Secure IoT Device Lifecycle Management

Aaron Ardiri, CEO RIoT Secure

Jfokus 2015

Feasibility of Security in Micro-Controllers

Aaron Ardiri

Chief Technology Officer - Evothings AB

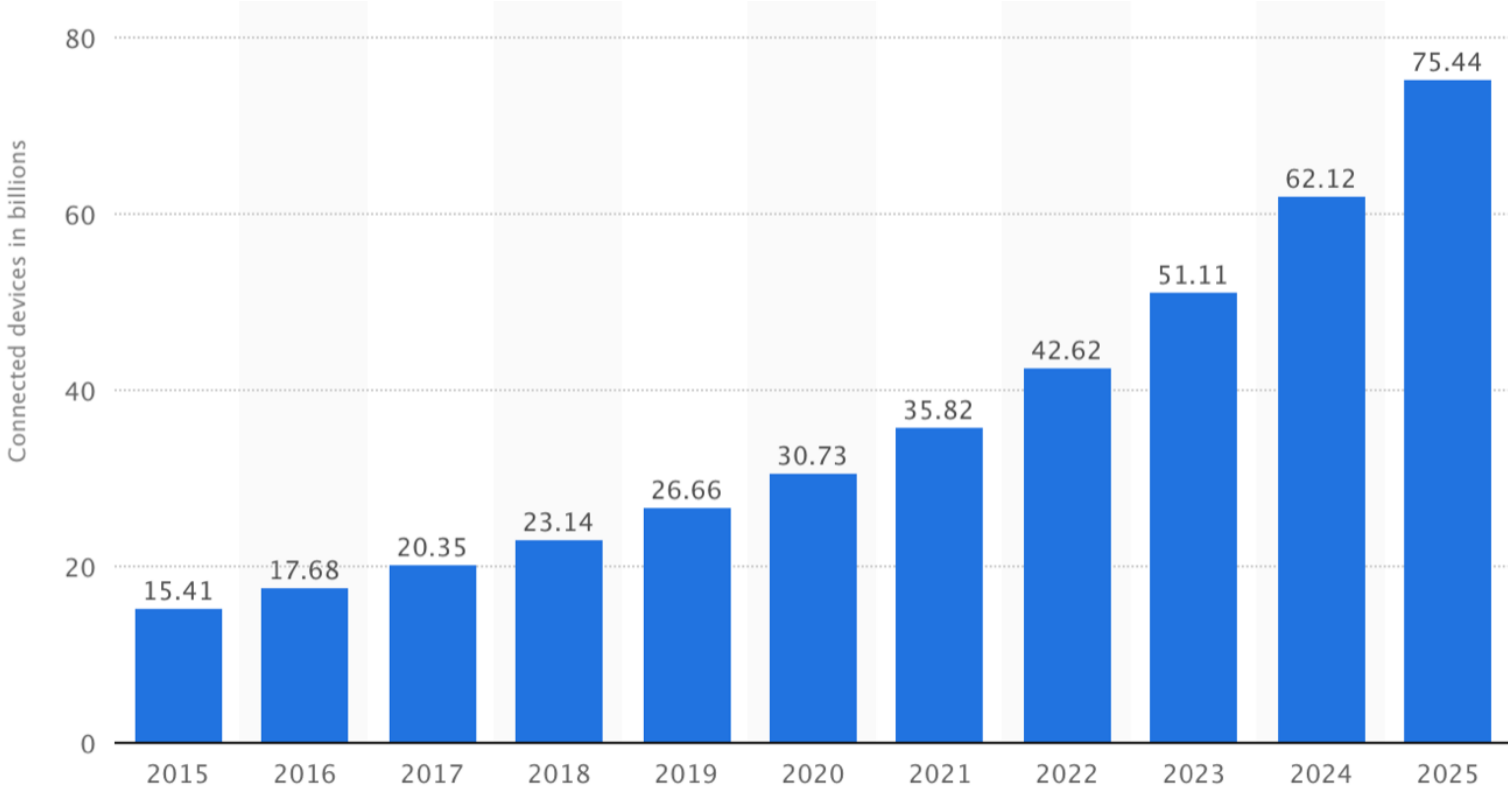
jFokus IoT, Stockholm

3rd February, 2015

https://www.jfokus.se/iot15/talks.jsp#Isitpossibletosecure

THE PROBLEM

75 Billion Devices by 2025



ref: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

Market Analysts

TOP 10 IoT Technologies for 2017 / 2018

- IoT Security
- IoT Analytics
- IoT Device Management
- Low-Power, Short-Range Networks
- Low-Power, Wide-Area Networks
- IoT Processors
- IoT Operating Systems
- Event Stream Processing
- IoT Platforms
- IoT Standards and Ecosystems

Gartner

ref: https://www.gartner.com/webinar/3435117

IoT Security Audit

REPORT AFTER SECURITY AUDIT OF 100 IOT SOLUTIONS

The Top #5 most common vulnerabilities found in connected objects:

1. unsecured updates: no encryption or signature for firmware updates
2. use of default keys and passwords: even in production environment
3. unsecured communications: weak or no encryption and integrity checks
4. data stored in plain text: no encryption used for local data storage
5. presence of debugging interfaces (UART, USB) on production hardware

ref: https://econocom.com/en/news/communiques-de-presse/report-after-security-audit-100-iot-solutions-digital-security

