# Making bare-metal testing accessible to every developer

Martin Roukala (néé Peres), Valve contractor

# Who am I?

- Martin Roukala (used to be Martin Peres)
- Freelancer at MuPuF TMI
- Valve contractor
- Previous projects: Intel GFX, Nouveau

# Why?

# Why build a bare-metal test farm?

- GitLab brought building and unit-testing capabilities for all drivers \o/
    - Won't prevent all regressions though...
        - Hardware can be emulated, but models are not imperfect
        - Users run your driver on hardware, not your model
        - Integration testing requires HW missing from GitLab public runners
- Your farm increases your productivity:
    - Never lose your context when your change crashes your machine
    - You can test your changes on different generations of HW in one go
    - You can debug issues using interactive sessions, like you would on your PC
    - You can let colleagues test their changes on rare HW

Because it's good for you...

and the project!

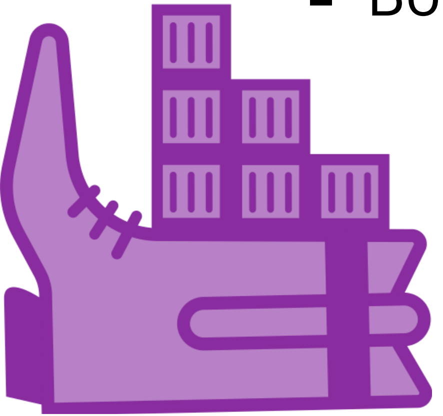
# High level requirements

- Needs to be more convenient than whatever developers currently use:
  - Easy/fast to add the machine to the farm
  - Easy/fast to switch hardware without needing reconfiguration
  - Full flexibility of deployment
  - The work done by someone shouldn't influence your work
  - Maintenance time: ~1h per week
- High availability: Resilient to short power / network outages
- Minimal security risks (no botnet, crypto mining, home spying, ...)
- Minimal risks for the flat/house/building

# Solution #1: Use containers, not OSes

**Benefits:**

- Go distro-less with boot2container, an initramfs with a declarative interface
  - No need to install/maintain/repair your test machine's distros!
- Make every boot fresh, one job cannot influence the next one
  - Use volumes to cache data between executions (backed-up in MinIO)
- Fast boot by caching all the container layers on the test machine
  - Only download what has changed, not the whole disk image!
- Re-use the same containers across all machines
  - Bonus points for re-using the ones you already made for your CI



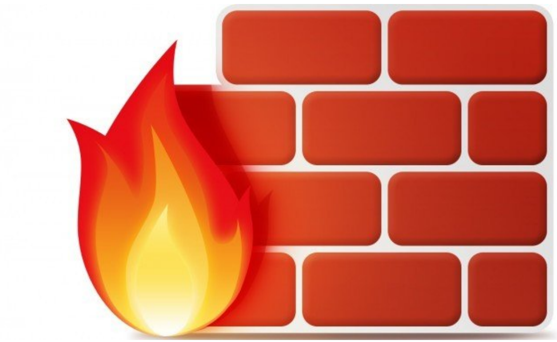
# Solution #2: Automate everything

- Auto-deploy using PXE/network boot
- Auto-enrolling:
  - Auto-discover the hardware and assign tags
  - Auto-test the hardware's boot reliability
  - Auto-expose the machine on GitLab when passing the test
  - Auto-re-enrollment if the machine changed its tags
- Auto-discovery:
  - Serial port -> machine (SALAD)
- Self-tests: Make it clear when assumptions are broken

