lite	517 lines	1.6 %
libfixmath	34 lines	0.2 %
lwip	767 lines	1.3 %
micro-ecc	14 lines	0.8 %
spiffs	284 lines	5.5 %
tweetnacl	33 lines	3.3 %
u8g2	421 lines	0.3 %

Network-level Interoperability



Wired & Bus
- CAN
- Ethernet

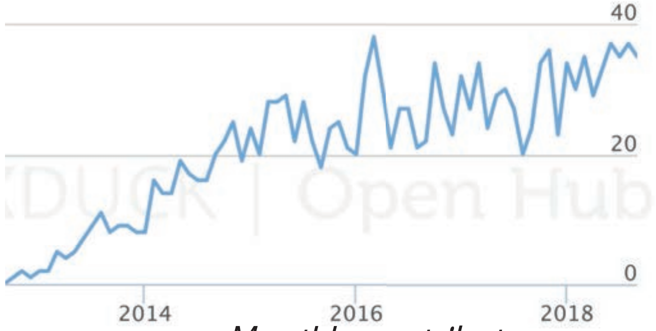
Low-power wireless LAN & WAN
- IEEE 802.15.4
- LoRa package
- BLE (work-in-progress)

IP Protocols Stacks
- Default stack (GNRC)
- Thread (package)
- lwIP (package)
- OpenWSN (in progress)

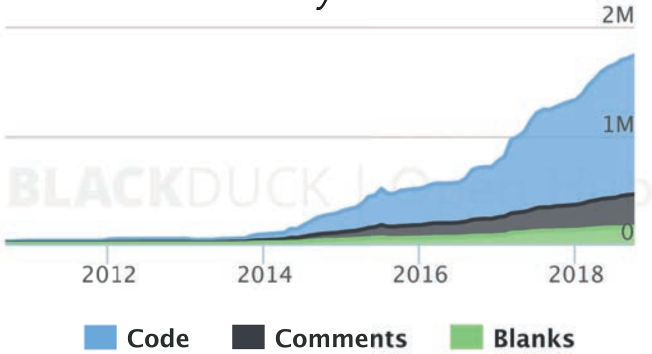
Experimental stacks
- CCN-lite (package)
- NDN-RIOT (package)

Large Open-Source Community

GitHub
github.com/RIOT-OS/RIOT



Monthly contributors



Source: www.openhub.net/p/RIOT-OS

- 2013: started as French-German research project
- 2018: ~200 contributors worldwide
  - 20,000 commits and ~8,500 Pull Requests
  - First products shipping RIOT last year
  - Hundreds of related scientific publications

E. Baccelli et al. '*RIOT: an Open Source Operating System for Low-end Embedded Devices in the IoT,*' IEEE Internet of Things Journal, 2018.

