UNIVERSIDAD
COMPLUTENSE
MADRID

Firmware Updates (OTA)

NP-2

Based on "Software update for IoT" from Chris Simmonds (2net)

Why is it necessary?

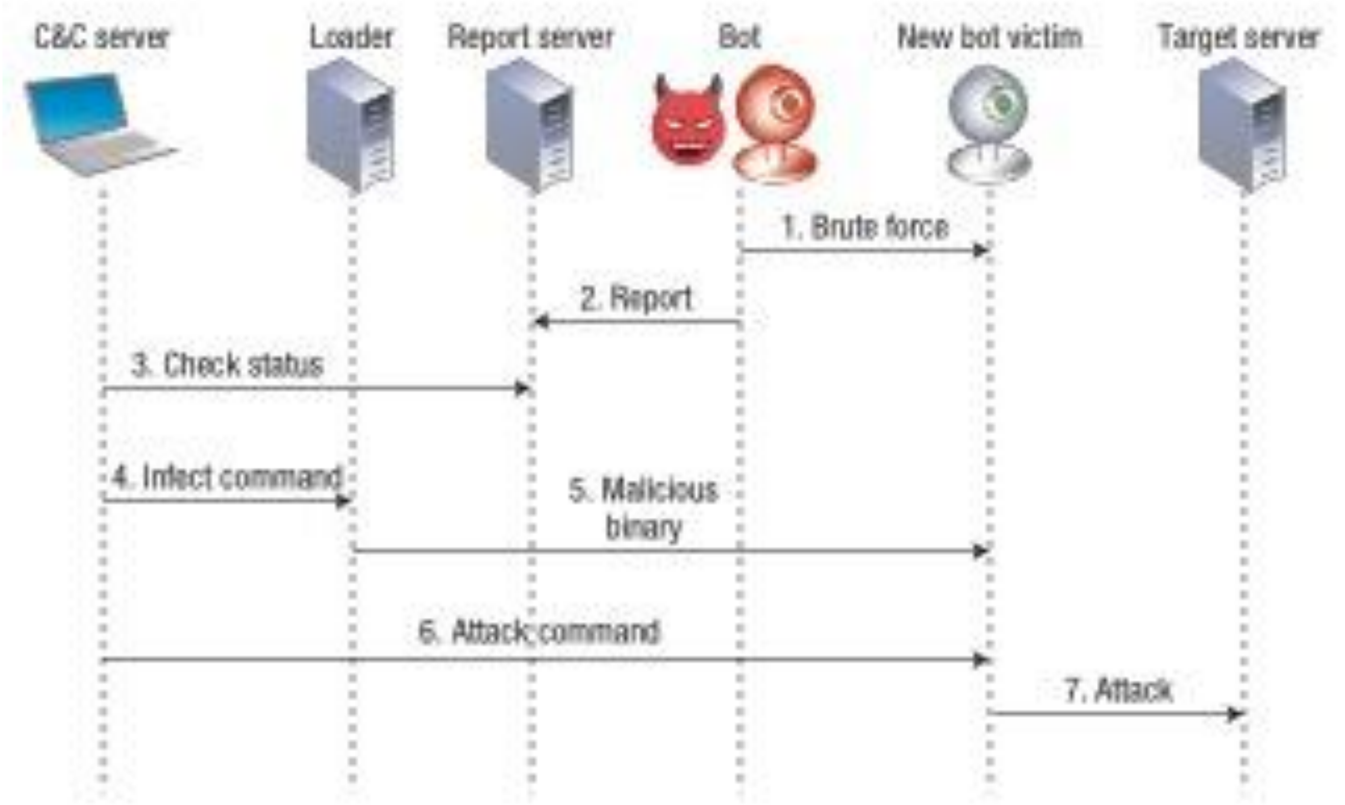
- Problem 1
  - Software is far from trivial: it has bugs!!
  - Connected devices: vulnerabilities
    - Bugs are not isolated, they can be remotely exploited
- Problem 2
  - Necessity of incorporating new functions, improving performance, reducing energy…
- Conclusion
  - Remote updates are necessary

Vulnerabilities

- Mirai botnet
  - DDoS attack - 600Gbps (2016)
  - Target: IP Cameras Duhua IP
  - Very easy:
    - Scanning telnet port
    - Brute force attack
      - (62 possible user-pass)

