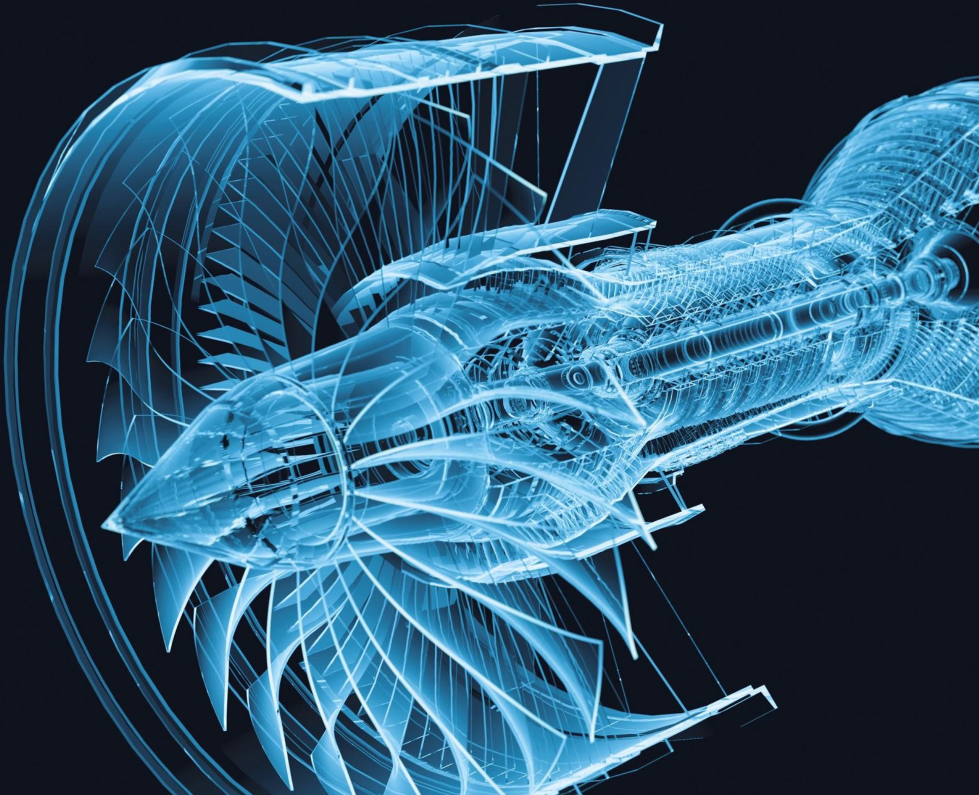
JETSTACK

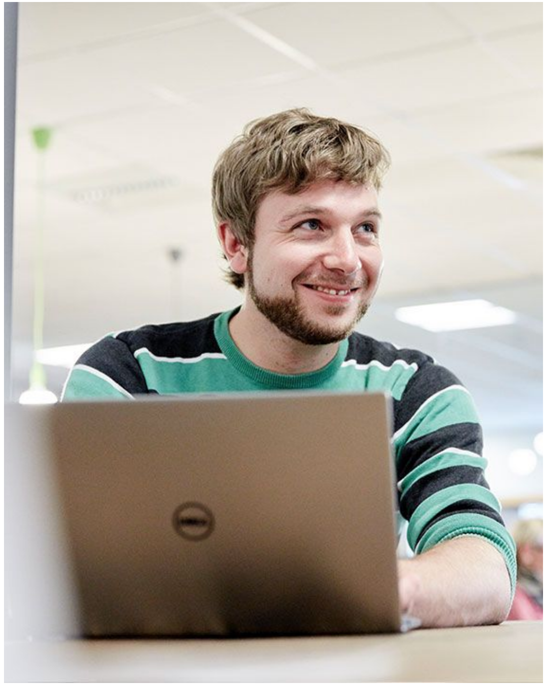
Presented by
Christian Simon

Why Running ***kubelet*** on your Vacuum Robot is (not) a Good Idea



jetstack.io

WHO AM I?

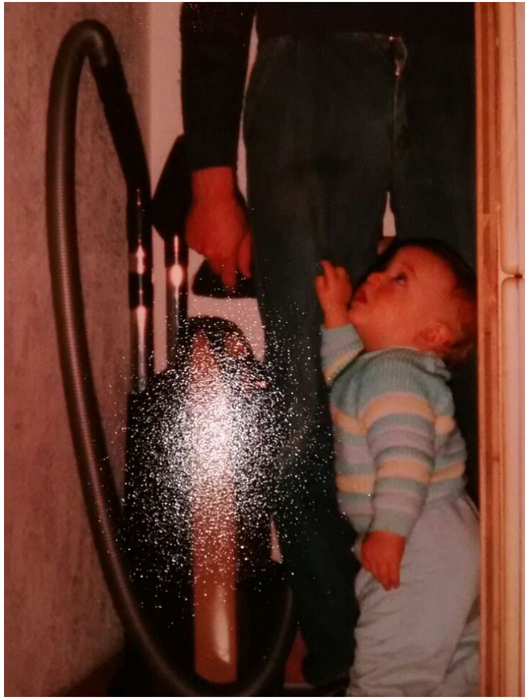


Christian Simon

- Using Kubernetes for 3+ years
- Started *kube-lego*
- Using vacuum cleaners for 25+ years

🐦 simonswine
🐙 simonswine

WHO AM I?



