



# Smart Door Lock

## HSLU - IOT

Boas Meier, Niklas Tanner, Lukas Mettler, Tobias Heller

# Table of Contents

01

## USECASE

Why is a smart door lock needed?

02

## SOLUTION

How did we solve it and what were the biggest challenges?

03

## NEXT STEPS

What are our next steps?

04

## DEMO

How does our product work?



# 01 USECASE

# Features



## Door lock

Lock, Unlock  
Remote access



## Door bell

Ringling  
Remote Push



## Security

Alarms, Intrusion  
Suspicious Activities  
Remote Push

# User Interfaces



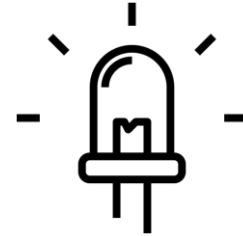
## Web Application

State of door  
Previous actions  
Remote actions



## Mobile App

Open with NFC  
Receive notifications  
Unlock remotely



## Human Machine Interface

LED Red / Green  
Button



# 02

## SOLUTION

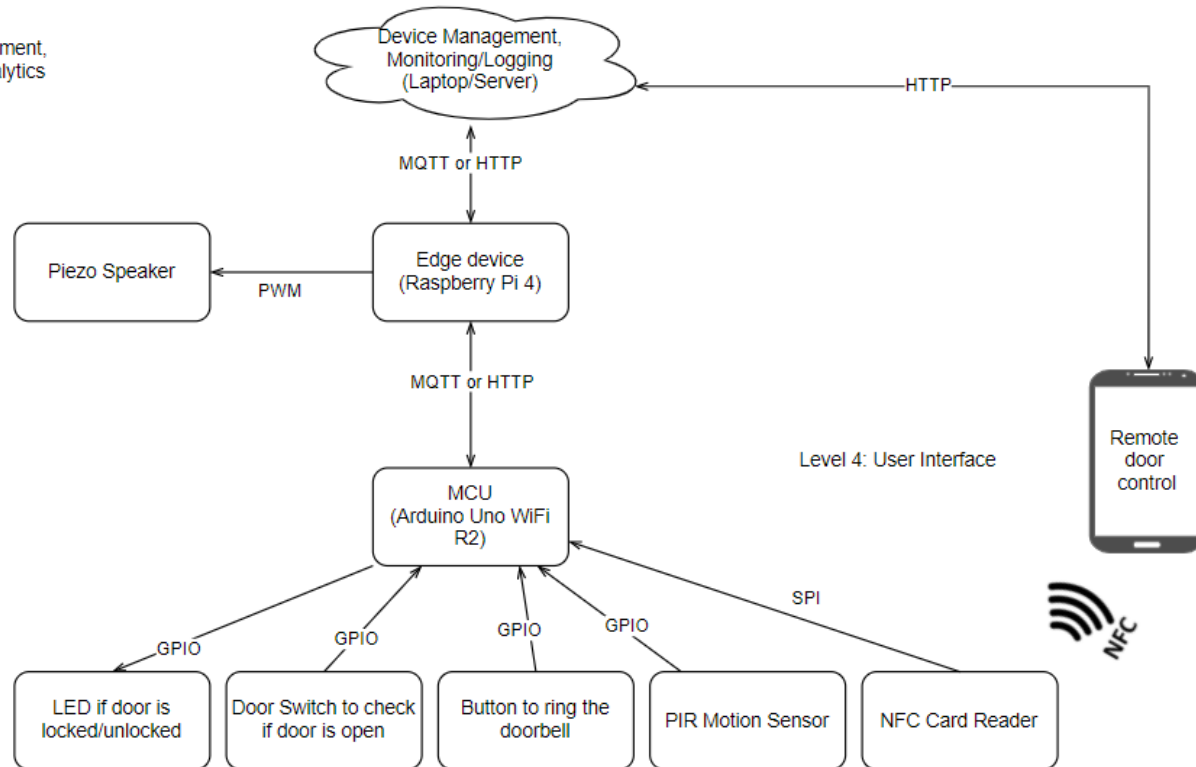
# System overview

Level 3: Device Management,  
Monitoring, Logging, Analytics

Level 2: Supervisory,  
Control, Aggregation

Level 1: Direct Control

Level 0: Field Level

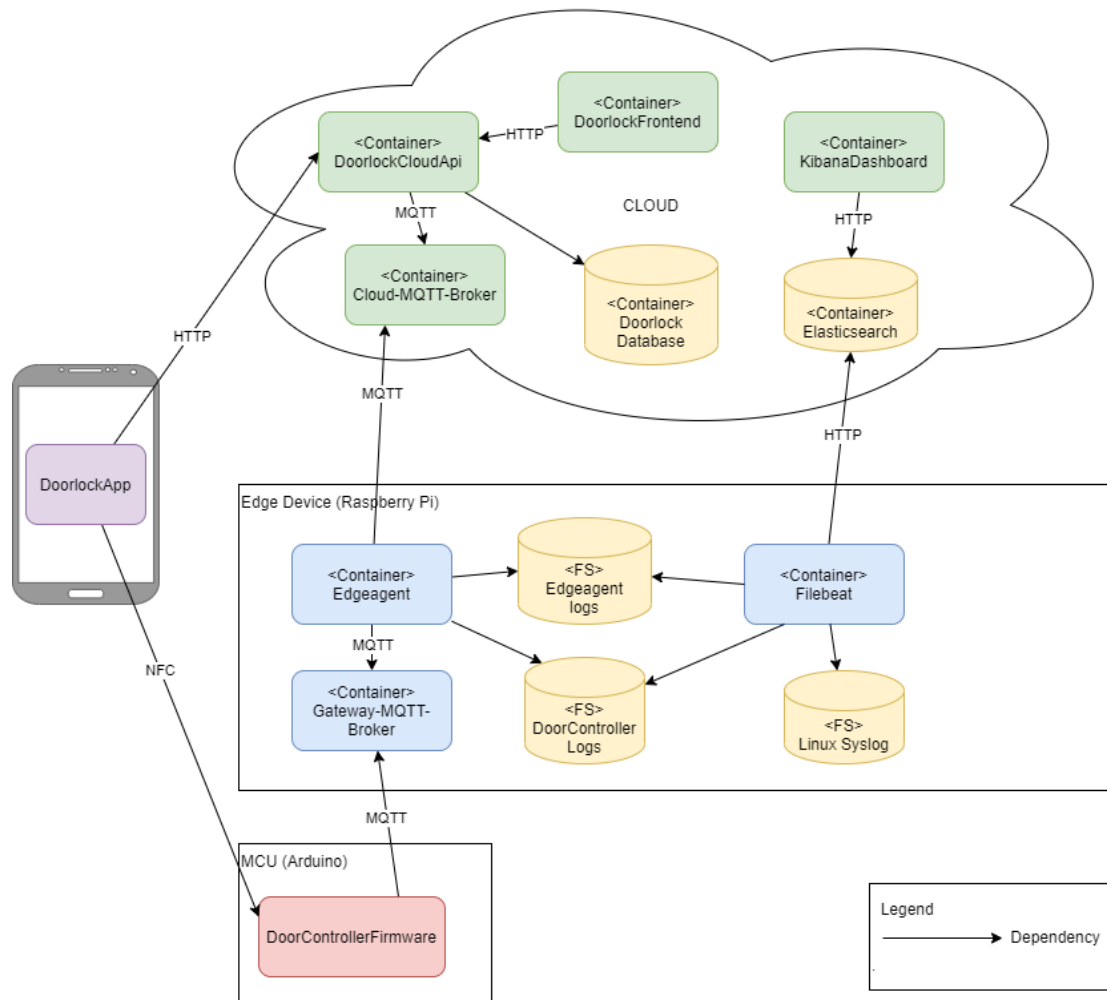




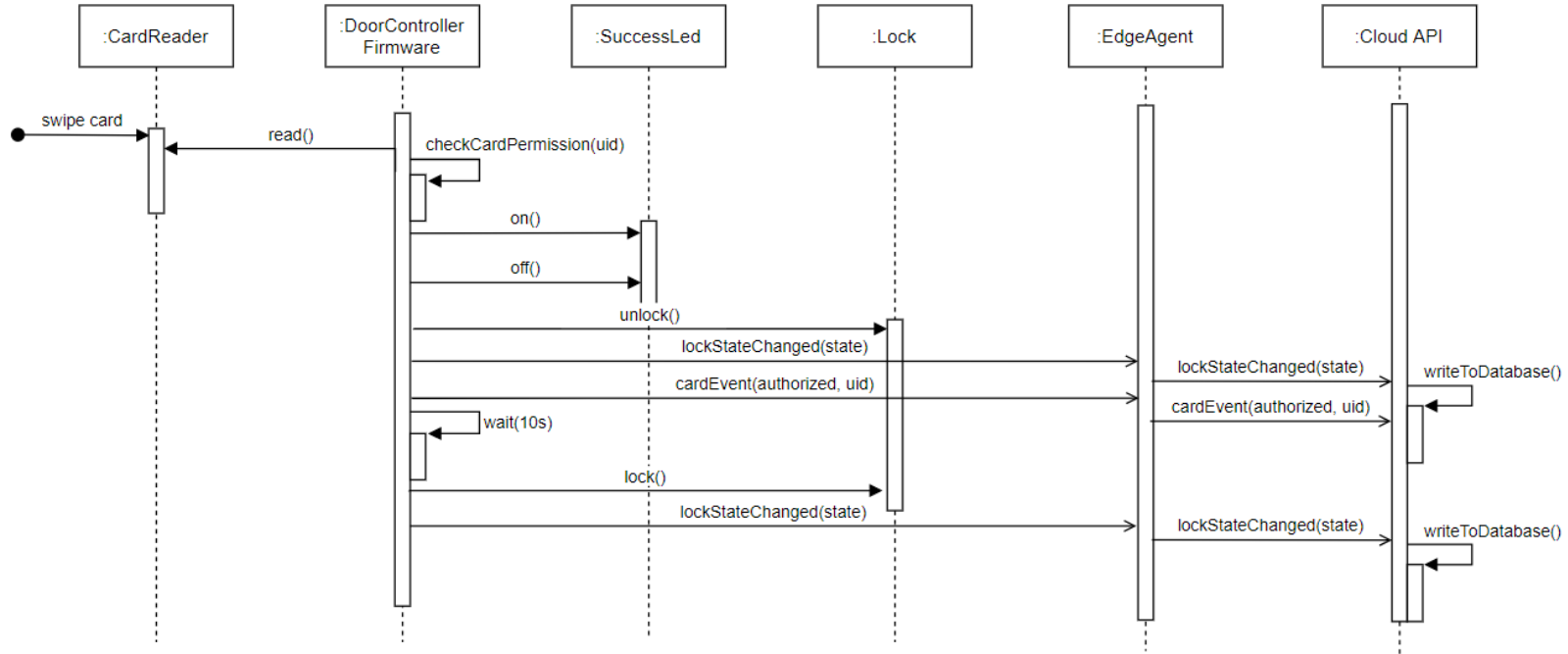
# 11

Number of main software components





# Functional Software Architecture

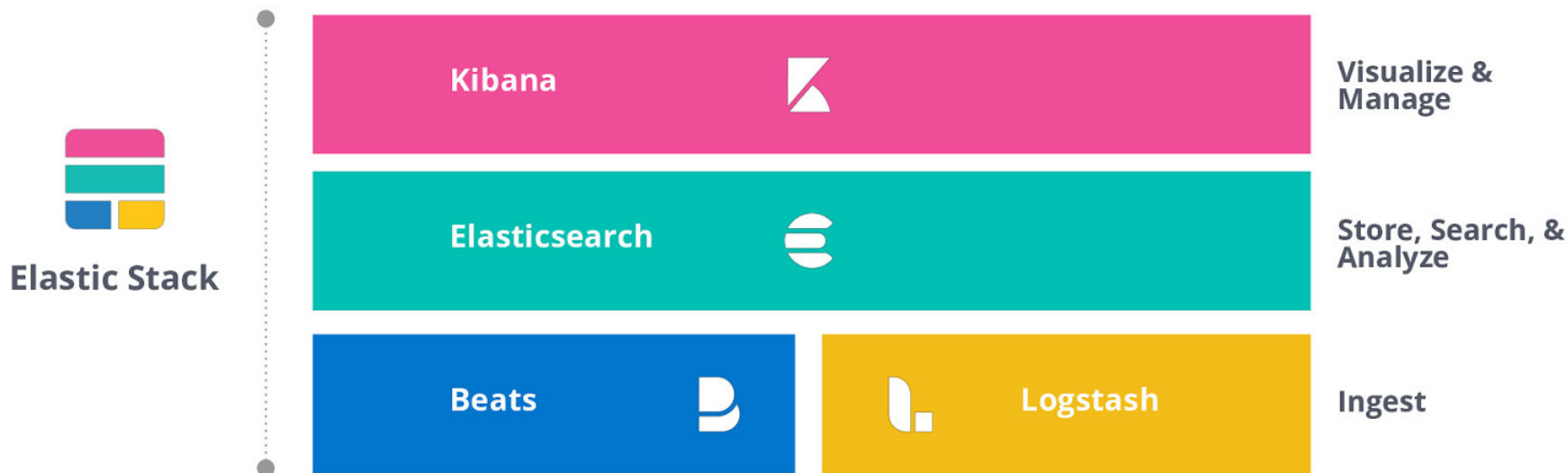


