WalT: A Reproducible Research Platform for Networks and Distributed Data

Etienne Dublé (LIG / CNRS), Franck Rousseau (LIG / Grenoble INP)
and Drakkar team of LIG lab

Grenoble, February 2021

WalT for real...

This is a typical **wireless** network experiment! ;)

Agenda

Motivation

Sample use case

Architecture

WalT under the hood

News, On-going work and Perspectives

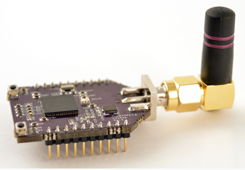
Existing platforms

Large scale testbeds (e.g. IoT-Lab)

- Limited hardware choice
- Static environment & network topology
- No physical access (for debugging)
- Booking system sometimes crowded

Handmade platform “on your own desk”

- Does not scale
- Is hardly reproducible



WalT platform objectives

WalT stands halfway between those 2 categories.

Just as a handmade platform,

- WalT can be lightweight (backpack-compatible, for a demo)
- You can build several instances of WalT
- You can set up one on your desk (for debugging)
- You can easily move nodes (CPS, wireless networks)

Just as a large-scale platform,

- You can make it bigger (you can deploy it in a building)
- It is easy to use

WalT project objectives

We provide:

- Open source code
- Installation images
- Documentation (how to install and use)
- Support (walt-users@univ-grenoble-alpes.fr)

We do not provide:

- A publicly available platform (you should build your own)

Agenda

Motivation

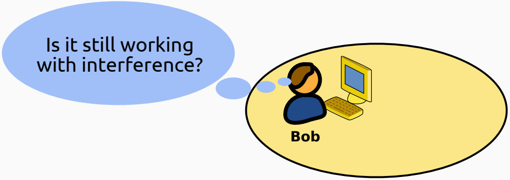
Sample use case

Architecture

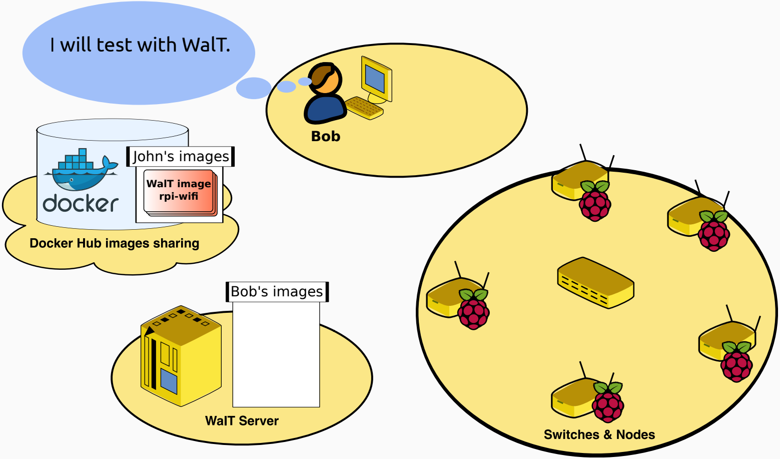
WalT under the hood

News, On-going work and Perspectives

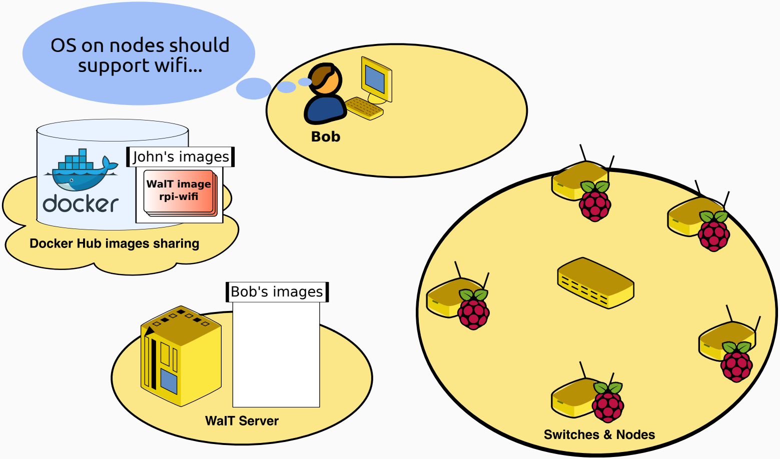
Sample use case



Sample use case



Sample use case



OS on nodes should support wifi...

Bob

John's images

WalT image rpi-wifi

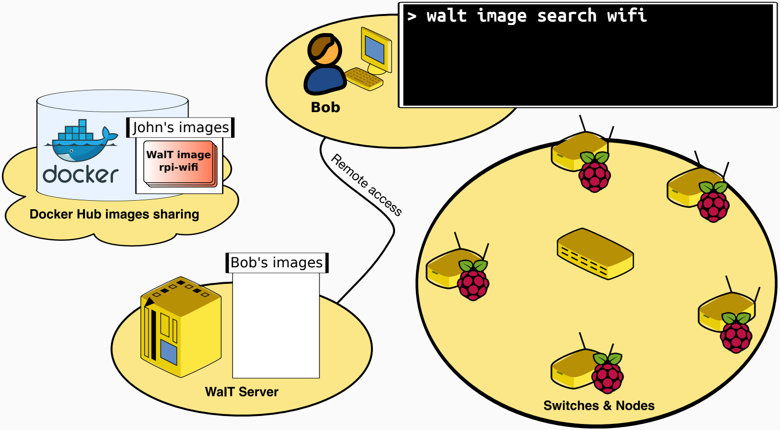
Docker Hub images sharing

Bob's images

WalT Server

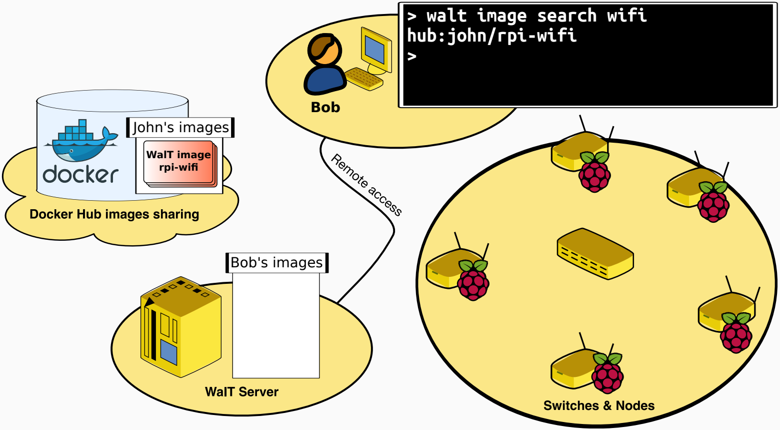
Switches & Nodes

Sample use case



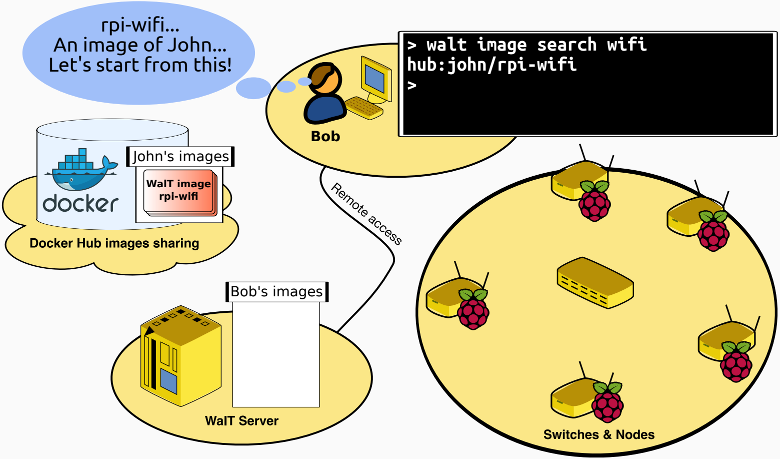
```
> walt image search wifi
```

Sample use case



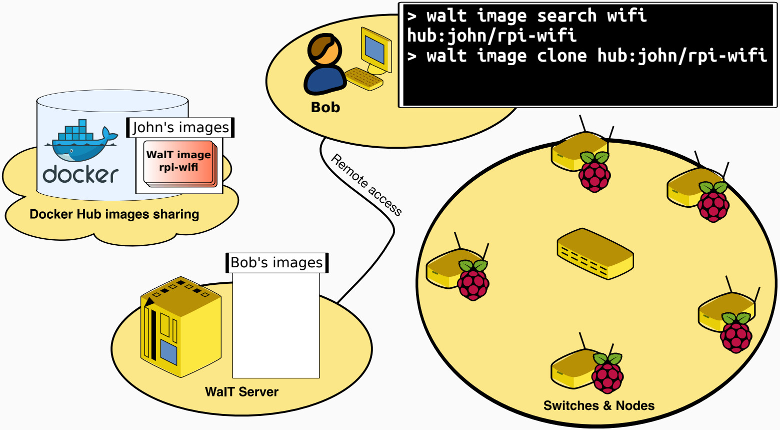
```
> walt image search wifi
hub:john/rpi-wifi
>
```

