a precursor infrastructure of slices RI

YOUPI by citi Lab

from YOUPI - Edge/Fog Computing platform towards SLICES-FR - Computing Continuum

Frédéric Le Mouël, Olivier Nicolas, Pascal Girard, Régis Rousseau, Oscar Carrillo
Univ. Lyon, INSA Lyon, CITI Lab

École d'été SLICES-FR
10/07/2025

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Inria INVENTEURS DU MONDE NUMÉRIQUE

citi Center of Innovation in Telecommunications and Integration of Services lab

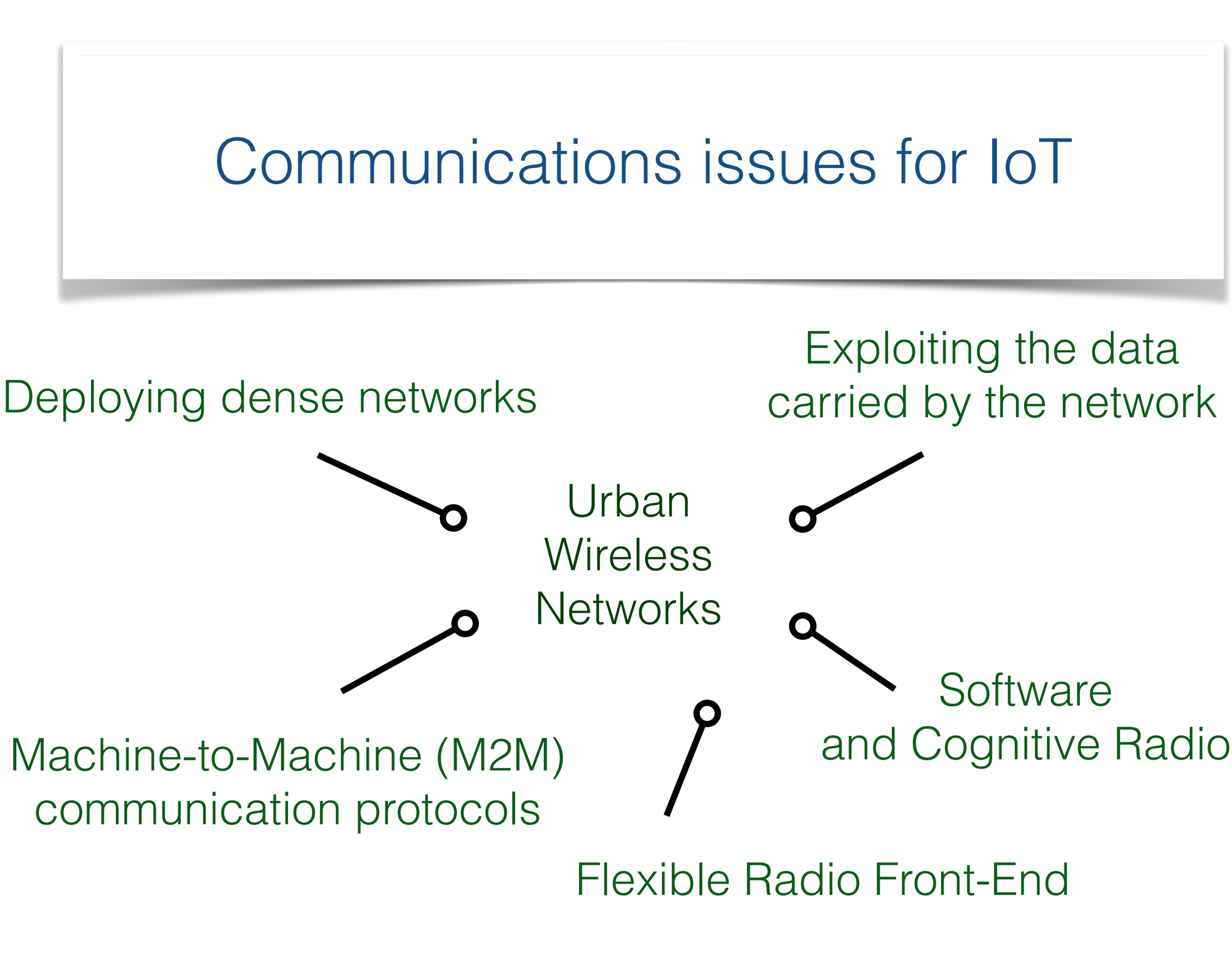
Télécommunications, Services & Usages

TC INSA LYON

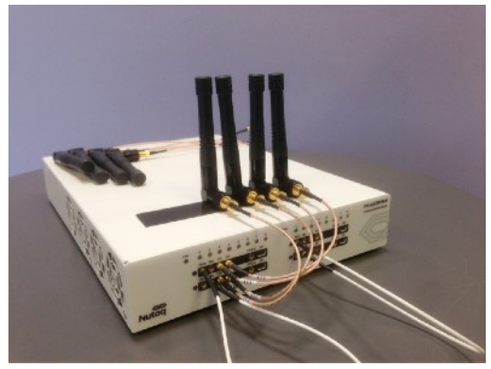
SPIE

Red Hat

Communications issues for IoT



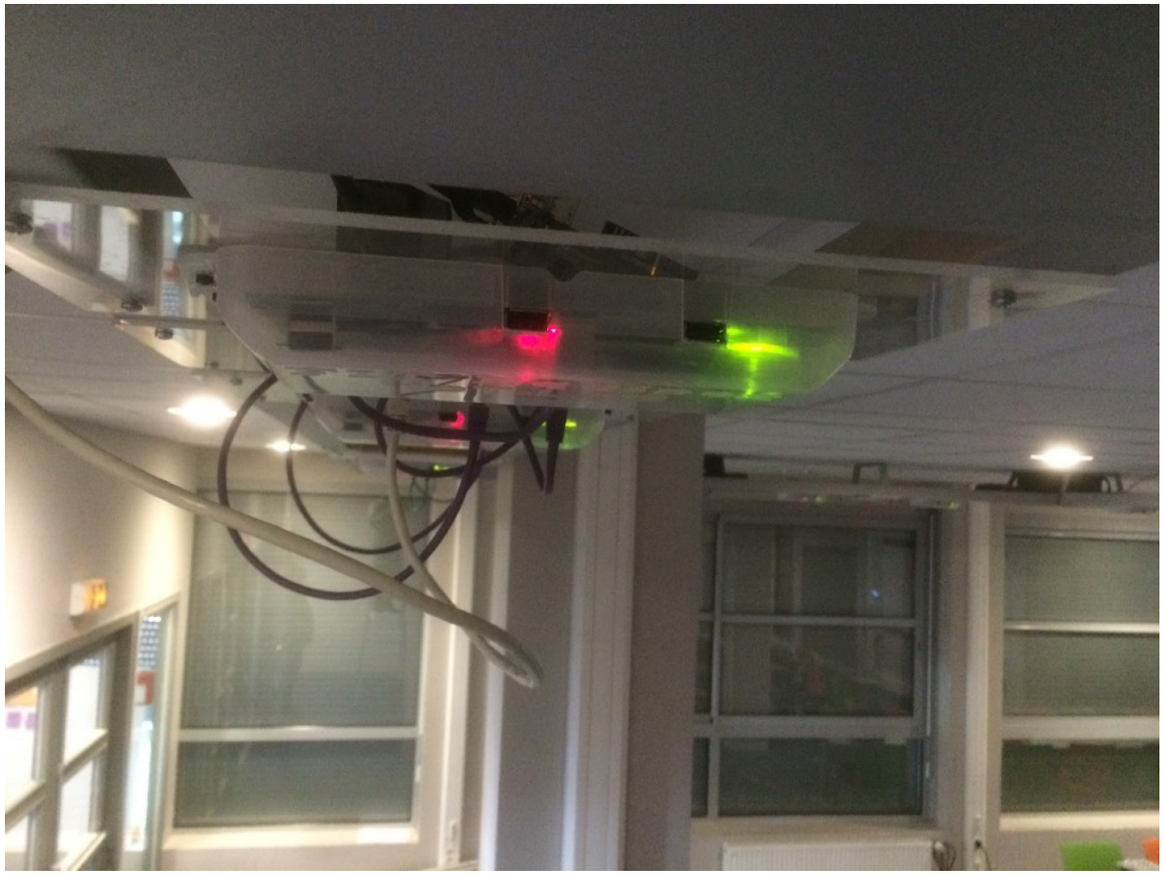