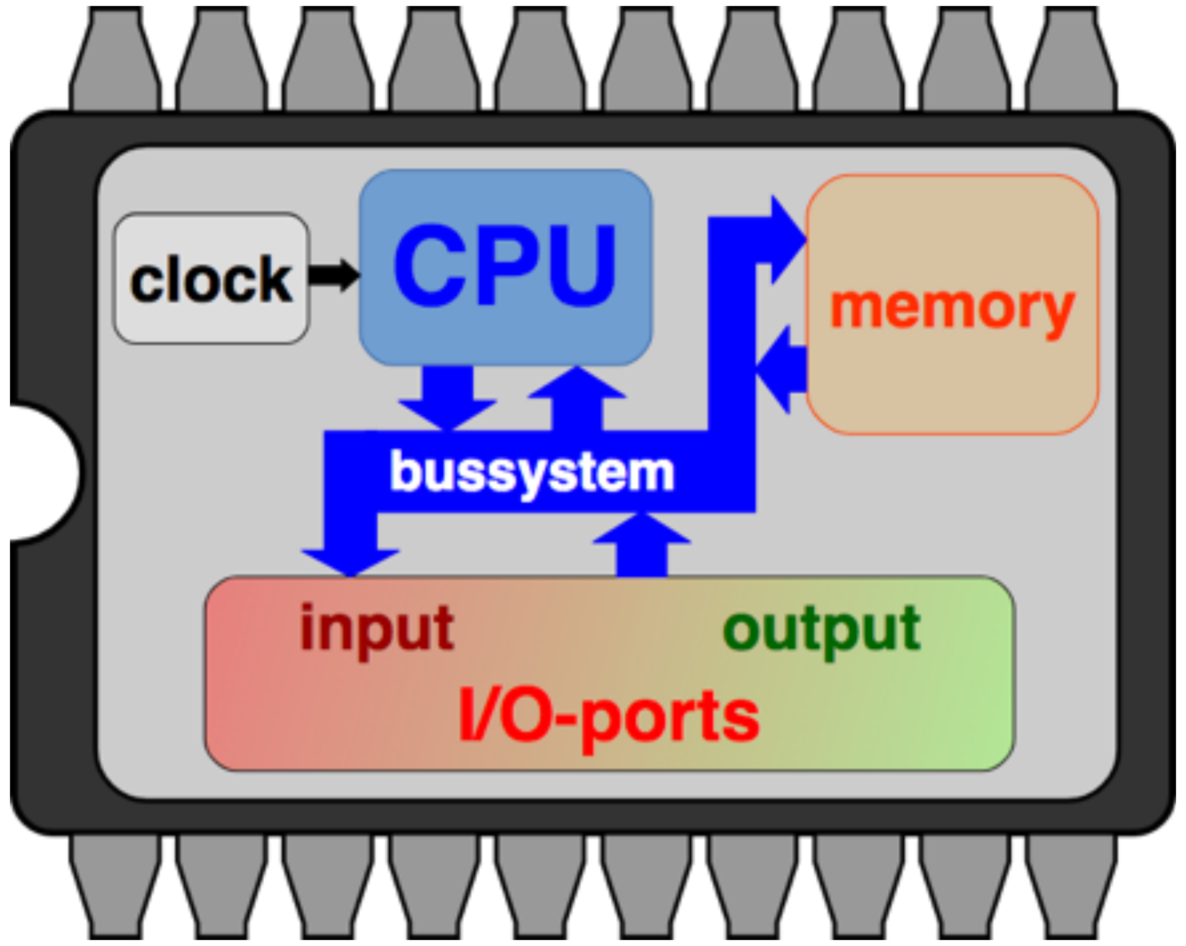
THE REASON

As hackers find new ways to attack IoT devices and protocols, long-lived things may need updatable hardware and software to adapt to their lifespan.

Nick Jones, Gartner Analyst

IoT Devices

Super Computers



Micro Controllers

Resource Constraints

Arduino UNO
16Mhz 2KB RAM 32KB FLASH

First of all, computation of RSA1024 key is quite CPU intensive, so it will require up to 30 mins to perform single RSA1024 encryption on a 16Mhz micro-controller. … coupled with RAM requirements - it seems RSA1024 on Arduino UNO - it is just not technically possible :)

ref: linkedin message from offshore recruitment house

possible - absolutely; feasible - probably not in practice

ref: https://evothings.com/is-it-possible-to-secure-micro-controllers-used-within-iot/

Developer Skills Shortage

Computer Programming has changed since the first computers; with murphy's law making entry level computers more powerful with unlimited resources - languages have severed hardware links

- 1950's Autocode, Fortran
- 1960's Algol
- 1970's Pascal, C
- 1980's C++, Perl, BASIC
- 1990's Python, Java
- 2000's C#, JavaScript
- 2010's Swift, Clojure, Scala

"640K ought to be enough for anybody."

ref: 1981, Bill Gates - founder of Microsoft

ref: https://www.thesoftwareguild.com/blog/history-of-programming-languages/

Fragmentation

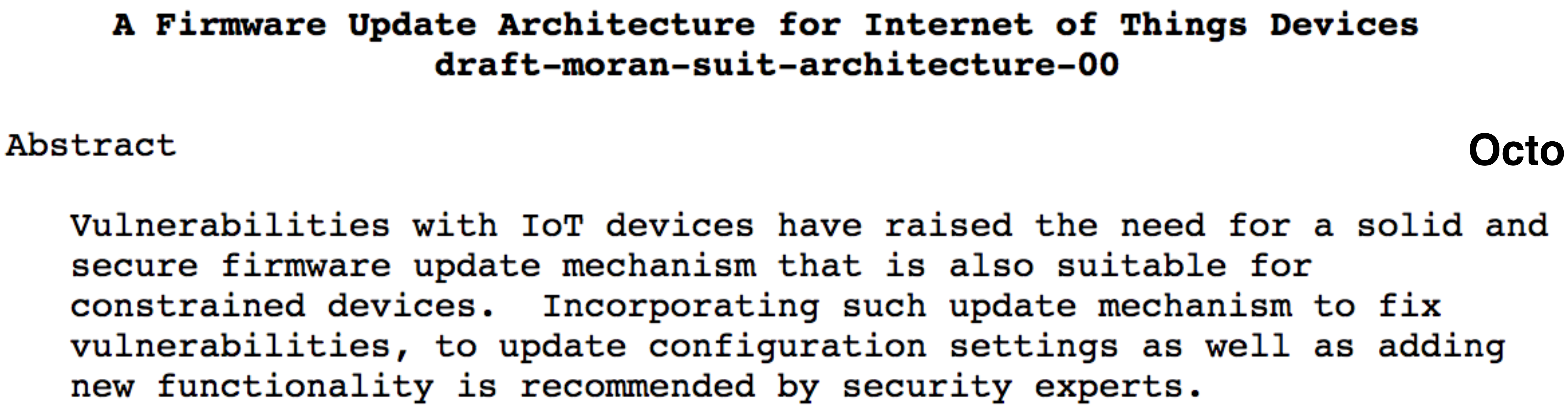
Do you think android fragmentation is bad?

Fragmentation?

Just wait until you see all the various micro-controllers, communications technology and extreme variety of sensor and actuators hardware and protocols that exist on the market. Many firmwares are written in C or assembler; unless an an interpreted / scripted environment is available to the developer; adding another layer of security vulnerabilities.

Dealing with IoT in a device agnostic manner will ease management of such devices.

Lack of Standards



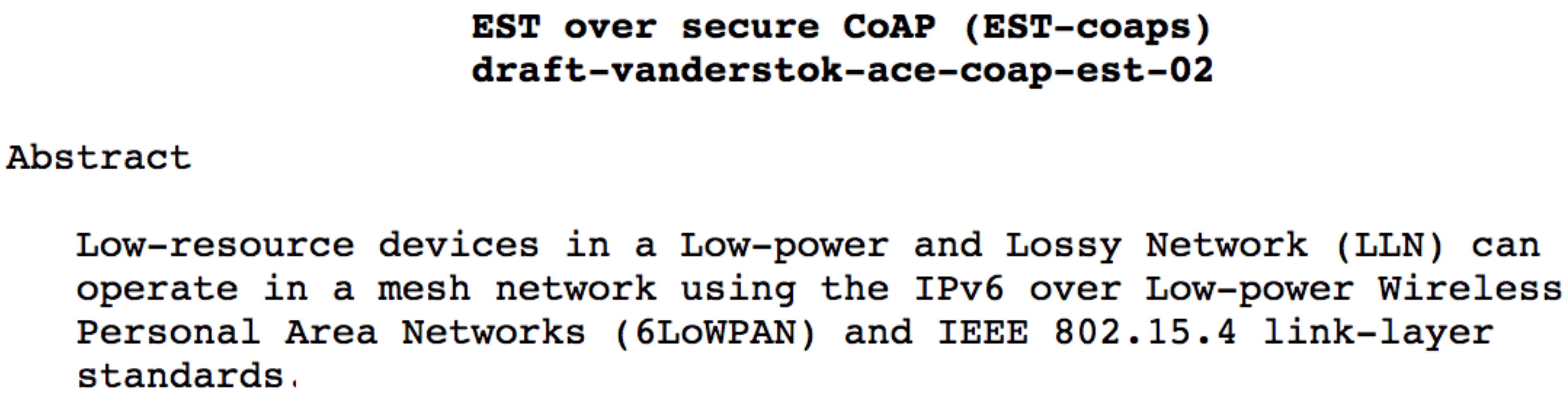
A Firmware Update Architecture for Internet of Things Devices
draft-moran-suit-architecture-00

Abstract

Vulnerabilities with IoT devices have raised the need for a solid and secure firmware update mechanism that is also suitable for constrained devices. Incorporating such update mechanism to fix vulnerabilities, to update configuration settings as well as adding new functionality is recommended by security experts.