Sample use case



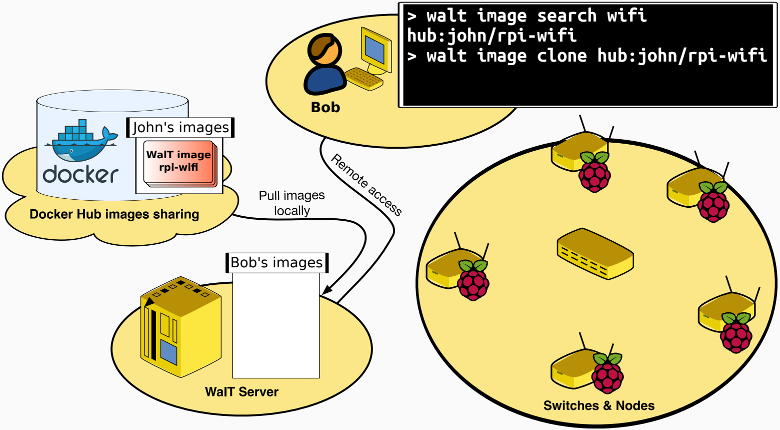
```
> walt image search wifi
hub:john/rpi-wifi
>
```

Sample use case



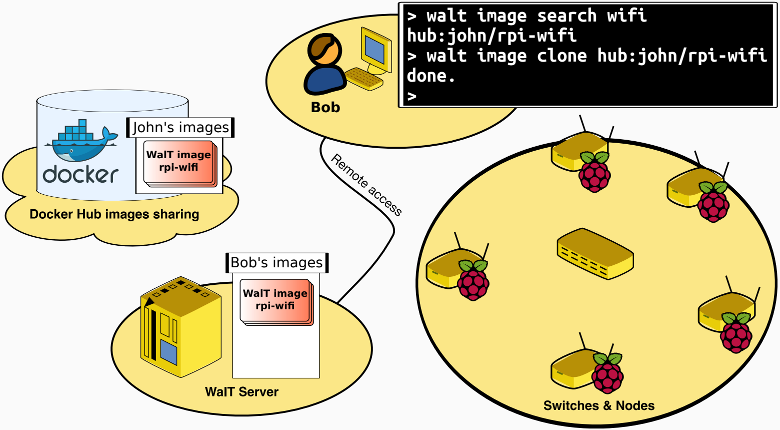
```
> walt image search wifi
hub:john/rpi-wifi
> walt image clone hub:john/rpi-wifi
```

Sample use case



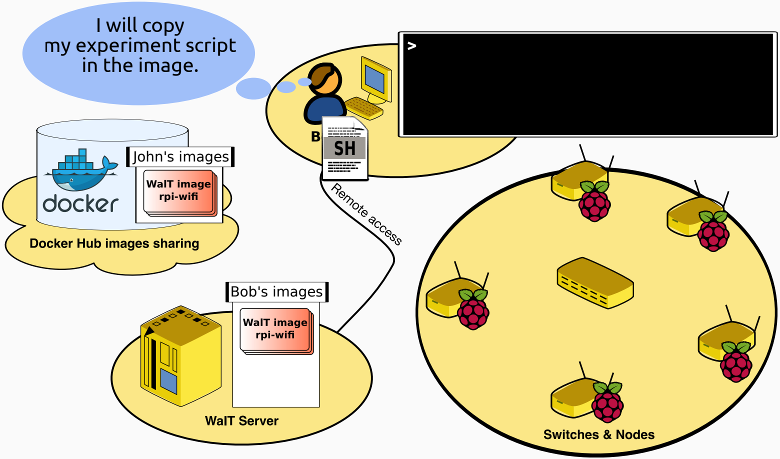
```
> walt image search wifi
hub:john/rpi-wifi
> walt image clone hub:john/rpi-wifi
```

Sample use case

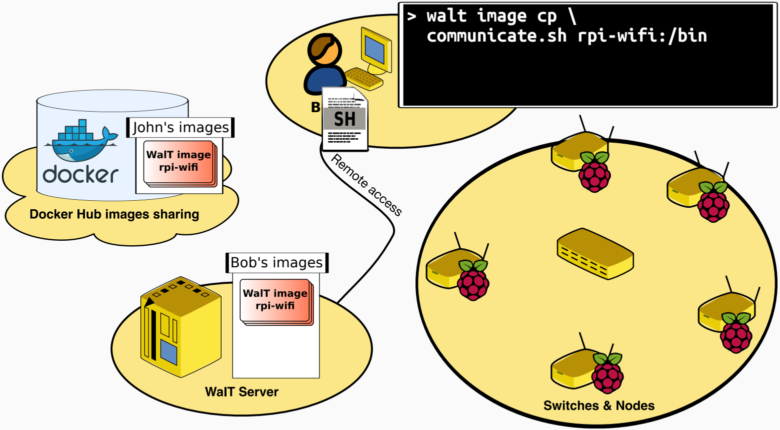


```
> walt image search wifi
hub:john/rpi-wifi
> walt image clone hub:john/rpi-wifi
done.
>
```

Sample use case

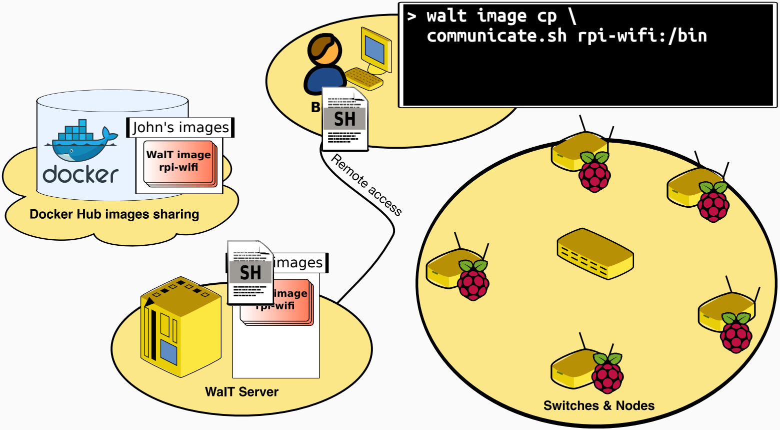


Sample use case



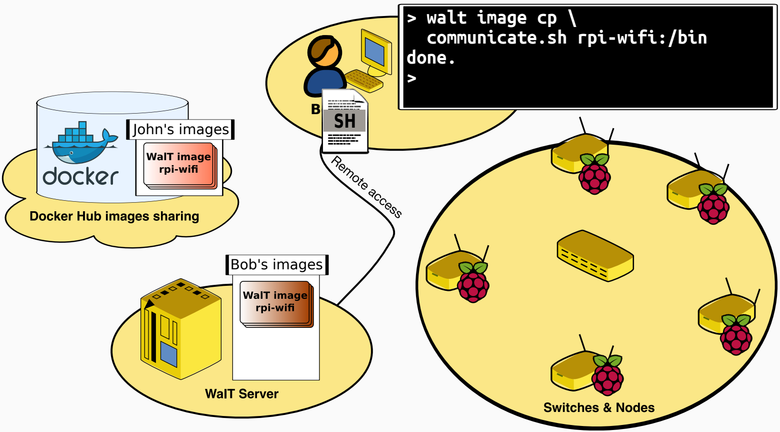
```
> walt image cp \
  communicate.sh rpi-wifi:/bin
```

Sample use case



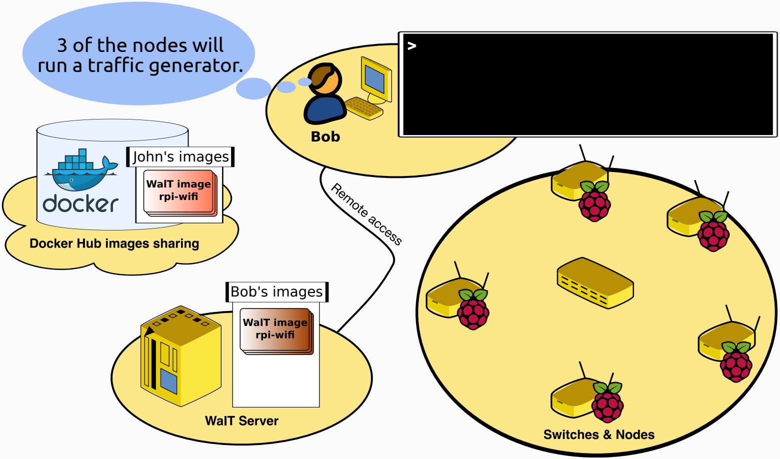
```
> walt image cp \
  communicate.sh rpi-wifi:/bin
```

Sample use case

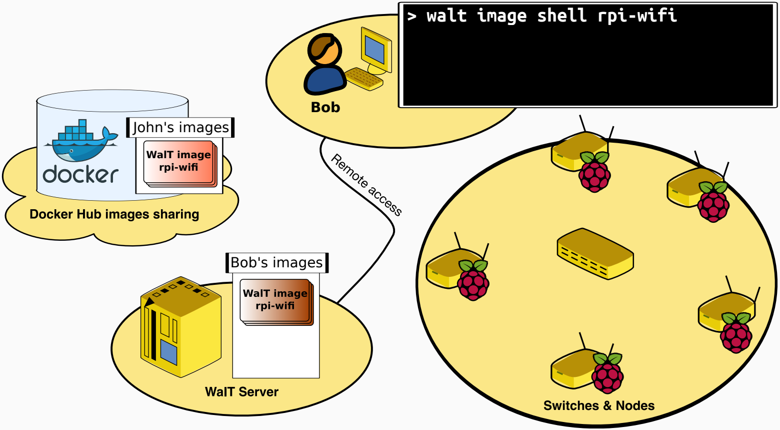


```
> walt image cp \
  communicate.sh rpi-wifi:/bin
done.
>
```