Christian Simon

- Using Kubernetes for 3+ years
- Started *kube-lego*
- Using vacuum cleaners for 25+ years

[simonswine](https://twitter.com/simonswine)
[simonswine](https://github.com/simonswine)

AGENDA

- Introduction
- Hardware capabilities
- Software stack
- Cloud Liberation
- Demo
- Further work

FOUNDATIONS

Standing on the shoulder of giants

- Dennis Giese & Daniel Wegemer (34C3)
  Unleash your smart-home devices: Vacuum Cleaning Robot Hacking

- github.com/marcelrv/XiaomiRobotVacuumProtocol
  Xiaomi Robot Vacuum Protocol description

- github.com/rytilahti/python-miio
  This library (and its accompanying cli tool) is used to interface with devices using Xiaomi's miIO protocol.

IDEAS

“Your talk submission has been accepted”

- Running a HA control plane
  - ✗ doable, does not fill a Kubecon time slot
- Running actual workload containers
  - ✗ quite hard, not that much resources available
- Vacuum robot combat league
  - ✗ really interesting, but expensive and no value provided
- Liberate from vendor’s cloud
  - ✓ interesting task, might provide wider value

HARDWARE

Xiaomi Mi Robot

- Cheap ($250-350)
- Hardware revision v1 / v2 exist
- In March a new firmware released which backported v2 features to v1
- Wifi chip is 2.4 GHz band only
- Battery powers the SoC for ~ 2 days (UPS)
- USB-OTG port

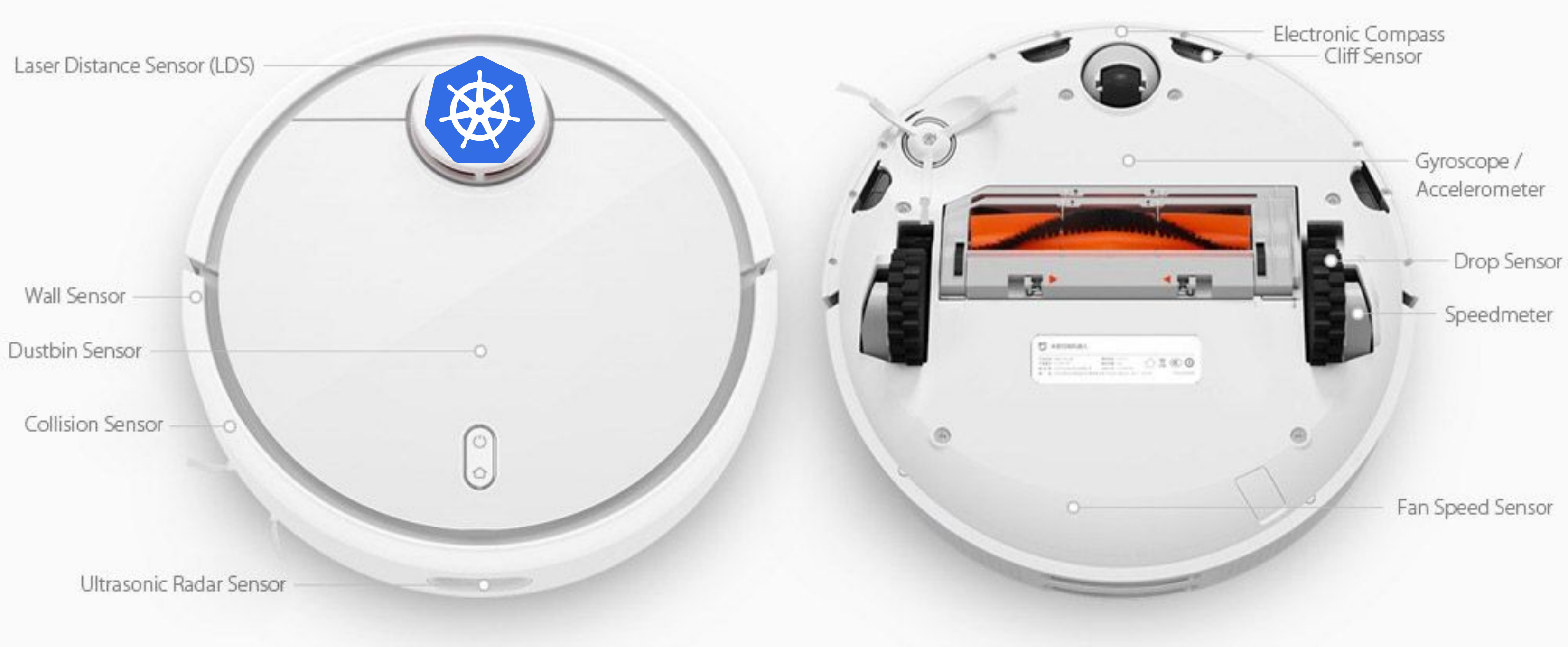
HARDWARE

CPUs

- SoC Allwinner R16
  Quad-Core Cortex-A7 ARM CPU
  512 MB RAM / 4GB Flash
- STM32F103 Microcontroller
- TMS320F28026 Piccolo Microcontroller