- Limits?
  - How to turn on/off the machine

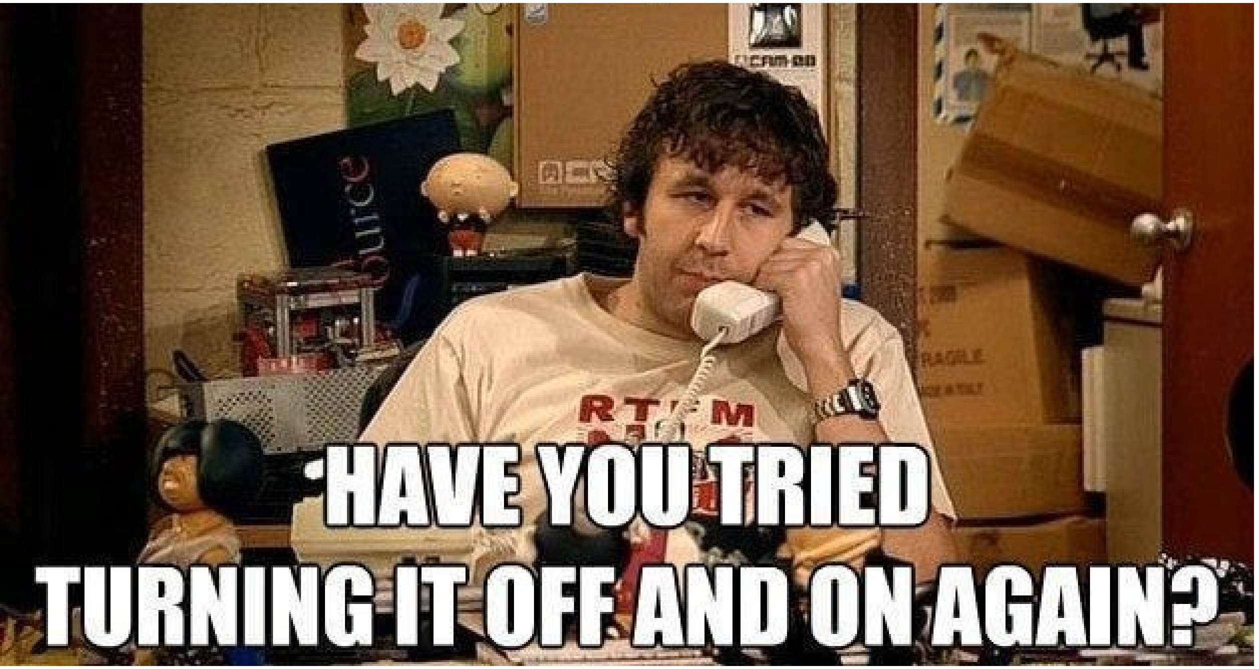
# Solution #3: Network security

- Put your CI infra in a separate network from your office/home
- Use a VPN to connect to the farm's gateway
- Block all un-needed ports and protocols at the router's level
- Whitelist the accepted domain names/IPs:
  - *.freedesktop.org / distro repo / docker hub / Quay / ...
- Give two network adapters to your gateway:
  - Public: Connected to your PDU and the internet
  - Private: Connected to your test machines (no routing to public)