Sample use case

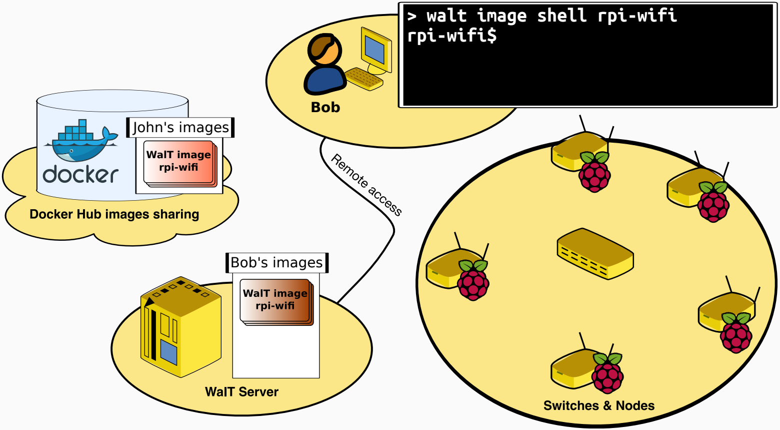


Sample use case



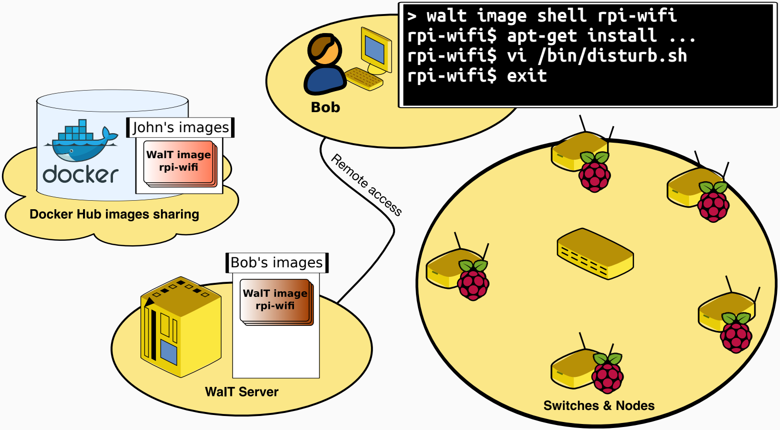
```
> walt image shell rpi-wifi
```

Sample use case



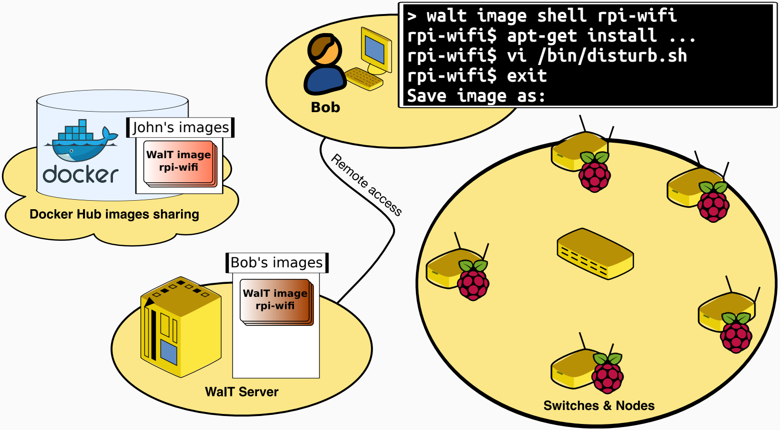
```
> walt image shell rpi-wifi
rpi-wifi$
```

Sample use case



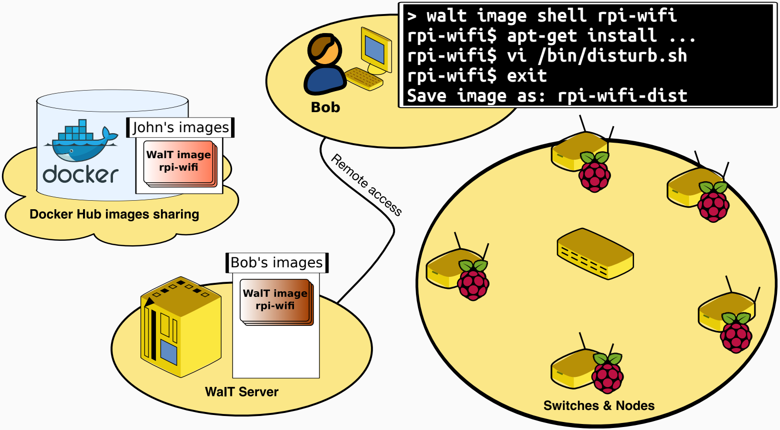
```
> walt image shell rpi-wifi
rpi-wifi$ apt-get install ...
rpi-wifi$ vi /bin/disturb.sh
rpi-wifi$ exit
```

Sample use case



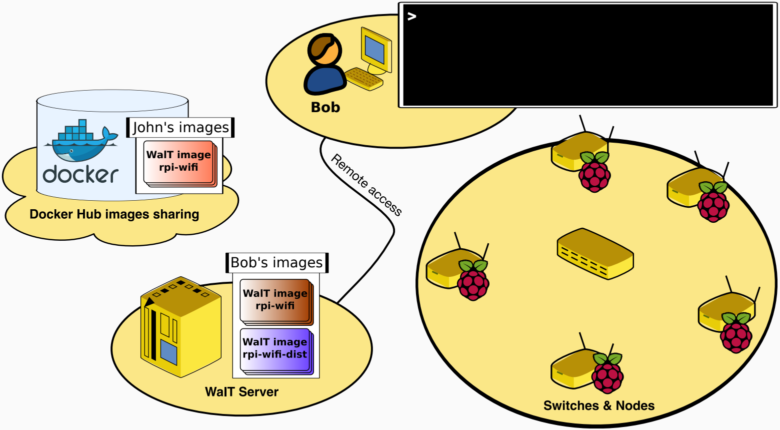
```
> walt image shell rpi-wifi
rpi-wifi$ apt-get install ...
rpi-wifi$ vi /bin/disturb.sh
rpi-wifi$ exit
Save image as:
```

Sample use case



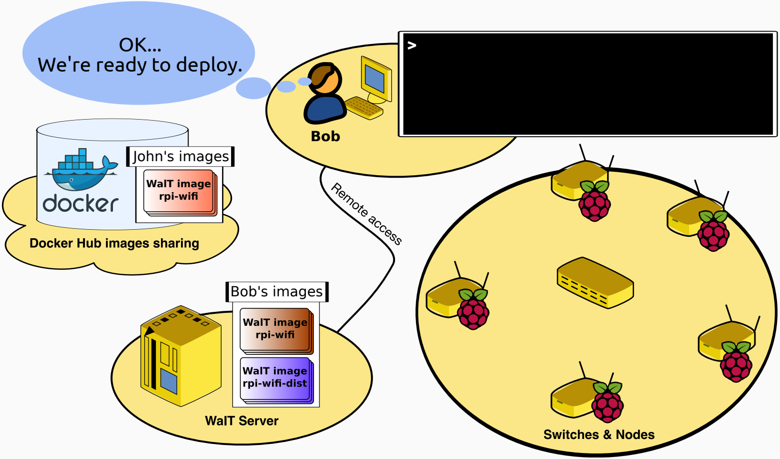
```
> walt image shell rpi-wifi
rpi-wifi$ apt-get install ...
rpi-wifi$ vi /bin/disturb.sh
rpi-wifi$ exit
Save image as: rpi-wifi-dist
```