HARDWARE

Sensors



(from Xiaomi datasheet)

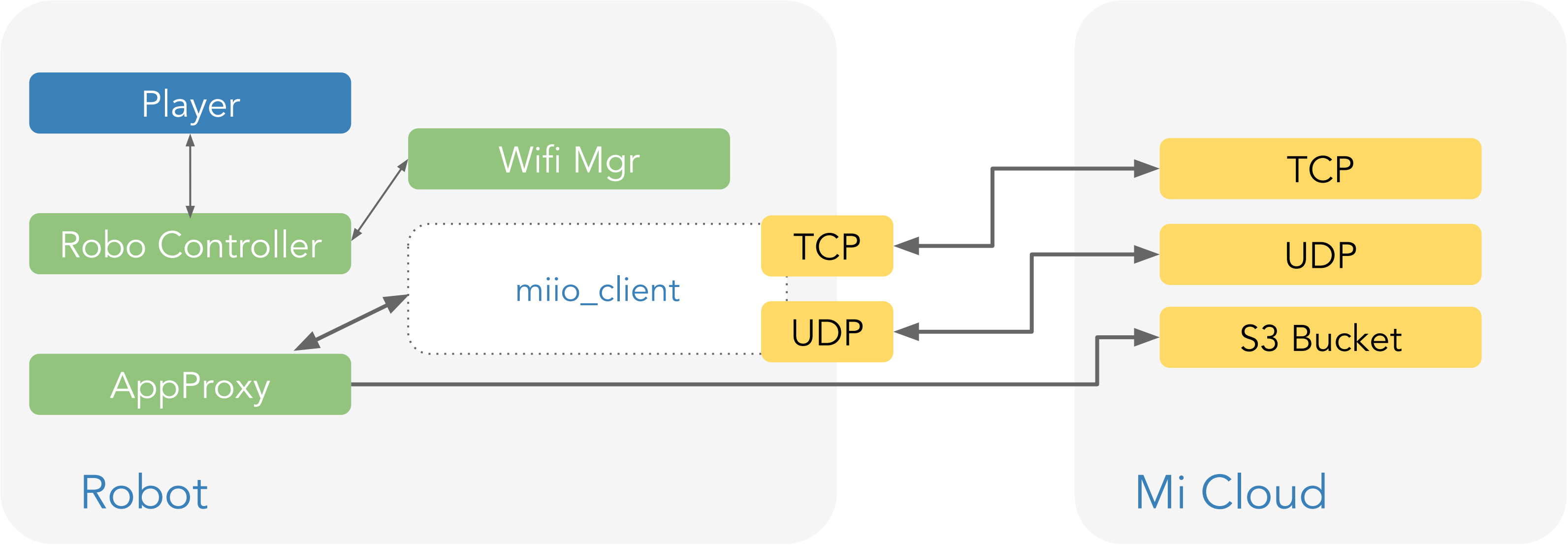
SOFTWARE

Stack

- UBoot
- Kernel 3.4.39 - with Android patches
- Ubuntu 14.04.3 LTS
- Player 3.10-svn robot interface server (including proprietary modules)
- Proprietary software
    - Communication with Cloud
    - Management of robot
    - Driving of the vacuum