FIT/CorteXLab - http://www.cortexlab.fr



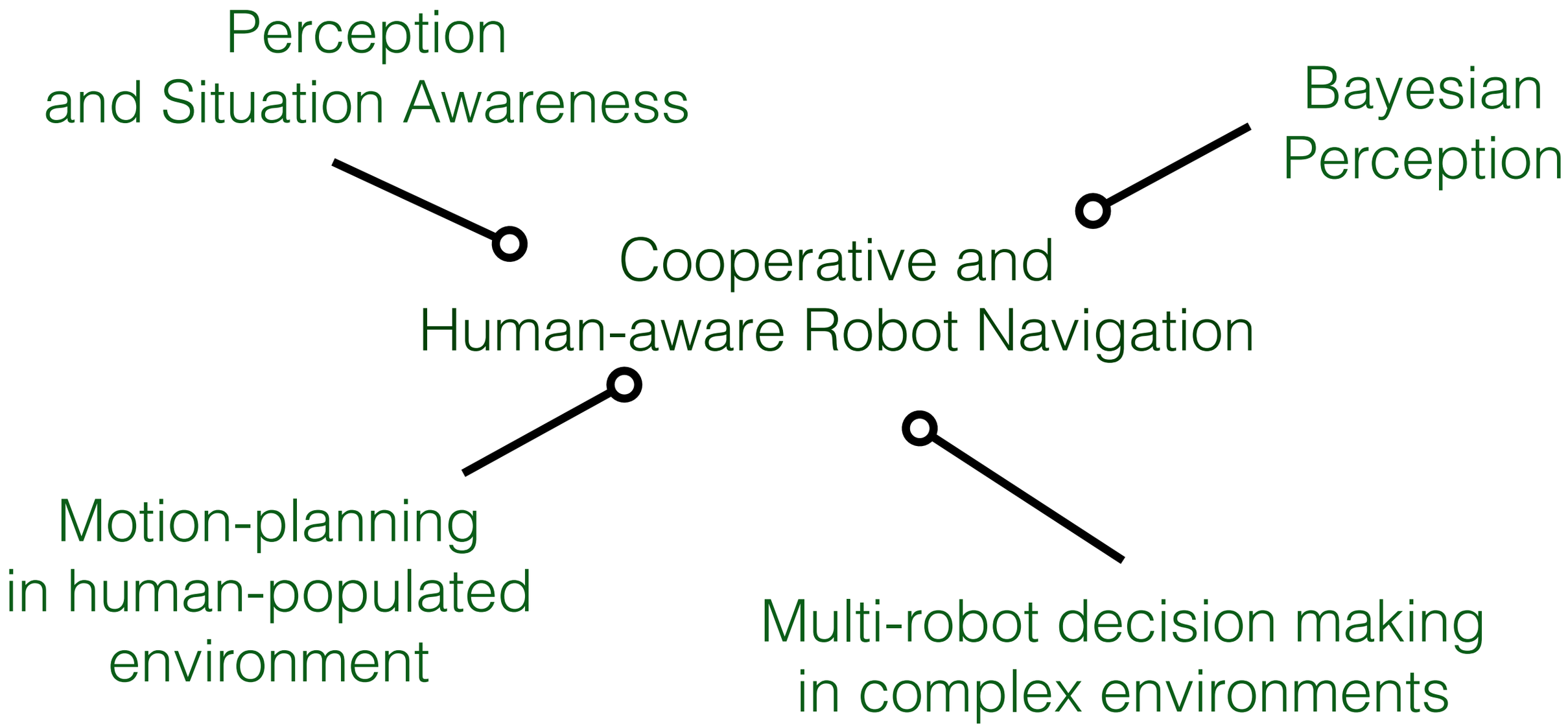
22 USRP + 16 PicoSDR nodes

http://www.cortexlab.fr/?page_id=387

FIT/IoTLab - https://www.iot-lab.info/



Cooperation / Mobility issues for Robots

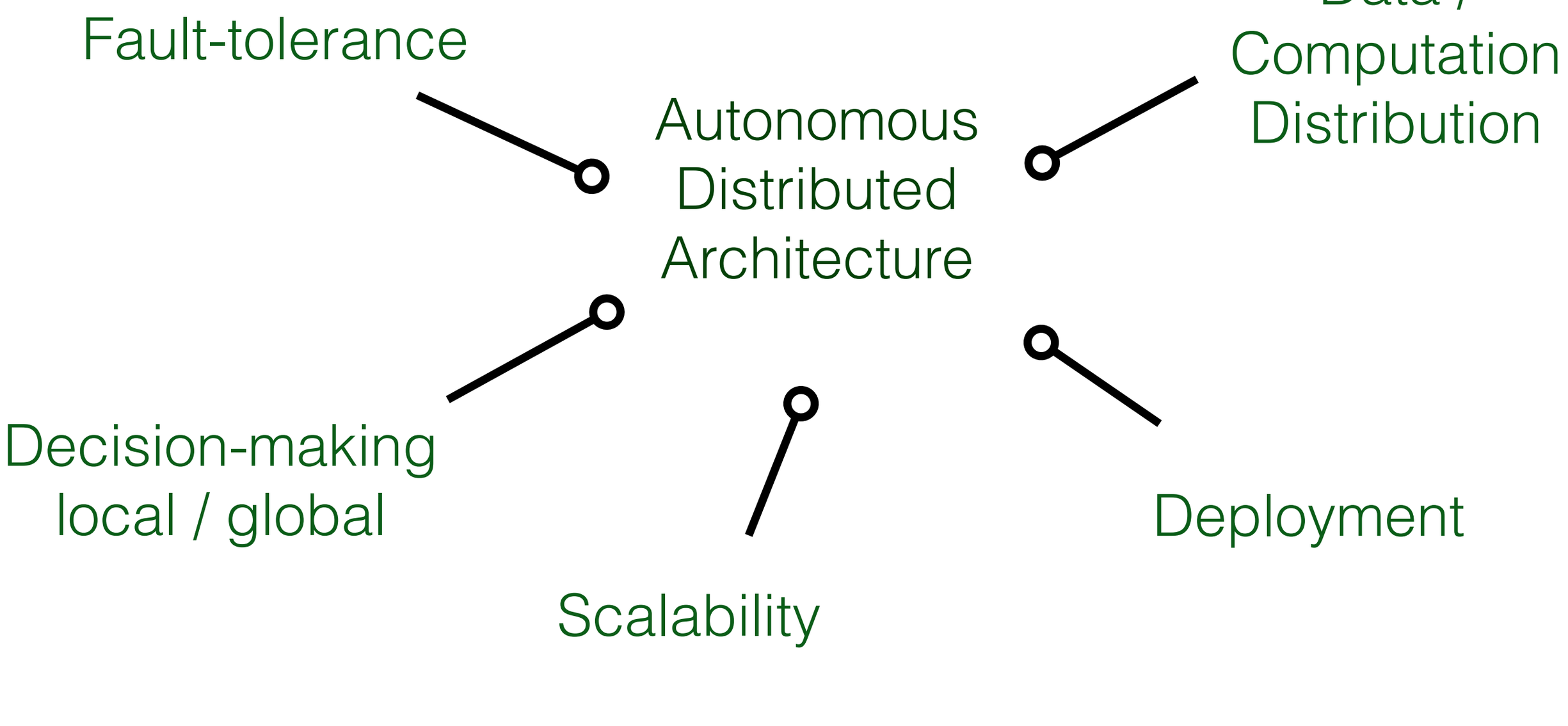


Télécommunications,
Services & Usages

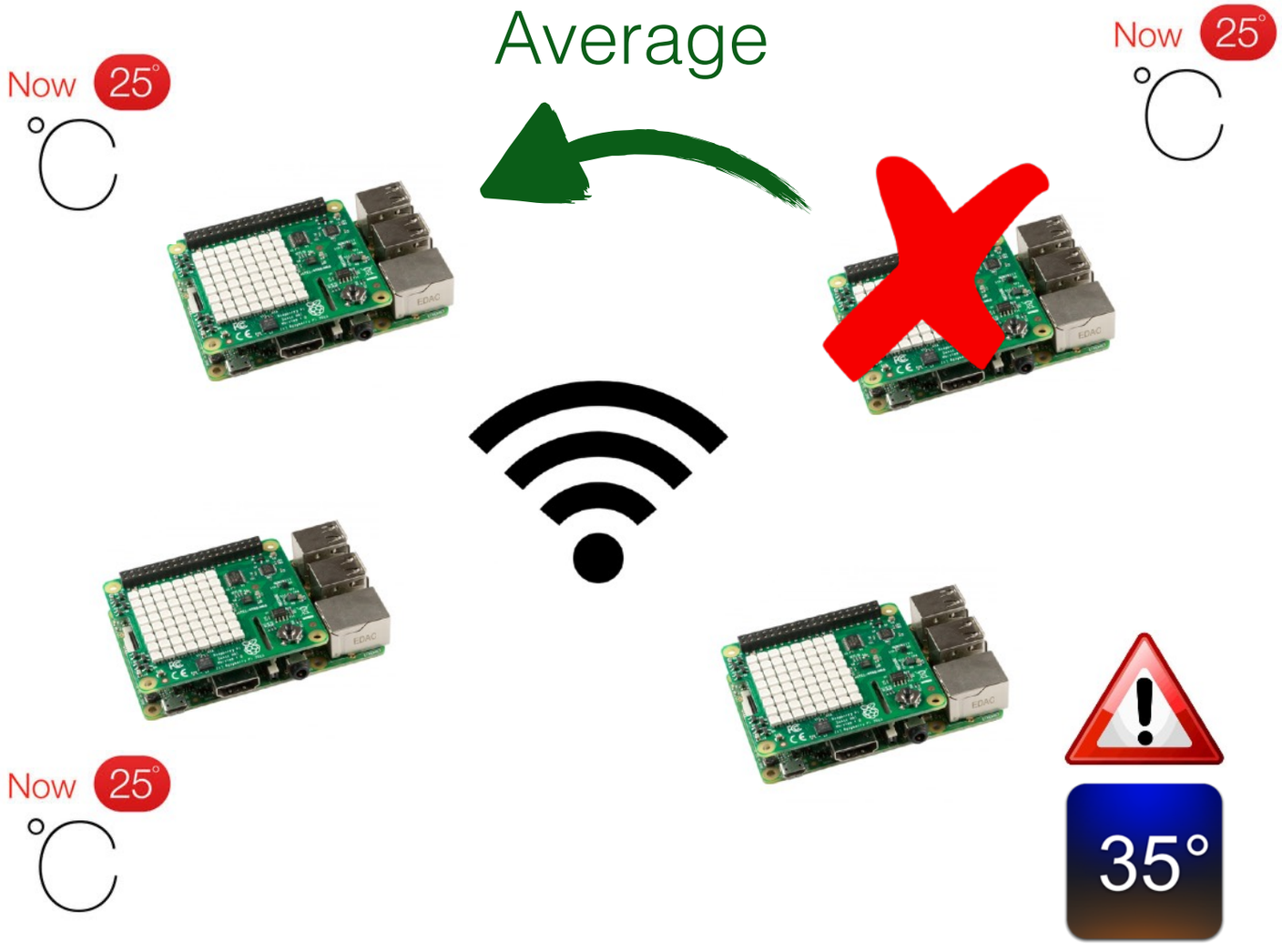
