

Raspberry Pi on the edge

configuration mgmt with ansible for on-site computing equipment

Peter

- Computer scientist
- 10+ years in the business
- Husband and Daddy
- Skater
- Currently starting my own business



Karl

- DevOps engineer at synyx
- Works with ansible, golang, RPis, Gitlab etc.
- "From the first screw til the last byte"
- Also engaged in HR and Sales

synyx *code
with
attitude*

Advertisement!

We're hiring! Have a look at
www.synyx.de/jobs

synyx code
with
attitude

Project Environment

- Container-Logistics
- Different sites along the Rhine
- Hundreds of km between sites
- Very diverse infrastructure



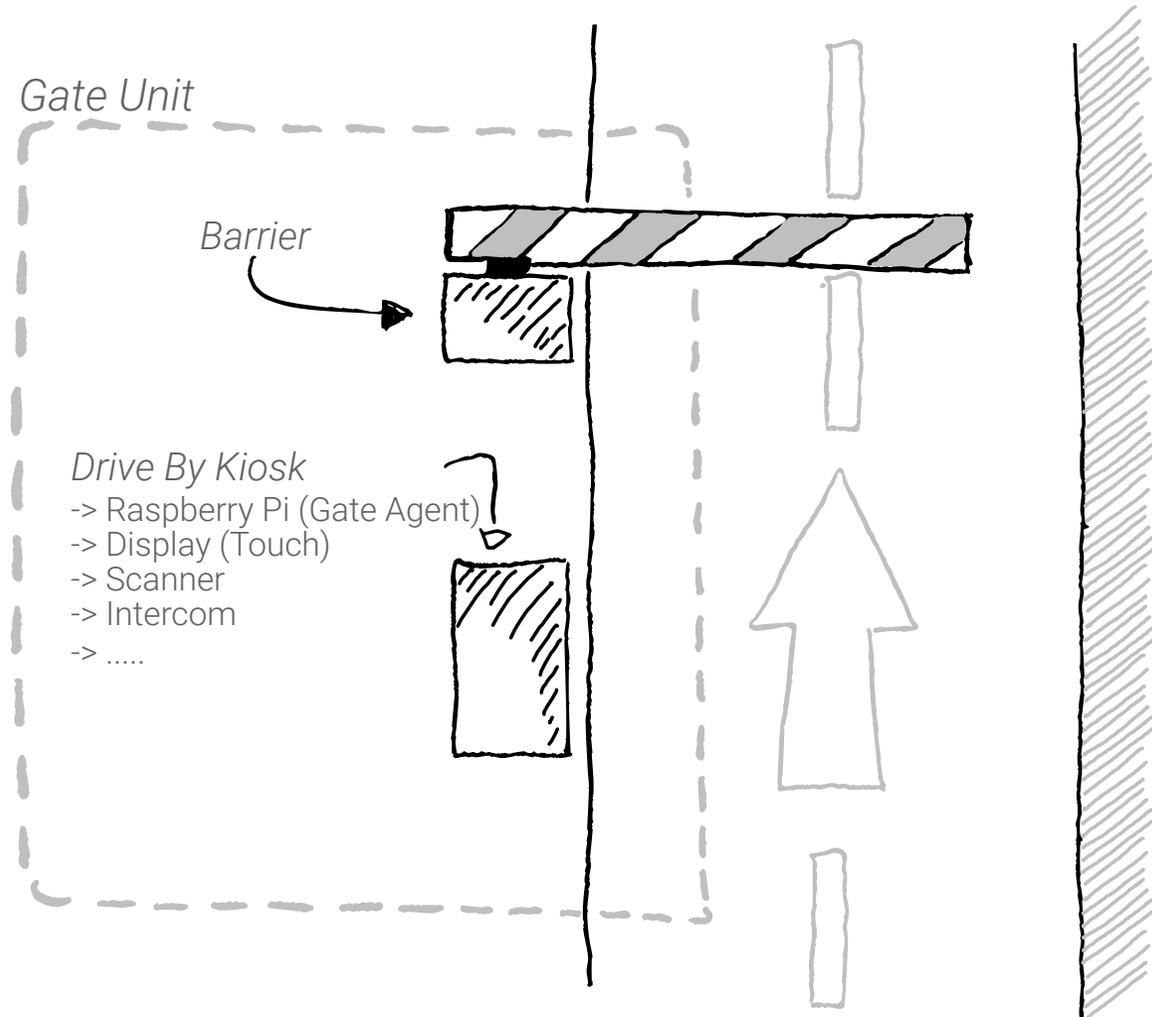
Tasks at hand

1. Trucker check-in kiosk
2. Dashboards for process and statistics monitoring
3. Truck entry and exit

Idea:

- Same base for all three use-cases
- Raspberry PIs: commodity-hardware, cheap, battle proven,
- Ansible for provisioning: Easy to use and learn, lightweight, extensible

=> Focus of this talk: Truck entry and exit



Truck entry and exit in a nutshell

- Truck arrival at drive-by kiosk
 - Trucker scans QR code
 - Check of terminal permission
 - Show information on display
 - Print receipt (exit)
 - Grant access by opening barrier
 - Inform downstream systems
 - ...
- => Powered by the "Gate Agent"

Challenges

"Never block the process"

"It's all about reliability"

Reasons for blocked processes

1. Hardware related Problems
=> Environment is rough
2. Bugs and other software related problems
=> We all know, they're right around the corner

Damages are going to happen, deal with it!





How to provide proper reliability

- "Help yourself philosophy"
 - Spare parts available
 - Kiosk: robust, simple, sufficient space, modular, customizable
- We need metrics for temperature, software, load etc.

Operations

- Metrics like temperature collected by telegraf :)
- Business Metrics
- Plug and Play (Help yourself philosophy)
- Reliability
- Software Upgrades

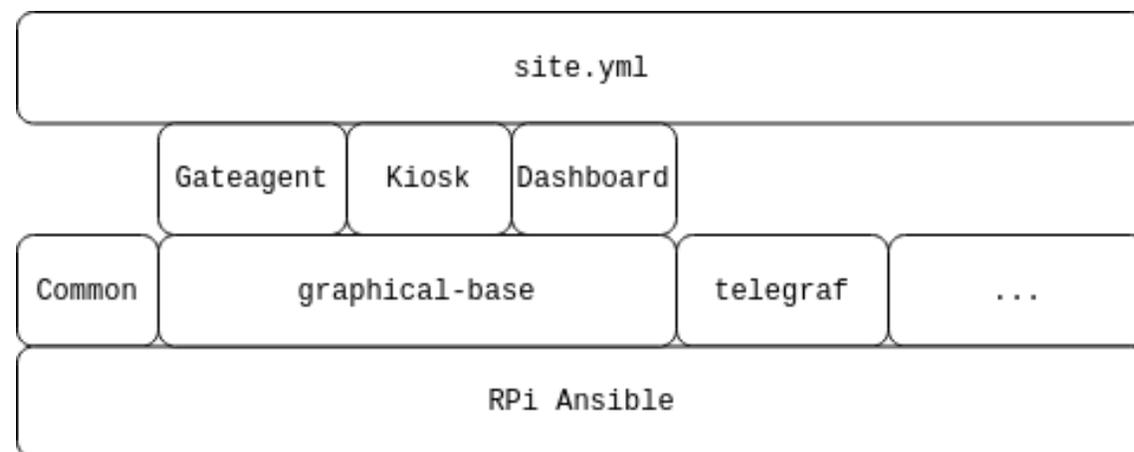
Reliability

- Read Only File system
- Replacement Hardware
- PXE Boot
- Initrd rescue mode
- Watchdog
- Agent is a stateless system

All with ansible <3

Provisioning Philosophy

- small composable roles
 - Kiosks
 - Dashboard
 - Gateagent
- Composed by site.yml