**But what  
if it  
doesn't?**

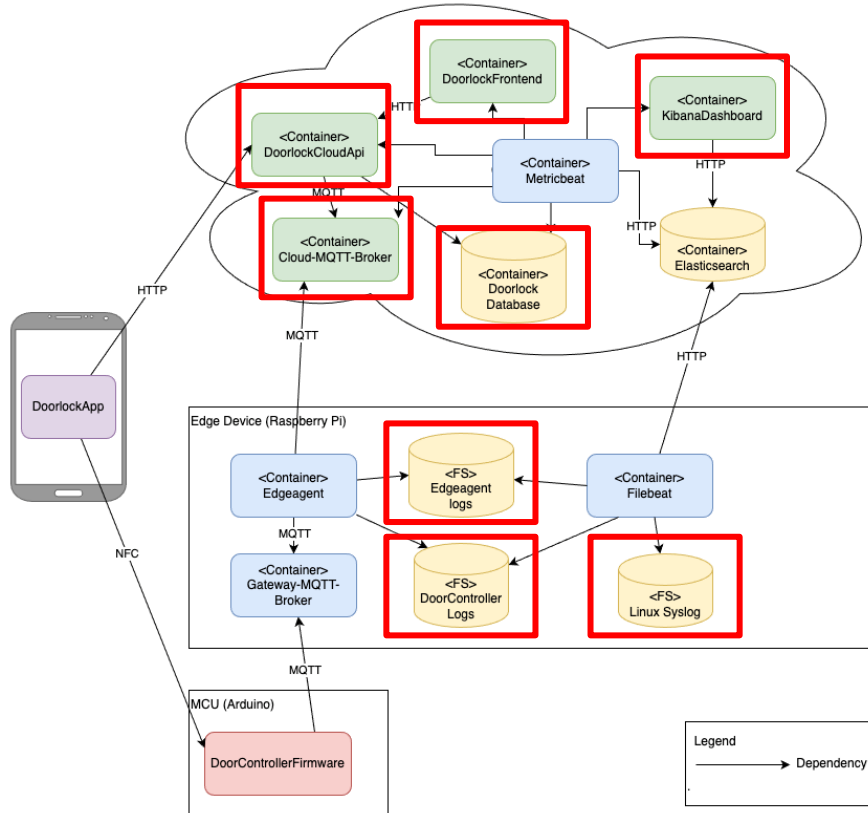
# Elastic Stack

Elasticsearch, Kibana and Integrations

# What is the Elastic Stack?



# What we use it for



- Log collection at Edge Device (Edge Agent, Door Controller, Edge Device itself (filesystem logs))
- Log collection (metric data) at Cloud / Backend (all containers)

Why?