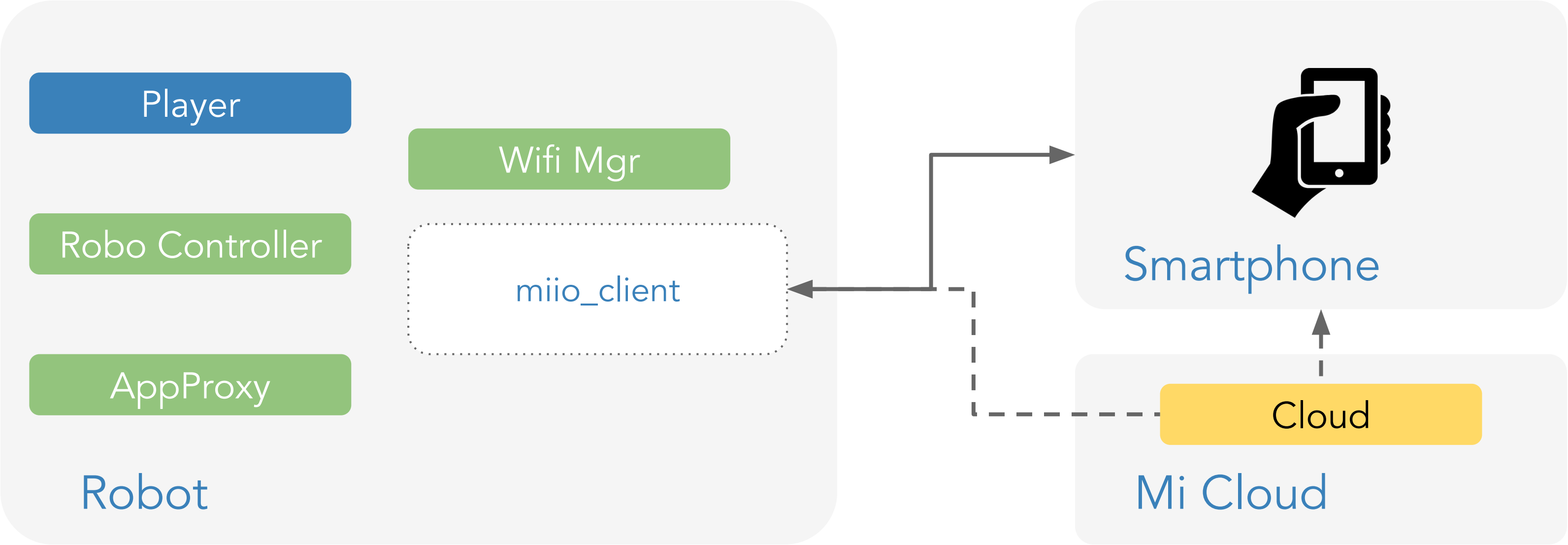
SOFTWARE

Communications Architecture



SOFTWARE

Communications Architecture



SOFTWARE

Protocols

- Xiaomi Robot protocol
  - JSON based
  - Request, Response
- Cloud protocol
  - Binary protocol, encrypted using AES
  - Wraps robot protocol over insecure channels
  - Contains sequence id
  - Supports UDP and TCP

CLOUD LIBERATION

- No control over data
- 34C3 discovered questionable remote debug commands
- Long terms support and EOL handling
- Data protection not mentioned in the manuals
- Potentially no data protection legislation enforceable in vendors jurisdiction

What about ☸ ?

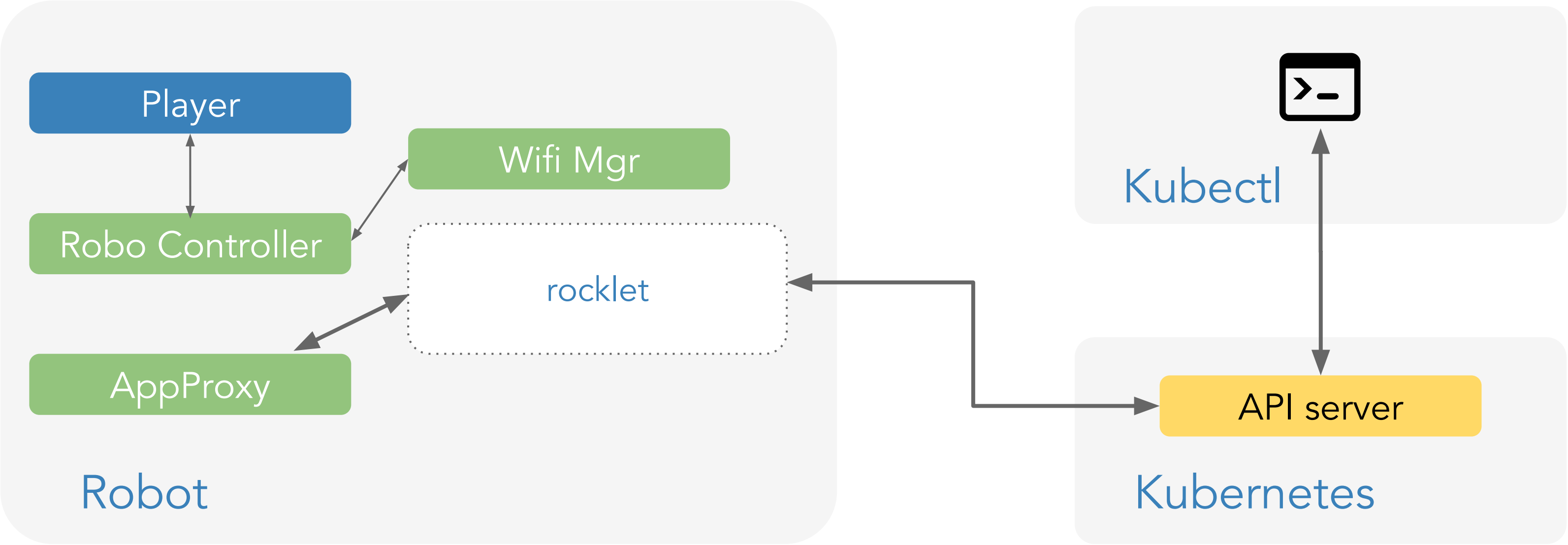
CLOUD LIBERATION

Cloud Native Landscape

- Using Kubernetes as a reference implementation of a distributed system
- Many areas of the Kubernetes code base reusable/extensible
- Benefit from tools out of the landscape
  - Monitoring using Kubernetes
  - Using kubectl to discover data and send commands
  - Kubernetes API powerful Auth(z)

CLOUD LIBERATION

Revised Architecture



CLOUD LIBERATION

Rocklet - “Fake Kubelet”

- A mock kubelet, not actually executing any container
- Taint nodes to not attract any “real” Pods
- Allows using higher level resources like Jobs
  (esp. CronJobs are covering interesting use cases)
- Conditions can get reported