October 30, 2017

ref: https://tools.ietf.org/html/draft-moran-suit-architecture-00



EST over secure CoAP (EST-coaps)
draft-vanderstok-ace-coap-est-02

Abstract

Low-resource devices in a Low-power and Lossy Network (LLN) can operate in a mesh network using the IPv6 over Low-power Wireless Personal Area Networks (6LoWPAN) and IEEE 802.15.4 link-layer standards.

June 12, 2017

ref: https://tools.ietf.org/html/draft-vanderstok-ace-coap-est-02

0-day Vulnerabilities

code injection

object use after free

local privilege escalation

buffer overflows

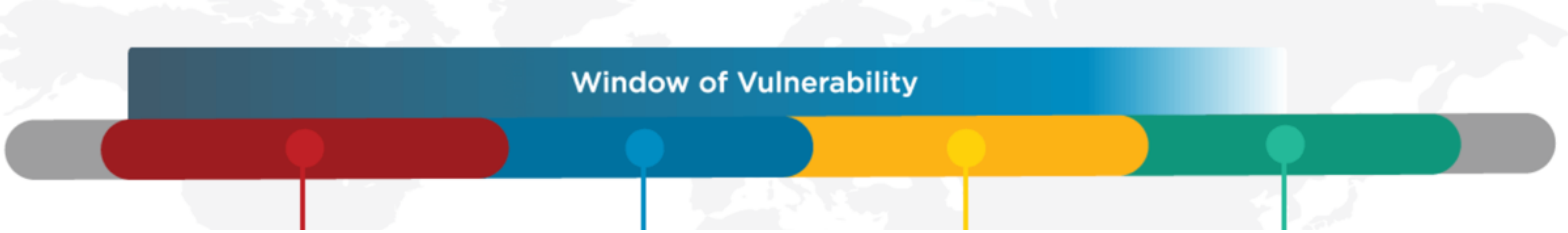
Linux

Raspberry Pi

beaglebone

intel Edison

Onion



discovery

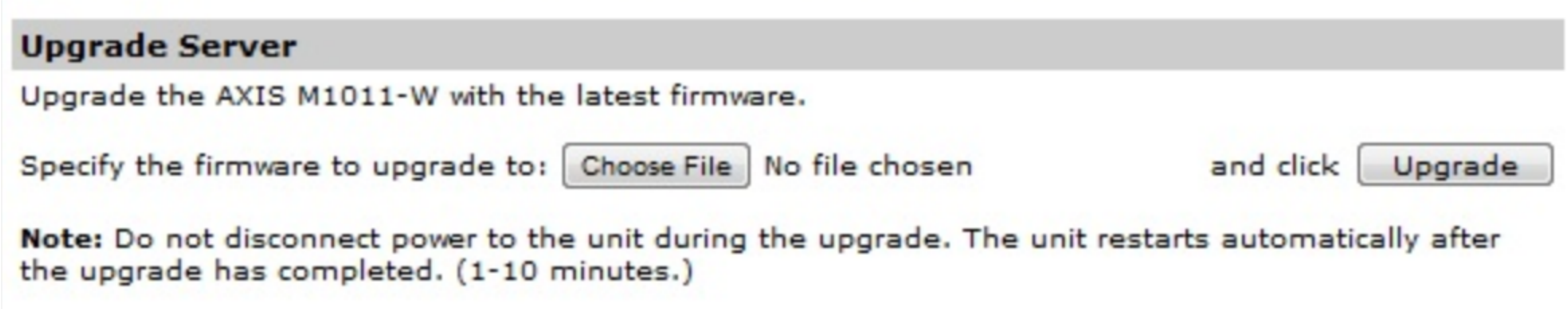
disclosure

vendor fix

patch rollout

Security = Users Problem

Product manufacturers have jumped blindly into the IoT ecosystem providing products with default passwords, passwords or firmware that are not changeable or require the end user to have technical expertise to install updates to the products themselves.



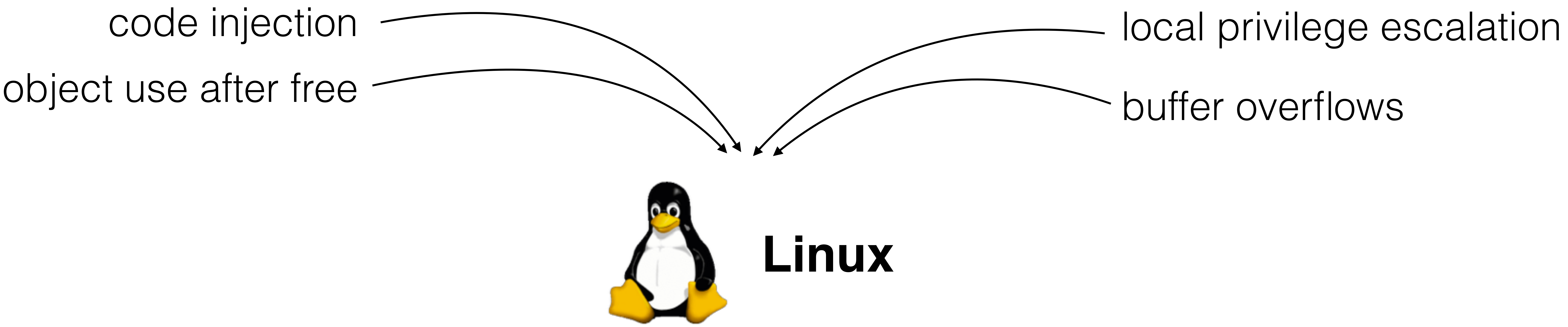
Upgrade Server

Upgrade the AXIS M1011-W with the latest firmware.

Specify the firmware to upgrade to: Choose File No file chosen and click Upgrade

Note: Do not disconnect power to the unit during the upgrade. The unit restarts automatically after the upgrade has completed. (1-10 minutes.)