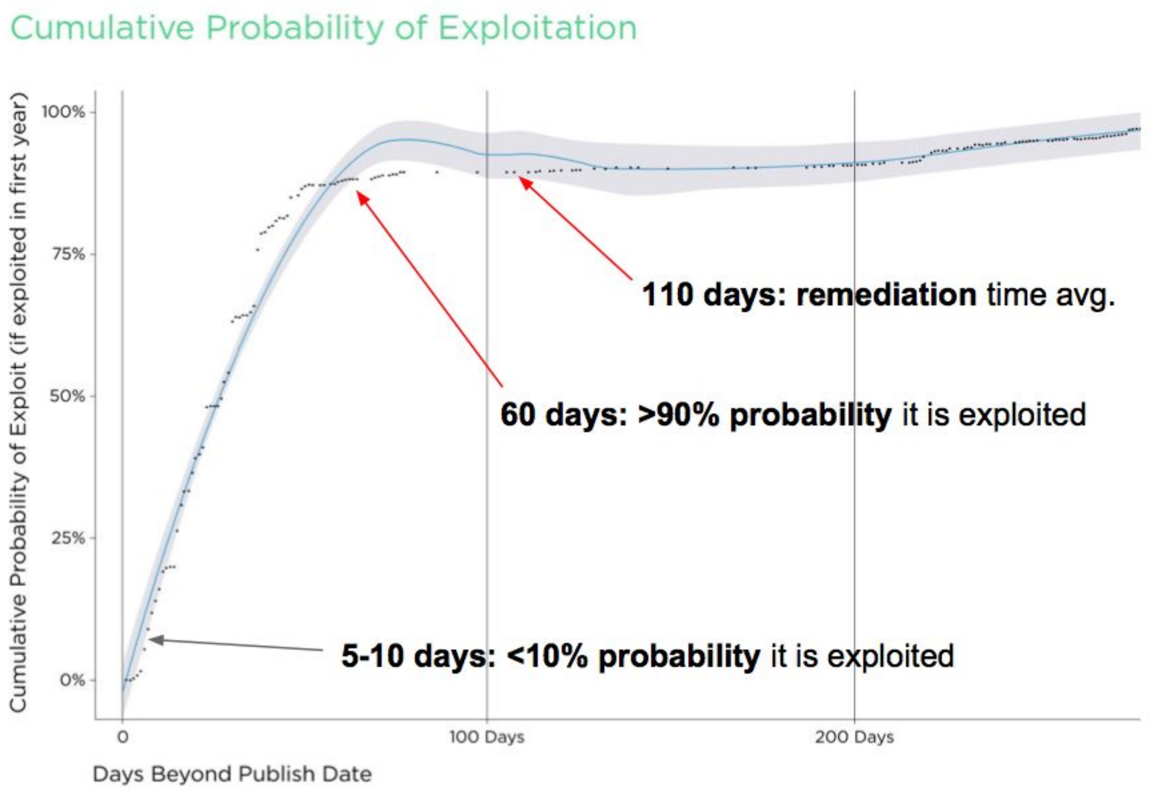
Vulnerabilities

- Mirai botnet



Reaction speed is key

- Companies usually act too late



Source: How the Rise in Non-Targeted Attacks Has Widened the Remediation Gap, Kenna Security

UNIVERSIDAD COMPLUTENSE MADRID

Examples

- 2014: Tesla was one of the first considering OTA

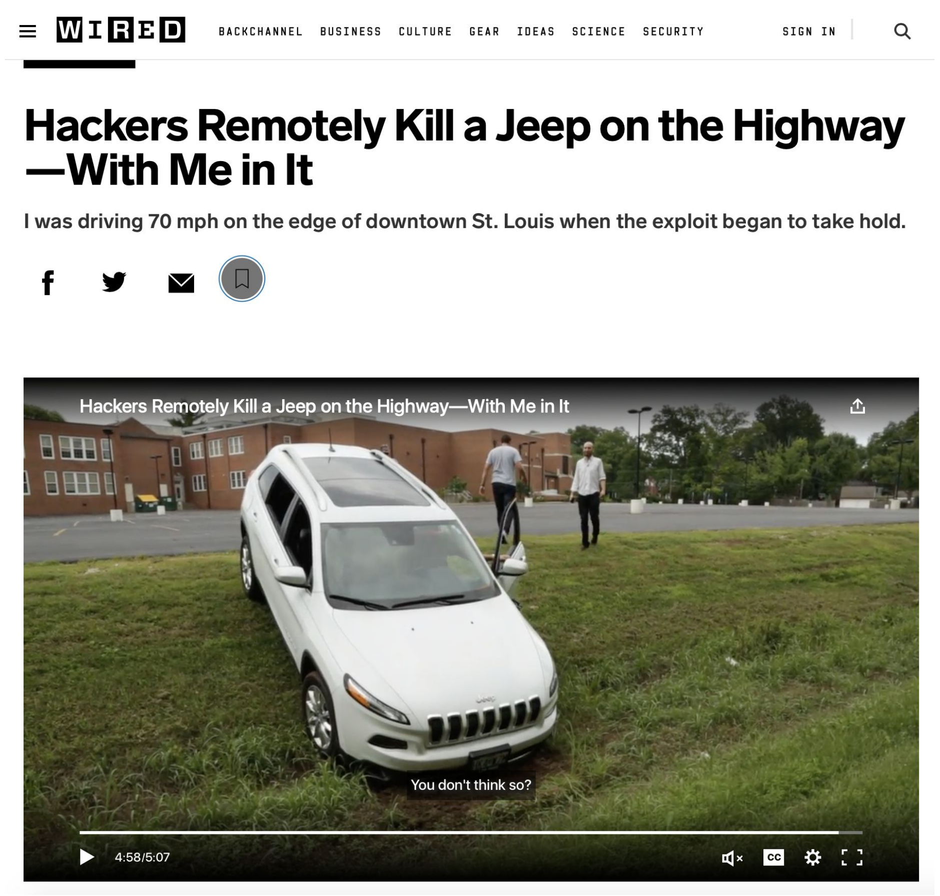
TESLA'S OVER-THE-AIR FIX: BEST EXAMPLE YET OF THE INTERNET OF THINGS?

Image: jurvetson/Flickr

The National Highway Traffic Safety Administration recently published two recall announcements, one from Tesla Motors and one from GM. Both are related to problems that could cause fires. Tesla's fix can be conducted as an "over the air" software update and doesn't require owners to bring their cars to the dealer. For that reason, we have a new precedent for what constitutes an automotive recall.

Examples

- Jeep vs Tesla (2015)