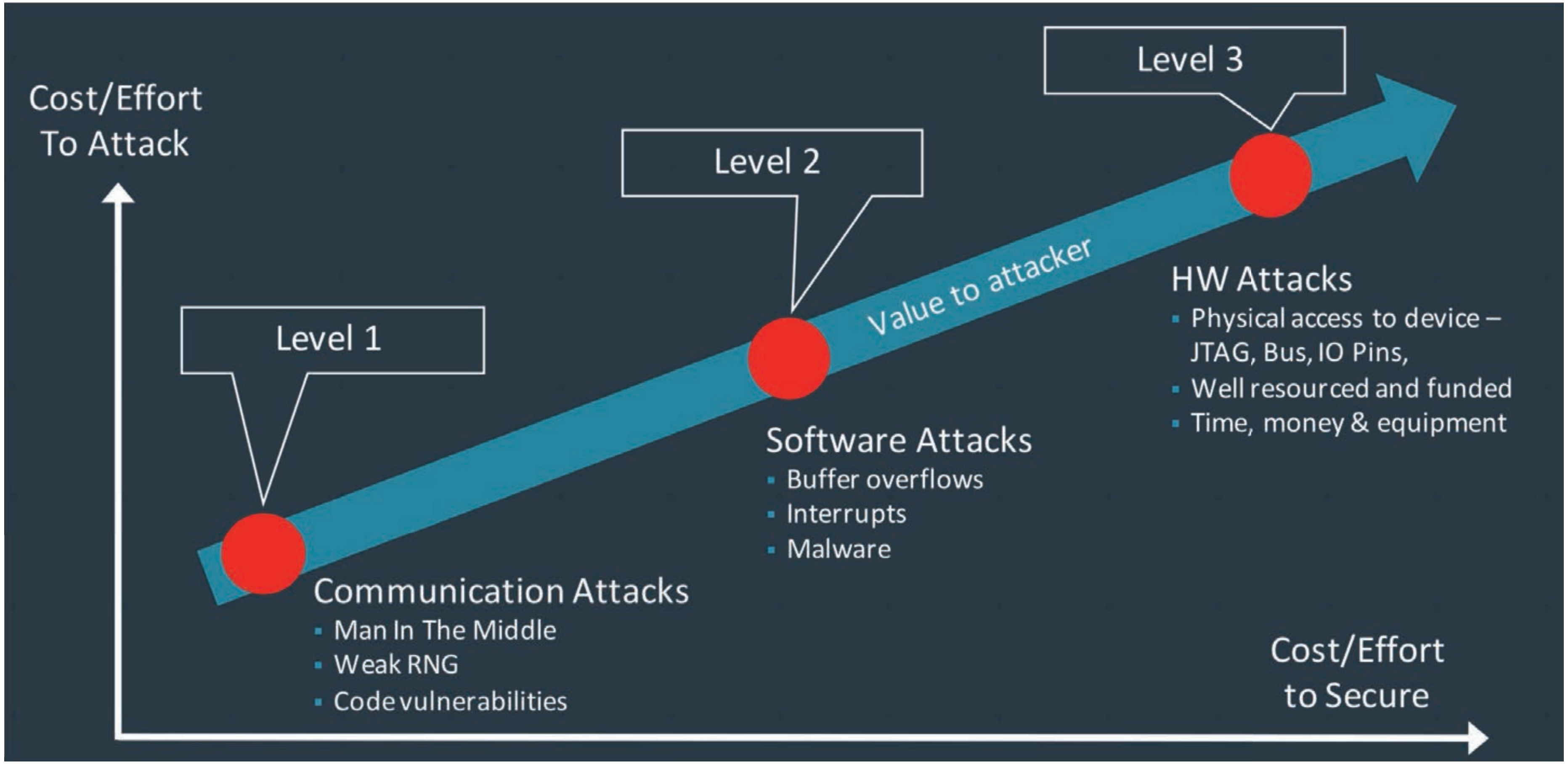
  - Yearly RIOT Summit conference

    Last in Amsterdam, Sept. 13-14, see http://summit.riot-os.org

Agenda

- Context
- Improving IoT functionalities
- Mitigating risks: areas of work
    - (Trusting IoT Hardware)
    - Crypto Primitives for IoT
    - Secure IoT Software
    - Secure IoT Networking
    - IoT Software Updates
- RIOT-FP

(Trusted IoT Hardware)



Slide borrowed from Hannes Tschofenig, ARM, at ENISA Summer School 2018

(Trusted IoT Hardware)

- Trend: **secure area of the microcontroller** for isolated execution, integrity of applications & confidentiality of their assets
  - **Sancus*** on MSP430 16-bit microcontrollers (automotive context)
    - Prototype **isolating software components** via memory curtaining
    - **Remote attestation** & authenticates comm. with software component
  - Similar on ARM Cortex-M 32-bit microcontrollers: **TrustZone** (commercial)
    - Upcoming Cortex-M33 and Cortex-M23 micro-controllers

* J. Noormans et al. '*Sancus 2.0: A Low-Cost Security Architecture for IoT Devices*', ACM Transactions on Privacy and Security, 2017

Agenda

- Context
- Improving IoT functionalities
- Mitigating risks: areas of work
  - (Trusting IoT Hardware)
  - Crypto Primitives for IoT
  - Secure IoT Software
  - Secure IoT Networking
  - IoT Software Updates
- RIOT-FP

IoT Crypto Primitives

- IoT devices deployed will last for years (maybe decades!)
- Current cyphers are typically slow + big on low-end IoT devices

$\Rightarrow$ Need for faster, smaller asymmetric cyphers

$\Rightarrow$ Need for stronger cyphers (quantum resistance)

  $\Rightarrow$ new techniques for asymmetric crypto (ongoing NIST competition)

  $\Rightarrow$ upgrade symmetric crypto, sometime down the line (e.g. double key size)

