RPi Fan Control

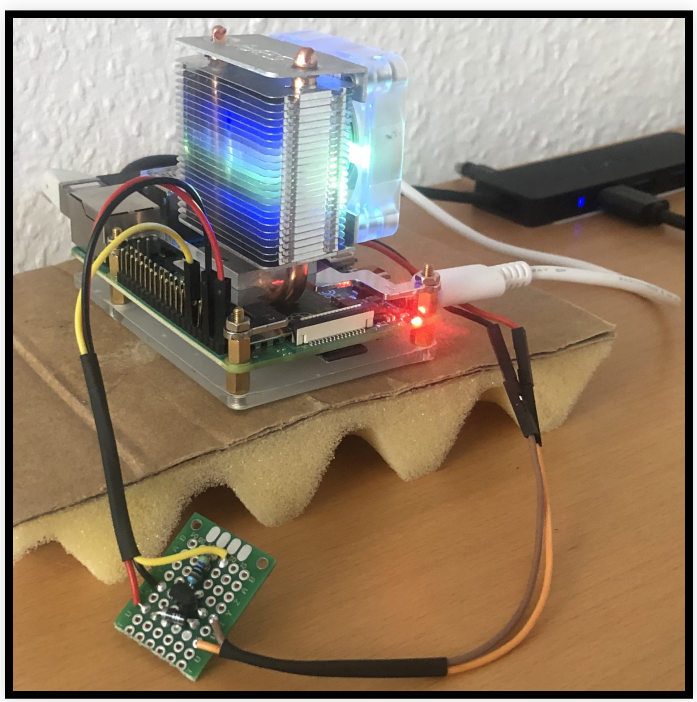
(RPiFC)

Gerrit Klein

Motivation

Situation

- Existing **fan control software** for headless RPi's
  - Controls fan duty cycle based on CPU temp



Current limitations

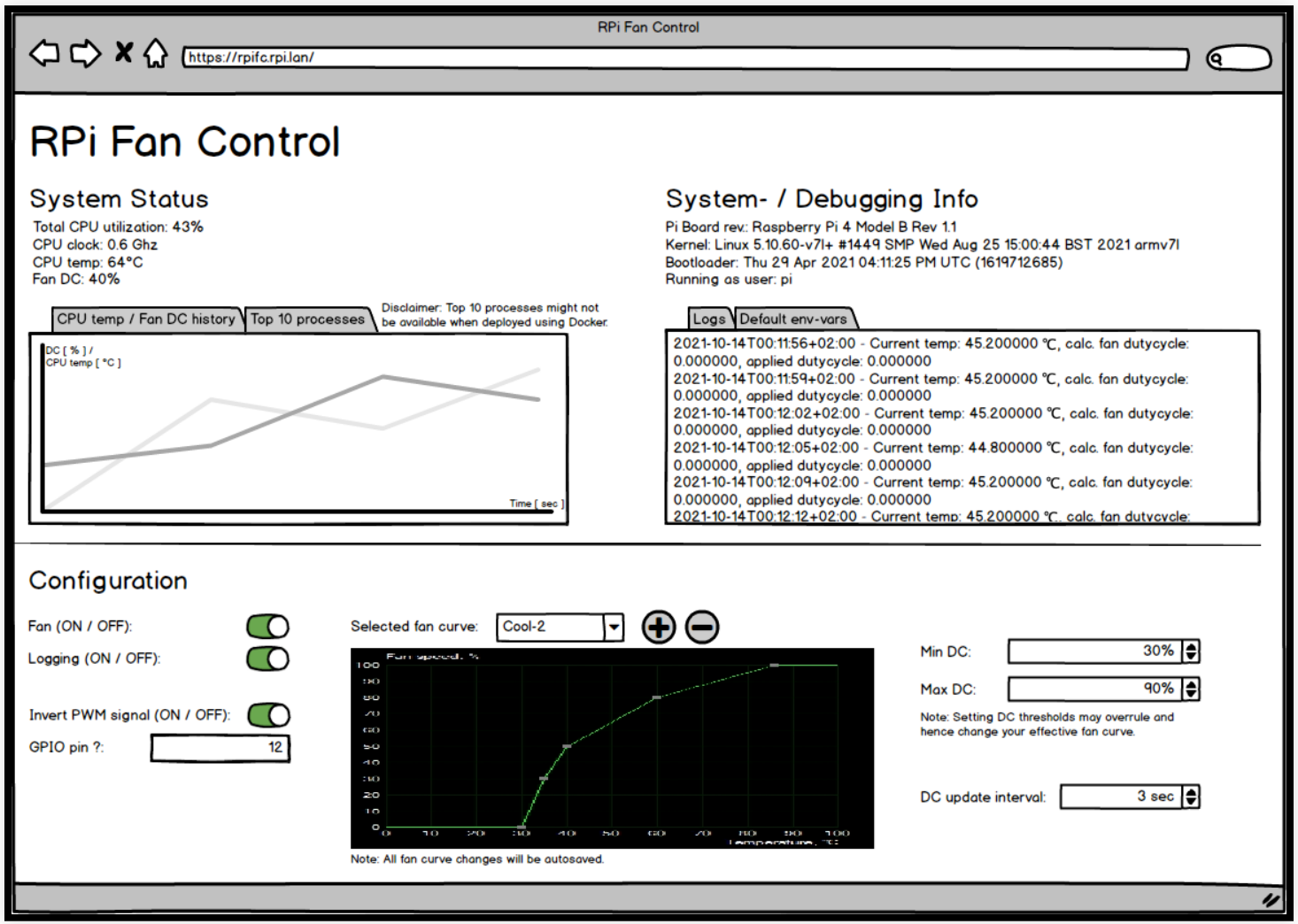
- **Cumbersome configuration**
  - Currently: `ssh` → change env-vars → restart app
- **No direct access** to logs / stats
  - Currently: Only via `ssh` / Portainer