INSA LYON

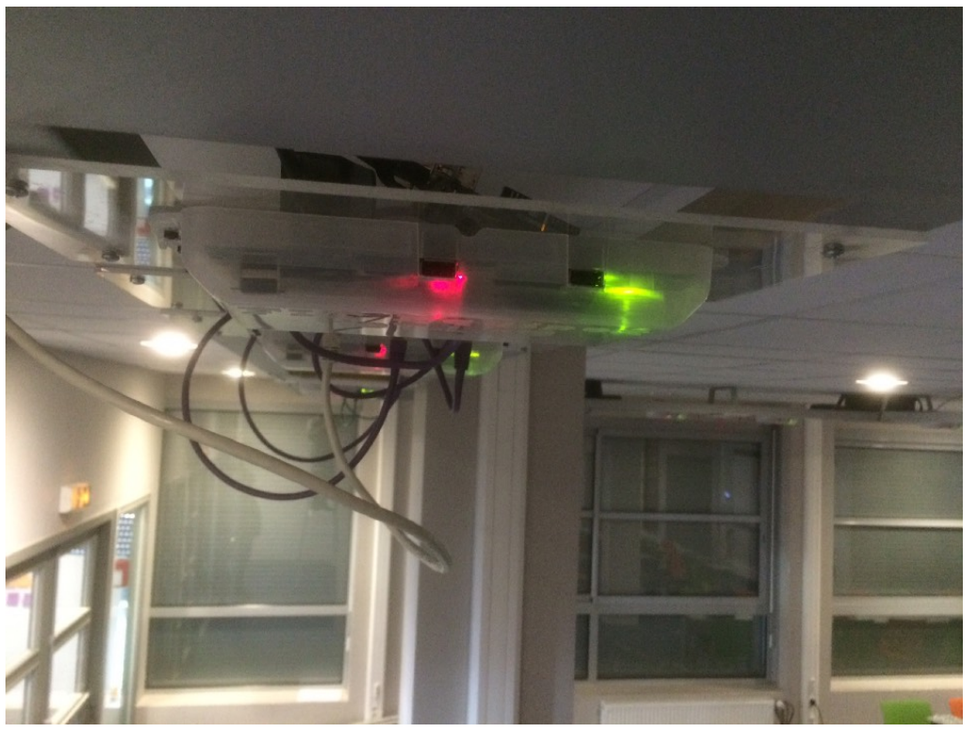
12 TurtleBot 2
Drone Parrot Bibop

Software / middleware issues for IoT



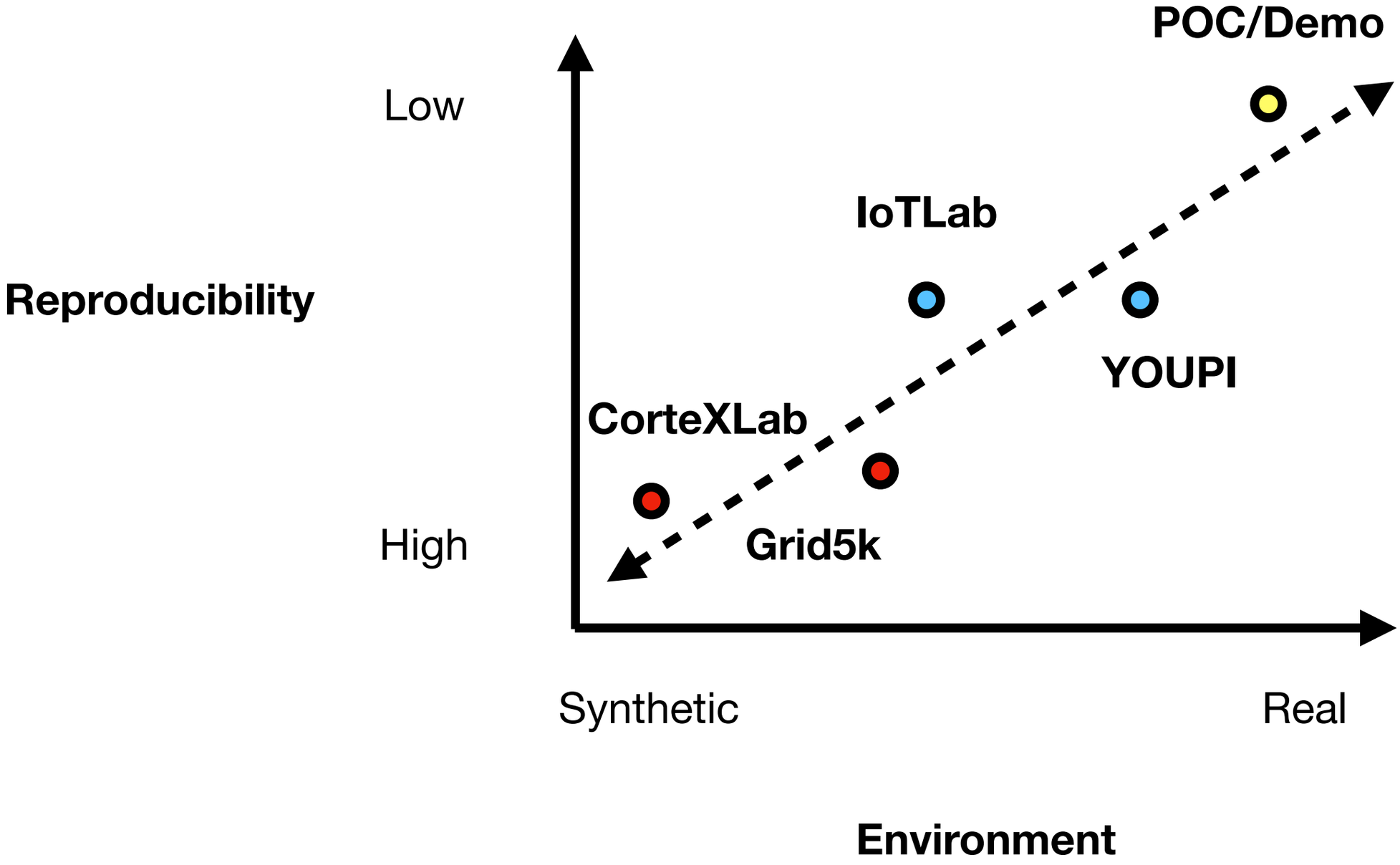
YOUPI - Edge/Fog Computing



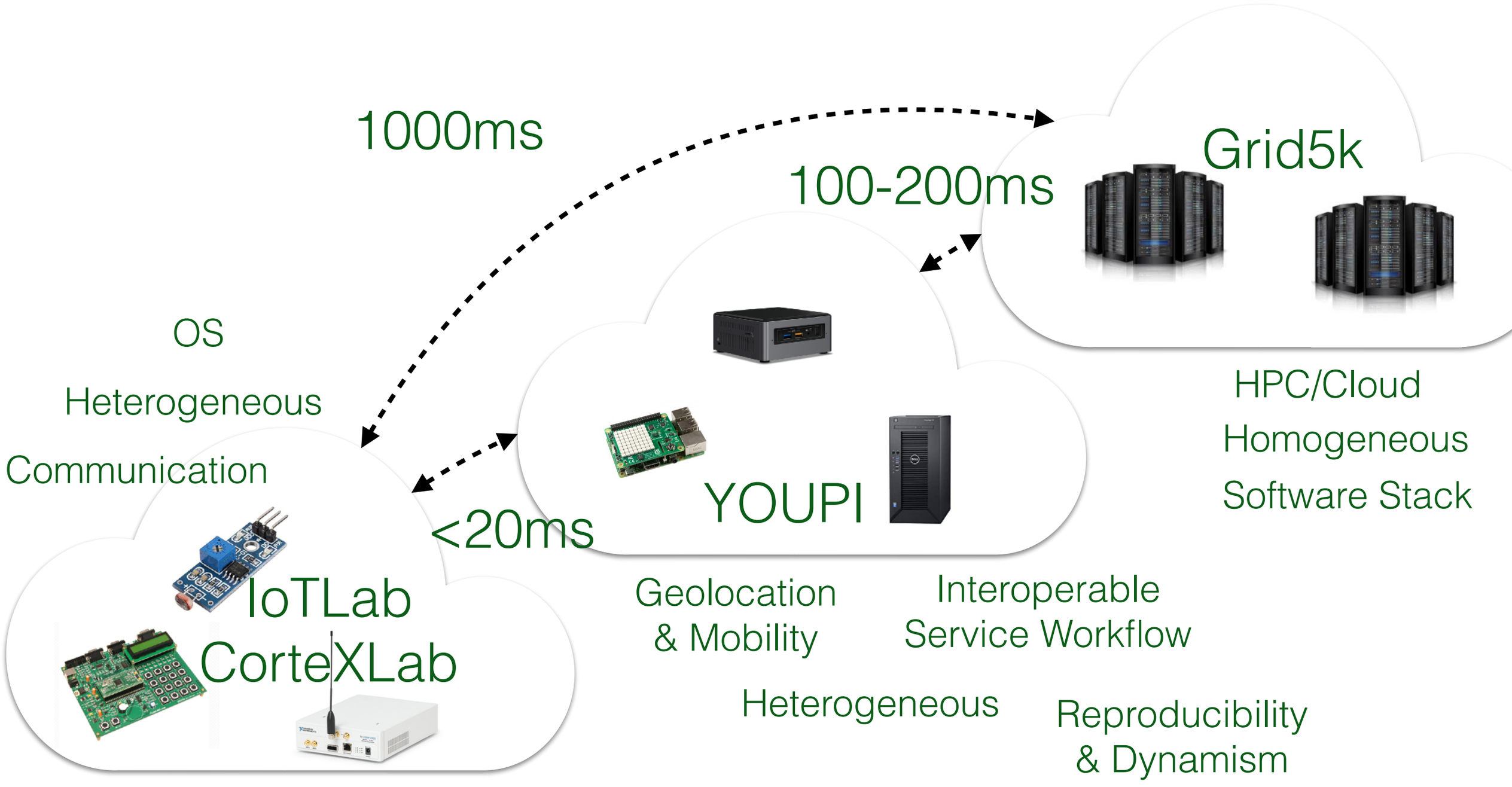


25+ RPI
3 NUC
4 Nvidia

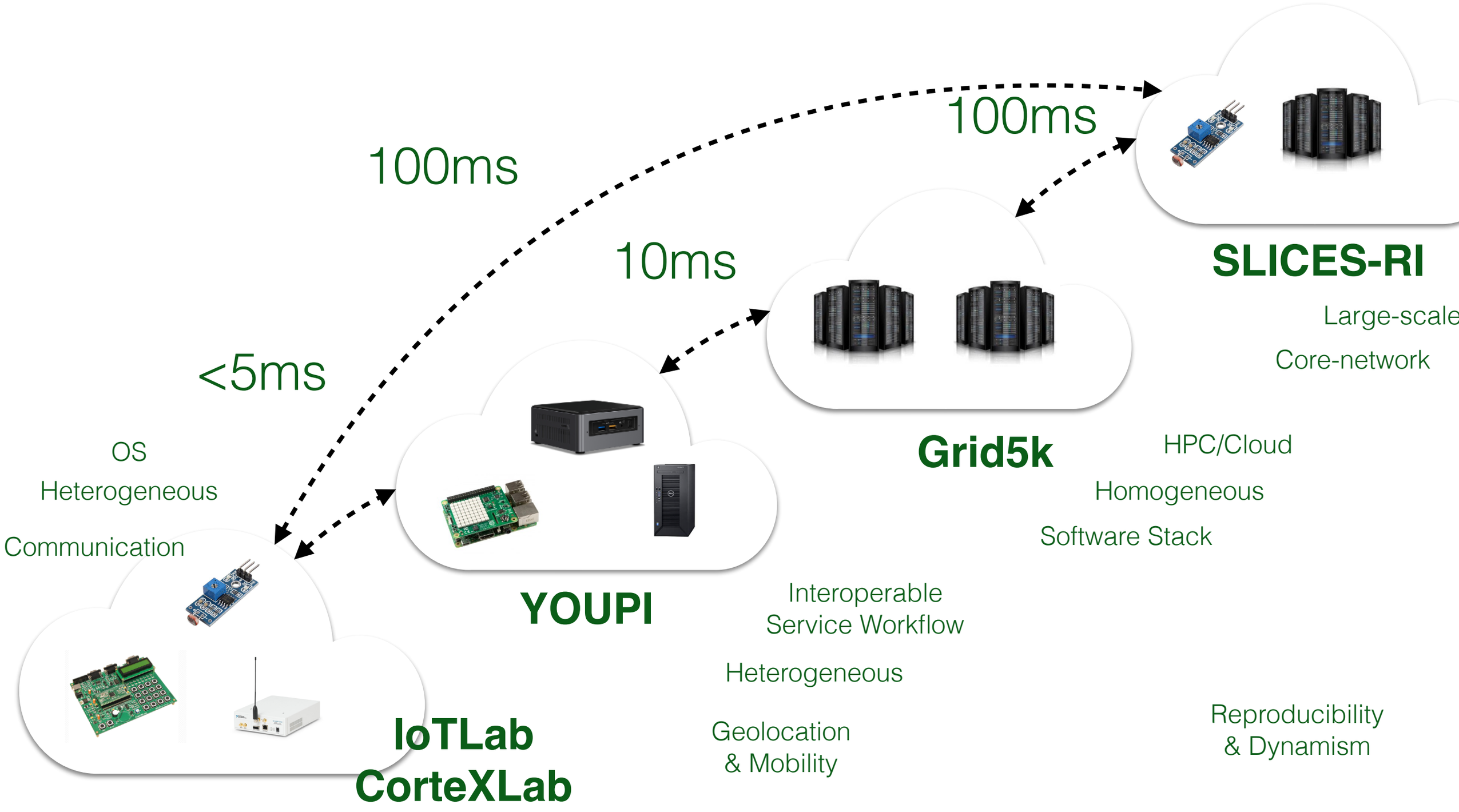
Reproducibility ?