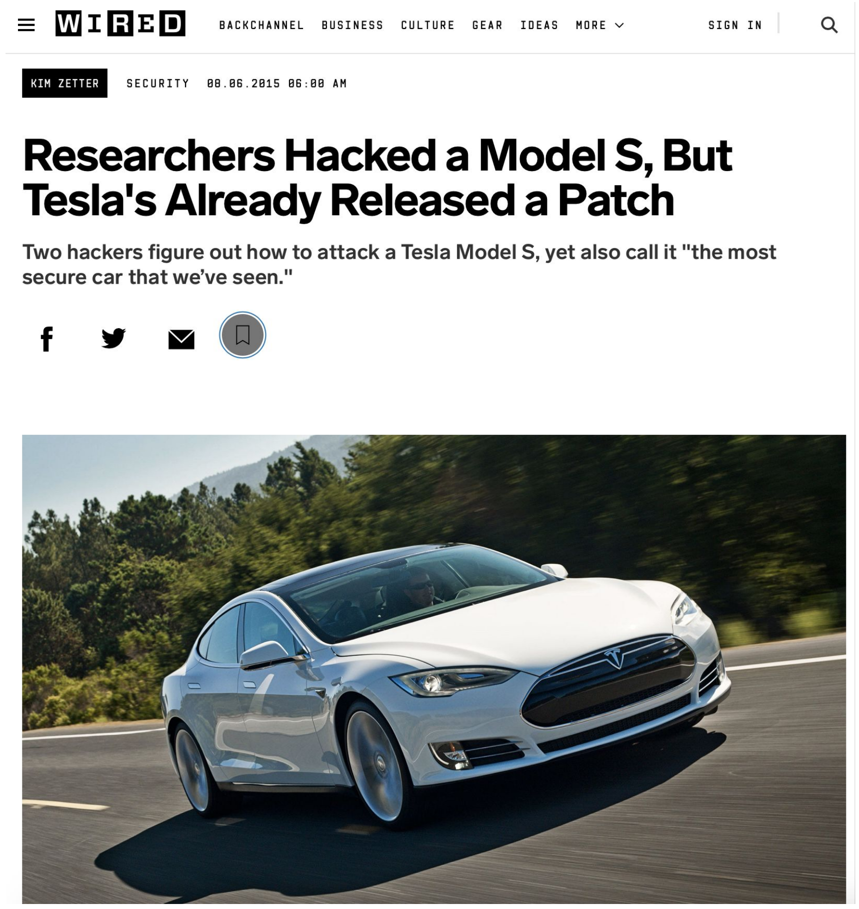


Hackers Remotely Kill a Jeep on the Highway—With Me in It

I was driving 70 mph on the edge of downtown St. Louis when the exploit began to take hold.

Hackers Remotely Kill a Jeep on the Highway—With Me in It

You don't think so?



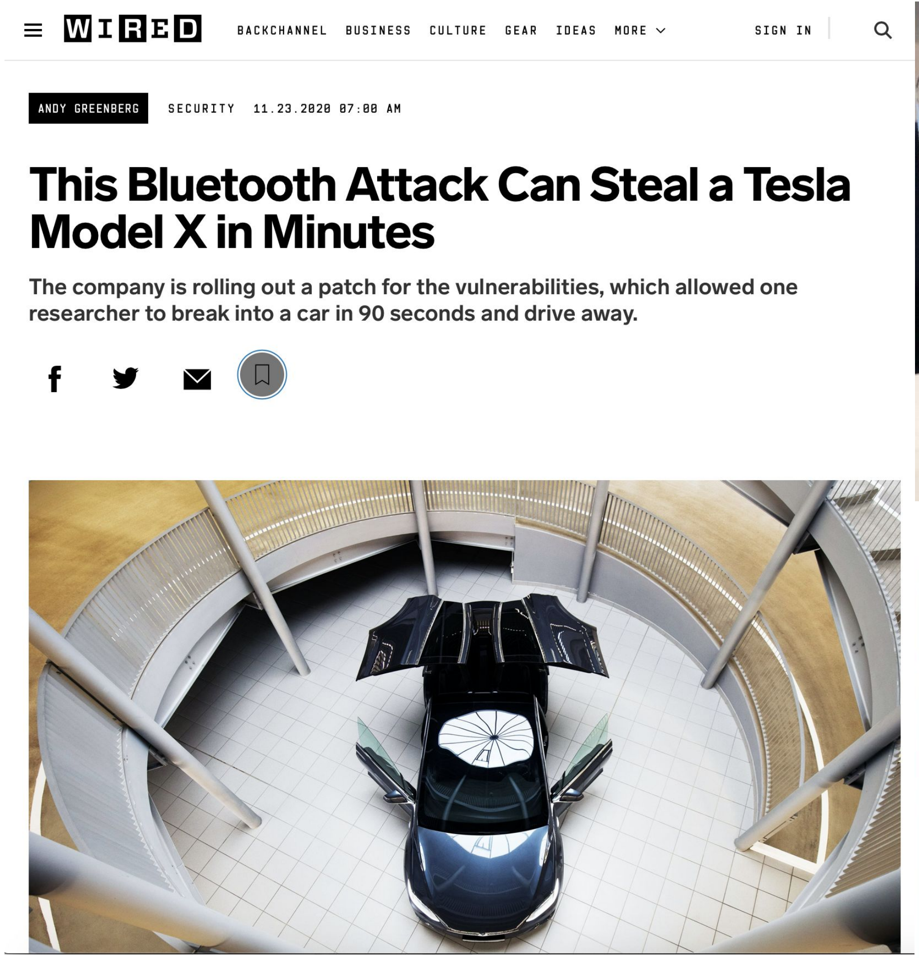
KIM ZETTER SECURITY 08.06.2015 06:00 AM

Researchers Hacked a Model S, But Tesla's Already Released a Patch

Two hackers figure out how to attack a Tesla Model S, yet also call it "the most secure car that we've seen."