0-day Vulnerabilities



code injection

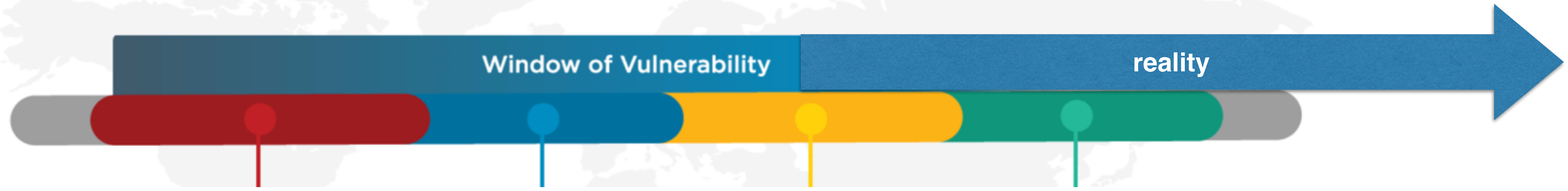
object use after free

local privilege escalation

buffer overflows

Linux

Raspberry Pi   beaglebone   intel Edison   Onion



discovery   disclosure   vendor fix   ~~patch rollout~~

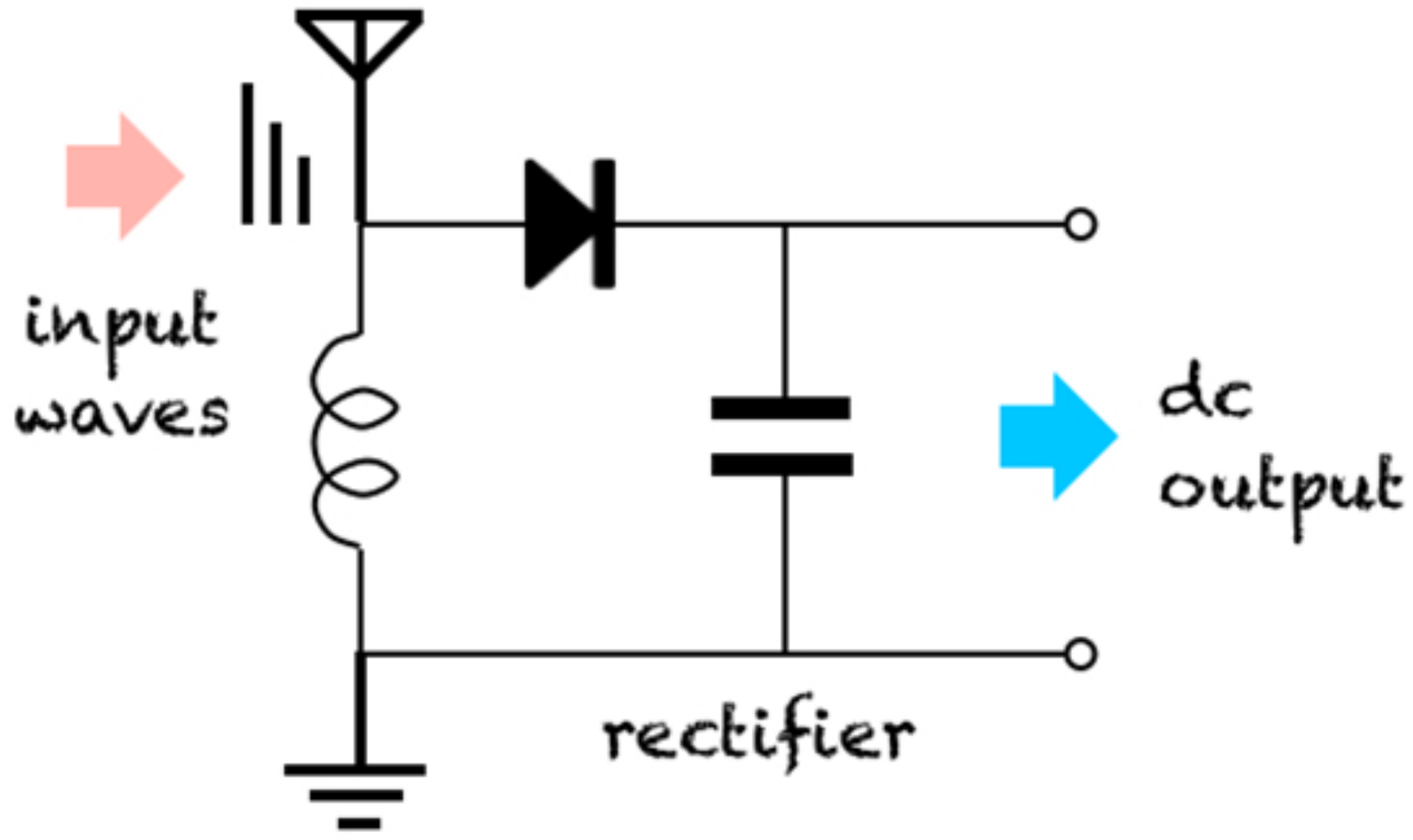
Energy Harvesting

Current Status

- considered a ‘black magic’ realm
- use ambient fields to produce power
- limited or low power throughput

Implications

- standard power is not be available everywhere
- battery solutions are expensive; harvesting is required
- low powered, resource constrained devices are here to stay



Buzzword Bingo

Sales Reps have ruined User Perception

- ‘years of battery life’
- ‘secure’ with ‘OTA updates’
- ‘IPv6 communication’, ‘mesh networking’
- automatic network association

Solutions = Compromise

- it is impossible to tick all check boxes
- each vertical will have different requirements
- real life is completely different from sales pitch

LoRa / Sigfox != Solution

Features

- long range, low power, low cost
- end-to-end encryption (XOR, AES, pre-shared keys)
- designed for low power devices that have basic sensors

Problems

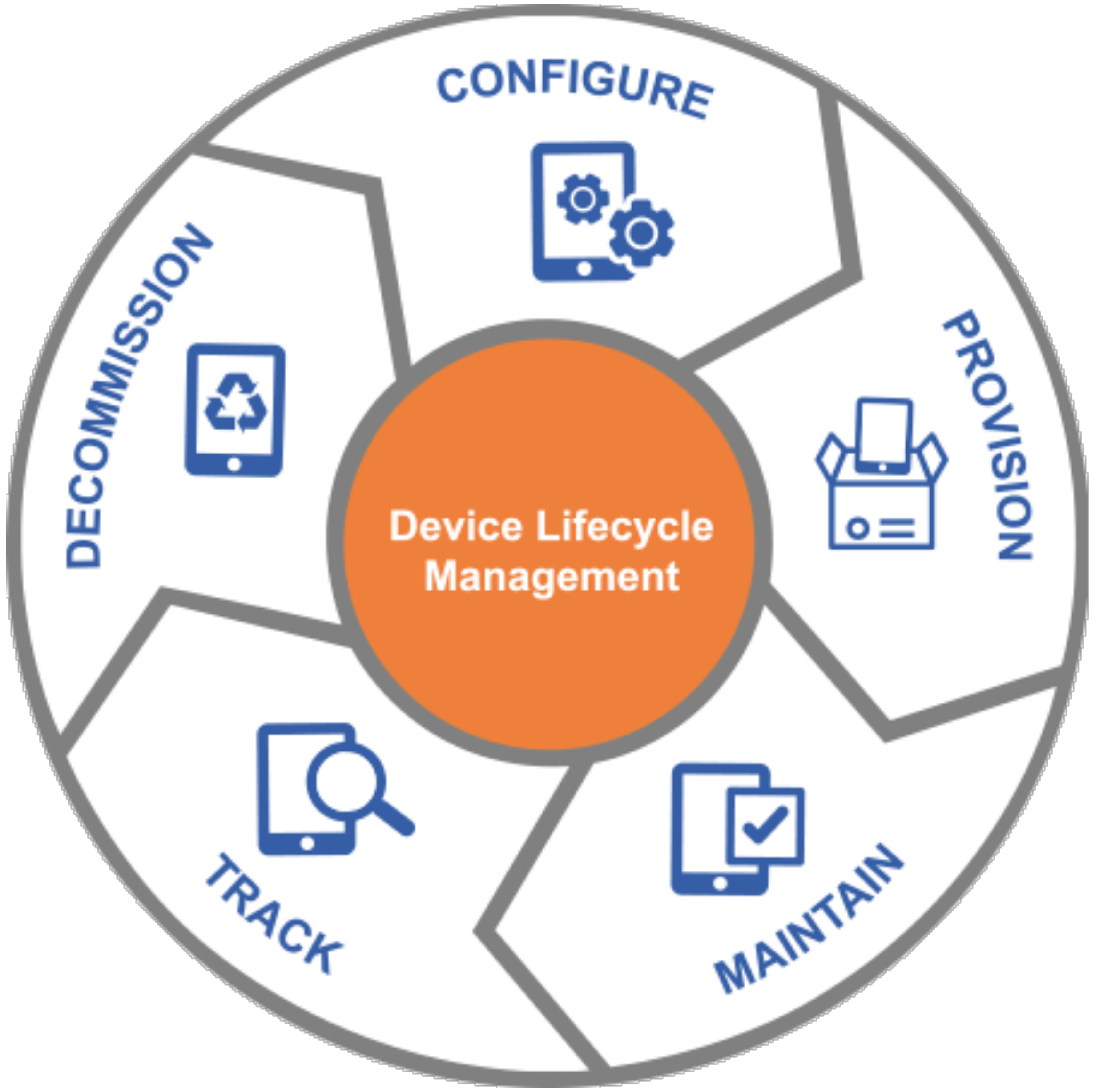
- extremely limited bandwidth, (x times day, max y bytes) - shared gateways
- no established network; user must provide gateways, frequency based on region
- bi-directional communication* - uplink/downlink not truly designed for it
- impossible to implement firmware updates OTA, due to limited bandwidth