YOUPI vs FIT(IoTLab/ CortexLab) vs Grid5000 ?



Towards SLICES-FR



Goals

- YOUPI: Yet Another *PI platform
    - Off-the-shell hardware and software
    - Capturing dynamic properties of the environment - not reproducible, but traceable - wireless connections/ interferences, mobility, user behaviour
        - Replay/analyse offline

Heterogeneous Hardware

- Microcontrollers — Sensing — IoT SASTRESS Node – 4-parts PCB:
  - **Seismic sensor** (omnidirectional geophone) + **amplification and filtering board** (band-pass 10 Hz – 215 Hz, max gain +58 dB at 40 Hz) + buffer stage to MCU ADC
  - **MCU**: MDBT50Q-1MV2 module based on nRF52840 + 62 Mbit EEPROM + DS3231 RTC + SHT31 temperature/humidity sensor
  - **LoRa module**: RFM95W + u.fl connector
  - **Power supply**: 3.7 V – 6000 mAh Li-Ion polymer battery + 80x100 mm 1 W solar panel + power management (MCP73831 + RT9078)

Heterogeneous Hardware

- Microcontrollers — Sensing/Computing — IoT TRALALA Node:
  - **CPU**: Raspberry Pi Zero board, Broadcom BCM2835 (ARM11, 1 GHz, monocore), RAM : 512 MB
  - **LoRa module**: Adafruit LoRa Radio Bonnet with OLED – RFM95W
  - **Solar panel**: 6 V – 9 W
  - **Battery**: 3.7 V – 4400 mAh
  - **Power management**: Solar Pi Platter board (available on request via Tindie / product link)

Heterogeneous Hardware

- 16 RPI4 + 15 RPI3 + 10 RPI2 (+ 2 BananaPI)
    - RPI4: **ARM** Quad Core **1.5GHz** Broadcom BCM2711 64bit CPU, **2-8GB RAM**
    - RPI3: **ARM** Quad Core **1.2GHz** Broadcom BCM2837 64bit CPU, **1GB RAM**
    - RPI2: **ARM** Quad Core **900MHz** Broadcom BCM2836 32bit CPU, **1GB RAM**

Heterogeneous Hardware

- 2 TPUs

  - Tinker Edge T : NXP i.MX 8M Quad-core **ARM** Cortex-A53 **1,5GHz, 1G RAM, Google Edge TPU** GC7000 Lite

Heterogeneous Hardware

- 3 NUCs
  - NUC8I7HVK: **Intel** Kaby Lake G Core i7-8809G **3.1 GHz Quad-Core 8 thread** / GPU Radeon RX Vega M GH Graphics 4 Go HBM2 (24 Compute Units, **1063 MHz**) / **32 Go** DDR4-SODIMM **2400 MHz**

  - NUC7I3BNH: **Intel** Core i3-7100U Dual-Core **2.4 GHz - 4 threads** / GPU Intel HD Graphics 620 (24 Compute Units, **300 MHz**) / **16 Go** DDR3-SODIMM **2133 MHz**

Heterogeneous Hardware

- 4 Nvidias
  - Jetson TX1: **256-core** NVIDIA Maxwell™ GPU **998 MHz**
  - 3 Nano: **128-core** NVIDIA Maxwell™ GPU **921 MHz**

Heterogeneous Wireless Communication Interfaces

- Bluetooth Low Energy
- 6LoWPAN
- 802.11b/g/n/ac
- LoRaWAN

Short

Mid

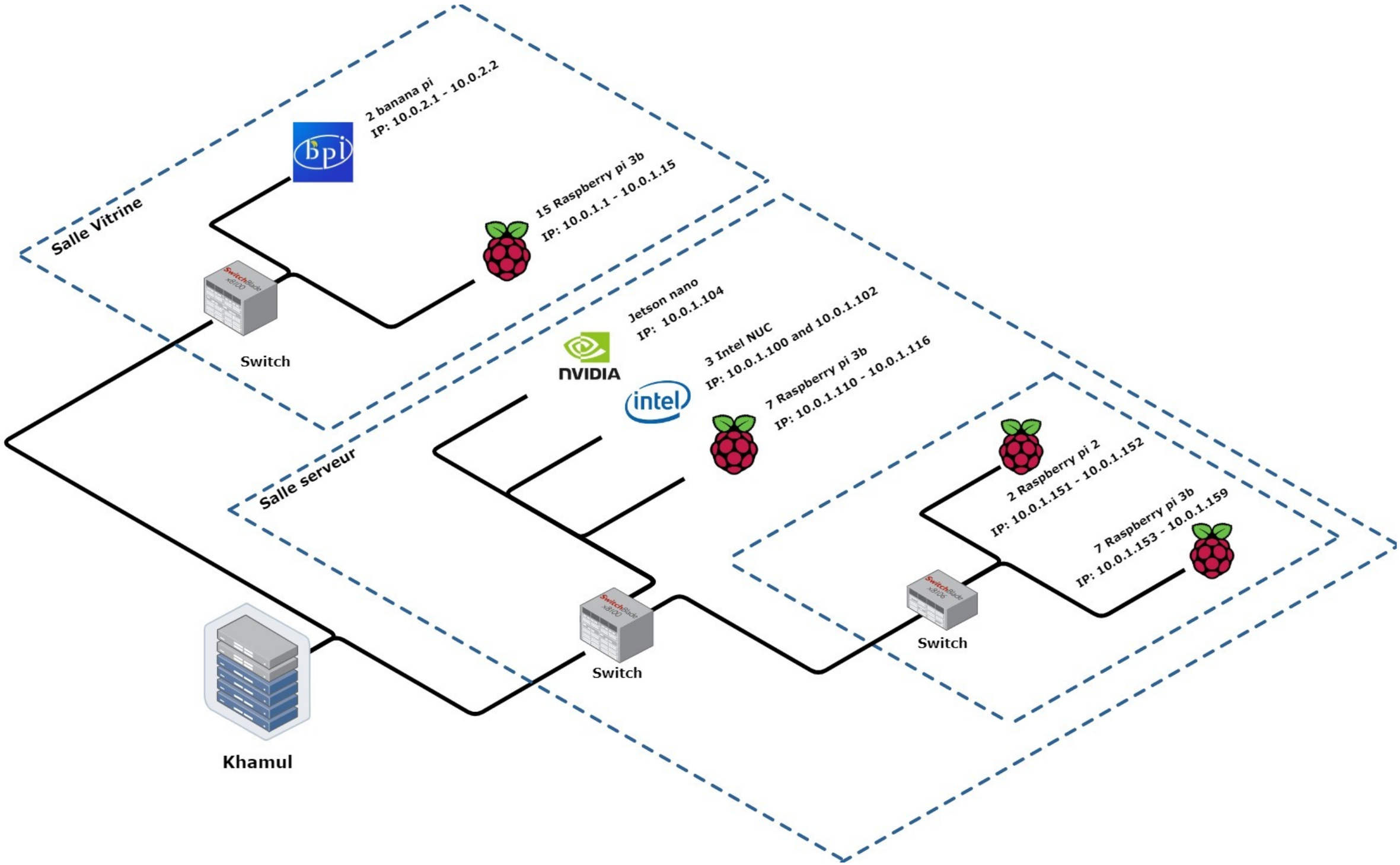
Range

Long

Energy Management

- Power-Over-Ethernet Switch / POE splitter
- Off-the-shell electric plugs: high failure probability
- Energy monitoring, switch on/off nodes

Network Topology



Software Management

- Image Server
  - Boot / NFS mount / reflash node / specific configuration
- Off-the-shell containers
  - Docker
  - Kubernetes

Node management / flash

YOUPI reflash system - Implementation

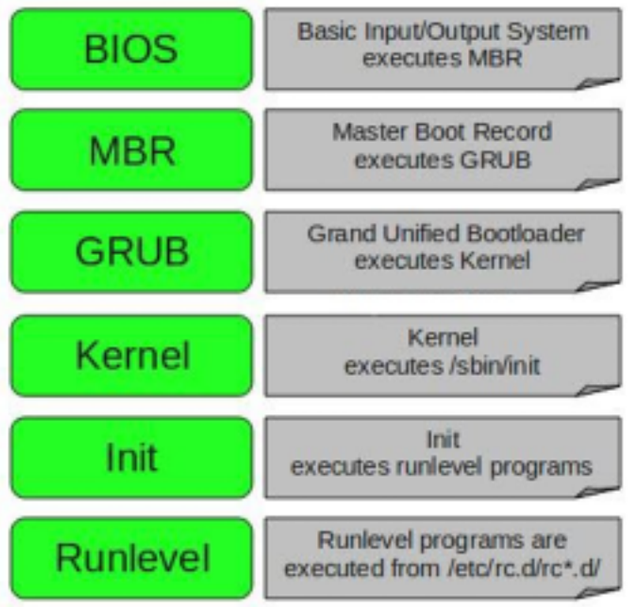
The Flashing Process explained

The Linux Boot Process

Note: the terms initrd and initramfs are synonymous/interchangeable.
initramfs stands for INITial RAM File System.