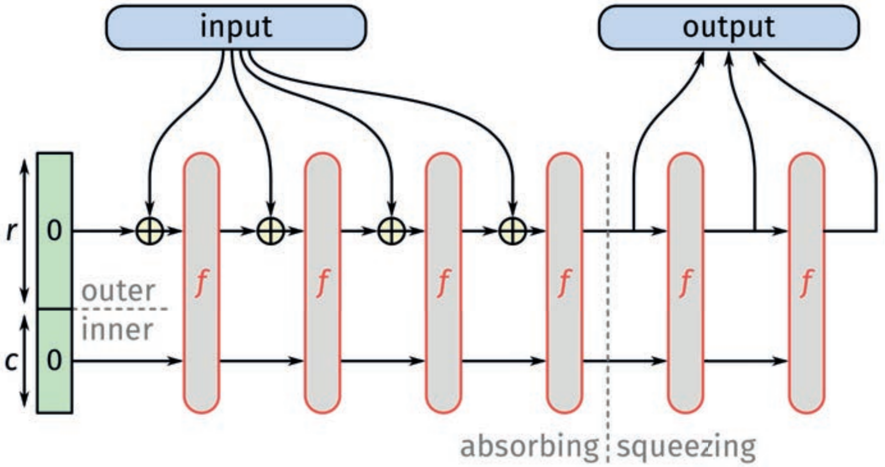
IoT Symmetric Crypto

More flexible primitives:

SHA-3's **sponge construction*** for hashing

- Output: infinte length (on demand)
- Shared code can provide other functions
  - Pseudo-random number generator
  - Message authentication code (MAC)
  - Stream encryption
  - (more with the duplex construction)

- On-going experimental work to evaluate this prospect on top of RIOT



G. Van Assche *'Permutation-based cryptography for the IoT,'* RIOT Summit, 2017.

* G. Bertoni et al. *'Cryptographic sponge functions'*, 2011.

IoT Asymmetric Crypto

- Smaller code:
  - **tweetnacl** (Bernstein et al.): Source fits in 100 tweets, using curve25519

- More efficient algorithms:
  - **uKummer** and **qDSA** *: smarter use of algebraic geometry
    - software-only hyperelliptic cryptography on constrained platforms
    - on Cortex M0+, qDSA is ~50x faster ~10x smaller stack compared to ed25519 (tweetnacl implementation)
    - qDSA is available in RIOT as a package

- Stronger algorithms:
  - On-going **NIST post-quantum competition** (on low-end IoT: pqm4)

* J. Renes, B. Smith *‘qDSA: Small and Secure Digital Signatures with Curve-based Diffie–Hellman Key Pairs’*, ASIACRYPT 2017.

Agenda

- Context
- Improving IoT functionalities
- Mitigating risks: areas of work
  - (Trusting IoT Hardware)
  - Crypto Primitives for IoT
  - Secure IoT Software
  - Secure IoT Networking
  - IoT Software Updates
- RIOT-FP

Secure (IoT) Software: What of Open Source?

- Security by obscurity? Not much.
  - Thousand eyes are better than a couple (or none)

- Still, some vulnerability were in plain sight for years (e.g. Heartbleed)

Formally Verified IoT Software

- Producing more robust IoT code… without too much performance cost ?
  - Radical approach: **(re)implementation in specific language**
    - F* code then transformed and compiled in C with KReMLin
    - Kernel re-implementation in Rust*
  - Soft approach: **advanced static analysis of existing C code**
    - E.g. use formally verified analyzer Verasco
  - Middle-ground: **annotate existing C code, on which proofs are then possible**
    - Contiki linked-list module verified with Frama-C**

* A. Levy et al. '*Multiprogramming a 64 kB Computer Safely and Efficiently*,' ACM SOSP, 2017.
** A. Blanchard et al. '*Ghosts for Lists: A Critical Module of Contiki Verified in Frama-C*', NFM 2018

Formally Verified IoT Software

- Steps towards formally verified software modules in RIOT
  - **HACL crypto library: written in F* formal language**
  - F* code verified for
    - memory safety,
    - mitigations against timing side-channels,
    - functional correctness
  - F* code then compiled to readable C code with KReMLin, preserving proofs*

- Current work on optimizing HACL memory + speed of ed25519 signatures

* JK Zinzindohoué et al. *'HACL*: A verified modern cryptographic library,'* ACM CCS, 2017