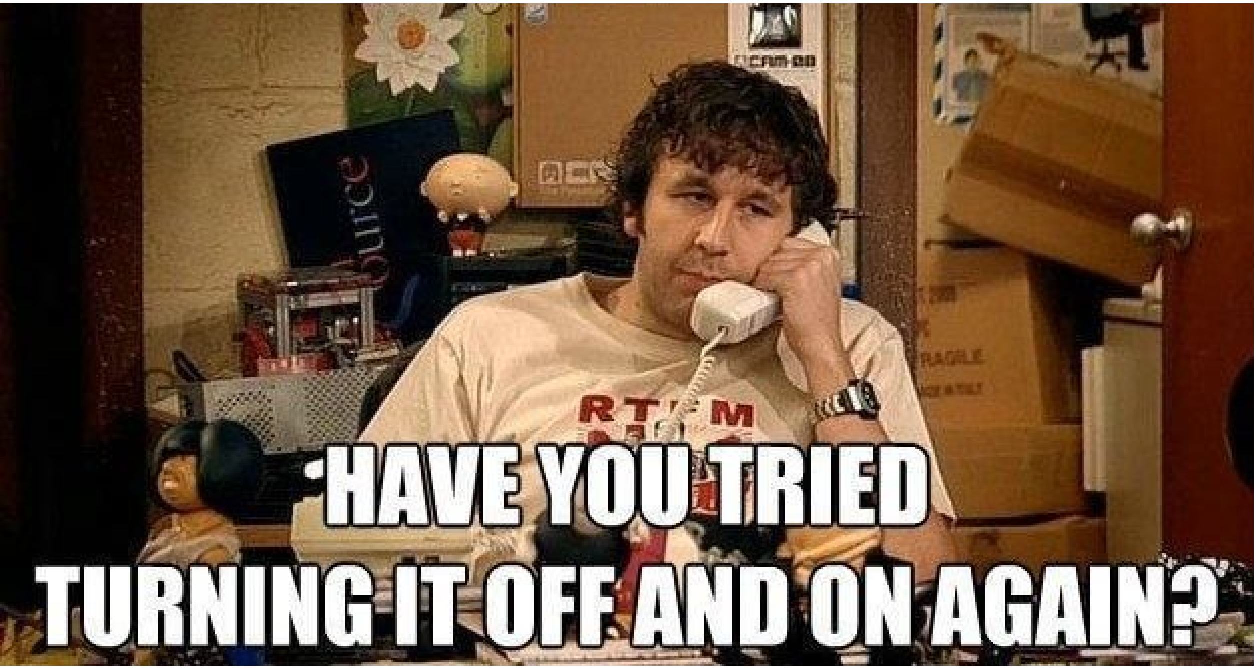
# Solution #4: Power cutting

Works around pesky hardware state
by always cold-booting

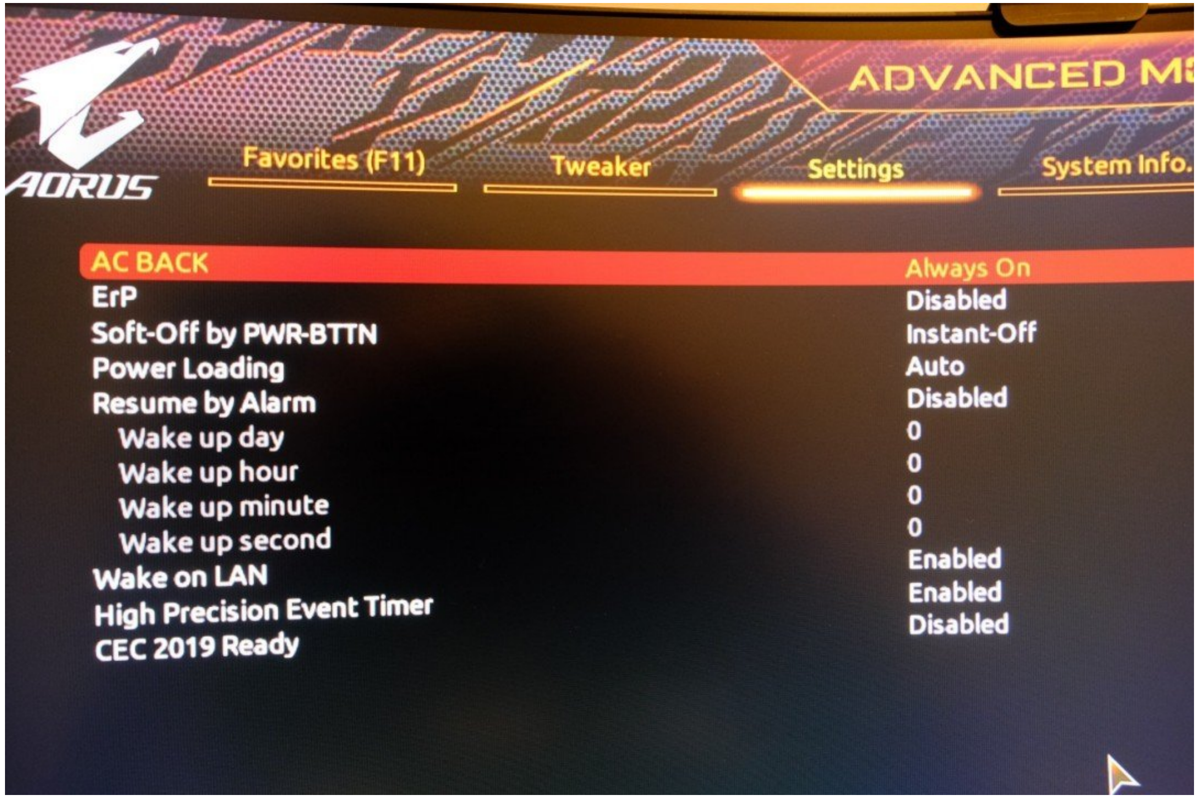


# Solution #4: Power cutting

Works around pesky hardware state by always cold-booting



Can also be used as a "let's boot up" signal



```
AC BACK                          Always On
ErP                              Disabled
Soft-Off by PWR-BTTN             Instant-Off
Power Loading                    Auto
Resume by Alarm                  Disabled
   Wake up day                   0
   Wake up hour                  0
   Wake up minute                0
   Wake up second                0
Wake on LAN                      Enabled
High Precision Event Timer       Enabled
CEC 2019 Ready                   Disabled
```