Examples

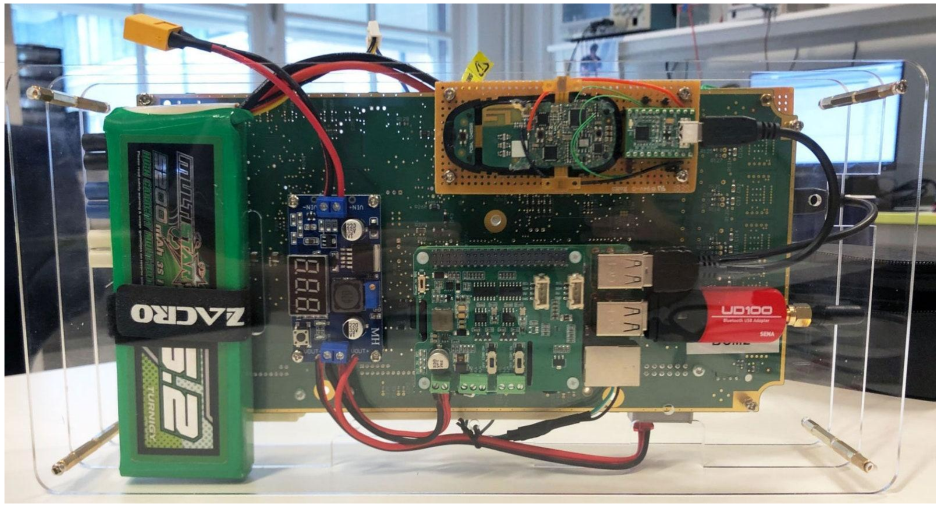
- But OTA can also be dangerous (2020)



ANDY GREENBERG SECURITY 11.23.2020 07:00 AM

This Bluetooth Attack Can Steal a Tesla Model X in Minutes

The company is rolling out a patch for the vulnerabilities, which allowed one researcher to break into a car in 90 seconds and drive away.



```
Open Trunk: 6d1290edc24d2ba49edd51001fbd566f
Lock all: 0a442a25759aabd38590d57d37b60312
Ready to unlock the target car...
Unlocking doors
Opening left falcon
```

COSIC post

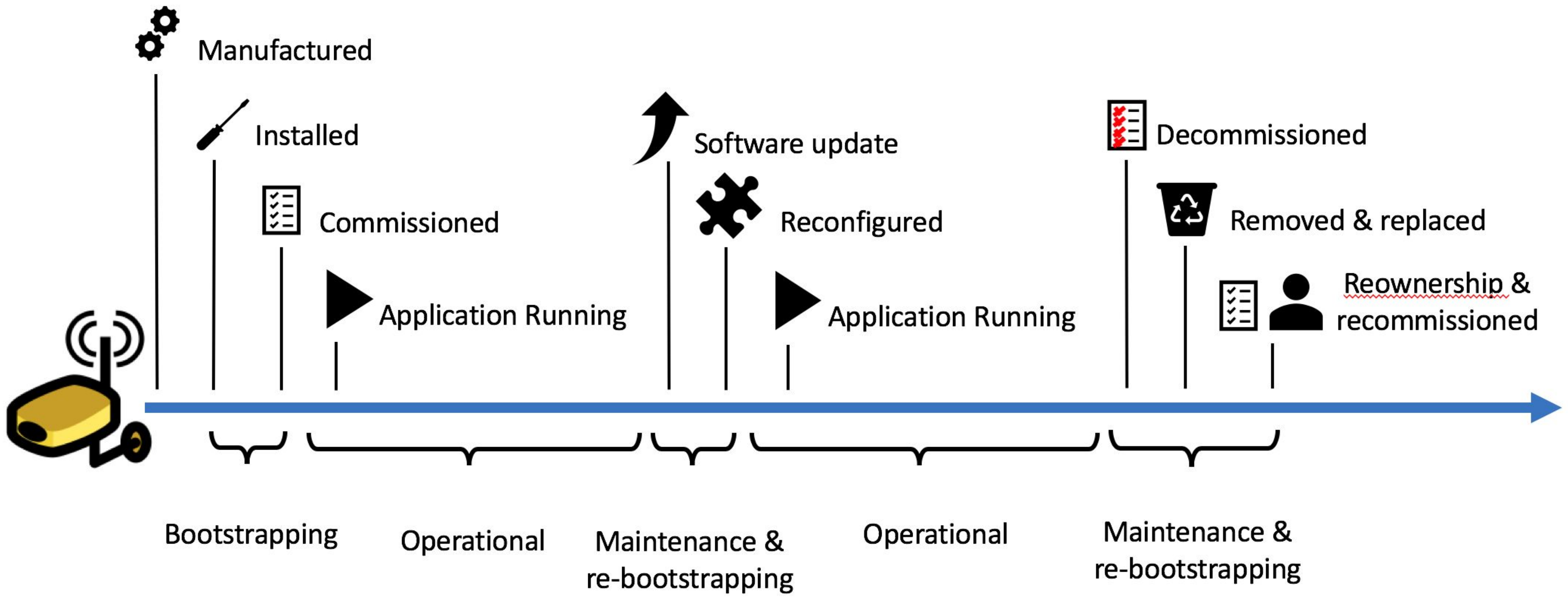
UNIVERSIDAD COMPLUTENSE MADRID

Importance of updating

- Design requisite of first order
  - Necessary to consider it from the beginning of the IoT solution development
- They can be an attack vector
  - Ej. Zigbee Worm
- Necessity of standardized mechanisms and tools
  - Avoid ad-hoc solutions

The Thing Lifecycle

- IoT Security: State of the Art and Challenges
  - RFC-8576



Requisites of update process

- OTA must guarantee:
  - <u>Security</u>: to avoid being an attack vector (device kidnapping)
  - <u>Robustness</u>: avoid device malfunction (permanent)
  - <u>Atomicity</u>: avoid partial updates
  - <u>Fault tolerance (fail-safe)</u>: returning to a safe status in case of error
  - <u>Energetic efficiency (and also memory and computation)</u>:
    - Specially important in “constrained devices” (CD)