In a nutshell:
When using initrd, the system typically boots as follows:

1. the boot loader loads the kernel and the initial RAM disk
2. The kernel converts initrd into a “normal” RAM disk and Frees the memory used by initrd
3. root device is mounted…
4. init mounts the “real” root file system
5. init places the root file system at the root directory
6. init execs the /sbin/init on the new root filesystem, performing the usual boot sequence
7. the initrd file system is removed



Node management / flash

The Flashing Process itself

During this whole process, we actually "hijack" the init process to perform the following duties:

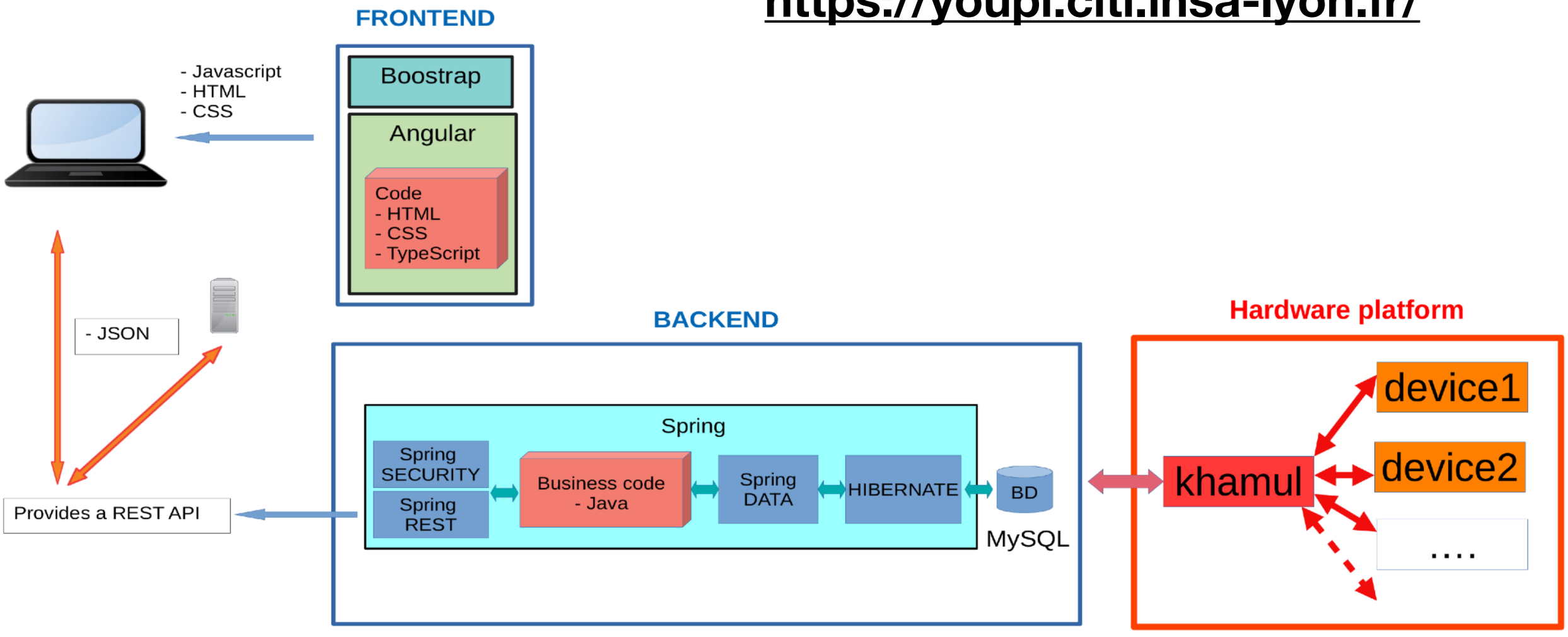
1. Get an IP address via DHCP
2. Mount the nfs directory for the device type we want to flash
3. Execute our flashing script, using the image on the nfs mounted partition
4. Reboot once dd has finished its job and we have have resized the drive to full capacity, bypassing the last 2 steps of the process described previously

Architecture - Example for Raspberry-3B device type

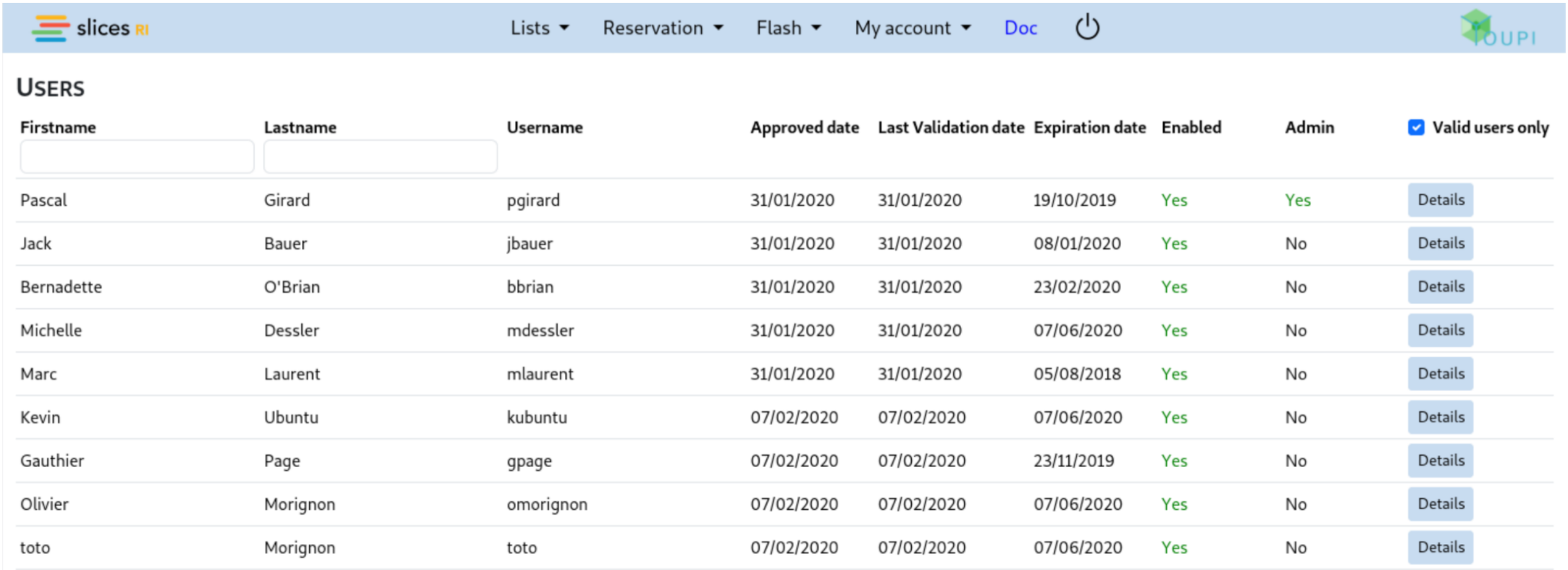
```
reflash
├── devices
│   └── raspberry-3B
│       ├── b8-27-eb-57-7b-e7
│       │   ├── init-script.sh -> ../scripts/flash-script.sh
│       │   └── reflash.img -> ../../../images/raspberry-3B/raspbian-jessy.img
│       ├── b8-27-eb-6c-8f-6a
│       │   ├── init-script.sh -> ../scripts/flash-script.sh
│       │   └── reflash.img -> ../../../images/raspberry-3B/raspbian-buster.img
│       └── scripts
│           └── flash-script.sh
├── images
│   └── raspberry-3B
│       ├── raspbian-buster.img
│       └── raspbian-jessy.img
├── initramfs
│   ├── devices
│   │   └── raspberry-3B
│   │       ├── boot
│   │       │   ├── cmdline.txt
│   │       │   ├── config.txt
│   │       │   └── initramfs -> ../initramfs.d/initramfs
│   │       └── initramfs.d
│   │           ├── additions
│   │           │   ├── etc
│   │           │   │   └── ld.so.conf.d
│   │           │   │       └── 00-mylibs.conf
│   │           │   ├── newbins
│   │           │   ├── newlibs
│   │           │   ├── config
│   │           │   ├── init
│   │           │   ├── init.sh
│   │           │   └── udhcp-script.sh
│   │           └── initramfs
│   └── utils
├── flash-device.sh
└── target-script.sh
```