Formally Verifying IoT Software

- D. Knuth 1977: "Beware of bugs in the above code; I have only proved it correct, not tried it." *
- SafeRTOS formally proven & certified: vulnerabilities found recently**
- C. Bormann 2017: " A security proof can be very useful — as useful as the model against which the proof was written."
- …

* http://www-cs-faculty.stanford.edu/~knuth/faq.html

** https://blog.zimperium.com/freertos-tcpip-stack-vulnerabilities-put-wide-range-devices-risk-compromise-smart-homes-critical-infrastructure-systems/

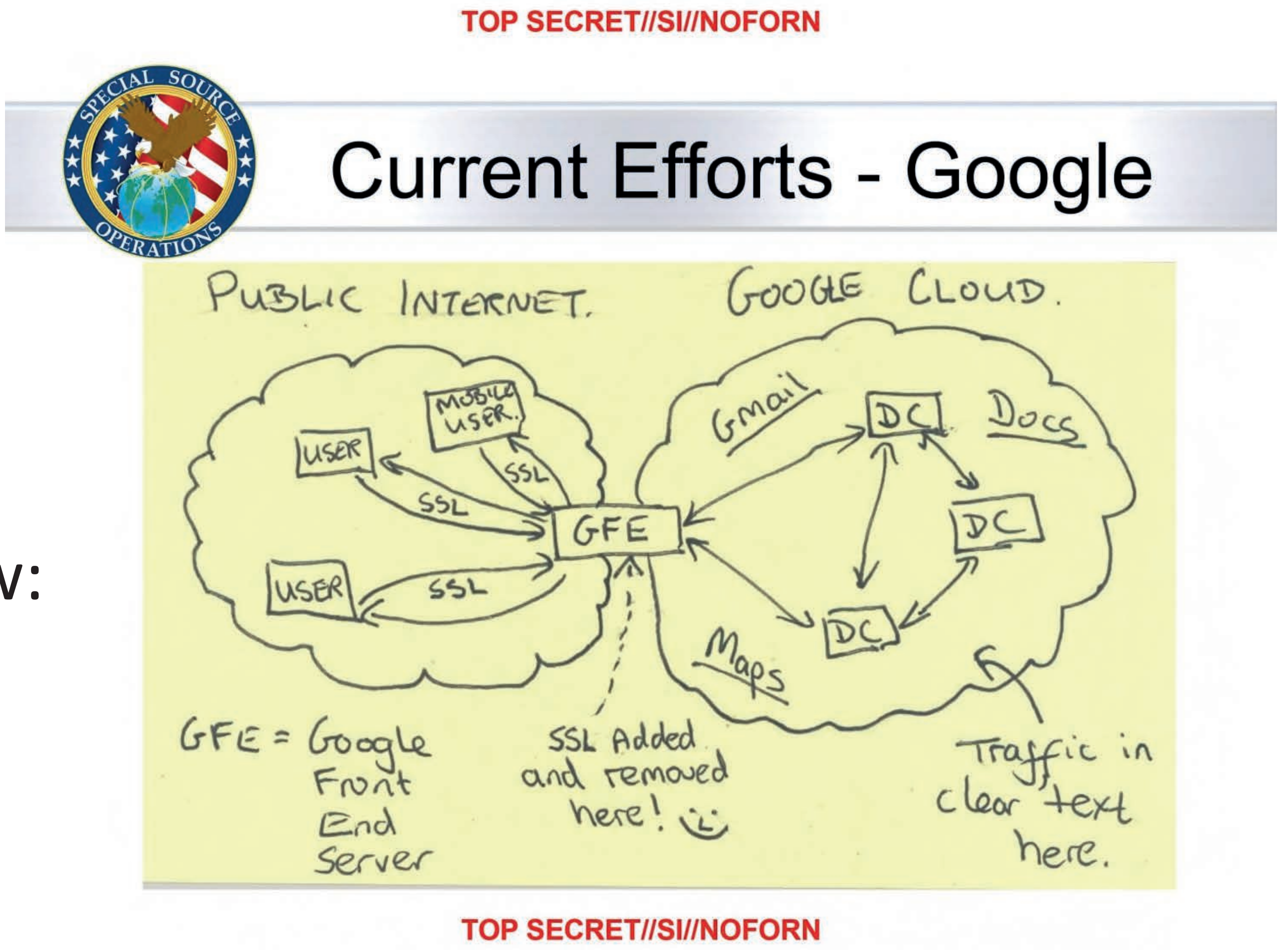
Agenda

- Context
- Improving IoT functionalities
- Mitigating risks: areas of work
  - (Trusting IoT Hardware)
  - Crypto Primitives for IoT
  - Secure IoT Software
  - Secure IoT Networking
  - IoT Software Updates
- RIOT-FP