Sample use case

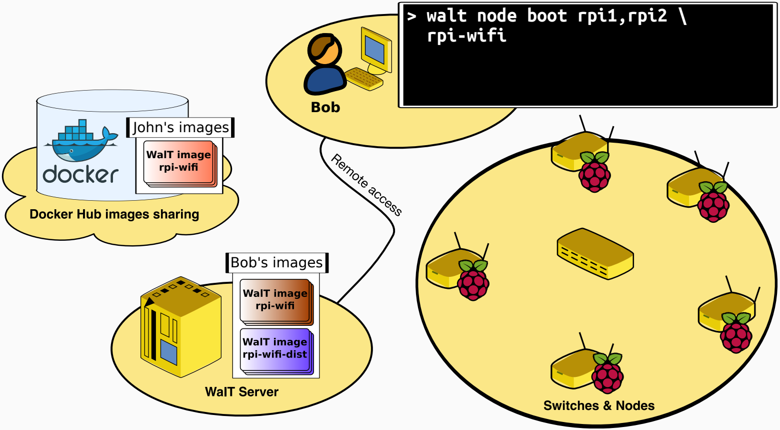


```
>
```

Sample use case

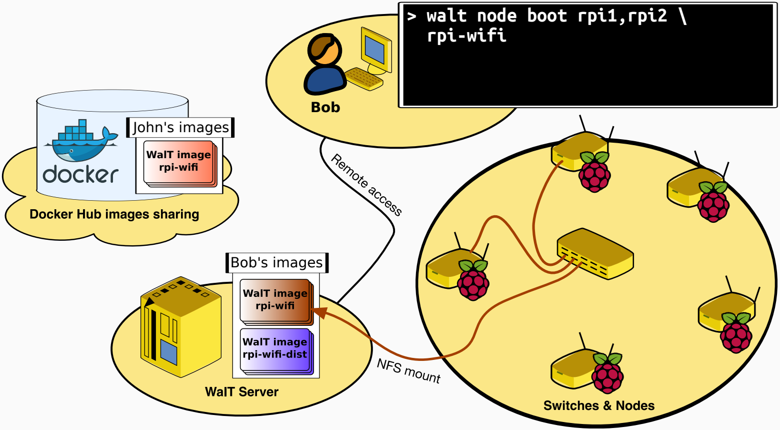


Sample use case



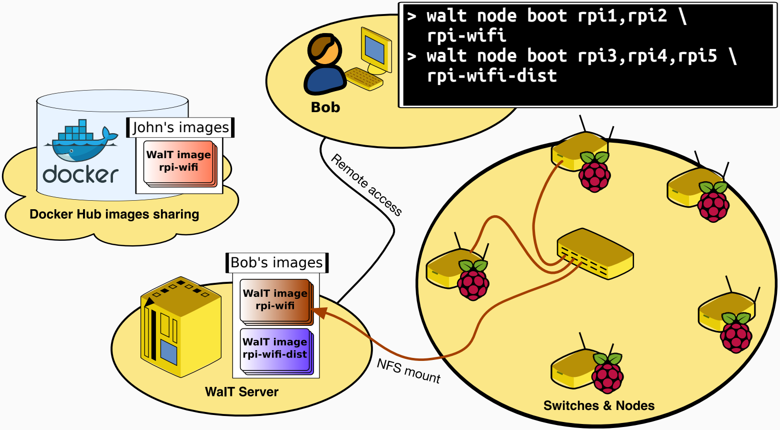
```
> walt node boot rpi1,rpi2 \
  rpi-wifi
```

Sample use case



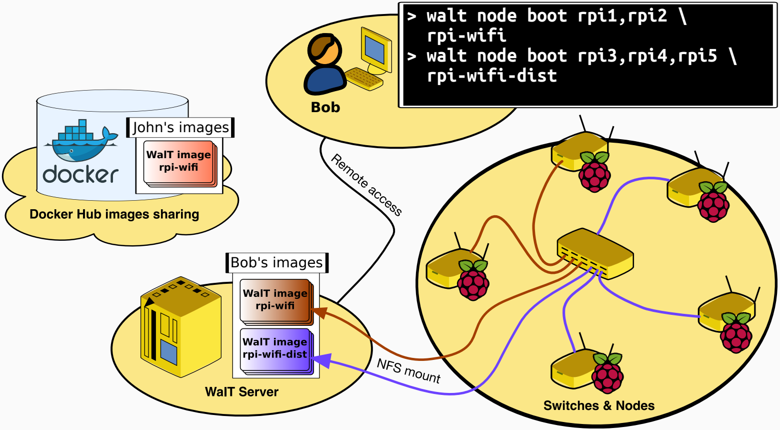
```
> walt node boot rpi1,rpi2 \
  rpi-wifi
```

Sample use case



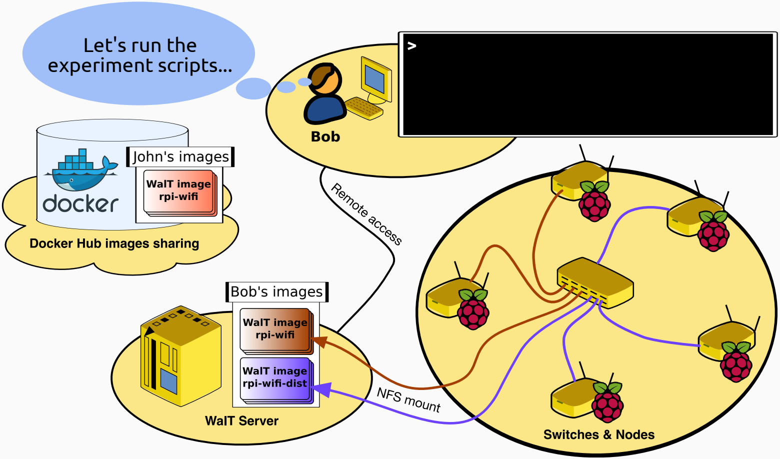
```
> walt node boot rpi1,rpi2 \
  rpi-wifi
> walt node boot rpi3,rpi4,rpi5 \
  rpi-wifi-dist
```

Sample use case

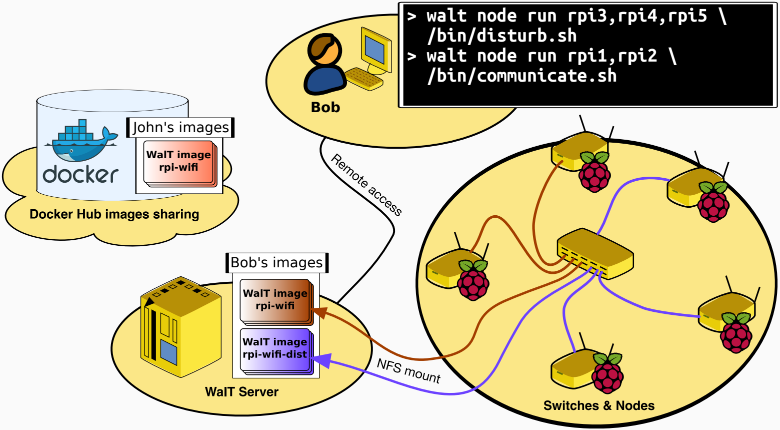


```
> walt node boot rpi1,rpi2 \
  rpi-wifi
> walt node boot rpi3,rpi4,rpi5 \
  rpi-wifi-dist
```

Sample use case

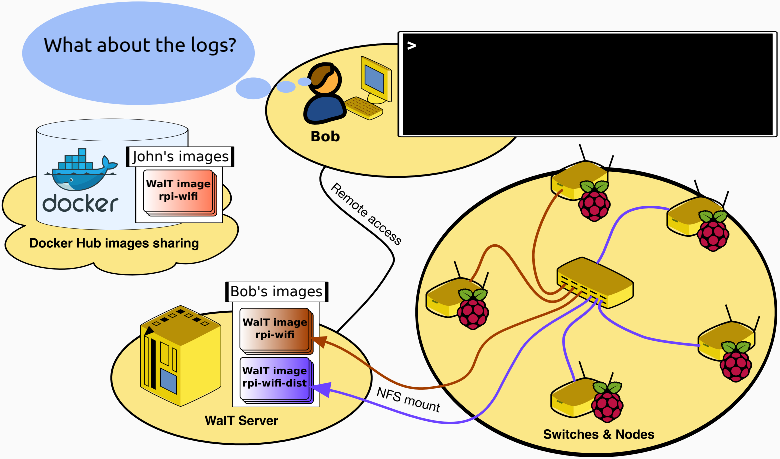


Sample use case

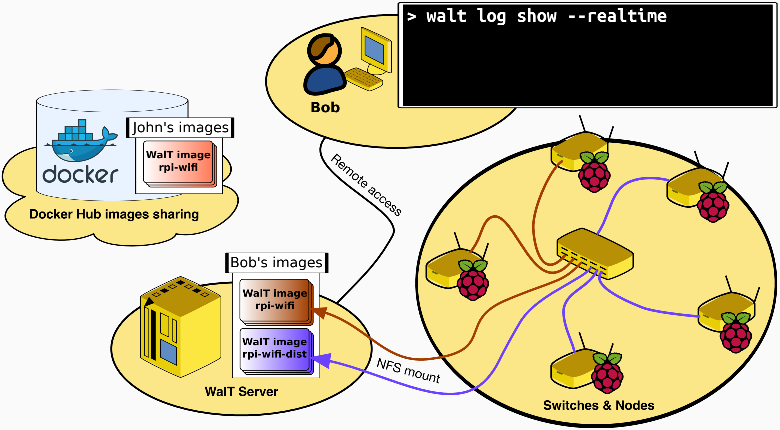


```
> walt node run rpi3,rpi4,rpi5 \
  /bin/disturb.sh
> walt node run rpi1,rpi2 \
  /bin/communicate.sh
```