# Solution #4: Power cutting

## Switchable Power Delivery Unit

**Pros:**

- Industrial grade
- Guaranteed switching cycles
- Controllable using SNMP

**Cons:**

- ~$500 new
- ~$200 on ebay

# Solution #4: Power cutting

## Ikea's TRÅDFRI

**Pros:**

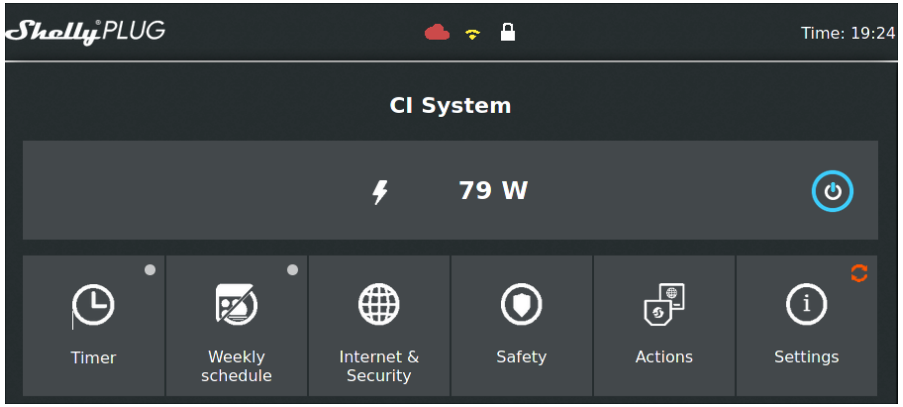
- Cheap
  - $35 for the gateway
  - ~$15 per socket
- Easy to find everywhere
- Protocol documented

**Cons:**

- Adding a new socket is annoying
- Wireless (could be a pro too)
- No rating on the switching cycles
- Inconvenient protocol

# Solution #4: Power cutting

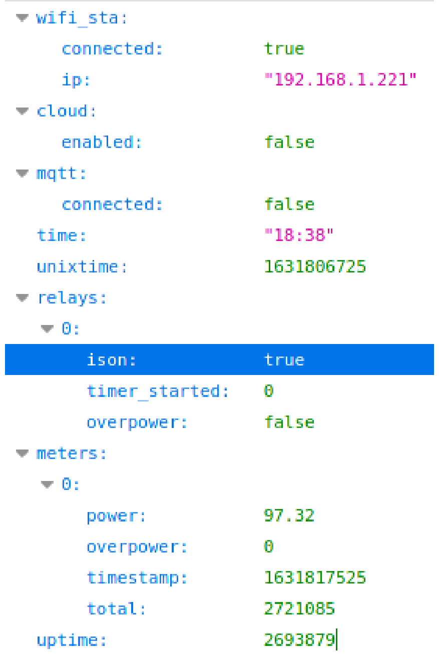
## Shelly Plugs



Time: 19:24

CI System

79 W



```
▼ wifi_sta:
    connected:          true
    ip:                 "192.168.1.221"
▼ cloud:
    enabled:            false
▼ mqtt:
    connected:          false
  time:                 "18:38"
  unixtime:             1631806725
▼ relays:
  ▼ 0:
      ison:             true
      timer_started:    0
      overpower:        false
▼ meters:
  ▼ 0:
      power:            97.32
      overpower:        0
      timestamp:        1631817525
      total:            2721085
  uptime:               2693879
```

**Pros:**

- Cheap (20-30 euros)
- Measures power usage
- REST / MQTT
- Easy to integrate with