CLOUD LIBERATION

Rocklet - Pod controller

- Pod Controller watches for scheduled pods
    - Forwards commands to the AppProxy
    - Reports status back like any other terminating pod

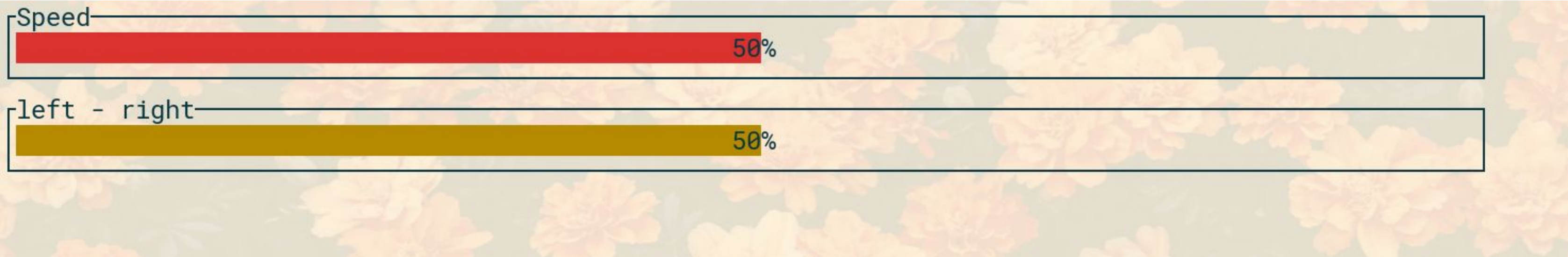
```
spec:
  containers:
  - image: app_goto_target
    args:
    - “[x,y]”
```

Schedule a cleaning

CLOUD LIBERATION

kubectl exec

- Allow low latency remote control of the vacuum
- Send direction commands to the robot
- Used for the contest later



```
┌Speed─────────────────────────────────────────────────┐
│                         50%                          │
└──────────────────────────────────────────────────────┘
┌left - right──────────────────────────────────────────┐
│                         50%                          │
└──────────────────────────────────────────────────────┘
```

CLOUD LIBERATION

CRD *Vacuum*

- Export actual map and path driven
- Report status of vacuum back
- spec could be used to configure parameter of the vacuum
  - audio volume, fan speed

CLOUD LIBERATION

CRD *Vacuum*

```
status:
  area: 2732500
  batteryLevel: 95
  charger:
    x: 125
    y: 79
  doNotDisturb: false
  duration: 4m53s
  errorCode: 0
  fanPower: 60
```

CLOUD LIBERATION

CRD *Cleaning*

- Exports the state from historic cleanings
  - Completed, Error code, duration, area

Maps

- Historic maps are read from SQLite database
- Live map is read once vacuum runs
- PNG is stored in the CRDs
- Path decoded and converted as value pairs