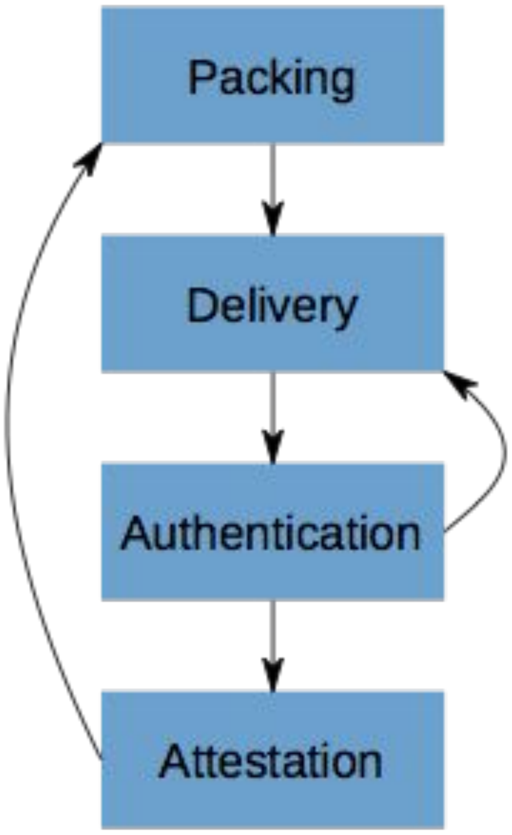
Update process

- Questions to solve:
  - How to distribute the SW update?
  - How to validate authenticity?
  - How to apply it on a device?
  - How to verify it is valid?

- Involved elements
  - SW design on the device
  - Backend framework (server)
  - Transport network (TCP/UDP, SSL, …)

Update process

- 4-phase model:
  - Packing and signature
  - Delivery
  - Authentication
  - Remote attestation



Resource dependency

- The update process depends on the available resources
  - *Embedded Linux (EL)*: memory, storage, computing (crypto) capabilities allow for complex mechanisms
  - *Constrained Devices (CD)*: few KiB RAM/Flash, few compute capabilities, simple crypto
    - RFC-7228 “Constrained Devices”

OTA - Embedded Linux

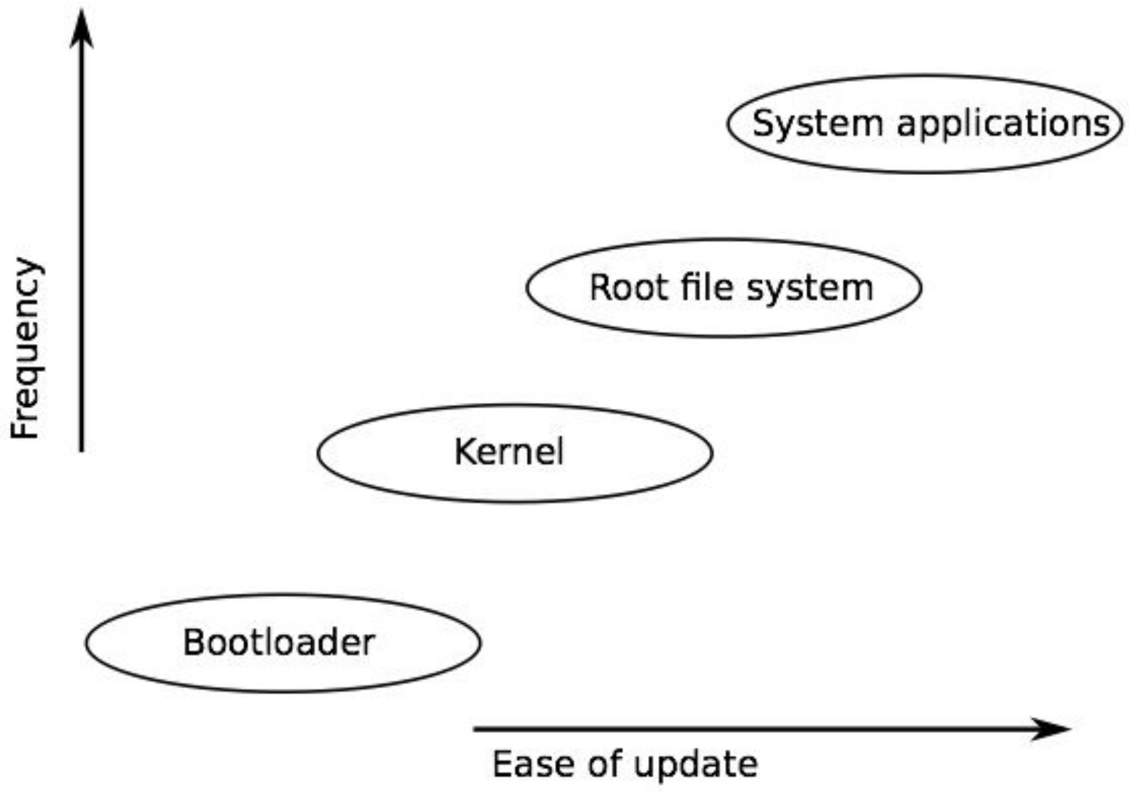
UNIVERSIDAD COMPLUTENSE MADRID

EL - Firmware Update

- Server update != device update
- Server
  - Safe environment, no failures on network or electricity supply
  - If the update fails, a human can intervene
- Device
  - Power supply and network can be unstable, interrupting the *update*.
  - A failure within the process is difficult to correct
- Update via packages (rpm/deb) is not atomic

What to update?

- Frequency, simplicity vs. importance



Granularity

- File:
  - Not an option: difficult to guarantee atomicity
- Package (e.g. RPM, deb):
  - apt-get can be enough in servers, not in devices
- Image of filesystem:
  - Common option: easy to implement and verify
- “Atomic differential update”
  - Uses FS tricks to update in an atomic fashion groups of files
- Container
  - Good idea…provided devices could run containers in a generalized manner

Main problems

- Authentication
  - Is the update legitimate?
- Security/integrity
  - Did the device receive what we sent to it?
- Roll-back
  - Upon an update failure, it must be possible to roll back to the previous version
- Monitoring
  - Monitor of the updates processes and versions on all devices in the deployment

Roll-back

- Limit in the boot counter
  - Feature of the bootloader (e.g. U-Boot)
  - Incremented upon bootloader init
  - Removed if success
  - If counter > 0, the bootloader knows there is a failed intent and loads an alternative image
- Watchdog hardware
  - If the boot is not successful, a timeout can occur and CPU is reinit
  - The bootloader checks the reason
    - If the reason is a watchdog, loads an alternative image