**Cons:**

- Wireless (can be a pro)
- No switching ratings

# Solution #4: Power cutting

## PoE Network Switch

**Pros:**

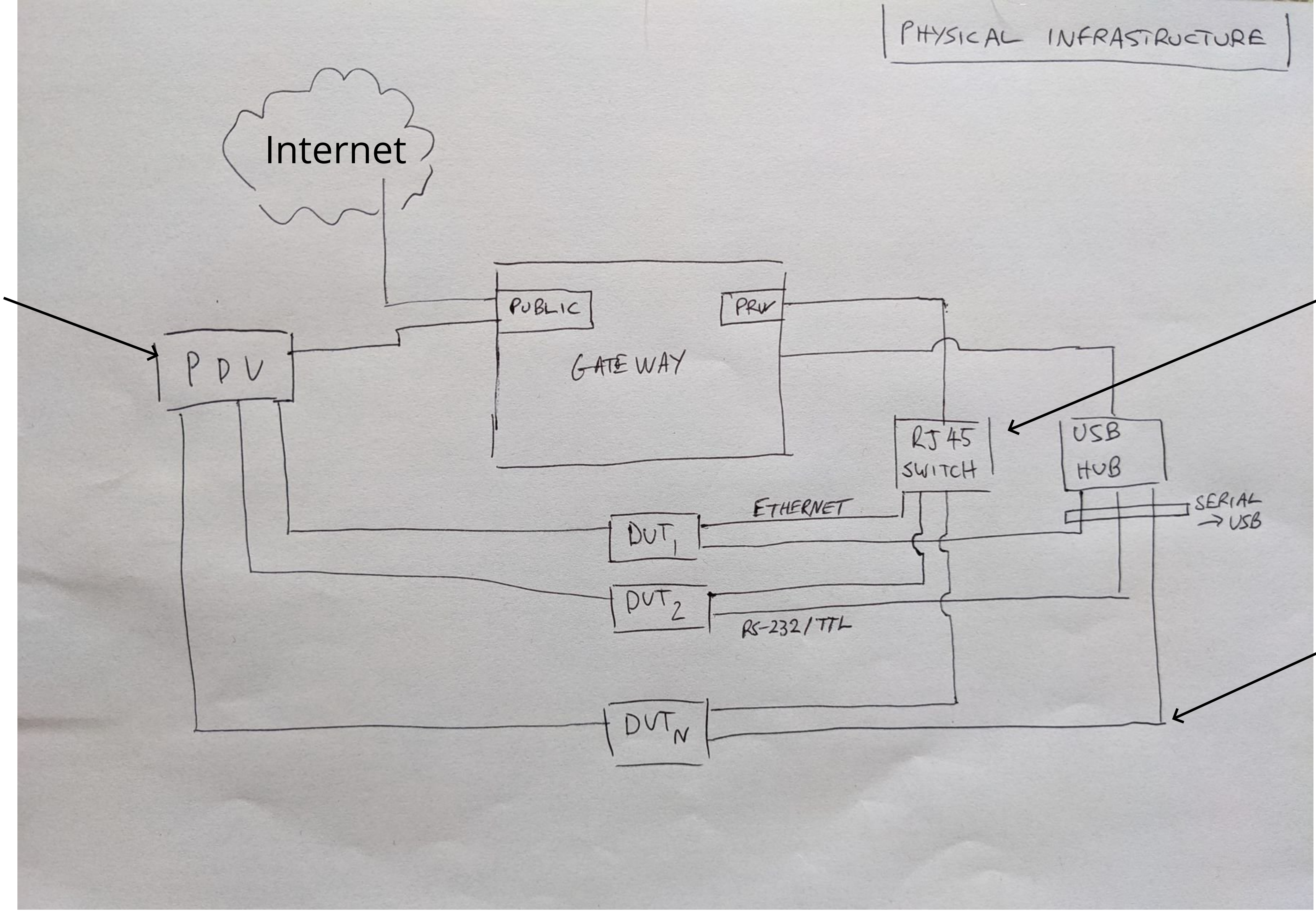
- One cable for both network and power

**Cons:**

- Only works for single-board computers
- May require an adapter on the receiving side
- Much more expensive switch

# Solution #5: Uninterruptible power supply

- Protects your hardware from surges and micro cuts
- To be used on all your networking equipment, and test machines
- WARNING: Check the power rating!