Sample use case

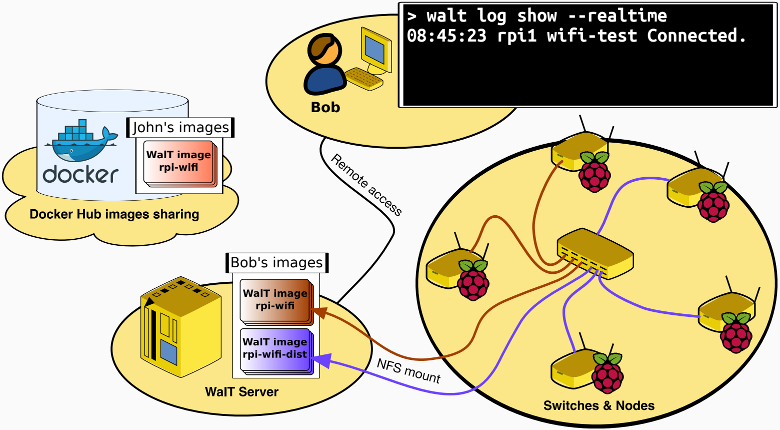


Sample use case



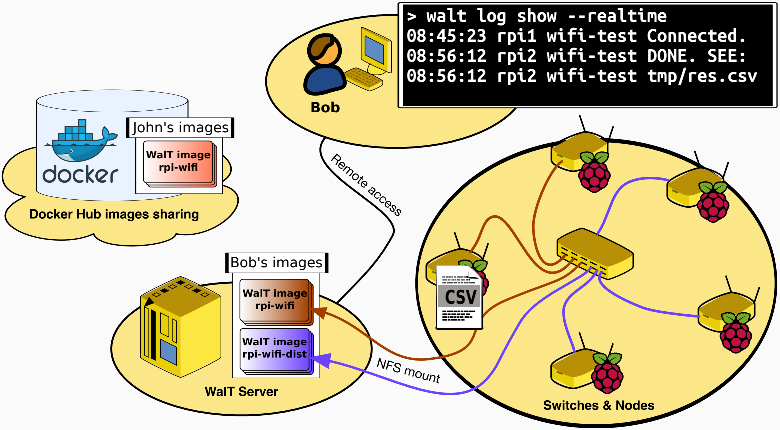
```
> walt log show --realtime
```

Sample use case



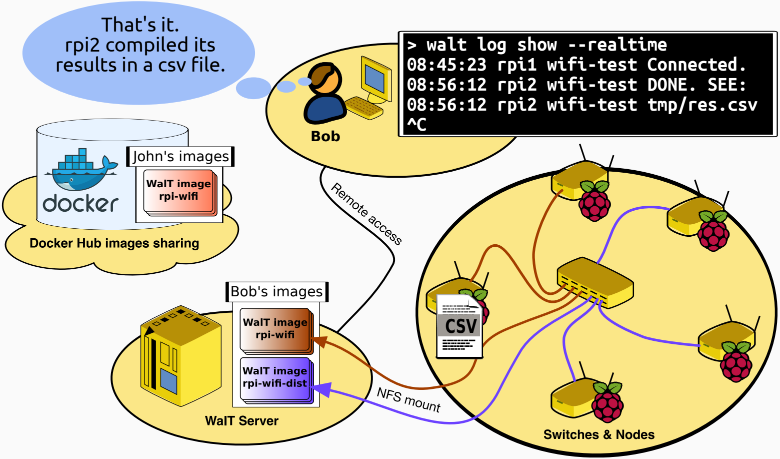
```
> walt log show --realtime
08:45:23 rpi1 wifi-test Connected.
```

Sample use case



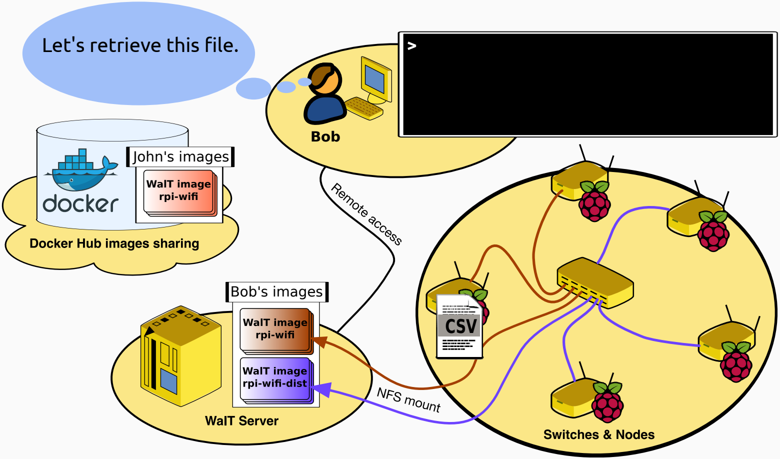
```
> walt log show --realtime
08:45:23 rpi1 wifi-test Connected.
08:56:12 rpi2 wifi-test DONE. SEE:
08:56:12 rpi2 wifi-test tmp/res.csv
```

Sample use case



```
> walt log show --realtime
08:45:23 rpi1 wifi-test Connected.
08:56:12 rpi2 wifi-test DONE. SEE:
08:56:12 rpi2 wifi-test tmp/res.csv
^C
```

Sample use case



Let's retrieve this file.

Bob

```
>
```

John's images

WalT image rpi-wifi

Docker Hub images sharing

Remote access

Bob's images

WalT image rpi-wifi

WalT image rpi-wifi-dist

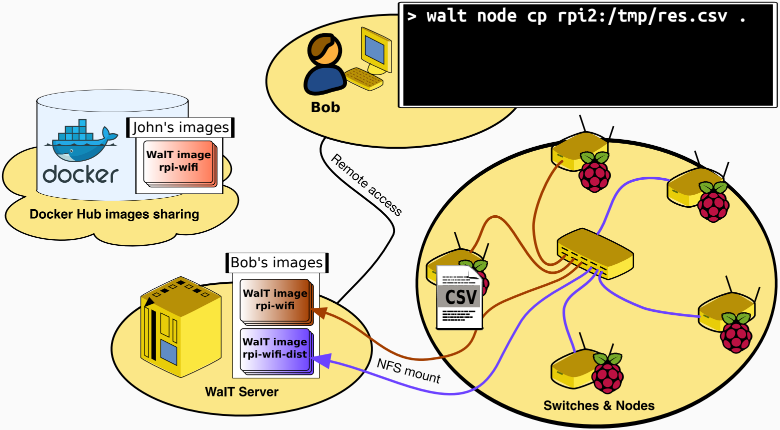
WalT Server

NFS mount

CSV

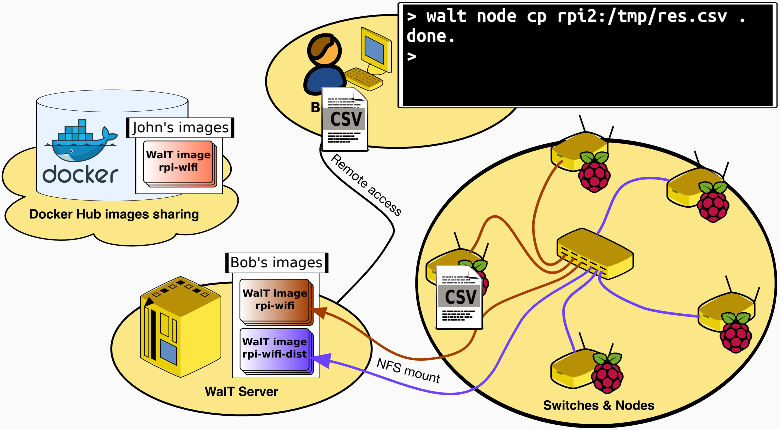
Switches & Nodes

Sample use case



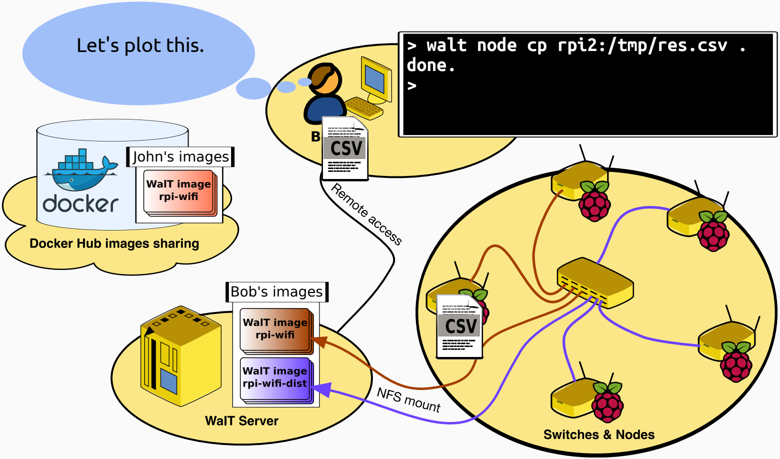
```
> walt node cp rpi2:/tmp/res.csv .
```

Sample use case



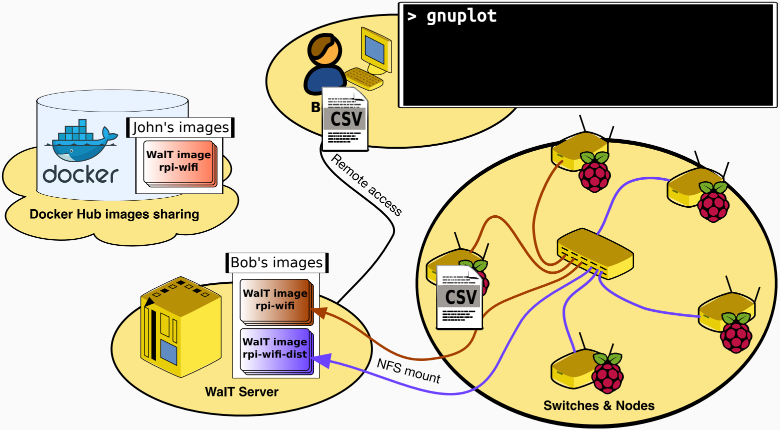
```
> walt node cp rpi2:/tmp/res.csv .
done.
>
```

Sample use case



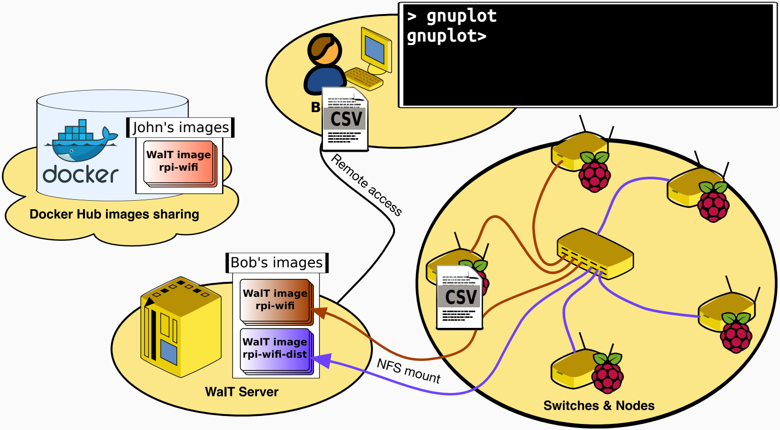
```
> walt node cp rpi2:/tmp/res.csv .
done.
>
```