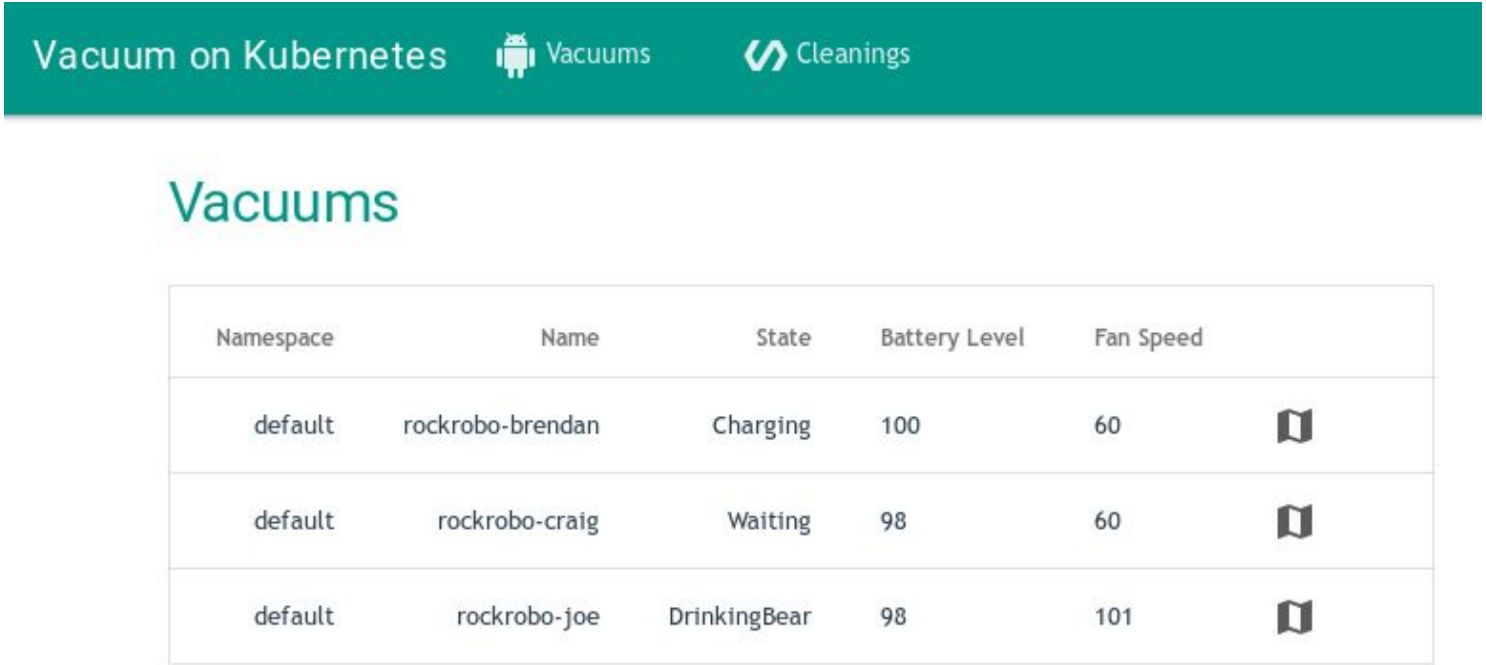
CLOUD LIBERATION

Web UI

- Backend uses client-go to cache CRDs
- ELM used for the frontend
  - Draws map using SVGs
  - Decodes and visualizes cached kubernetes API from backend
- Backend sends notifications to frontend on updates
- Material design lite CSS/JS

CLOUD LIBERATION

Web UI

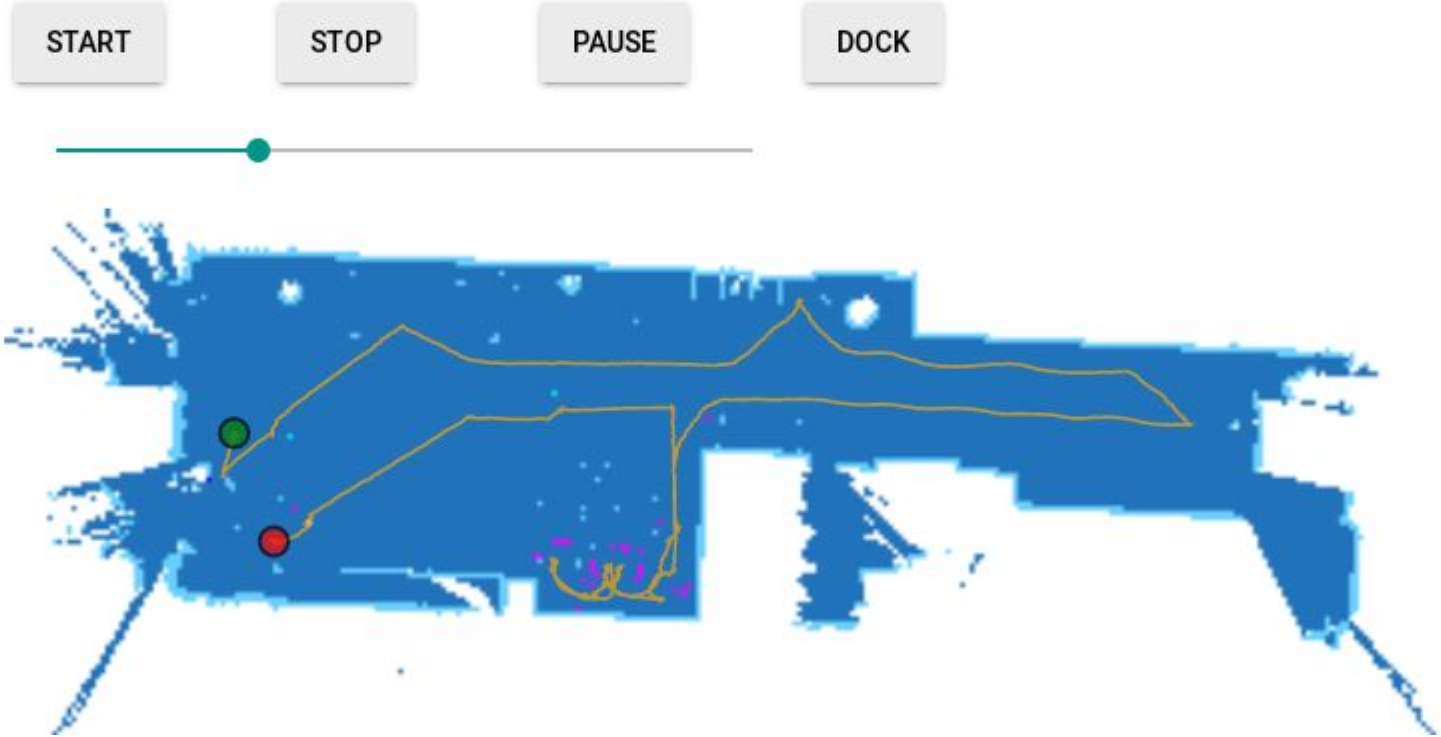


Vacuums

Namespace	Name	State	Battery Level	Fan Speed	
default	rockrobo-brendan	Charging	100	60	🗺
default	rockrobo-craig	Waiting	98	60	🗺
default	rockrobo-joe	DrinkingBear	98	101	🗺