Front-end admin

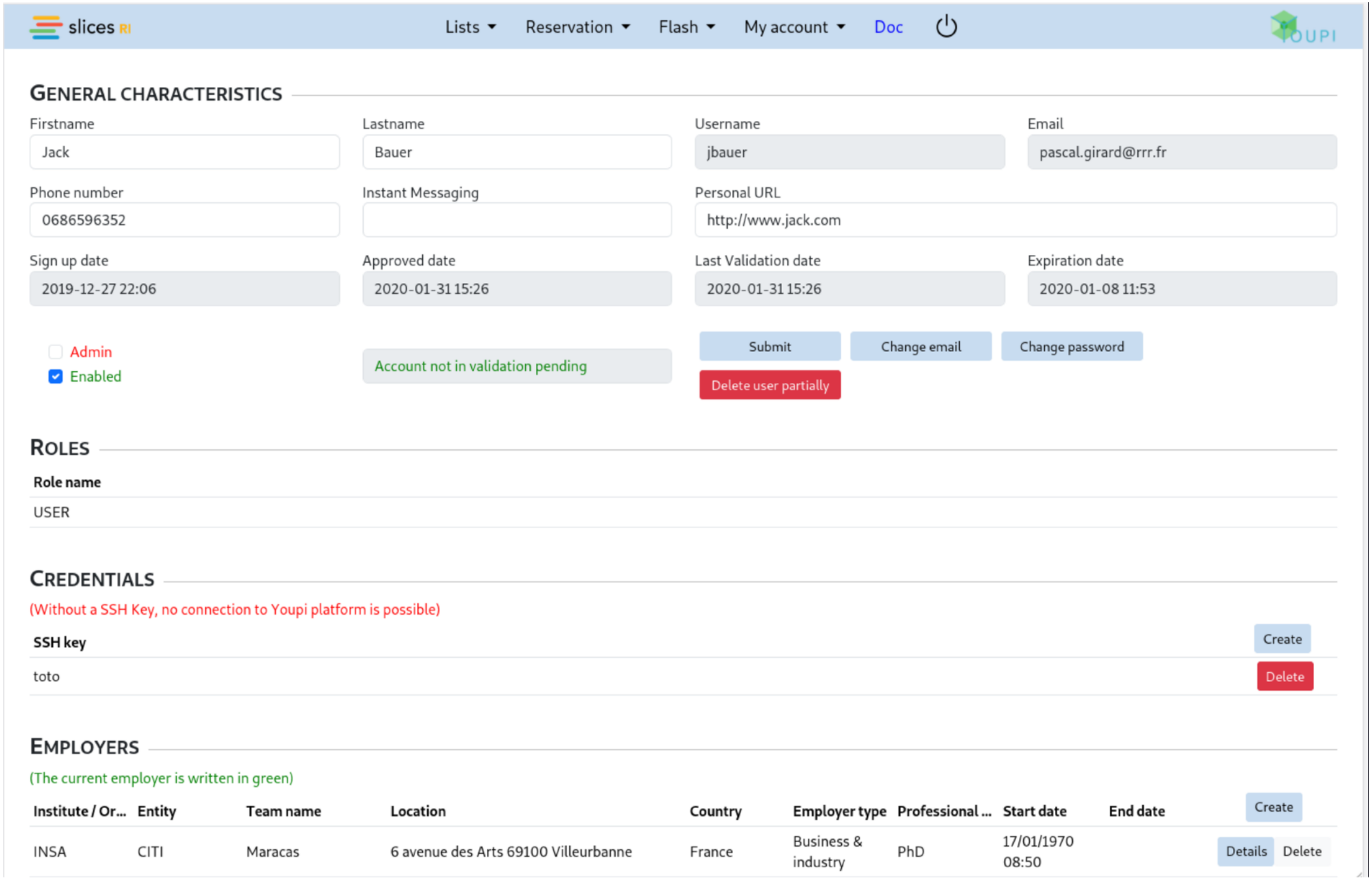
https://youpi.citi.insa-lyon.fr/



User management



User management



Booking management

slices RI | Lists | Reservation | Flash | My account | Doc | OUPI

MY RESERVATIONS

First Prev Next Last | 5 users per page | Current / future

Reservation nu...	Responsible	Users involved	Submission date	Start date	Duration	End date	Reservation de...	Devices	Type
165	Pascal Girard		2024-12-11 11:07	2024-12-12 13:07	01J00H00	2024-12-13 13:07	resa 4 pgirard	10.0.1.102 / 10.0.1.103	Shared
155	Pascal Girard	Jack Bauer, Kevin David	2024-06-03 14:30	2024-06-04 15:30	183J00H12	2024-12-04 14:42	resa 1 pgirard	10.0.1.31 / 10.0.1.32	Shared
160	Pascal Girard		2024-05-13 17:00	2024-05-18 19:00	205J17H04	2024-12-10 11:05	resa 3 pgirard	10.0.1.51 / 10.0.1.52 / 10.0.1.53 / 10.0.1.54	Exclusive
158	Pascal Girard		2024-05-01 15:00	2024-05-02 15:00	219J20H00	2024-12-08 10:00	resa 2 pgirard	10.0.1.21	Exclusive
168	Pascal Girard	Jack Bauer, Kevin David	2023-06-17 18:29	2023-06-23 19:29	01H02	2023-06-23 20:31	resa 5 pgirard	10.0.1.31 / 10.0.1.32	Shared

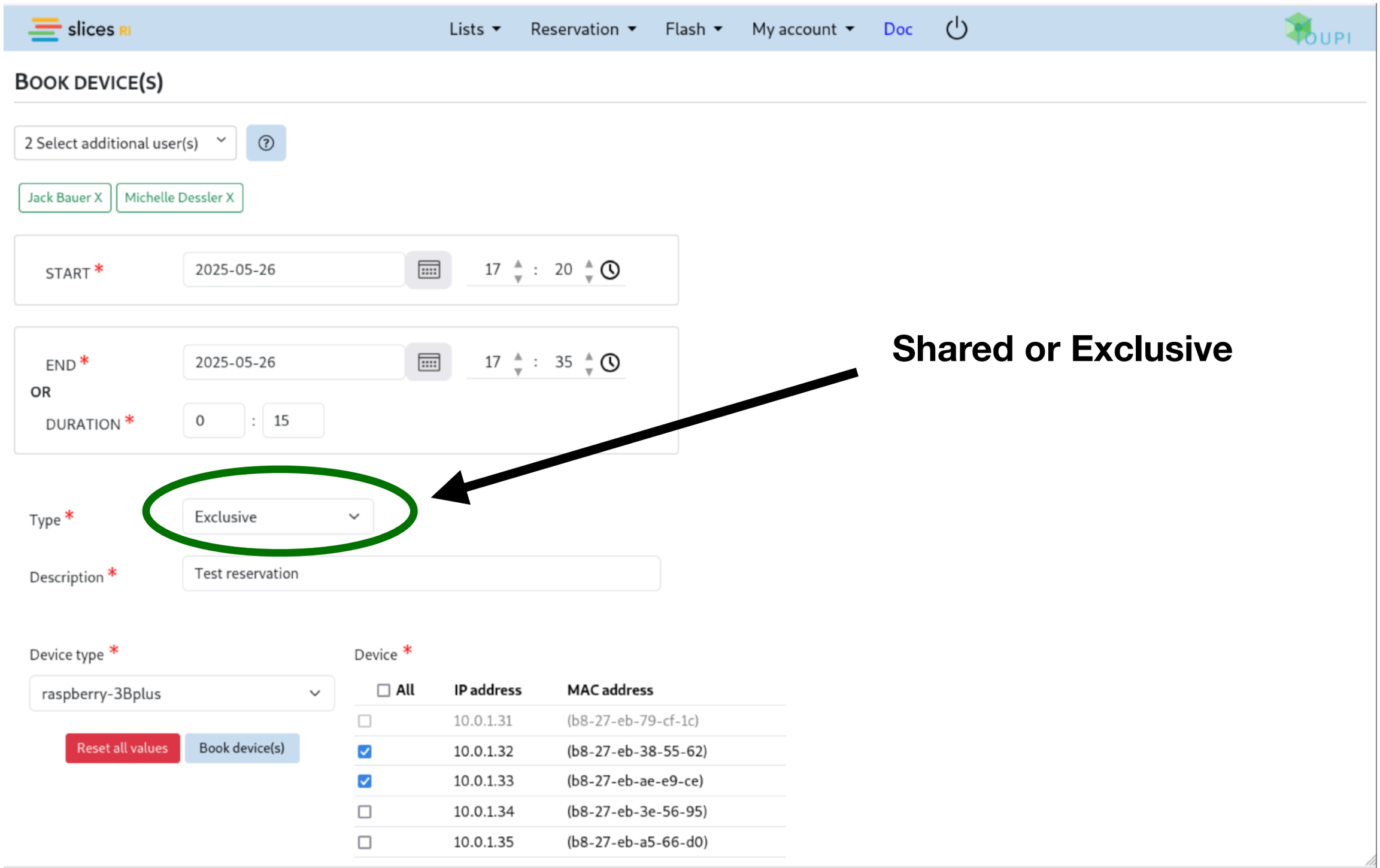
Book device(s)