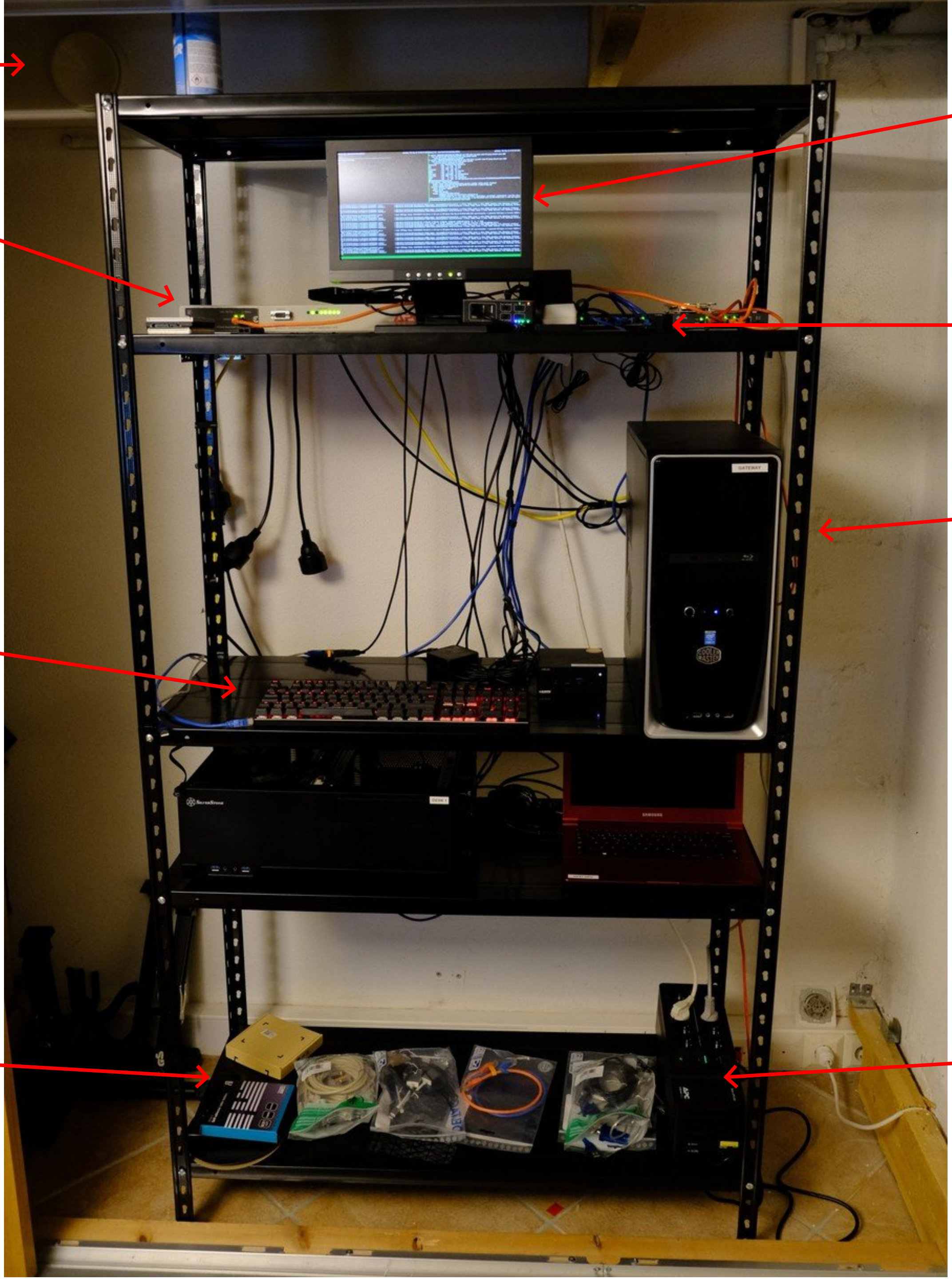
Internet

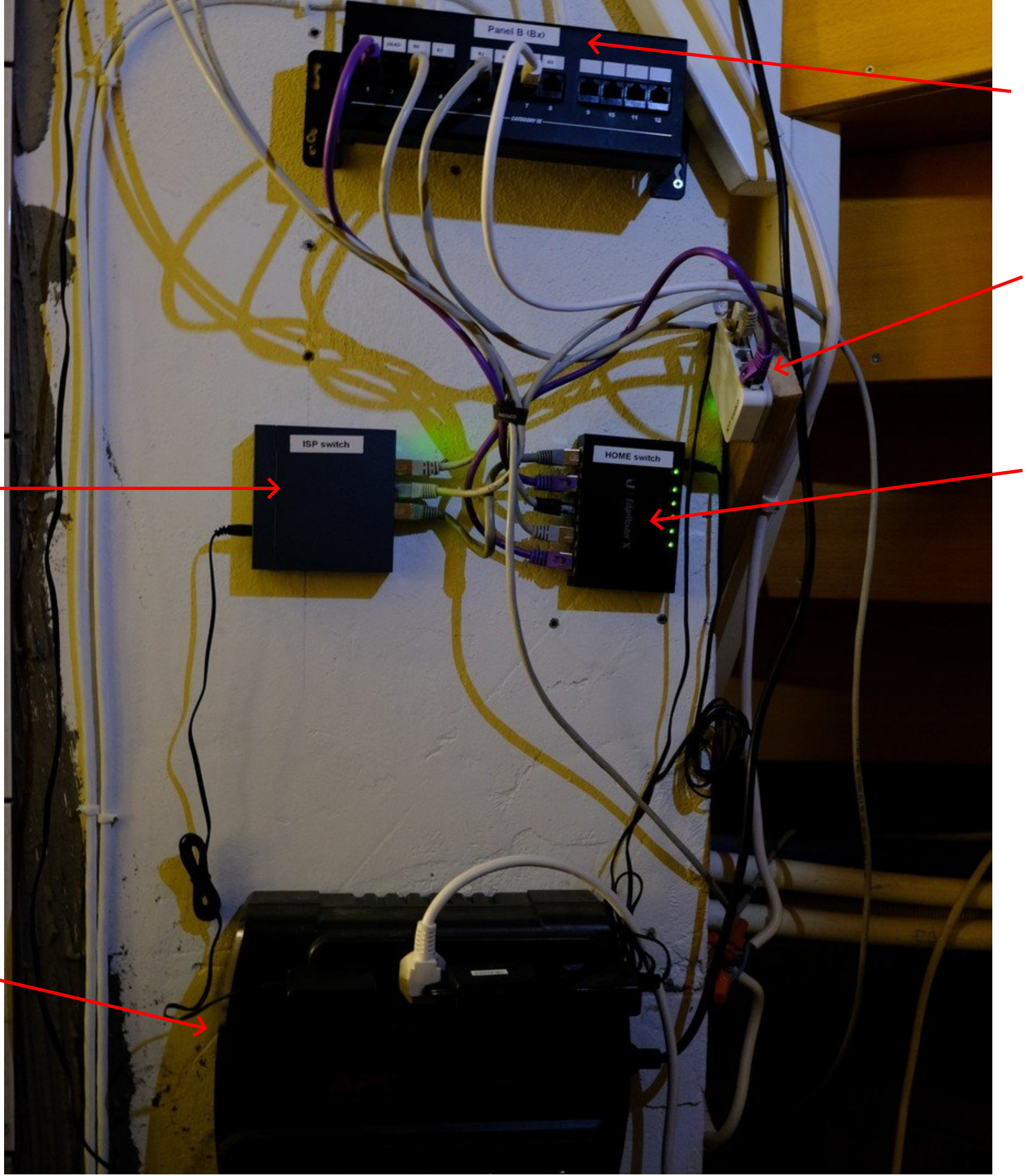
Switchable power supply

Networking

Serial console

# OK, but how does it look in practice?


Ventilation
Screen +
KVM
PDU
Network
switches
Metallic
shelf
Keyboard
Storage
UPS