Sample use case



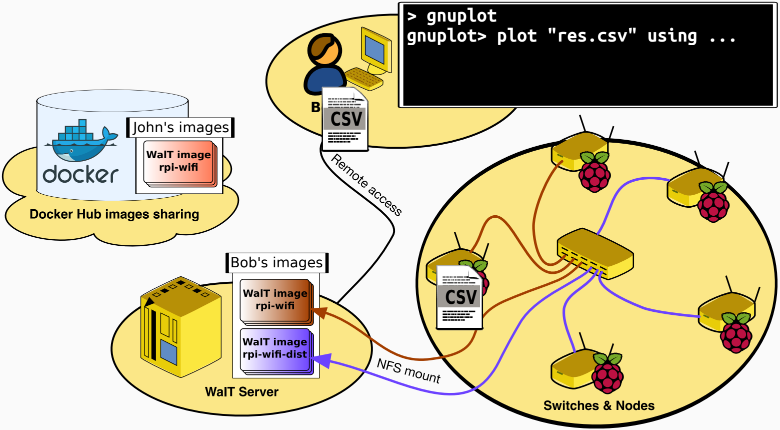
```
> gnuplot
```

Sample use case



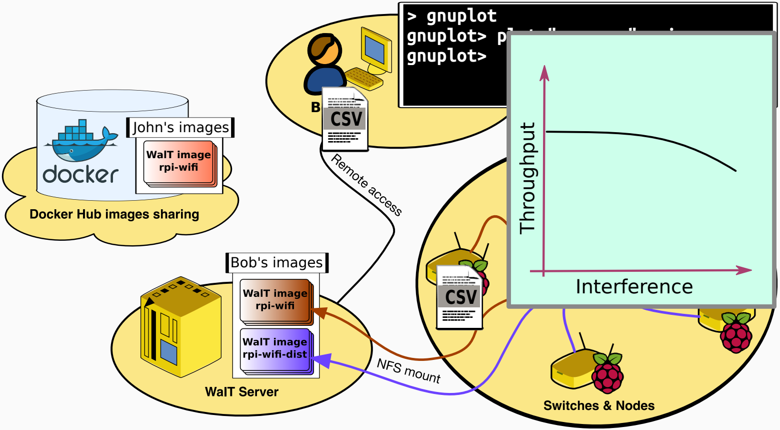
```
> gnuplot
gnuplot>
```

Sample use case



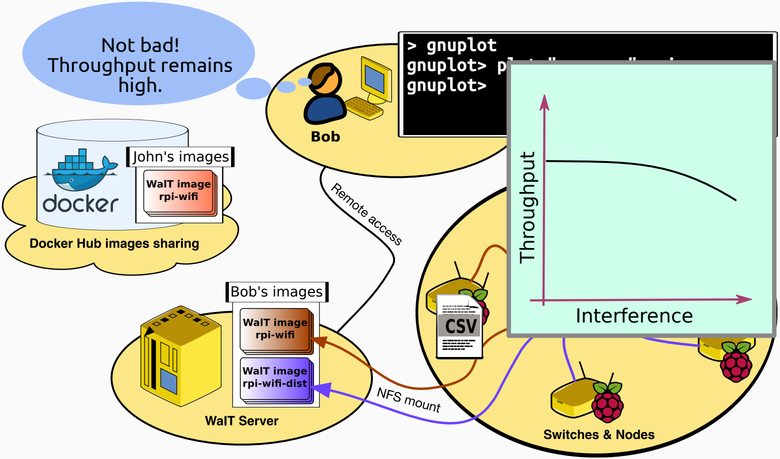
```
> gnuplot
gnuplot> plot "res.csv" using ...
```