Update agent

- Device code that manages the update process
- Tasks:
  - Receive the update from a server
  - Apply the update
  - Replace flag at bootloader
  - Requests for reboot

Image update

- Two alternatives:

Symmetric A/B (Android after Nougat)

Asymmetric normal/recovery (Android before Nougat)

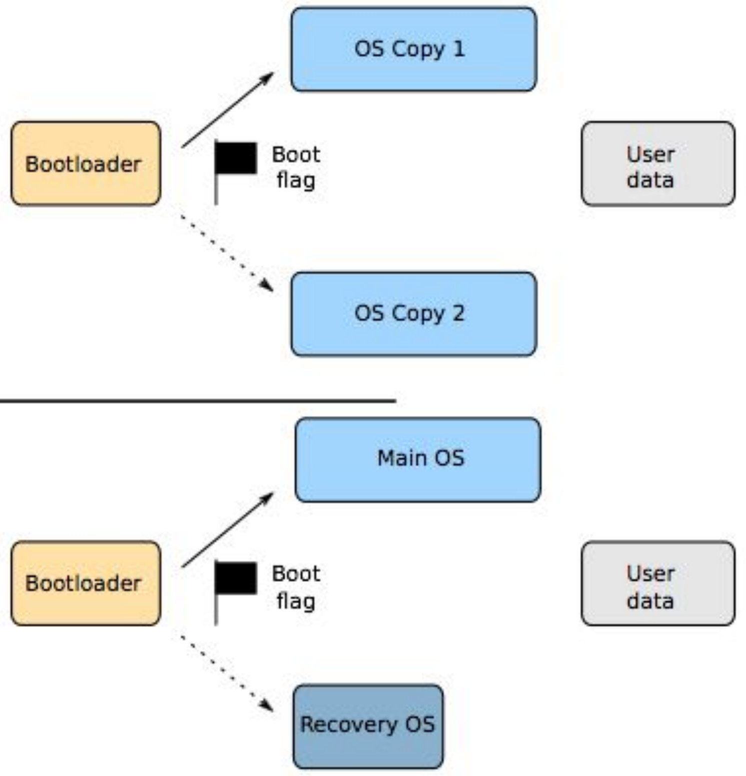


Image update

- Examples (update init by client)
  - swupdate
    - Update client for symmetric/asymmetric image update
    - Remote streaming via curl (http/https/ssh/ftp)
    - REST client for hawkBit
    - Bootloader: U-boot
  - RAUC (Robust Auto-Update Controller)
    - Update client for symmetric/asymmetric image update
    - Bootloader: grub, barebox
    - Remote streaming via curl (http/https/ssh/ftp)
    - REST client for hawkBit
    - Supports x.509

Image update

- Examples (update init by server)
  - Mender
    - Server and client for symmetric update
    - Bootloader: U-boot

Atomic differential update

- OSTree
  - Stores data in a repository similar to git
  - Filesystem linked with this repo
  - Atomic checkout
  - Pros:
    - Less storage, transfer time
  - Cons:
    - No backup system in case of corruption

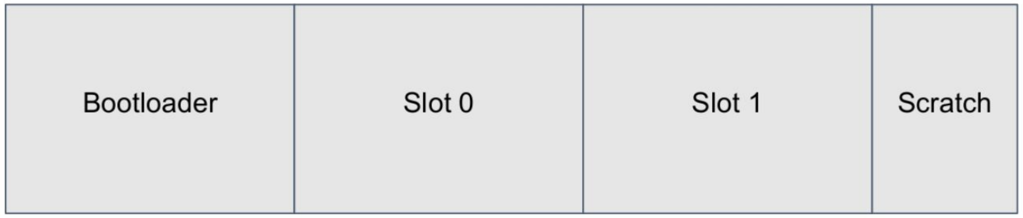
Container-based updates

- Examples:
  - Resin.io
    - Based on Docker containers
    - Integrated with a backend proprietary server
  - Snappy
    - Based on snap containers

OTA - Constrained Devices

Constrained Devices

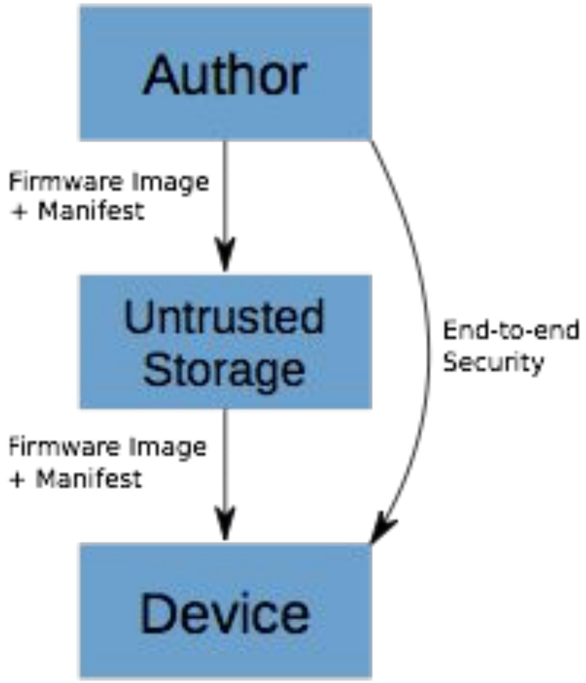
- Usually image-update
- Minimalist bootloader + flash slots
  - Problems on small flash