WHAT’S THE SOLUTION?

Experienced IoT security specialists are scarce.

Nick Jones, Gartner Analyst

Secure Device Lifecycle

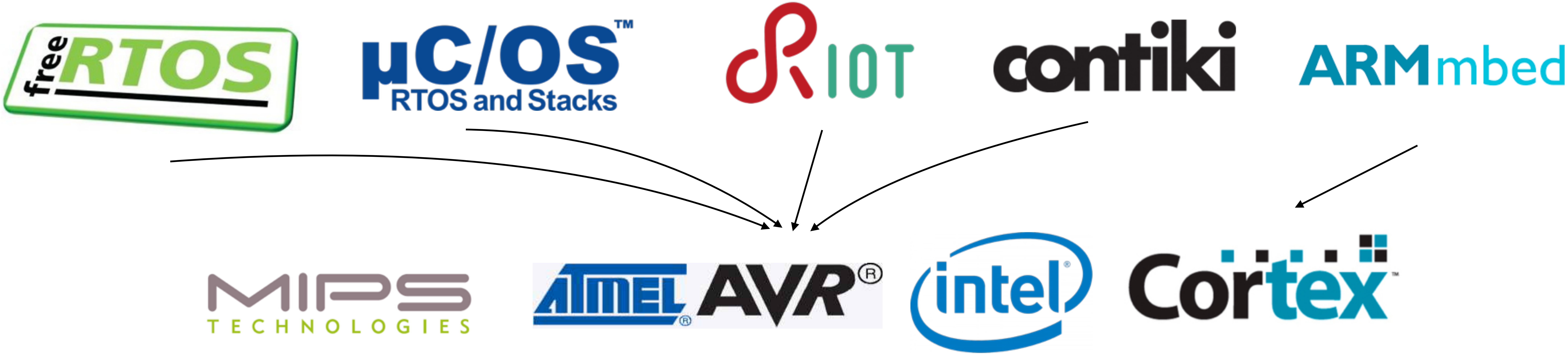


Benefits of Device Lifecycle Management

- provision devices with ease
- know your devices (status, whereabouts)
- improve asset utilisation
- reduce downtime and keep devices updated
- focus on customer experience/satisfaction
- fix the spark before it catches fire
- data monitoring and usage analytics
- decommission or reassign the devices purpose

Barebones or RTOS?

The development of IoT clients should be done as close to the hardware as possible - utilising “barebone” development practices or use thin streamlined RTOS that enable networking and/or multithreading capabilities on the underlying target hardware. No additional services to be exposed in a production environment to maximise security.



Less can be More

There is a notion in software development industry that there is a direct correlation between the number of lines in a product and the number of software bugs that exist. Open source projects tend to lack consistent coding styles and documentation.

A. **Industry Average:**
"about 15 - 50 errors per 1000 lines of delivered code."

B. **Microsoft Applications:**
"about 10 - 20 defects per 1000 lines of code during in-house testing, and
0.5 defect per 1000 lines of code in released product."

C. **Harlan Mills pioneered 'cleanroom development'**,
"as low as 3 defects per 1000 lines of code during in-house testing, and
0.1 defect per 1000 lines of code in released products.

ref: https://www.mayerdan.com/ruby/2012/11/11/bugs-per-line-of-code-ratio

Keep It Simple Stupid

WHAT IS OUR SOLUTION?

> **Vendors selling tools derived from MDM are inexperienced in IoT and may not provide appropriate pricing models**
>
> Nick Jones, Gartner Analyst

IoT Platform (PaaS)

Solution and Technology Overview

- secure end-to-end device lifecycle management solution with OTA update capability
- industry accepted cryptographic algorithms and cloud ready scalable architecture
- REST API for independent vendor integration and future expansion opportunities
- designed for IoT and resource constrained devices (memory, computational power)
- device and platform agnostic, IoT virtual machine for application development
- validated by independent third party IoT developers and solution providers
- built entirely from scratch for minimal code base maximal efficiency
- full ownership of intellectual property (IP), no third party components used