Secure IoT Networking

- Recently: (D)TLS 1.3
  - hooray!
- Security at transport layer & below: necessary but not sufficient!
  - In IoT, proxies are to expected

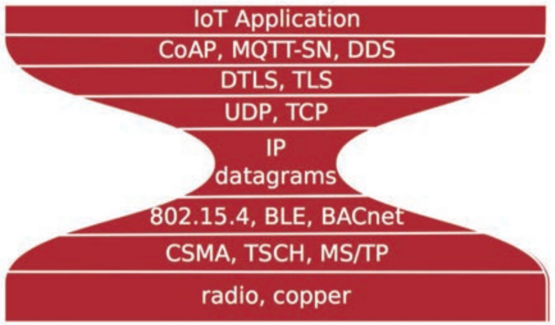
  ⇒(D)TLS termination!
- Object security is needed



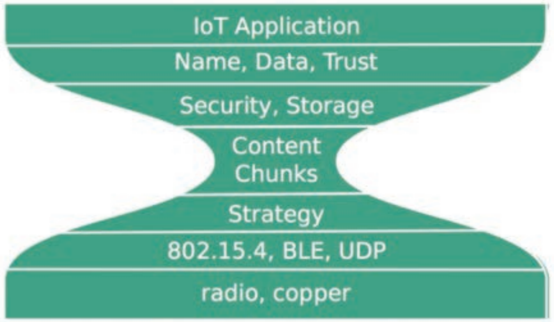
Secure IoT Networking

- Object security approaches:
  - Extending the standard 6LoWPAN/CoAP suite of protocols
    - COSE : CBOR consise serial data encryption and signature
    - OSCORE : in-layer security for CoAP over *foo*
    - …
  - Novel paradigms: named data networking on IoT*
    - instead of network focusing first on connecting machines, directly focus on accessing (named) data.
    - only two types of packets: *Interest* and *Data* (chunks)

    $\Rightarrow$ encryption etc. at *Interest* and *Data* level



IoT IP Stack



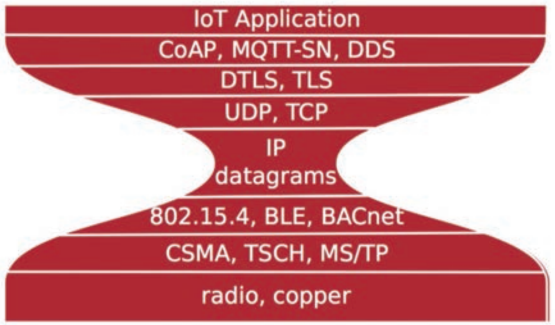
IoT NDN Stack

* E.Baccelli, C. Mehlis, O. Hahm, TC Schmidt, M. Wählisch, ''*Information centric networking in the IoT: Experiments with NDN in the wild*,'' in ACM ICN, Sept. 2014.

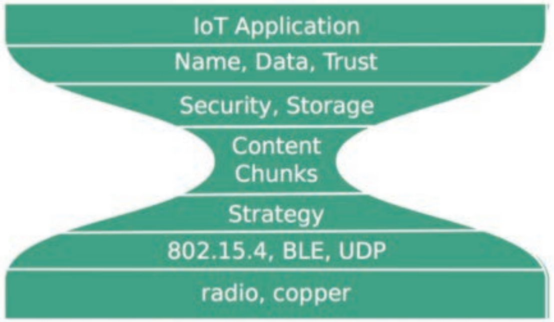
Secure IoT Networking

- Object security in RIOT
  - COSE supported
  - OpenWSN* stack support will bring OSCORE support
  - Experimental ICN stacks supported: CCN-lite, NDN-RIOT

- Next towards secure bootstraping in OpenWSN + RIOT
  - Zero-configuration, secure network join
  - PKI for low-end IoT