  - Examples: ESPER, MCUboot, RIOTboot

Backend Frameworks

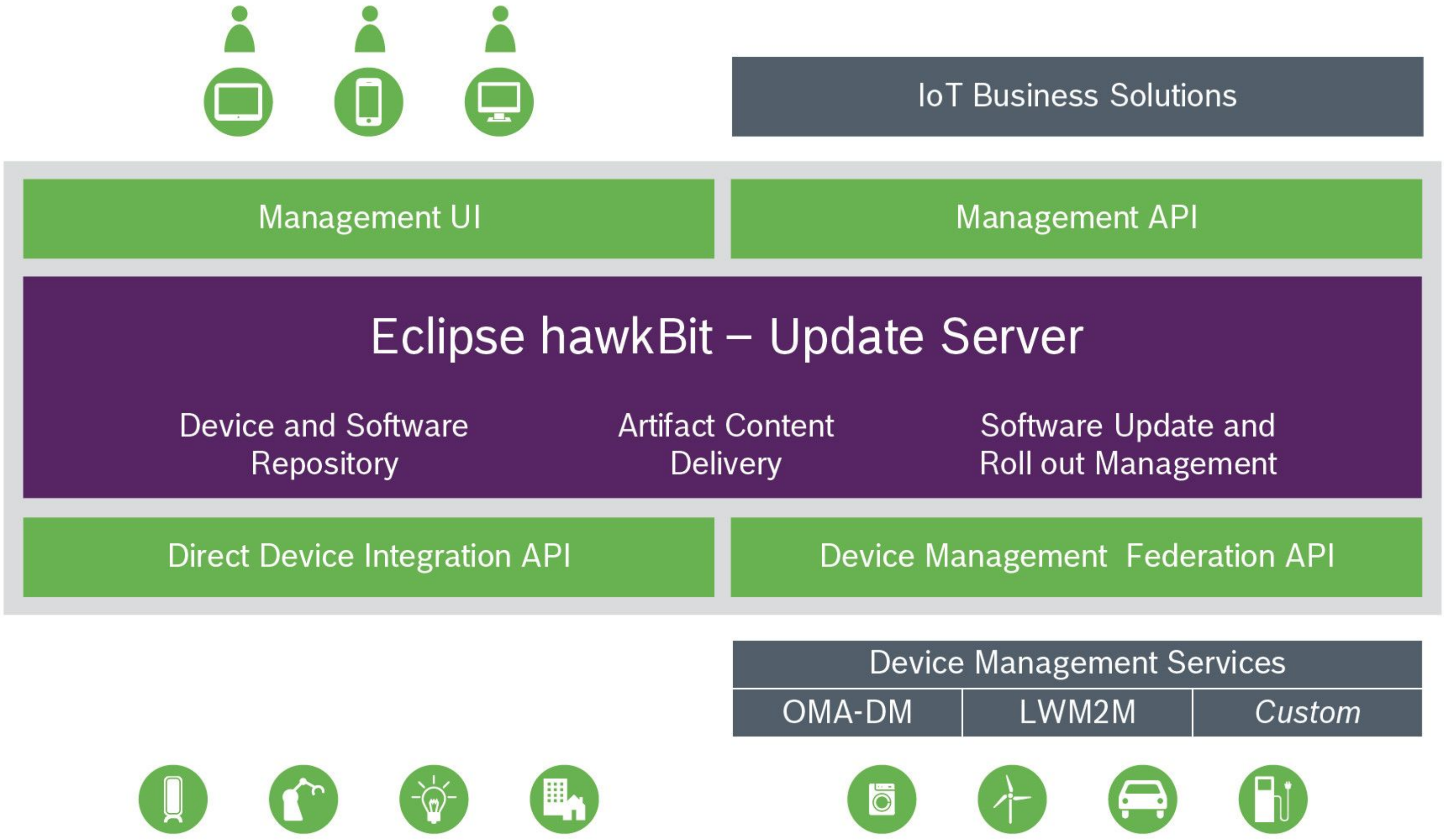
- Necessary end-to-end validation
- Use of metadata
- Firmware manifest:
  - SW autorship (signature)
  - SW integrity (hash)
  - SW version
  - Cypher/protection
  - Even package localization



Backend Frameworks - Open

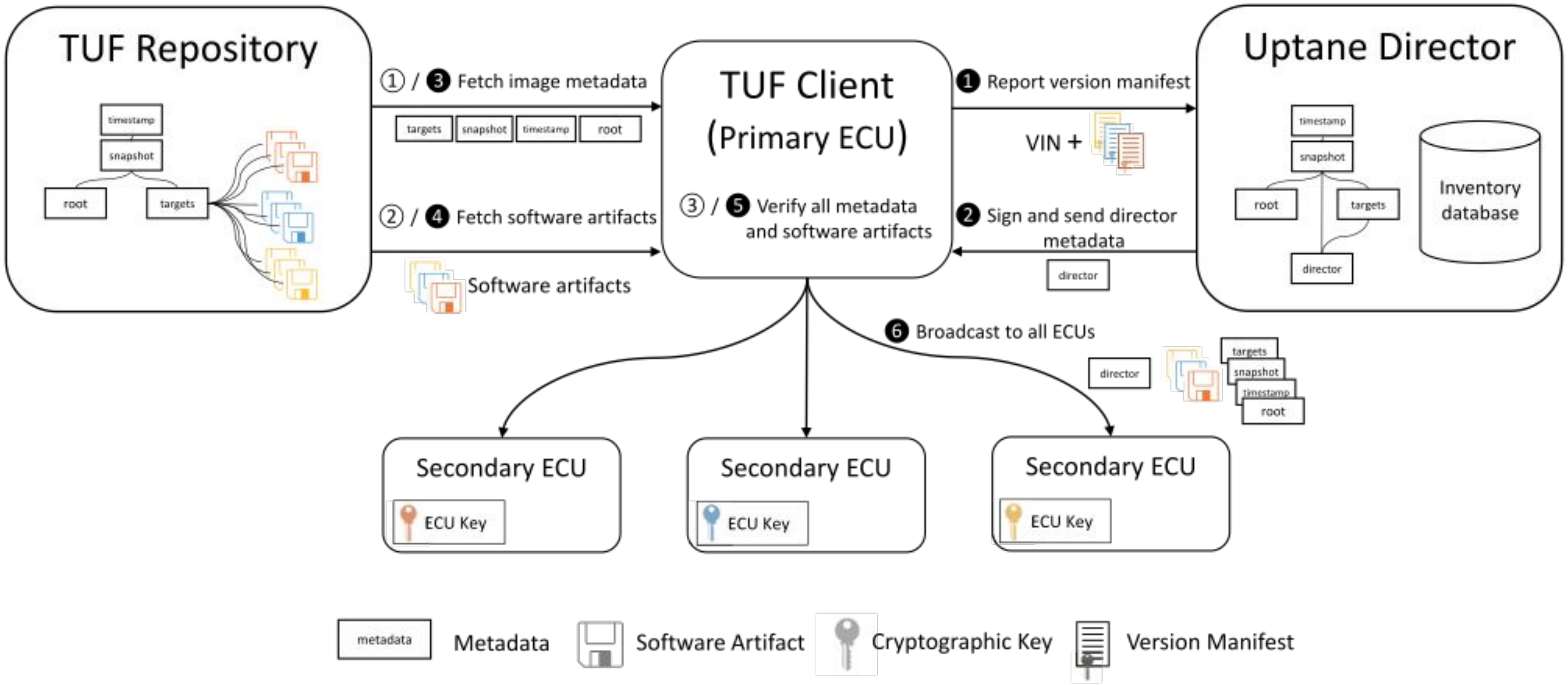
- hawkBit
- TUF/Uptane
- Mcumngr (download manager)
- LWM2M
- Newest:
  - ASSURED
  - CHAINIAC
  - UpKit