OTHER RESERVATION(S)

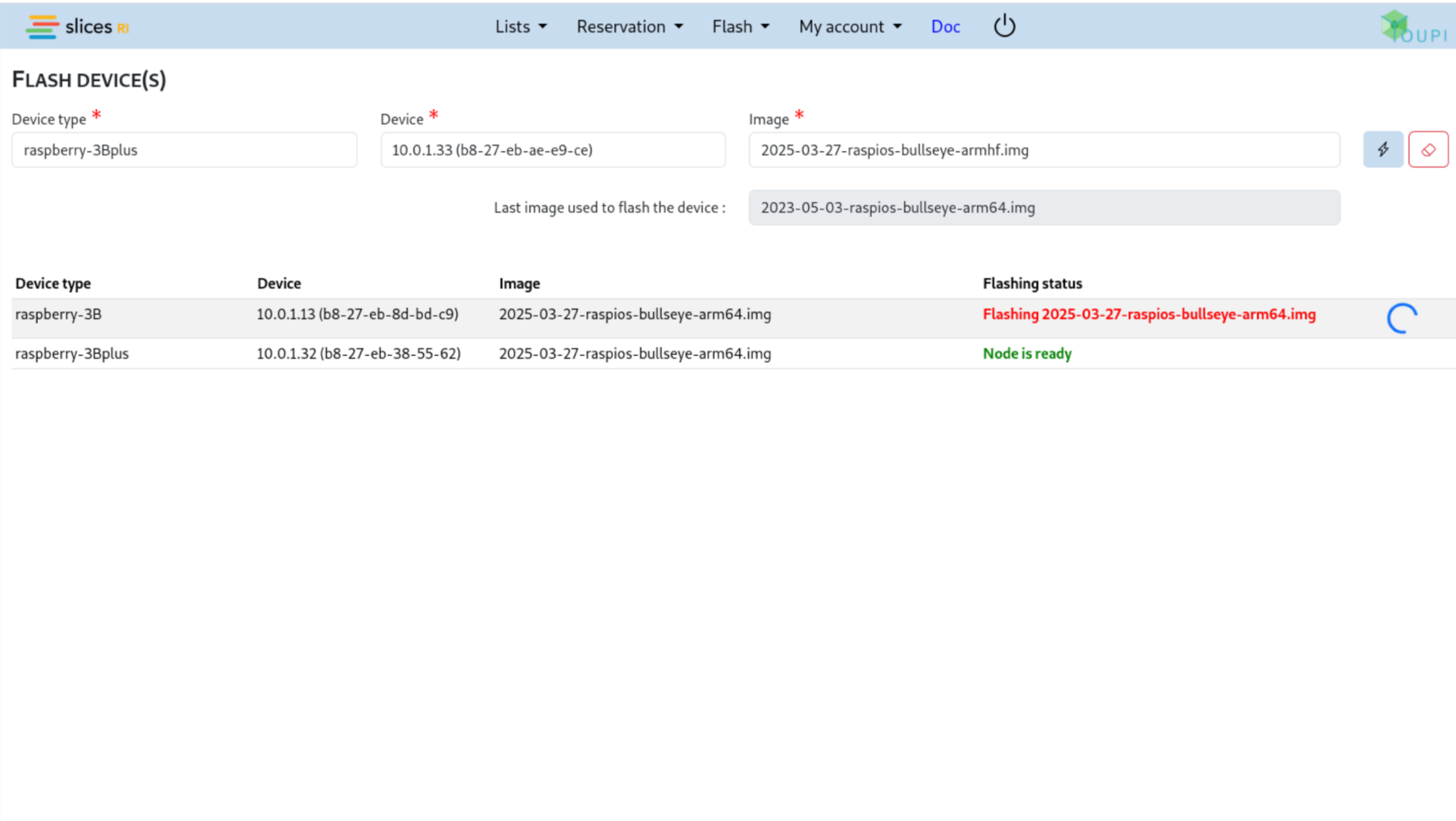
First Prev 1 Next Last | 5 users per page

Reservation nu...	Responsible	Users involved	Submission date	Start date	Duration	End date	Reservation de...	Devices	Type
177	Jack Bauer		2024-12-20 16:09	2024-12-22 18:12	03J00H00	2024-12-25 18:12	resa 3 jbauer	10.0.1.107 / 10.0.1.108	Exclusive
171	Jack Bauer		2024-12-13 15:07	2024-12-14 15:07	02J00H00	2024-12-16 15:07	resa 1 jbauer	10.0.1.31 / 10.0.1.32	Exclusive
174	Jack Bauer	Pascal Girard, Michelle Dessler, Marc Laurent, André Martin	2024-06-01 19:00	2024-06-02 20:00	200J00H08	2024-12-19 19:09	resa 2 jbauer	10.0.1.105 / 10.0.1.106	Exclusive

Booking management



Node management / flash



Future work - your needs ?

- Software integration
    - Booking tools / Resource allocation / Service Orchestration - different SLICES-FR hubs, APIs ?
        - OAR / kadeploy / GUIX contribution ?
- Scenario in 5G / IoT
    - SLICES-FR blueprint - computing/storing services handoff in multi-proximity clouds

Future work - your needs ?

- Off-the-shell electronic components: significant failure probability (electric plug, SD cards), normal behaviour - highly-dynamic local fault-tolerance at the Edge? vs Energy?
    - Energy Management
        - Manageable PDU
        - Manageable Switches -> front-end integration
- Dynamic Network Topologies
    - P4 switches: Tofino Wedge
    - Efficient cross-domain network management? IPv6 local interconnection
    - VXLAN, tunnelling SSH: encapsulation, not acceptable delay for Edge Computing, level 3 SLICES-RI connection

Future work - your needs ?

- Off-the-shell electronic components: significant failure probability (electric plug, SD cards), normal behaviour - highly-dynamic local fault-tolerance at the Edge? vs Energy?
- HardwareAsAService - in collaboration with WALTonTheRoof
  - Comparison specific IoT design vs Generic nodes
  - Energy footprint
  - On demand wake up

Future work - your needs ?

- Off-the-shell electronic components: significant failure probability (electric plug, SD cards), normal behaviour - highly-dynamic local fault-tolerance at the Edge? vs Energy?
    - Quantum Computer / Quantum switches at the border
        - https://www.spinquanta.com/products/superconducting-quantum-products
        - https://meetiqm.com/products/iqm-spark/
        - https://www.pasqal.com/fr/solutions/hardware/
        - https://www.quandela.com/products-and-services/mosaiq/
        - https://eurohpc-ju.europa.eu/eurohpc-quantum-computers/our-quantum-computers_en

        - https://qutech.nl/business-innovation/academic-collaborations/
        - https://quantuminternetalliance.org/products-and-platforms/

Discussions ?