IoT IP Stack



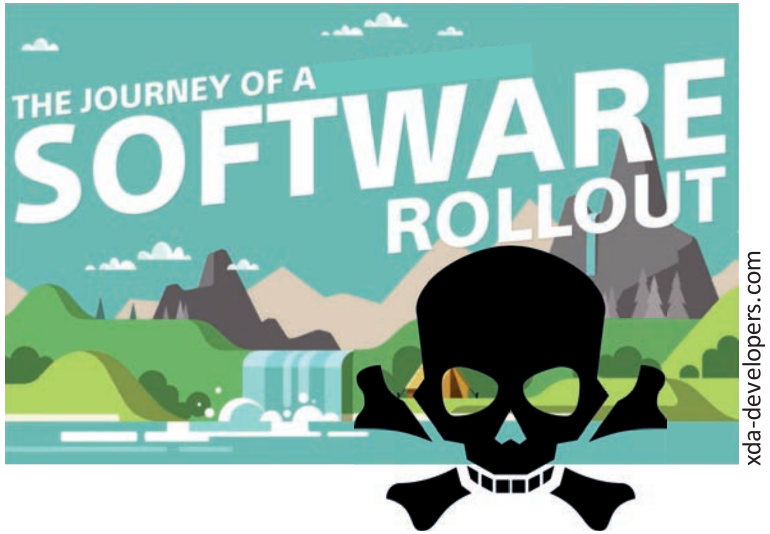
IoT NDN Stack

* T. Watteyne, et al. "*OpenWSN: a standards-based low-power wireless development environment*." Transactions on Emerging Telecommunications Technologies, 2012.

Agenda

- Context
- Improving IoT functionalities
- Mitigating risks: areas of work
  - (Trusting IoT Hardware)
  - Crypto Primitives for IoT
  - Secure IoT Software
  - Secure IoT Networking
  - IoT Software Updates
- RIOT-FP

Secure IoT Software Updates



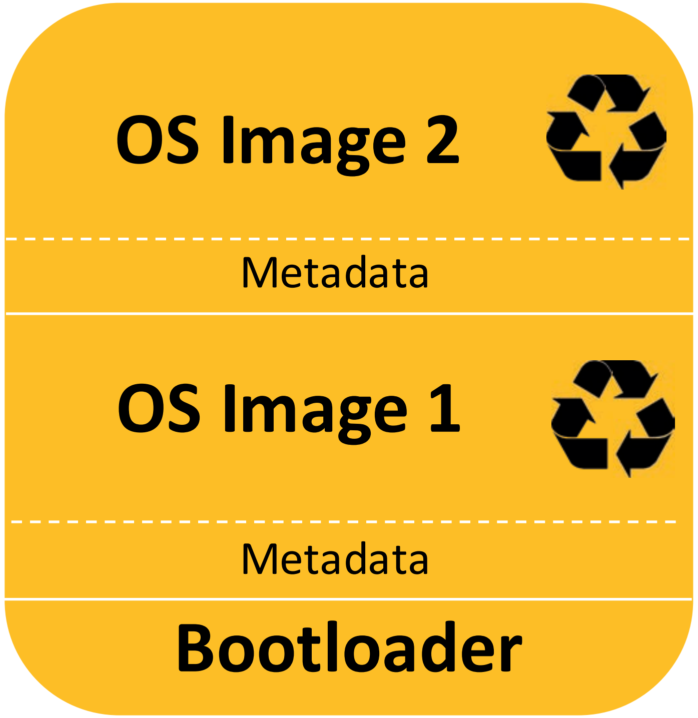
- Internet age: you can't secure what you can't update!
- Internet age: software updates are an attack* vector!

⇒ Enabling legitimate software updates is crucial & difficult

* www.wired.com/story/ccleaner-malware-supply-chain-software-security

Full IoT Software Updates: Firmware Updates

You thought you were tight w.r.t. memory?