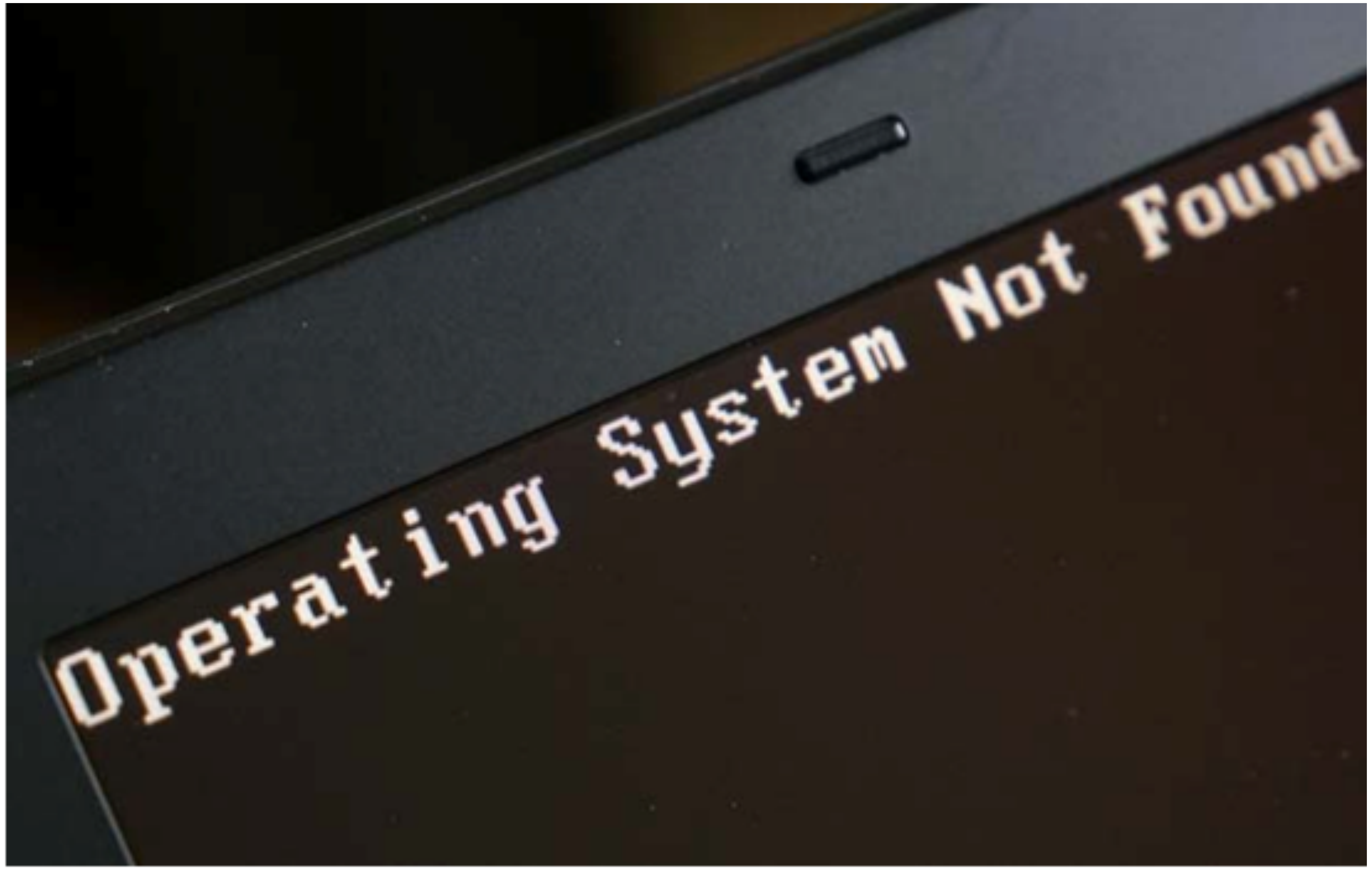
Barebones / RTOS

No Operating System

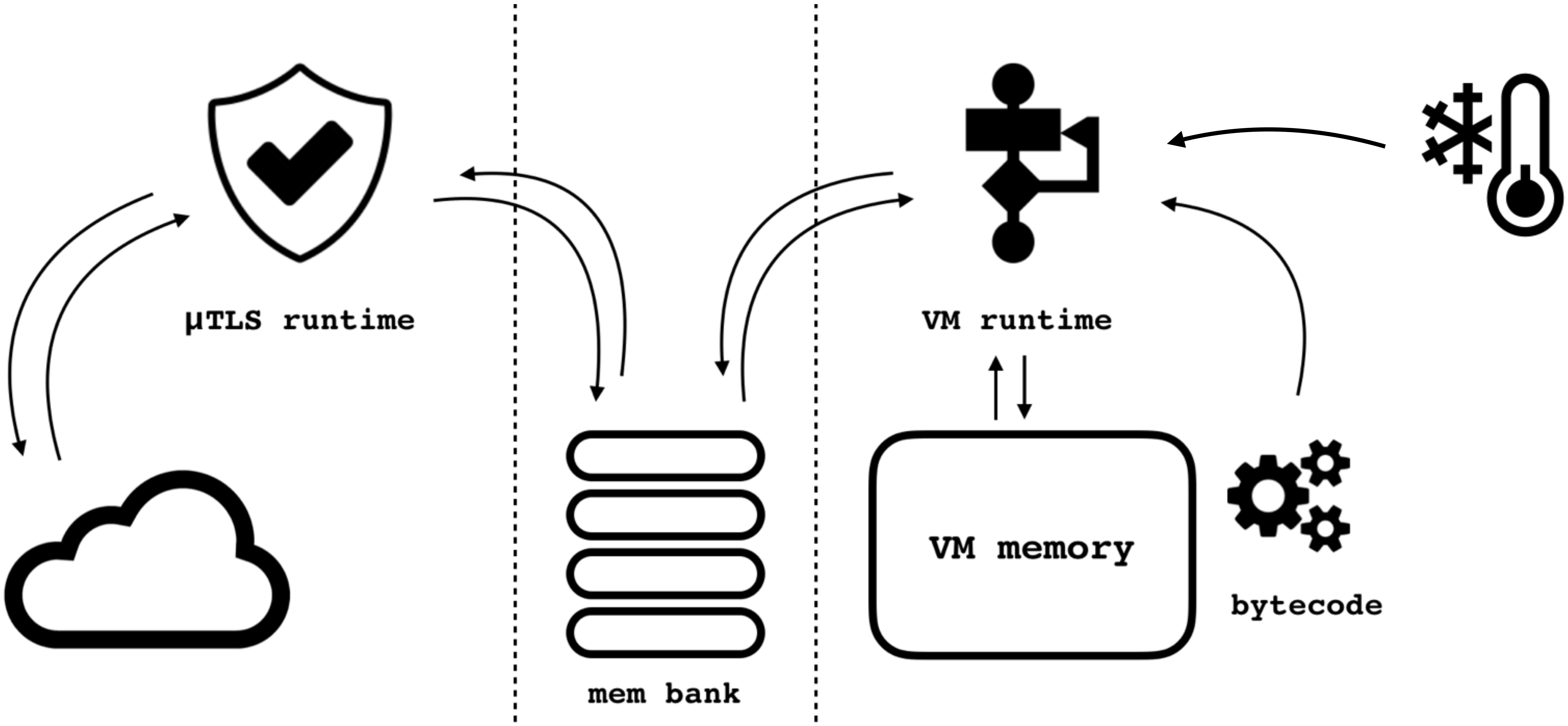
- firmware development
- minimise attack vectors
- removes 0-day vulnerabilities

Main Challenges

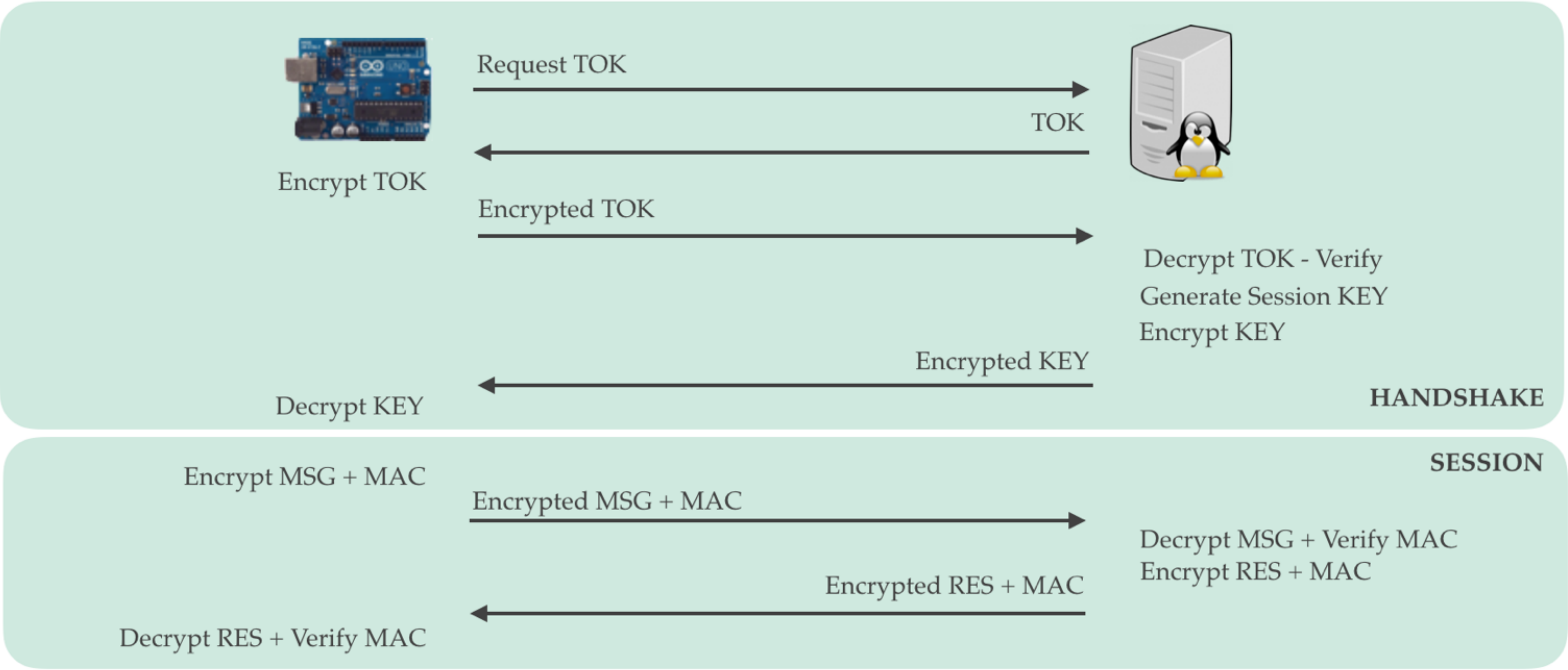
- very device specific
- build everything from scratch
- threading/multi tasking options