- [Eclipse hawkBit]()


IoT Business Solutions
Management UI
Management API
Eclipse hawkBit – Update Server
Device and Software Repository
Artifact Content Delivery
Software Update and Roll out Management
Direct Device Integration API
Device Management Federation API
Device Management Services
OMA-DM
LWM2M
Custom

TUF/Uptane

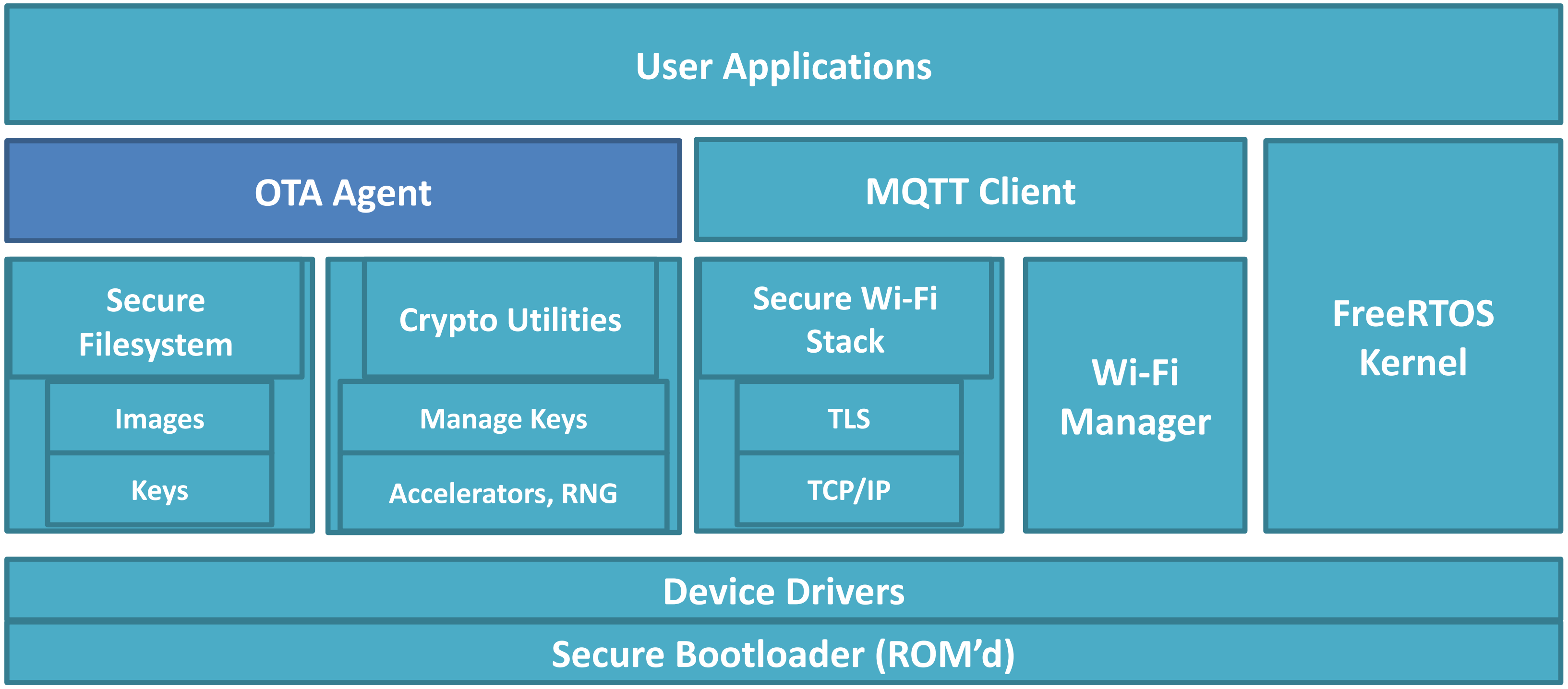
- The Update Framework (TUF) /Uptane



Backends frameworks - Proprietary

- Almost all frameworks offer OTA
  - Amazon AWS
  - Microsoft Azure
  - Google Cloud IoT
  - …
- Usually based on MQTT and proprietary libraries ☹

- Amazon FreeRTOS OTA Agent



IETF SUIT

- IETF Software Updates for Internet of Things
- 3 drafts
  - FU Architecture for IoT
  - Information mode for FU IoT Devices
  - CBOR based Serialization Format for the SUIT Manifest