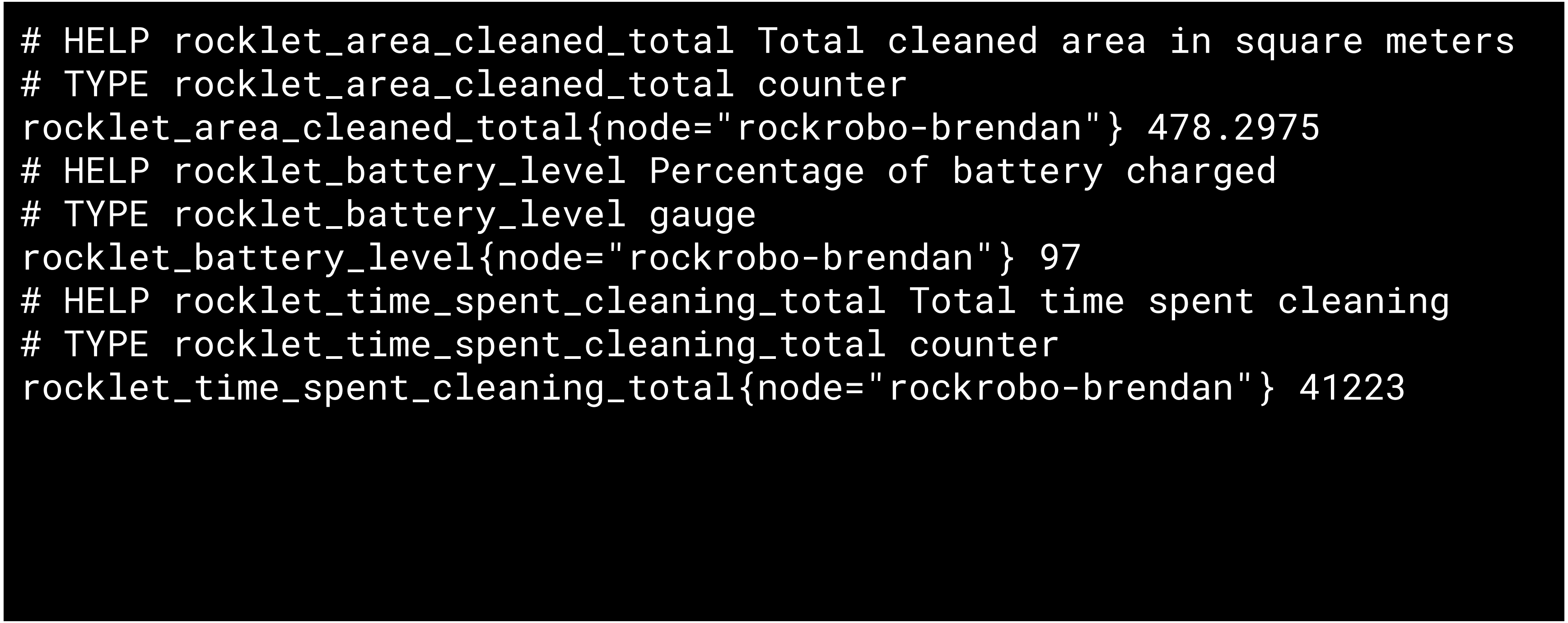
CLOUD LIBERATION

Web UI



CLOUD LIBERATION

Prometheus metrics



```
# HELP rocklet_area_cleaned_total Total cleaned area in square meters
# TYPE rocklet_area_cleaned_total counter
rocklet_area_cleaned_total{node="rockrobo-brendan"} 478.2975
# HELP rocklet_battery_level Percentage of battery charged
# TYPE rocklet_battery_level gauge
rocklet_battery_level{node="rockrobo-brendan"} 97
# HELP rocklet_time_spent_cleaning_total Total time spent cleaning
# TYPE rocklet_time_spent_cleaning_total counter
rocklet_time_spent_cleaning_total{node="rockrobo-brendan"} 41223
```