Sample use case



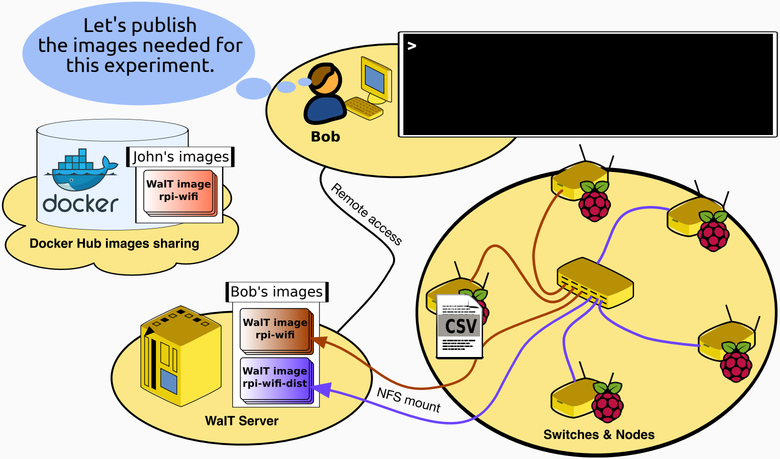
```
> gnuplot
gnuplot> p
gnuplot>
```

Sample use case



```
> gnuplot
gnuplot> p
gnuplot>
```

Sample use case



Let's publish the images needed for this experiment.

```
>
```

Bob

John's images

WalT image rpi-wifi

Docker Hub images sharing

Remote access

Bob's images

WalT image rpi-wifi

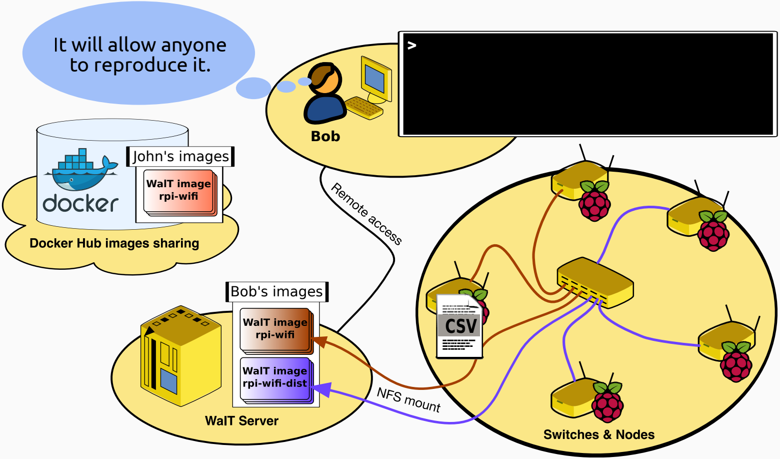
WalT image rpi-wifi-dist

WalT Server

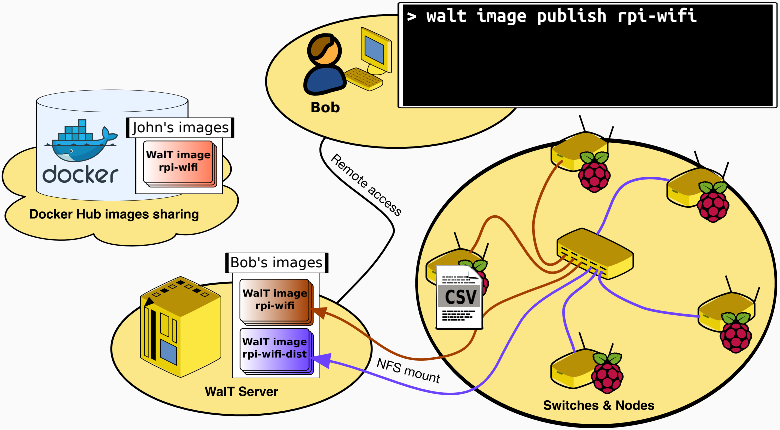
NFS mount

Switches & Nodes

Sample use case

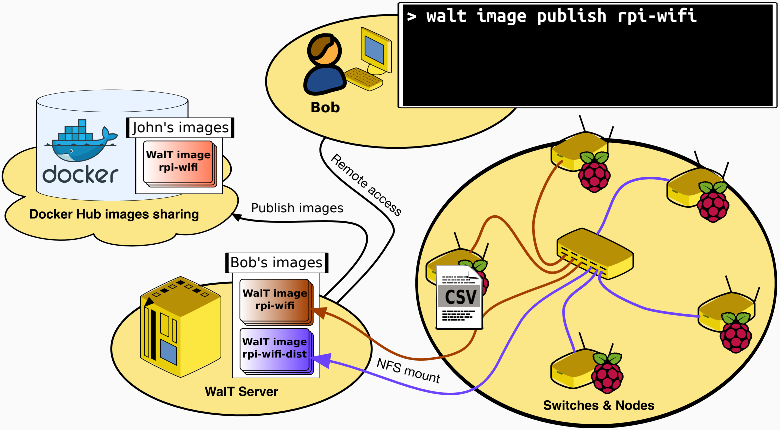


Sample use case



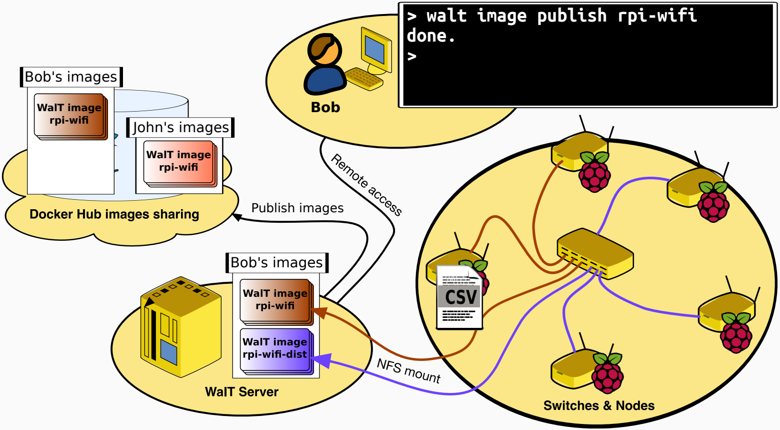
```
> walt image publish rpi-wifi
```

Sample use case



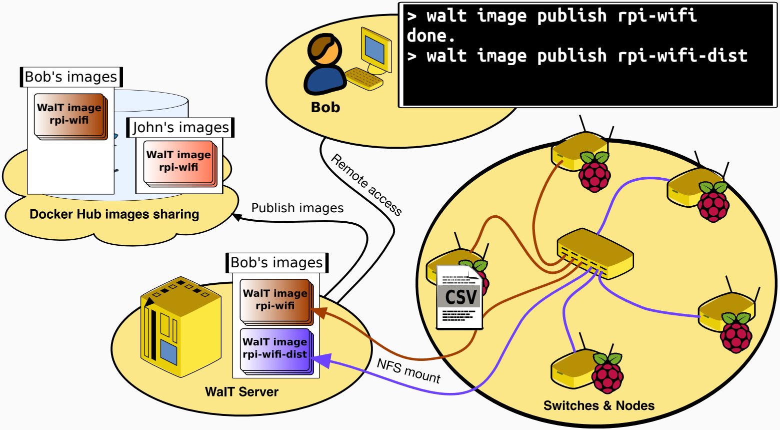
```
> walt image publish rpi-wifi
```

Sample use case



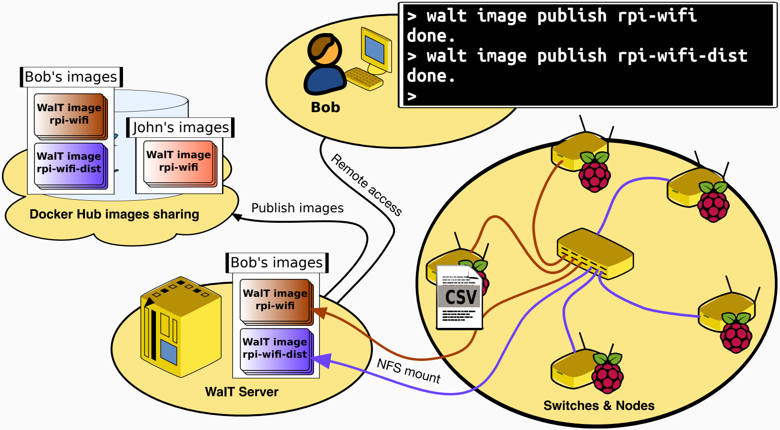
```
> walt image publish rpi-wifi
done.
>
```

Sample use case



```
> walt image publish rpi-wifi
done.
> walt image publish rpi-wifi-dist
```

Sample use case



```
> walt image publish rpi-wifi
done.
> walt image publish rpi-wifi-dist
done.
>
```

Sample use case

Agenda

Motivation

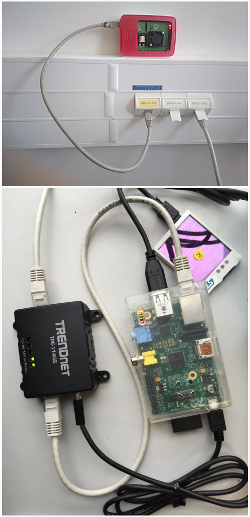
Sample use case

Architecture

WalT under the hood

News, On-going work and Perspectives

What is a WalT node?



- Historically, a WalT node was a **Single-Board-Computer**[1] preferably powered through PoE
- Its OS is a "**WalT image**" shared from the server through TFTP/NFS

2 cases:

1. WalT nodes run the experiment, **or**
2. WalT nodes control the target devices of the experiment (e.g., ST sensor board in the picture)

[1] We support **Raspberry Pi B/B+/2B/3B/3B+/4B** and **x86 boards**.

What is a WalT node?

WalT now supports *other kinds of nodes*:

- **Any PC** can become a WalT node,
  by booting a USB stick flashed appropriately
- One can also create **virtual nodes** (kvm)
- And walt also supports **distant nodes** (VPN-connected)

WalT platform network

Network isolation

- WalT wired network is **isolated** from internet by default. (→ improve reproducibility)
- If it's needed, **walt node netsetup** command can still toggle NAT mode on a node, to allow internet access.

VPN and distant nodes

- WALT embeds a **ssh-based VPN** for distant nodes (cf. LoRa, SigFox).
- Distant nodes are **Rpi3B+** nodes
- Their network boot procedure is augmented with a level of **virtualization** to support the VPN connection.

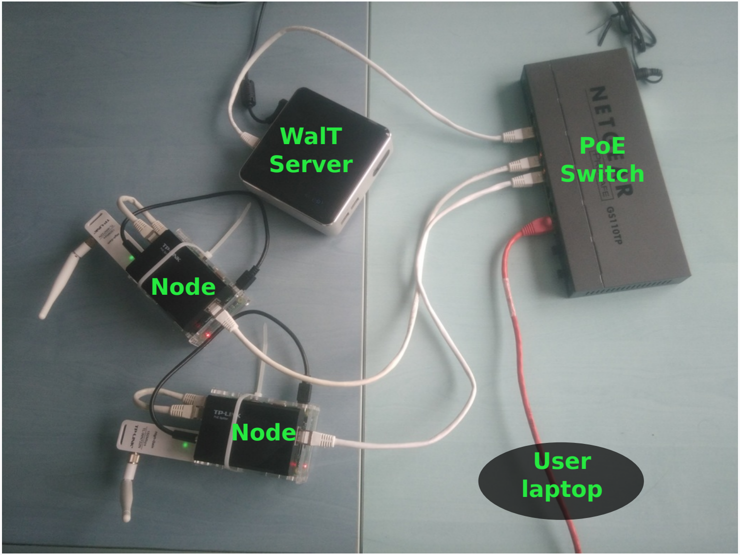
The *reproducible platform* concept

WalT is a *reproducible platform*[1]

- **Any researcher can build its own WalT platform**
- ... or several platforms of various scales (e.g., one for debugging, one for experiments)

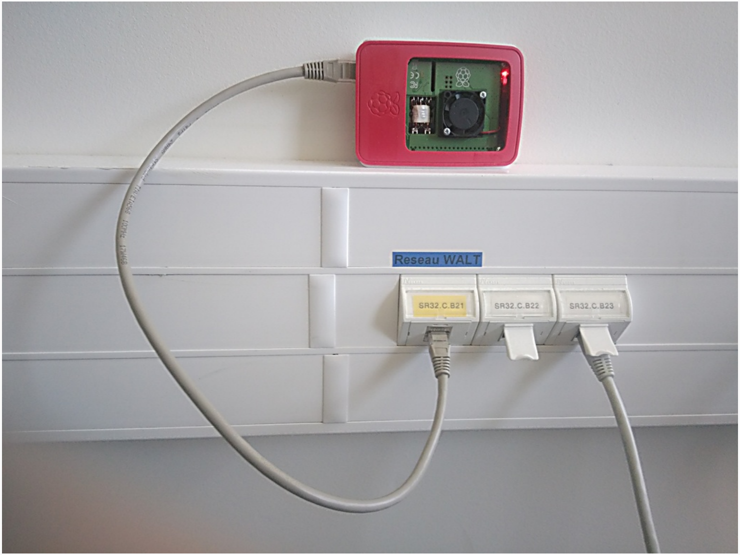
[1] P. Brunisholz, E. Duble, F. Rousseau, and A. Duda, "WalT: A Reproducible Testbed for Reproducible Network Experiments", in Proc. of the International Workshop on Computer and Networking Experimental Research Using Testbeds, 2016.

