SUIT-based firmware update architecture

Emmanuel Baccelli
Kaspar Schleiser
Koen Zandberg

Outline

- Intro
- Update architecture overview
- Device Software Components
- Bootloader
- Project Ideas

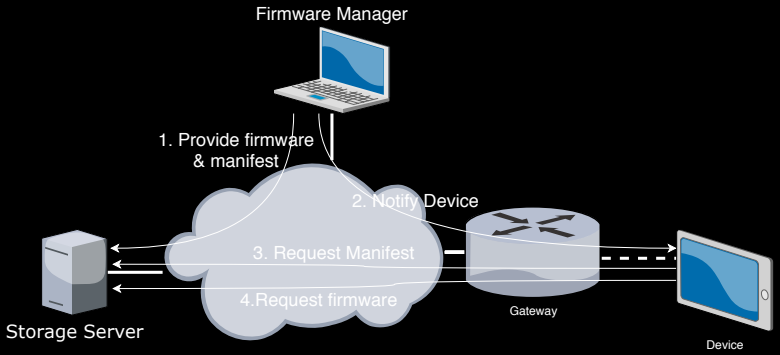
Intro

- 32/16/8 bit MCU
- Open standards
- *Internet of Things*

Intro

- RIOT SUIT example: https://git.io/suit-updater
- SUIT VM (3.3GB): http://demo-fit.saclay.inria.fr/vms/RIOT-VM.ova
- Instructions: https://git.io/suit-hackathon

Update architecture overview



Update architecture overview

Focus on the storage server

- Provides image and and manifest over CoAP

Update architecture overview

Management station:

- Allows remote firmware management and triggers deployments.

Update architecture overview

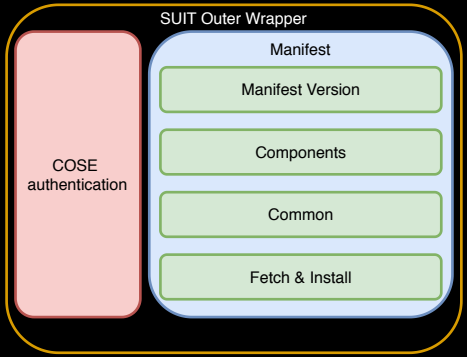
Node:

- Receives manifest trigger over CoAP
- Fetches manifest
- Fetches firmware update

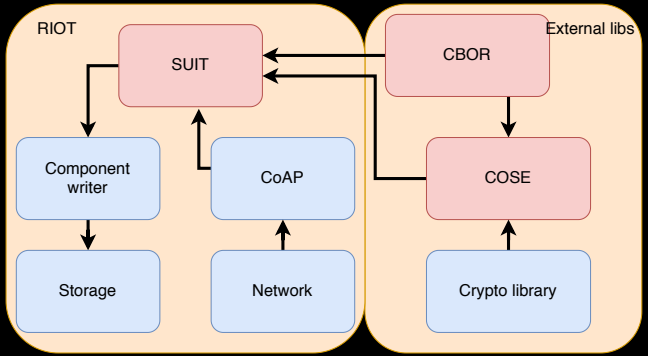
Device Software Components

- Manifest recap
- Software component overview
- Chosen implementations

Device Software Components



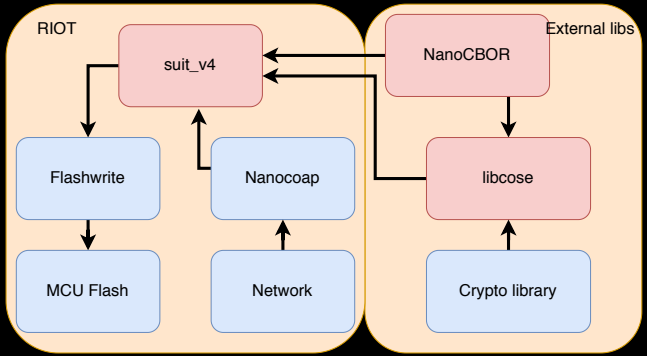
Device Software Components



Device Software Components

- SUIT parser
- libcose
- NanoCBOR

Device Software Components



Device Software Components

NanoCBOR

- Pull style parser
- 600B - 800B decoder
- Optimized for parsing known structures

https://github.com/bergzand/NanoCBOR

Device Software Components

Libcose

- Embedded COSE library
- Multiple crypto backend support

https://github.com/bergzand/libcose

Device Software Components

SUIT parser

- Based on NanoCBOR
- libcose for verification

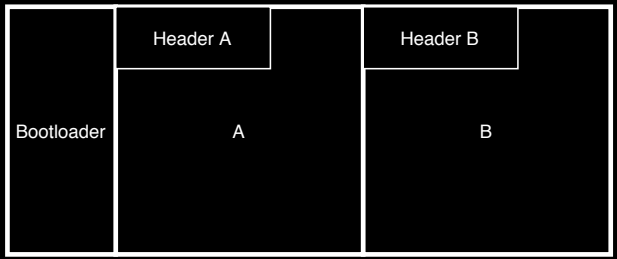
Device Software Components

SUIT parser design

- Iterate over the map
- Multiple jump table based on map key

Device Software Components

Flash layout



Device Software Components

Flash partition writer

```
typedef struct {
    uint32_t magic_number;      /**< Magic number    */
    uint32_t version;           /**< Sequence No     */
    uint32_t start_addr;        /**< Address         */
    uint32_t chksum;            /**< Checksum        */
} riotboot_hdr_t;
```


IOT

Bootloader

Just another RIOT application

1. Iterate over all headers
2. Determine whether the header is valid
3. Determine highest valid sequence number
4. Boot firmware

Project Ideas

- Extend manifest parsing
- MCUboot interoperability
- Infrastructure improvements