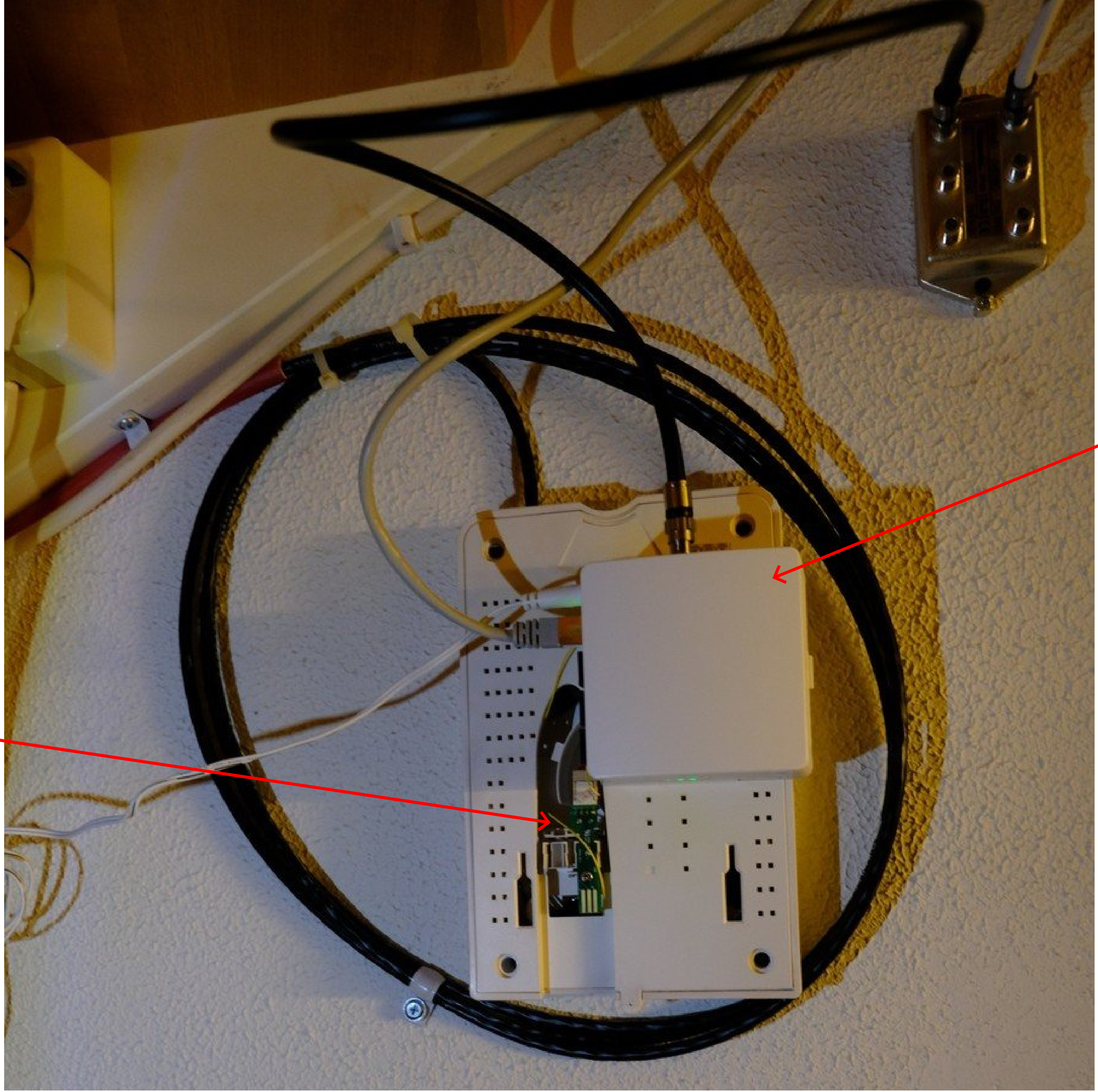
Patch panel

IOT switch

Home router

ISP switch

UPS



Fiber To The Home

Fiber modem

# What about the promised 1k€?

# Bill of material

- Gateway (525€):
  - UPS: 150€
  - USB Switch (10 ports): 25€
  - Network Router (wireguard capable): ~100€
  - Network switch: ~150€
  - 1TB NVME/SSD drive: ~100€
  - An old machine: Free
- Per machine (~100€):
  - Shelly plug: 30€
  - USB 2 RS232 adapter: 20€
  - Ethernet cable: 5€
  - Storage: 30€
  - Your DUT!
- Total: 525 + 5 * 100: ~1k€

# Questions?