Memory must be further split :

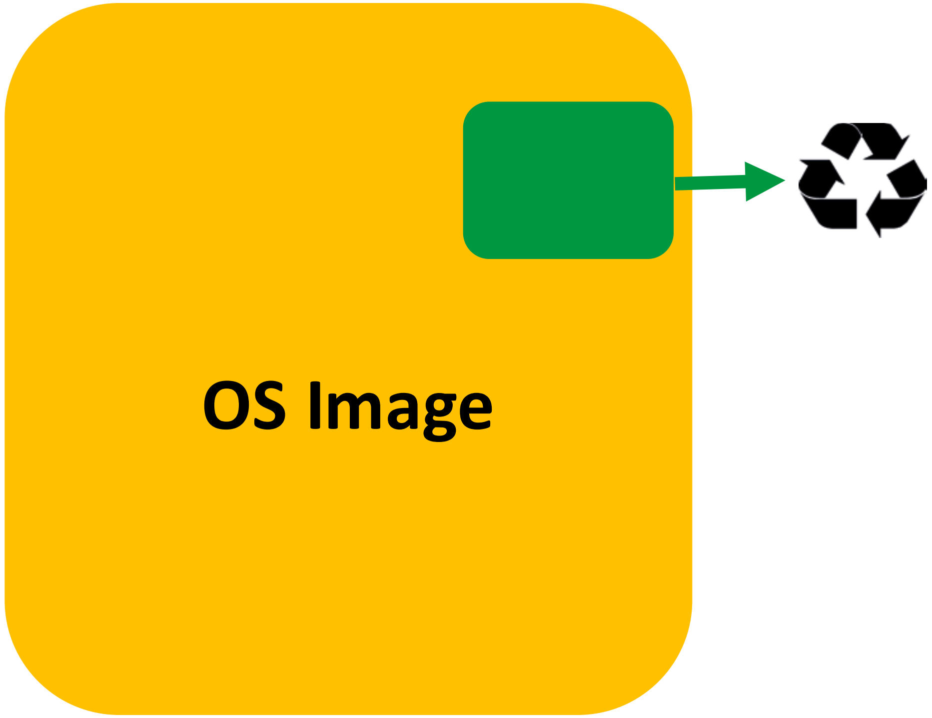
- Bootloader
  - (e.g. minimalistic config or RIOT)
- Several OS Images
  - Typically need >2 for roll-back
- Metadata
  - IETF working group SUIT working on standardizing metadata for IoT firmware *

* https://datatracker.ietf.org/group/suit/documents/

Partial IoT Software Updates, Multi-Stakeholder

1. Firmware updates: efficient ?
2. Multiple modules / stakeholders ?

   ⇒ need partial update



- **differential updates** of patching the binary

⇒ Efficient but risky

- **dynamic loading** of binary modules

⇒ More robust but more complex

- use **interpreted languages** (instead of compiled)

⇒ Elegant but interpreter overhead

Secure IoT Software Updates

- Steps towards secure RIOT updates
  - RIOT-based prototype of SUIT-compliant IoT software updates appeared this year *

⇒ next: contribute to SUIT standardization based on our experiments

  - Runtime .js container demonstrated to work on Cortex-M based low-end IoT devices with RIOT **

⇒ next: secure with COSE and explore sandboxing of this contruct

* SUIT 2018 Berlin Hackathon https://github.com/suit-wg/Hackathon-Interim-Berlin/blob/master/SUIT-Berlin-Hackathon-recap.pdf

** E. Baccelli et al. "*Scripting Over-The-Air: Towards Containers on Low-end Devices in the Internet of Things*," IEEE PerCom, 2018.

Agenda

- Context
- Improving IoT functionalities
- Mitigating risks: areas of work
  - (Trusting IoT Hardware)
  - Crypto Primitives for IoT
  - Secure IoT Software
  - Secure IoT Networking
  - IoT Software Updates
- RIOT-FP

RIOT-FP: Towards Future-Proof IoT Software

The RIOT-FP project aims to combine:

- **RIOT**: efficient, open source, deeply embedded IoT software
- **Next generation IoT crypto** primitives: small, fast and future-proof
- **Secure IoT networking**, secure bootstraping, and open protocol specs
- **Formally verified perimeter** for software within RIOT

- **Prototype securing IoT software maintainance** on low-end devices

Conclusions & Main Take-away

- IoT risk *vs* functionality tradeoff must improve

- Functionalites improving faster than security

- Security for IoT in practice means combining:
  - Open source (necessary but not sufficient)
  - Formally proven code (useful but not sufficient)
  - IoT crypto primitives (smaller, faster, stronger)
  - Secure IoT protocols above transport layer (TLS good but not sufficient)
  - Secure IoT software updates are necessary (but not easy)

Thanks! Questions?

Later per email: emmanuel.baccelli@inria.fr