Demo time 🤞🙏

FUTURE IDEAS

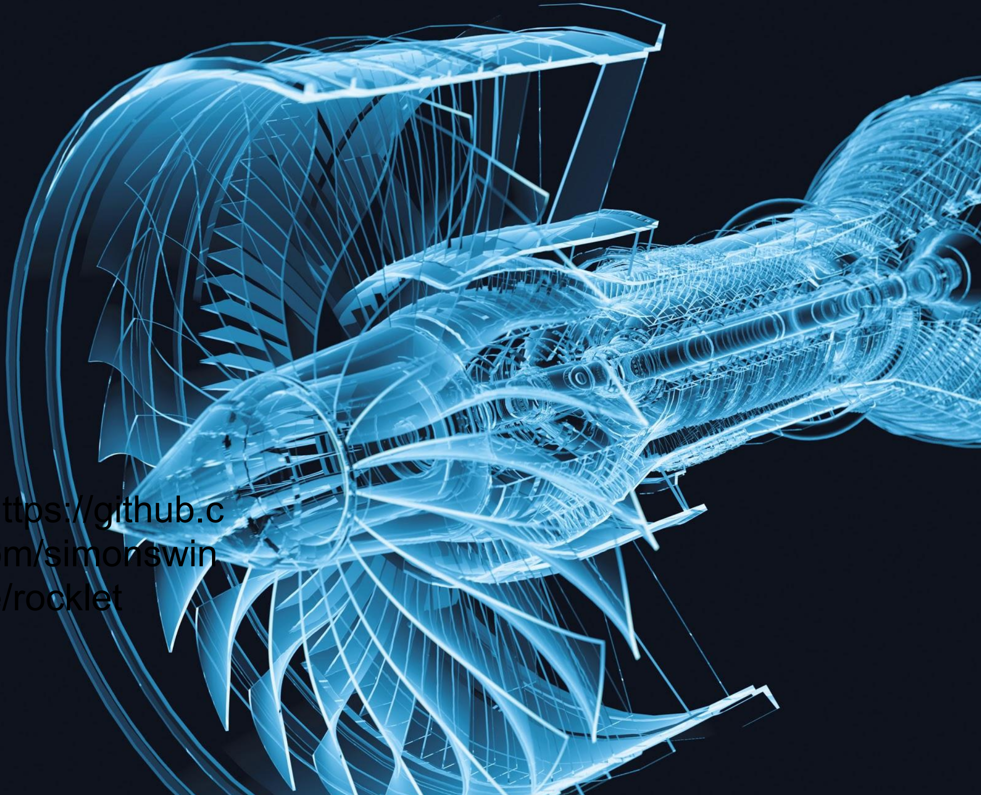
- Get rid of websockets from API server to robot
  - Are not really scalable
  - Require bidirectional communications
- Custom API server might be better suited
- Using namespaces for different tenants
- Vacuum Native Jobs and CronJob as CRDs
- Write a code generator for Elm
  - JSON decoder
  - GET, LIST operations

FUTURE IDEAS

- Implement graphs and more metrics in Prometheus
  - Consumables status for filters and brushes
- Use a controller to move the vacuums relative to each other

JETSTACK

Win one of the Xiaomi Robots at the Jetstack booth at 3pm



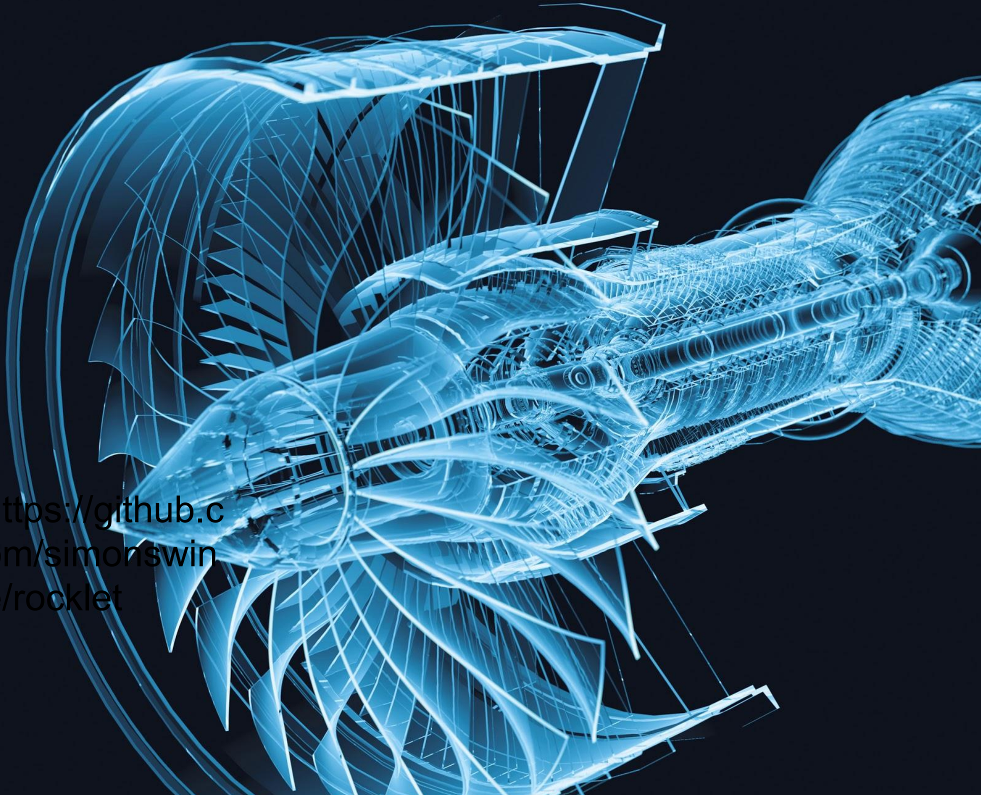
https://github.com/simonswine/rocklet

jetstack.io

JETSTACK

Source code

- github.com/simonswine/rocklet
- github.com/simonswine/rocklet-ui



https://github.com/simonswine/rocklet

jetstack.io

SOFTWARE

Enable a 5GHz Wifi adapter

- Kernel config was compiled into the kernel (extract-ikconfig)
- Kernel sources could be found on GitHub (GPL ftw)
  - An Android fork of upstream kernel 3.4
- Realtek USB adapter conflicted with built in wifi chip
- Some modifications necessary to compile the wifi module
  - Backport to 3.4 version
  - Modifications to cross-compile for ARM