Core: Standalone



Secure Comms (μTLS)



Request TOK

TOK

Encrypt TOK

Encrypted TOK

Decrypt TOK - Verify
Generate Session KEY
Encrypt KEY

Encrypted KEY

Decrypt KEY

HANDSHAKE

SESSION

Encrypt MSG + MAC

Encrypted MSG + MAC

Decrypt MSG + Verify MAC
Encrypt RES + MAC

Encrypted RES + MAC

Decrypt RES + Verify MAC

IoT Bytecode

```
CONST8, OUTPUT,
PINMODE, 13,    // pinMode(13, OUTPUT);
HALT,

CONST8, HIGH,
DIGOUT, 13,     // digitalWrite(13, HIGH);
CONST8, 1,
DELAYS,         // delay(1000);
CONST8, LOW,
DIGOUT, 13,     // digitalWrite(13, LOW);
CONST8, 1,
DELAYS,         // delay(1000);
HALT
```

IoT OPCODE (assembler)

IoT Compiler

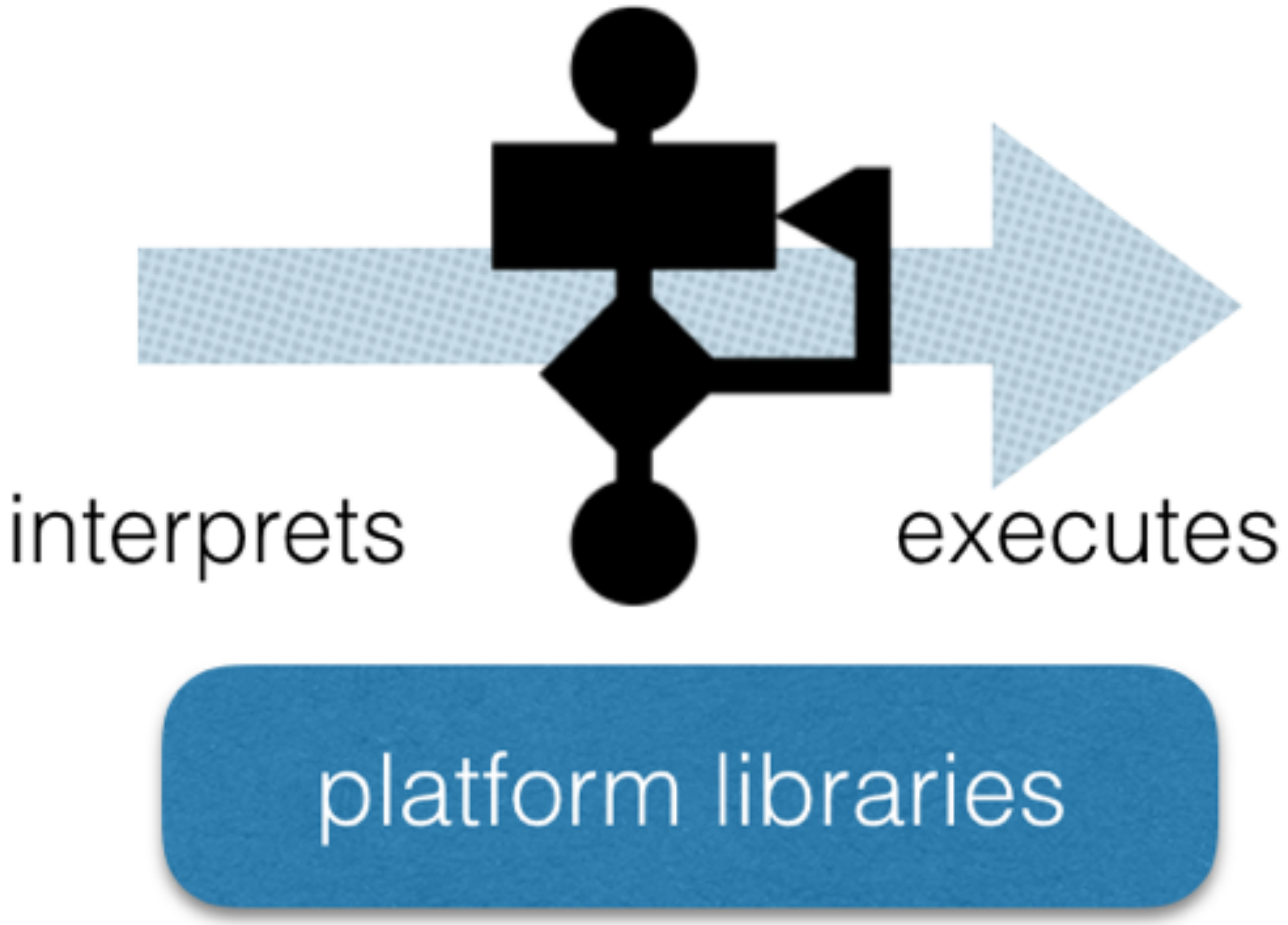
```
08 01 28 13
35


08 01 29 13
08 01 33
08 00 29 13
08 01 33
35

(20 bytes)
```

IoT BYTE CODE

IoT Virtual Machine

```
08 01 28 13
35


08 01 29 13
08 01 33
08 00 29 13
08 01 33
35

(20 bytes)
```