Solution: Web API

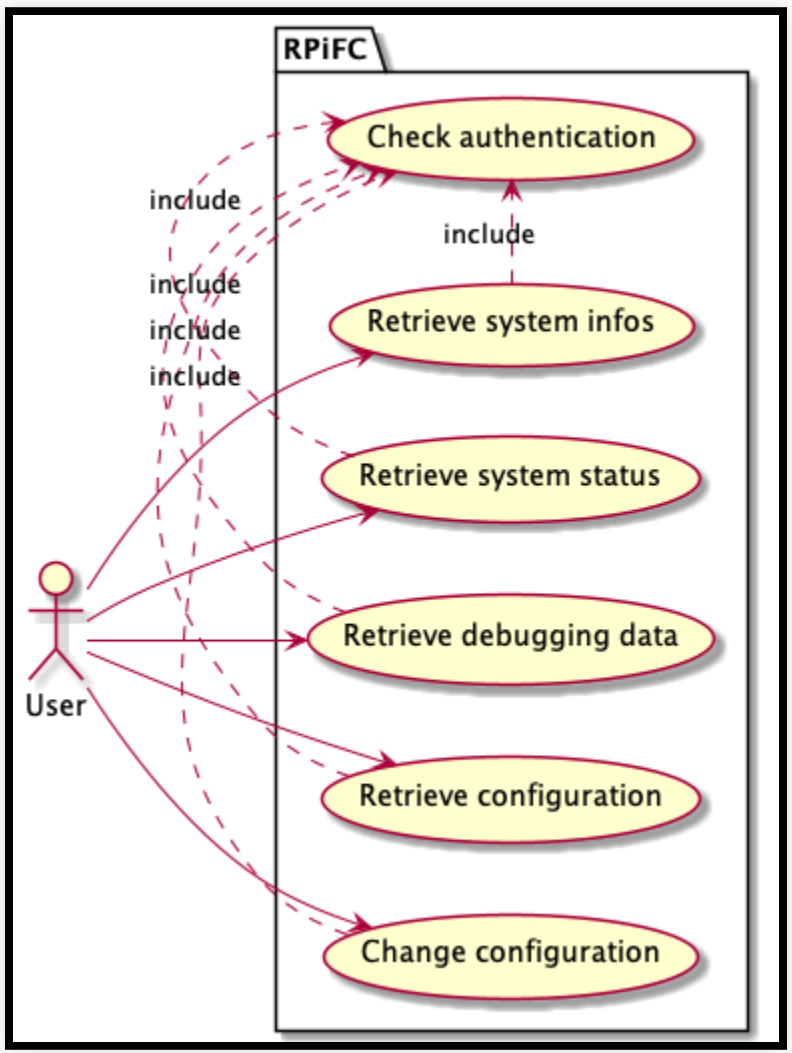
Requirements Specification

1. **Access system stats** / debugging info
    - E.g., CPU utilization, CPU temp. / Device info, Logs, current settings, etc.
2. Quickly **change configuration**
    - E.g., Logging enabled, Fan settings / thresholds
3. Authentication
4. Deactivate API