Sample mobile setup



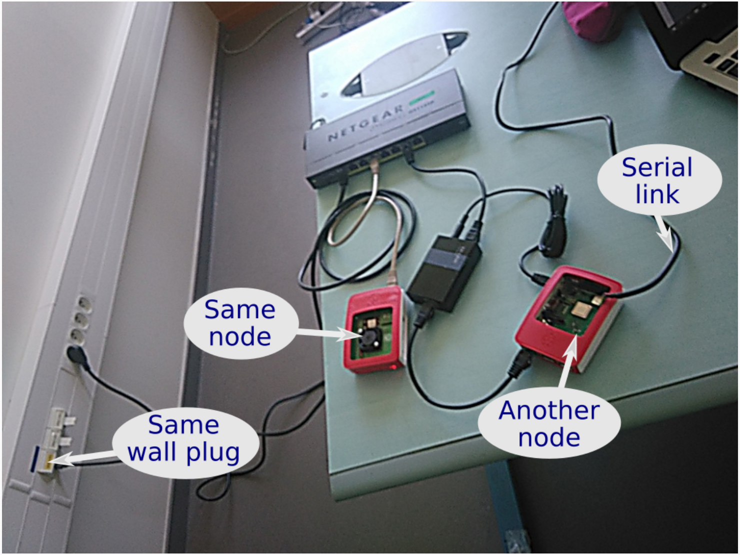
Minimal setup for an ANR Datatweet meeting

WalT versatility shown in two slides (1)



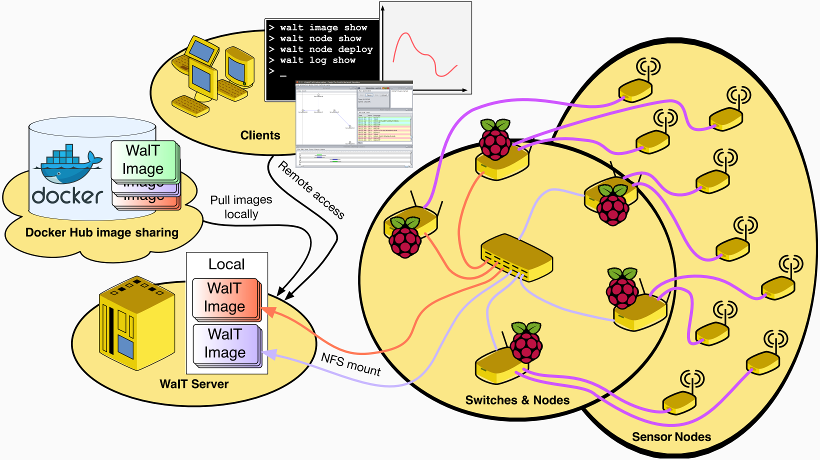
A node of the WalT network in IMAG building

WalT versatility shown in two slides (2)



Reusing the same wall plug for temporary low-level debugging

Architecture for WSNs



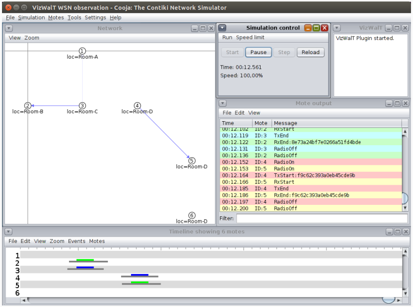
```
> walt image show
> walt node show
> walt node deploy
> walt log show
> _
```

Architecture for WSNs

In this case:

- We want to monitor *sensor nodes*
- WalT nodes are "*controller nodes*"
  (e.g., flash sensor nodes, save logs, debug using the serial link)
- A WalT image provides this support

VizWalt



Time: 00:12.561
Speed: 100,00%

VizWalT Plugin started.

```
Time       Mote   Message
00:12.102  ID:2   RxStart
00:12.119  ID:3   TxEnd
00:12.122  ID:2   RxEnd:8e73a24bf7e0266a51fd4bde
00:12.131  ID:3   RadioOff
00:12.136  ID:2   RadioOff
00:12.152  ID:4   RadioOn
00:12.153  ID:5   RadioOn
00:12.164  ID:4   TxStart:f9c62c393a0eb45cde9b
00:12.166  ID:5   RxStart
00:12.185  ID:4   TxEnd
00:12.186  ID:5   RxEnd:f9c62c393a0eb45cde9b
00:12.197  ID:4   RadioOff
00:12.200  ID:5   RadioOff
```

- A *Cooja* simulator plugin
- It makes *Cooja* show the behavior of real sensor nodes (instead of simulating)
- Input: logs coming from sensor nodes
- VizWalT *visualizes* what's happening in the network, *in real time* or *post mortem*

Agenda

Motivation

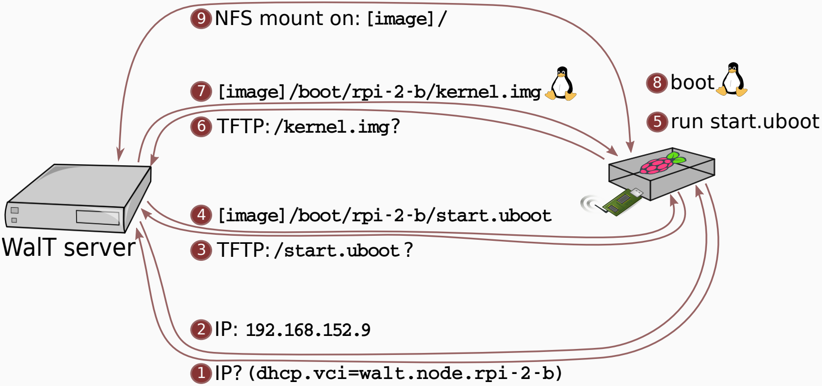
Sample use case

Architecture

WalT under the hood

News, On-going work and Perspectives

Node boot procedure



9 NFS mount on: `[image] /`

8 boot

7 `[image] /boot/rpi-2-b/kernel.img`

6 TFTP: `/kernel.img?`

5 run start.uboot

4 `[image] /boot/rpi-2-b/start.uboot`

3 TFTP: `/start.uboot ?`

2 IP: `192.168.152.9`

1 IP? `(dhcp.vci=walt.node.rpi-2-b)`

NFS reminder

- **NFS: Network File System**
- Allows to **expose locally** a view of the **files stored elsewhere**
- Each access to a file is translated into a request-response to NFS server
- Comparison:
  Local file system: requests go to disk controller
  Network file system: requests go to remote server
- **Consequence: WalT images stay on the server.**
  A WalT image is never transfered as a whole to the node.

Node boot procedure (continued)

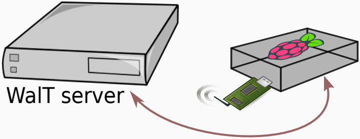
12 Start the rest of the OS:
`[image]/sbin/init`

11 Start walt-specific services
(e.g. reboot service)

10 Create an overlay in RAM:

`/ = UNION(nfs:[image]/,ram:<tmp>)`

=> file modifications stored in RAM
=> **`nfs:[image]`** can remain read-only



9 NFS mount on: **`[image]/`**

Node boot procedure (continued)

Benefits of this boot procedure:

- WalT images embed a kernel, an OS filesystem, bootloader scripts: *the whole software stack can be customized.*
- The SD card of a node just contains a network bootloader: => It remains read-only (this ensures **robustness** over time)
- *Nodes are completely stateless*:
  - At each reboot, they just download `/start.uboot` and run it (the server maintains redirections to the selected walt image)
  - File modifications (stored in RAM) are lost on reboot: when booting the same image, OS restarts exactly the same (this ensures **reproducibility**)

Technical highlights

Other technical highlights:

- "`walt image shell`" command:
    - Provides a shell session rooted inside the image
    - Allows to modify an existing image very easily
    - Behind the scenes, a *docker container* is created on server
    - *Transparent cross-CPU-architecture* is handled using `qemu`
- "`walt node boot`" command:
    - Exposes the content of a docker image as an NFS share
    - Lets the node(s) boot it
    - In short, this instantly *turns a virtual image into a real deployment*.

Agenda

Motivation

Sample use case

Architecture

WalT under the hood

News, On-going work and Perspectives

Recent news

Recent news

- WALT v7 is available since february 1, 2021
- A new WALT platform should be install at LAAS lab shortly (Toulouse)
- Schneider Electric has built a demonstrator based on WalT (with our help)
- Other platforms have been setup (Lyon, Istanbul)
- WalT is also used for wifi hands-on at Grenoble INP Ensimag
- We connected our WalT platform at IMAG to our data management platform (https://sakura-platform.liglab.fr/)

On-going work and Perspectives

On-going or for the near future

- Design of a more efficient network filesystem (wrt. NFS) for high-latency networks (e.g. VPN)
- Several network experiments (Drakkar team at LIG)
- Possible network monitoring experiments in the IMAG building[1]
- Help for new walt deployments (at LAAS (Toulouse), possibly at Bu-Ali Sina University (Iran))
- Several planned software improvements[2]

[1] with G. Enderlé, É. Jullien of Grenoble University

[2] https://github.com/drakkar-lig/walt-python-packages/issues

Getting help or more information

- get help:
  - register to the list: **walt-users@univ-grenoble-alpes.fr**
  - or checkout: `https://walt-project.liglab.fr`, section "Resources and Documentation"
  - or type: `walt help show`
- or just contact us:
  - email: walt-contact@univ-grenoble-alpes.fr

How to contribute

- Setup[1] and use your own platform, and
    - Share your WalT images on the *docker hub*
    - Add support for new kinds of nodes
    - Report bugs
- Discuss with us possible improvements

[1] Contact us (walt-contact@univ-grenoble-alpes.fr), we will assist you.

WalT credits

- Initial funding
    - Univ. Grenoble Alpes, Grenoble INP / UJF, AGIR WalT (2013-2014)
    - ANR IRIS (2011)
    - FP7 ICT CALIPSO - Connect All IP-based Smart Objects (2011)
    - ANR DataTweet (2013)
- Thanks
    - Pierre Brunisholz, Baptiste Jonglez, Narek Davtyan, Ayoub Bargach, Mehmet Tahir Sandikkaya, Qasim Lone, Takwa Attia, Rémy Grünblatt, Paul Grangette, WalT images design, development, testing & documentation
    - Bastien Faure, Pierre-Henry Frohring, Jorane Congio-Hollard, Cosmin Nichifor, core WalT
    - Jorge Luis Baranguan Castro, VizWalT
    - Henri-Joseph Audéoud, Elodie Morin, Timothy Claeys, Onss Bousbih, testing and demo setup
    - Joao Guilherme Zeni, Matheus Castanho, Iacob Juc, Liviu Varga, sensor integration & Contiki instrumentation

Questions?