- Monitoring
- Transparency
- Troubleshooting



# Technical challenges

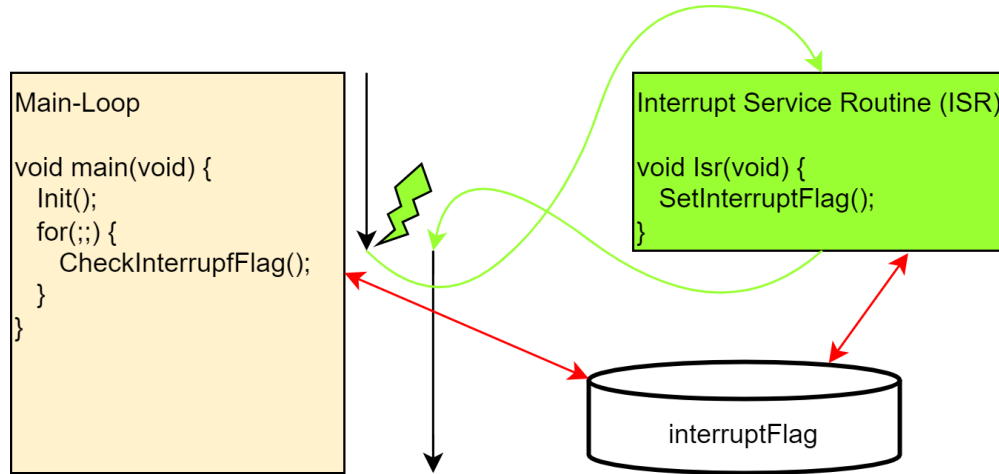
And how we solved them

# Interrupts and Reentrancy

Super-Loop vs. **Event-Loop** vs. RTOS

Mqtt messages and GPIO signals trigger interrupts

Actual work is done in main-loop and not inside ISR -> Set an interrupt flag (event)





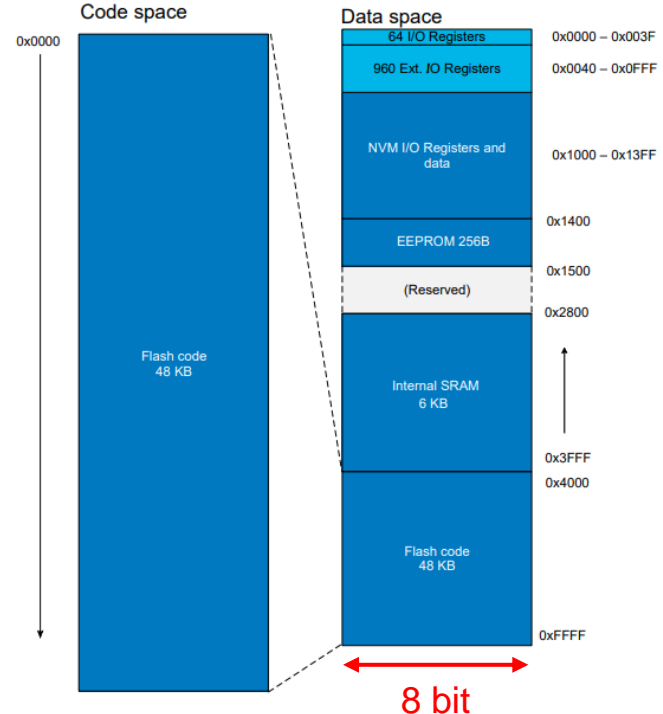
# Interrupts and Reentrancy

Use byte for atomic access, since Arduino Uno Wifi Rev2 uses an 8-bit mcu (ATmega4809).

```
static volatile byte interruptFlag = 0;
```

Use volatile to tell the compiler to not optimize any line of code containing the interruptFlag.

ATmega4808 Memory Map



# Interrupts and Reentrancy

```
#include <util/atomic.h>
void PahoMqttClient::handleMqttMessage() {
    int n = QUEUE_SIZE;
    ATOMIC_BLOCK(ATOMIC_RESTORESTATE) {
        if(mqttMessagePointerQueue[0] != -1) {
            MqttMessageHandler::handleMessage(mqttMessagePointerQueue[0]->getTopic(),
                                                mqttMessagePointerQueue[0]->getMessage());

            for(int i=0; i<n-2; i++) {
                mqttMessagePointerQueue[i] = mqttMessagePointerQueue[i+1];
            }
            mqttMessagePointerQueue[n-1] = (MqttMessage*) -1;
        }
    }
}
```

Disable Interrupts

Critical Section

Enable Interrupts again



# 03

## NEXT STEPS

# Next steps

- Features
  - Improved App (Android & iOS)
  - Camera Streaming
- Security (IAM, IDS/IPS, TLS, ...)
- Zero-Touch Deployment



# 04

## DEMO

# Local unlock

with card



# Local unlock

with phone



# Ring and remote unlock

with phone





# Suspicious activity

with rejected card



# Intrusion

with alarm



# Remote unlock

with web interface



# Filtering

with Kibana



# Questions?