UI Mockup

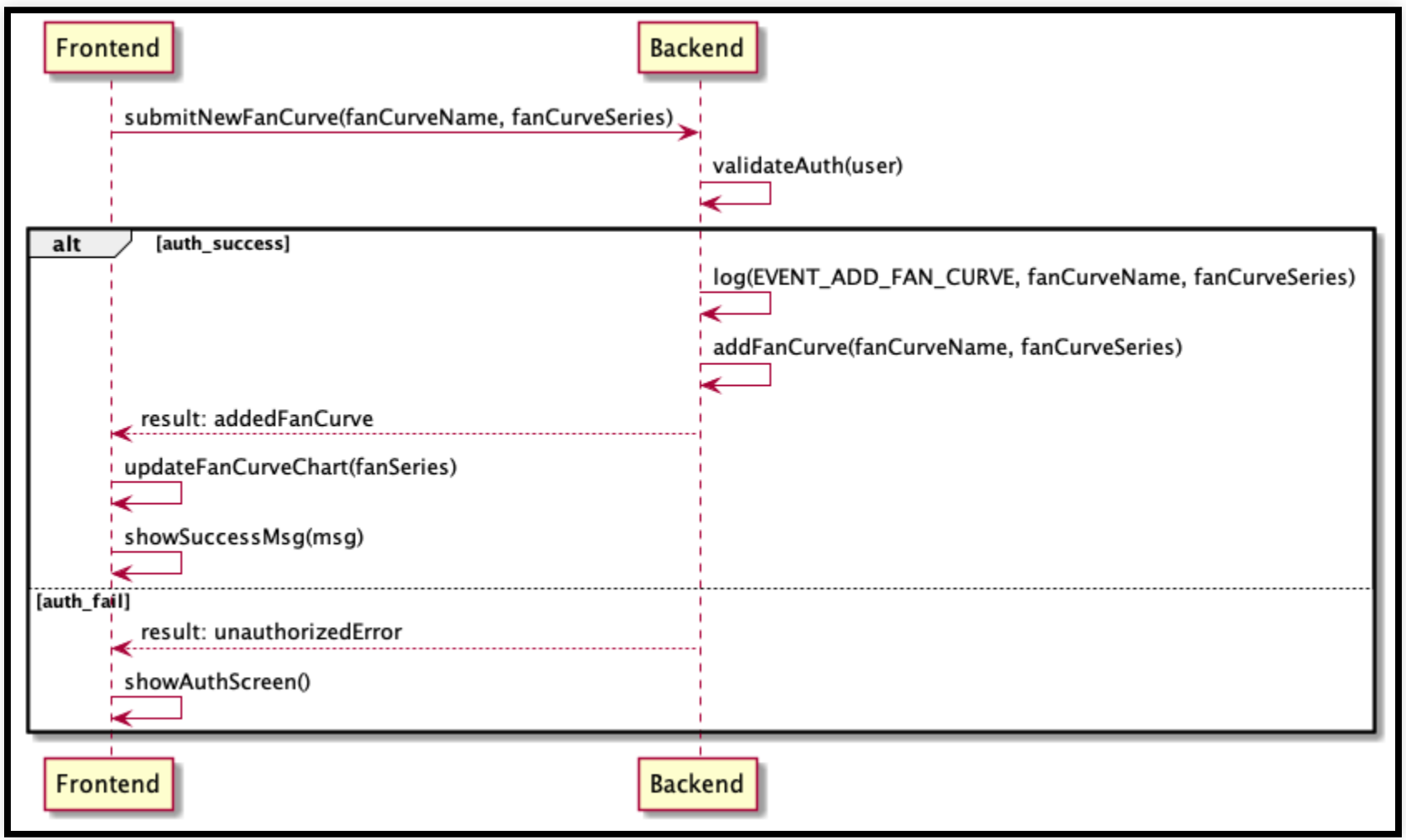

RPi Fan Control
https://rpifc.rpi.lan/
RPi Fan Control
System Status
Total CPU utilization: 43%
CPU clock: 0.6 Ghz
CPU temp: 64°C
Fan DC: 40%
CPU temp / Fan DC history
Top 10 processes
Disclaimer: Top 10 processes might not be available when deployed using Docker.
DC [%] /
CPU temp [°C]
Time [sec]
System- / Debugging Info
Pi Board rev.: Raspberry Pi 4 Model B Rev 1.1
Kernel: Linux 5.10.60-v7l+ #1449 SMP Wed Aug 25 15:00:44 BST 2021 armv7l
Bootloader: Thu 29 Apr 2021 04:11:25 PM UTC (1619712685)
Running as user: pi
Logs
Default env-vars
2021-10-14T00:11:56+02:00 - Current temp: 45.200000 °C, calc. fan dutycycle: 0.000000, applied dutycycle: 0.000000
2021-10-14T00:11:59+02:00 - Current temp: 45.200000 °C, calc. fan dutycycle: 0.000000, applied dutycycle: 0.000000
2021-10-14T00:12:02+02:00 - Current temp: 45.200000 °C, calc. fan dutycycle: 0.000000, applied dutycycle: 0.000000
2021-10-14T00:12:05+02:00 - Current temp: 44.800000 °C, calc. fan dutycycle: 0.000000, applied dutycycle: 0.000000
2021-10-14T00:12:09+02:00 - Current temp: 45.200000 °C, calc. fan dutycycle: 0.000000, applied dutycycle: 0.000000
2021-10-14T00:12:12+02:00 - Current temp: 45.200000 °C, calc. fan dutycycle:
Configuration
Fan (ON / OFF):
Logging (ON / OFF):
Invert PWM signal (ON / OFF):
GPIO pin ?:
12
Selected fan curve:
Cool-2
Fan speed: %
Temperature: °C
Note: All fan curve changes will be autosaved.
Min DC:
30%
Max DC:
90%
Note: Setting DC thresholds may overrule and hence change your effective fan curve.
DC update interval:
3 sec