IoT BYTE CODE

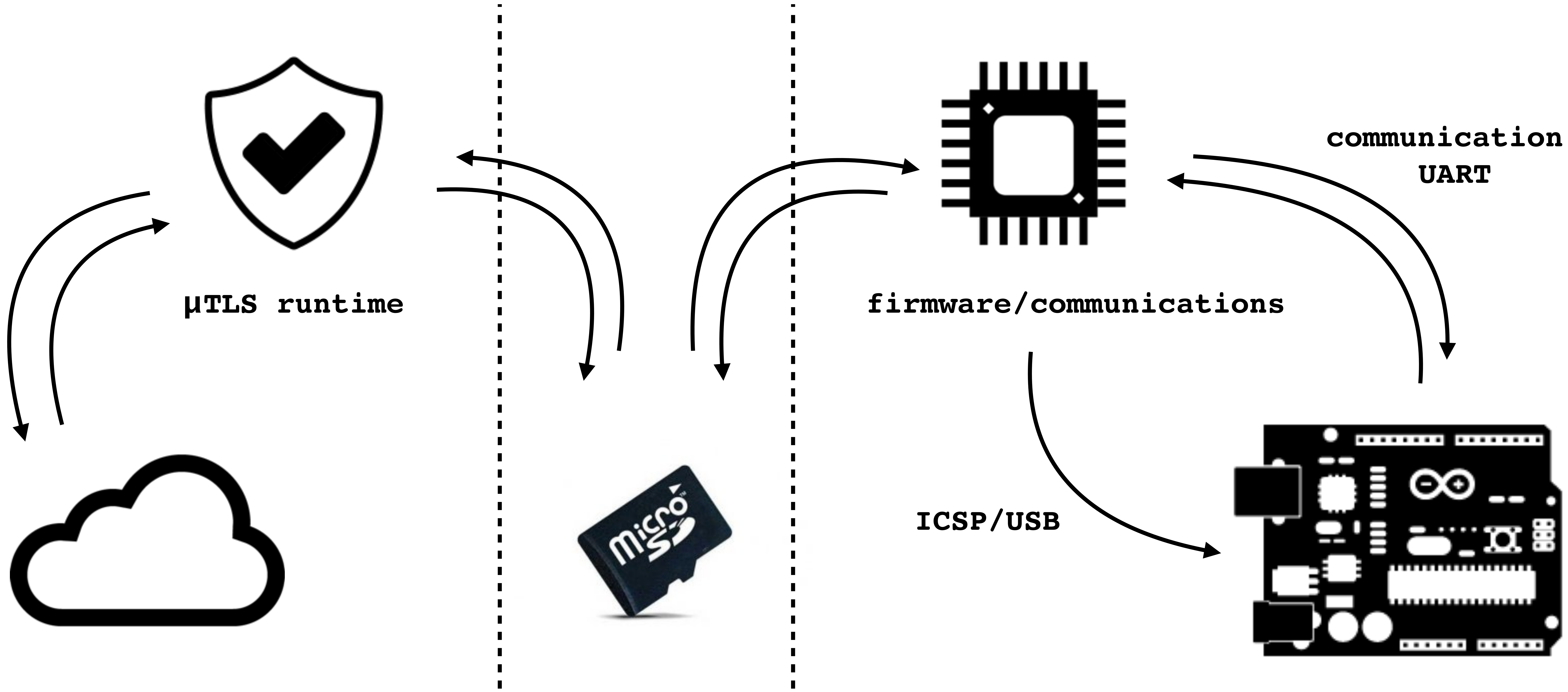


IoT Virtual Machine

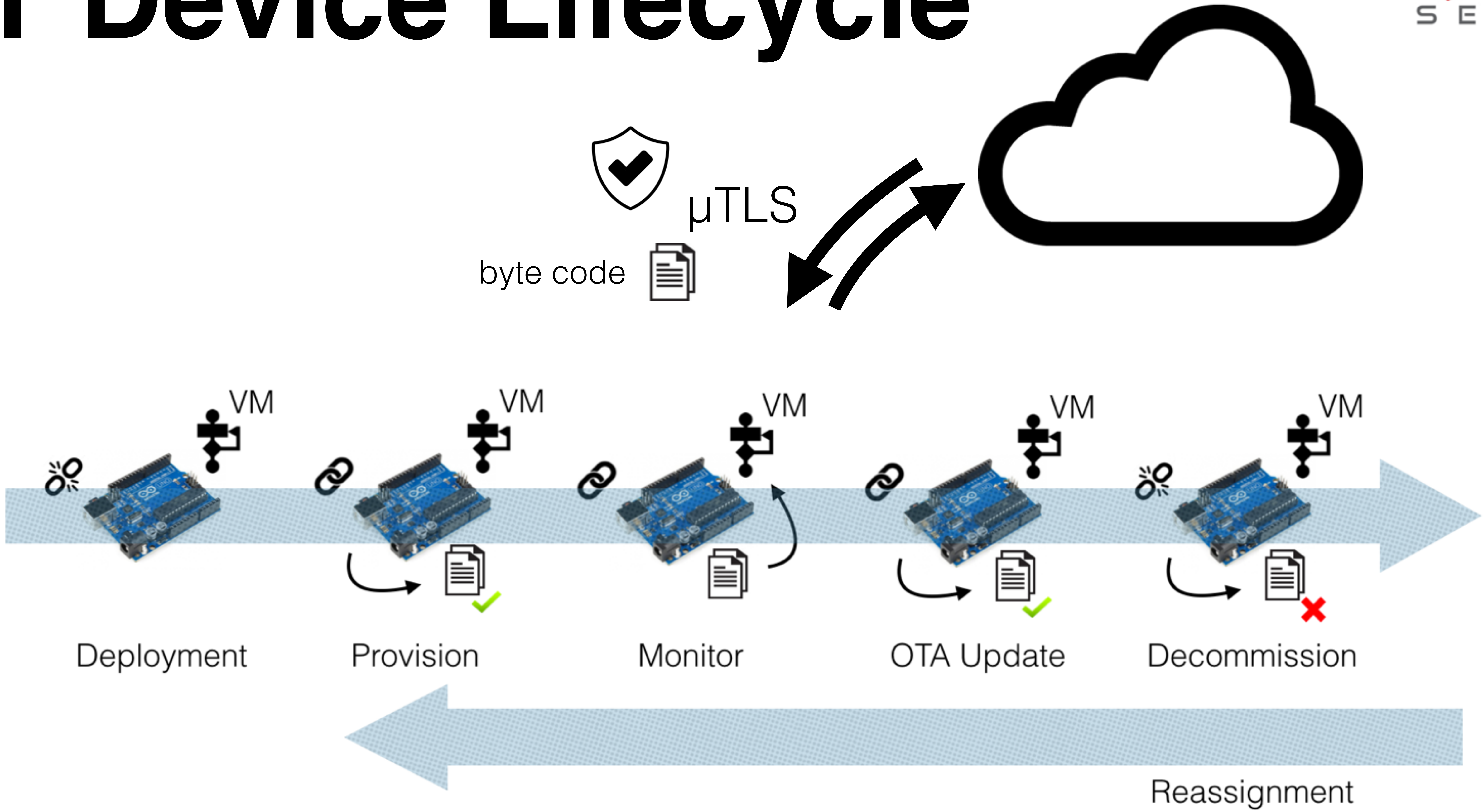
```
void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

Arduino C++

Core: Complimentary



IoT Device Lifecycle



RIoT Secure

Value Proposition

- securely track, manage and have and instant overview an control of IoT deployment
- reduce threat of cyber attacks and 0 day vulnerabilities due to underlying OS
- ensure sensitive sensor data is secure and sourced from a trustworthy location
- IoT bytecode encapsulated in secure delivery parcel ensuring no MiTM attacks
- IoT virtual machine provides secure developer sandbox to maximise product security
- software developers can implement IoT device functionality independent of platform
- architectured to co-exist with existing MDM solutions and third party services

DEMO : Arduino (LIVE)

THANK YOU

Contact Information

Aaron Ardiri
Chief Executive Officer

aaron.ardiri@riotsecure.se

🇸🇪 +46 70 656 1143
🇦🇺 +61 43 234 8856

http://www.riotsecure.se/ **http://www.riotsecure.se/blog/**