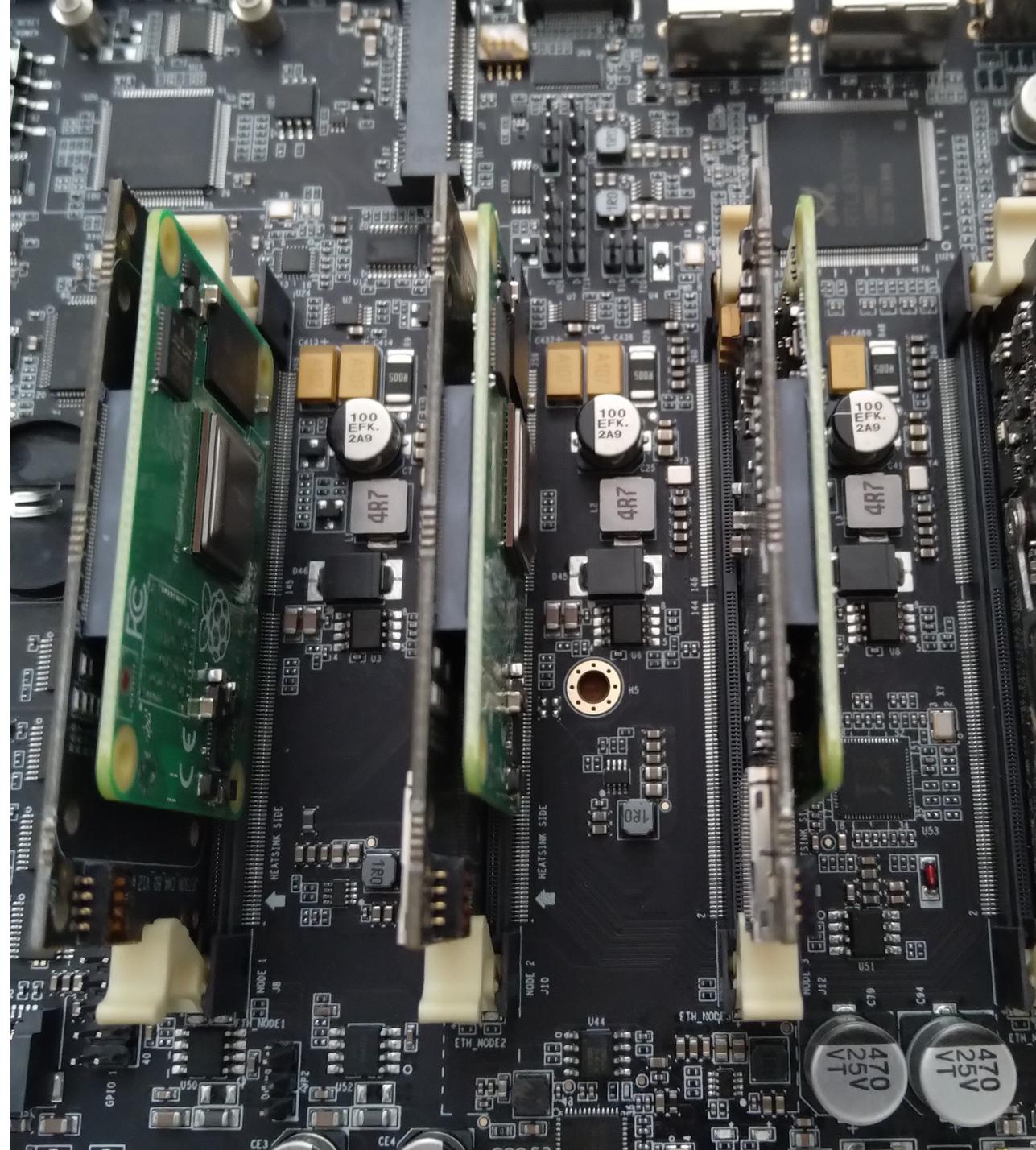
Inventory

- Yaml Format
- grouped by sites
- Single Source of truth
 - IP
 - printers
 - w3w location (geographical location)
 - ...
- Vars are converted to environment files for systemd or bashscripts

```
inventories > ! hostsPresentation.yml > {} all > {} children > {} gatecontrol > {} children > {} gatecontrol-deka > {} var
1  all:
2      children:
3          gatecontrol:
4              vars:
5                  kioskGateRole: gateagent
6                  printFetcherMode: client
7                  http_proxy: http://myproxy.net:8080
8                  monitoring: true
9                  readonlyFS: true
10                 gate_shutdown_timeout: 120
11                 gate_printer_timeout: 0
12                 gate_message_queue: secret
13                 url: http://localhost:8080
14             children:
15                 gatecontrol-deka:
16                     hosts:
17                         gatecontrol-agent-deka-entry:
18                             ansible_host: 1.1.1.1
19                             prod_ip: 2.2.2.2
20                             w3w_location: ///fragment.slice.detect
21                             gate_config_name: entry-west-1
22                             gate_purpose: entry
23                             gate_command: exit 0
24
25                         gatecontrol-agent-deka-exit:
26                             ansible_host: 3.3.3.3
27                             prod_ip: 4.4.4.4
28                             printerIp: 5.5.5.5
29                             w3w_location: ///guessing.skipped.lineno
30                             url: http://localhost:8080
31                             gate_config_name: exit-west-1
32                             gate_purpose: exit
33                             gate_command: exit 0
34
35                     vars:
36                         router: 1.1.1.1
37                         dns: 8.8.8.8
38                         os: raspbian.img.gz
39                         gate_location: DEKA
40
41
```

Continuous Integration

- We use gitlab for code/change review
- Every MR is build via ansible on test systems
- Other inventory is used
- Turing Pi as target for testing
 - More compact
 - On board management bus for power management



You build it you run it

- No regex (if possible)
- When something is missing, code it
 - Editing cmdline.txt was a real pain
- Software for controlling the barrier also written by us

Closing thoughts

- Ansible is integral part of our stack
 - Deployment of embedded devices on different sites
 - Continuous Deployment
 - Handling different environments
 - "Automate the shit out of it" :)