Use Case Diagram

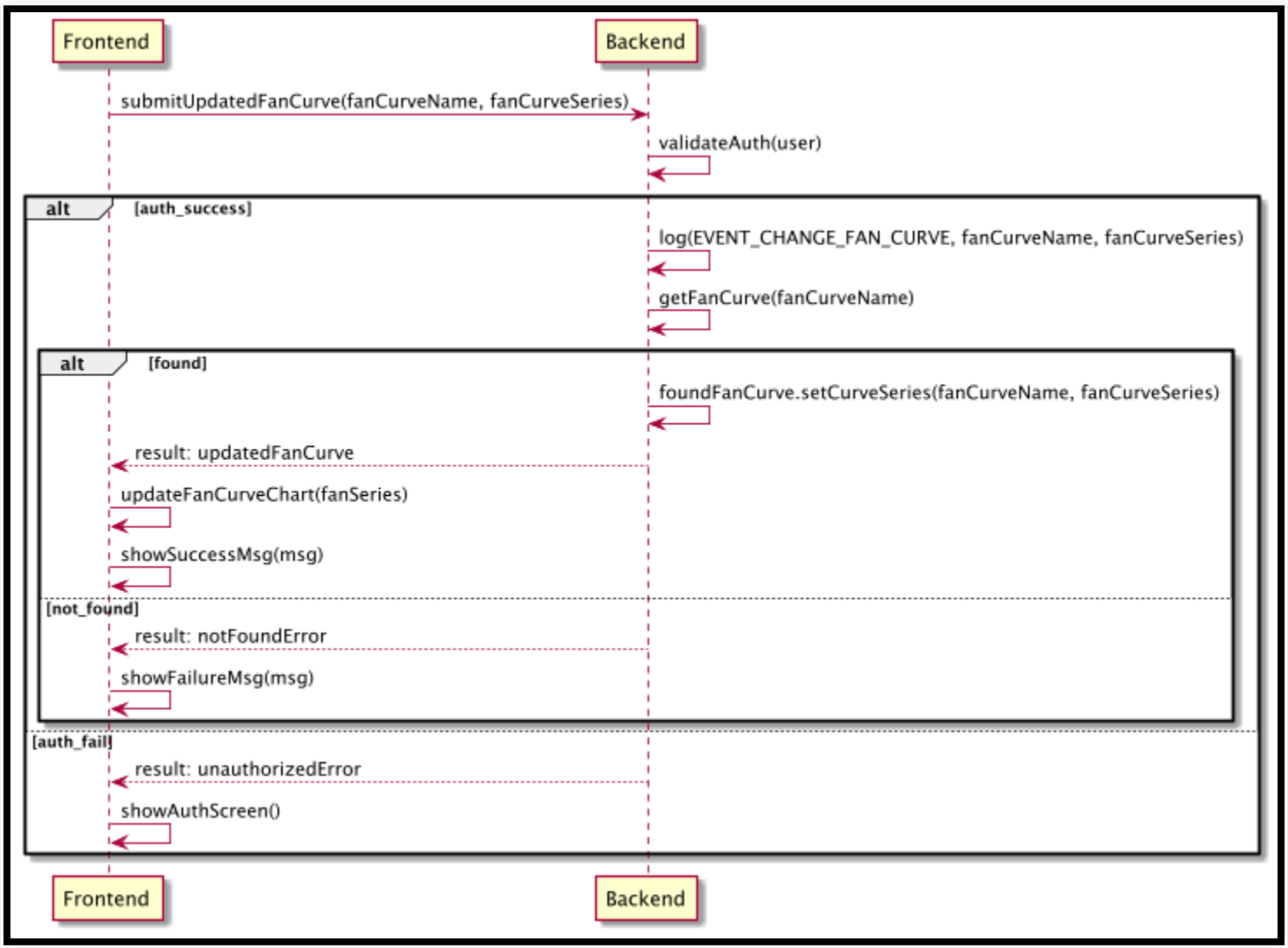


Sequence diagrams

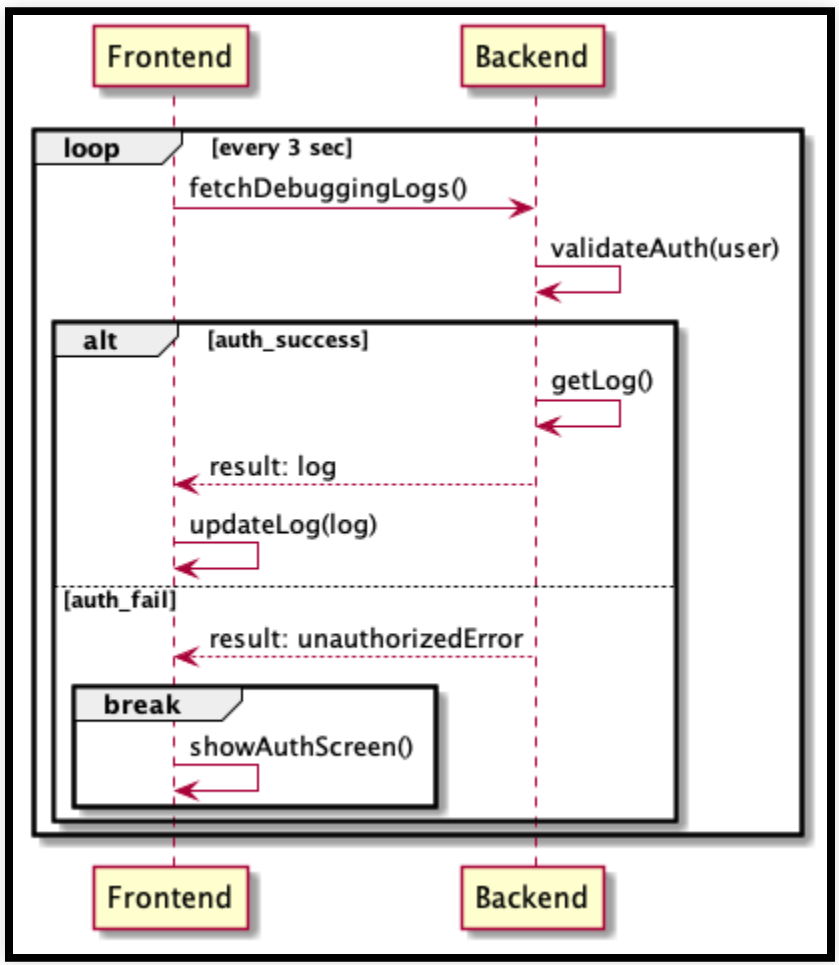
Add new fan curve



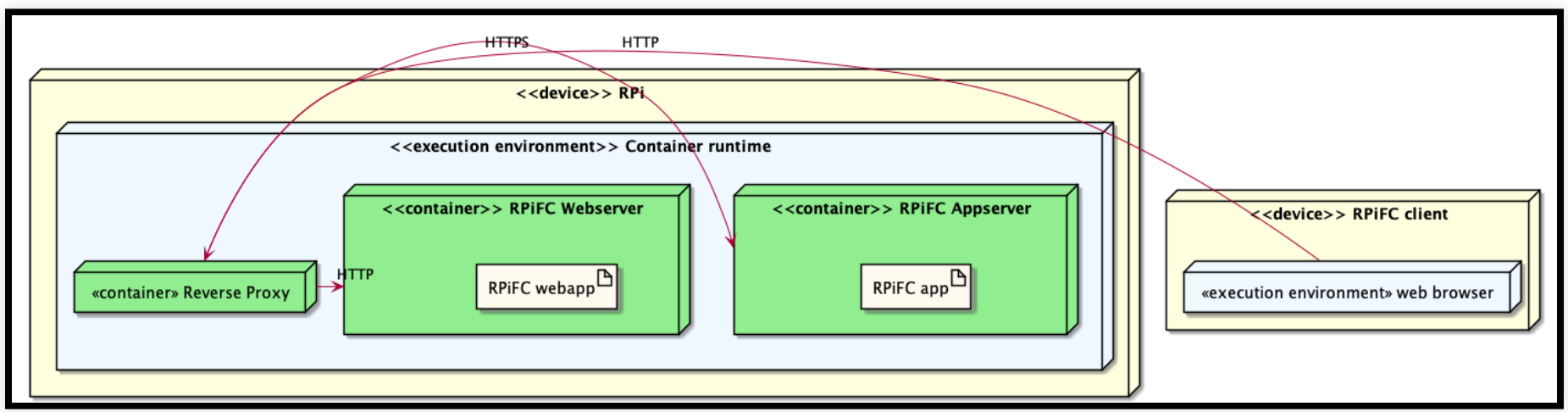
Change existing fan curve



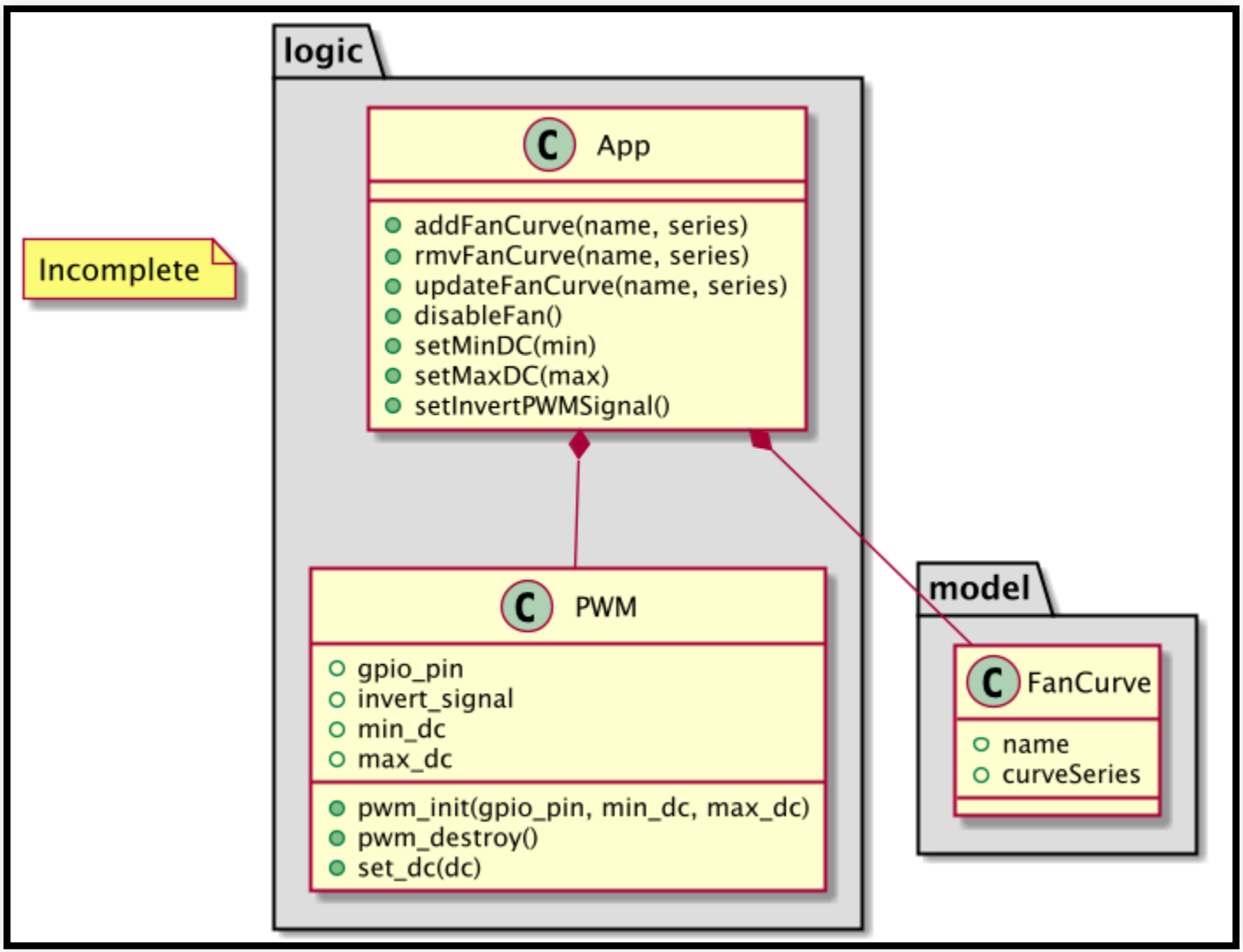
Update Web-UI (e.g., Log)



Deployment diagram



Class diagram



Business Model

1. Customer Segments

- IOT tinkerers
- Pi 4 Desktop users (w/o suitable case)

2. Value Propositions

- Improved performance (less throttling)
- Ensuring HW longevity
- Less noise (instead of running fan @ full speed)
- DIY experience (instead of buying case w/ fan)

3. Channels

- GitHub
- Docker Hub

4. Customer Relationships

- Feedback / Feature requests via GitHub (Issues)

5. Revenue Streams

- GitHub Sponsors
- Patreon

6. Key Activities

- Development
- Testing & Validating w/ real hw

7. Key Resources

- SW developers
- HW (soldered circuit board, Pi, Fan)

8. Key Partners

- GitHub
- Docker Hub

9. Cost Structure

- SW development expenses
- HW expenses