

# 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, September 2020

Wait 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

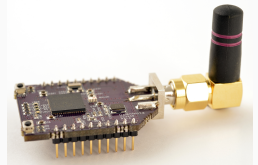
# Existing platforms

## 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



# WaT platform objectives

WaT stands halfway between those 2 categories.

## **Just as a handmade platform,**

- WaT can be lightweight (backpack-compatible, for a demo)
- You can build several instances of WaT
- 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

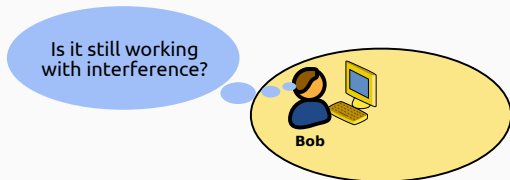
Sample use case

Architecture

WalT under the hood

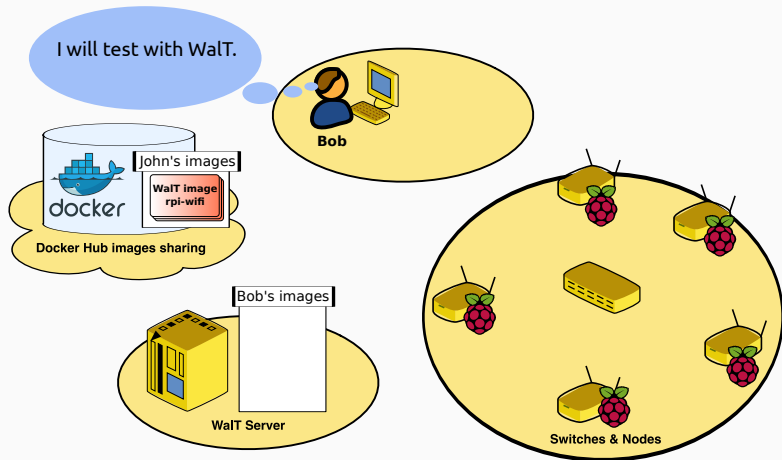
News, On-going work and Perspectives

# Sample use case

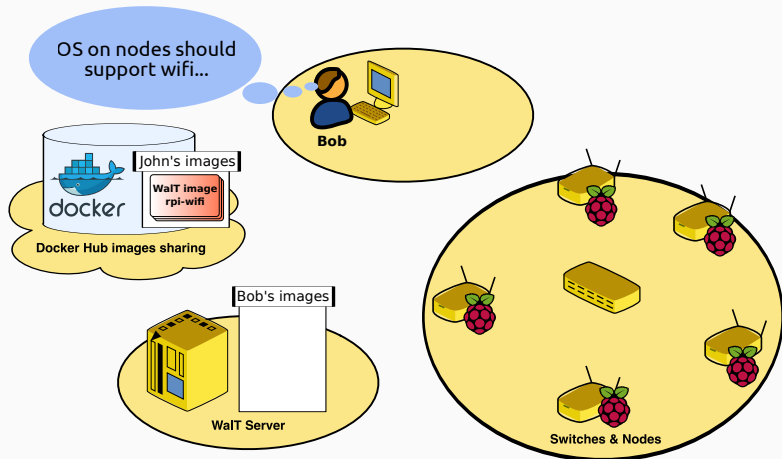




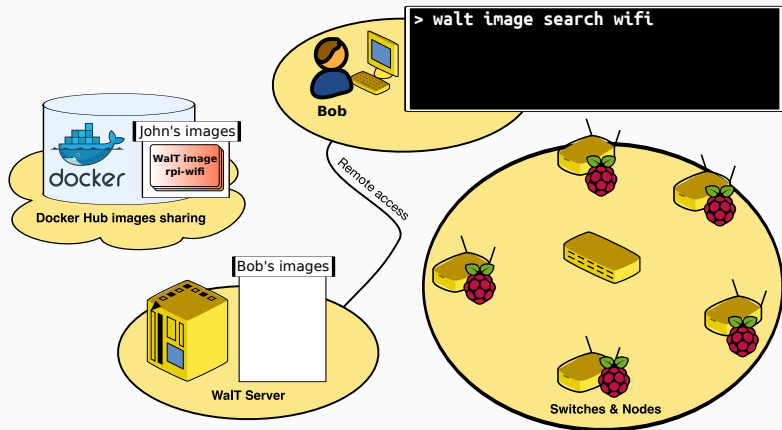
# Sample use case



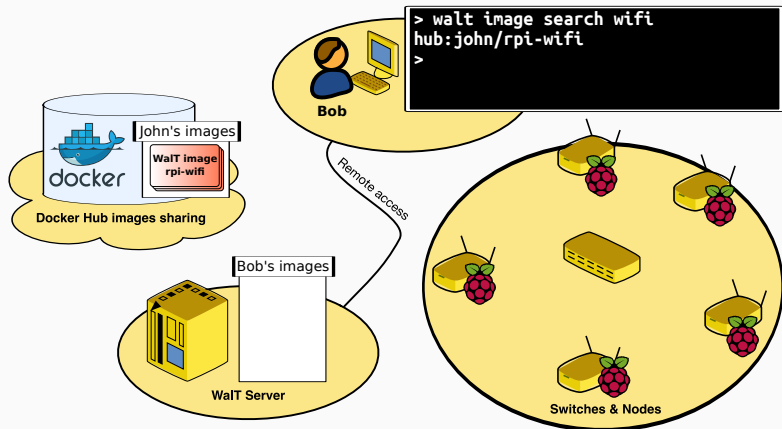
# Sample use case



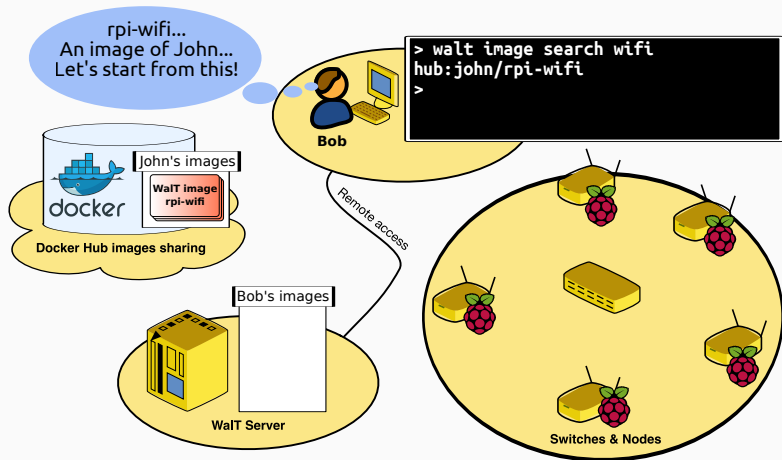
# Sample use case



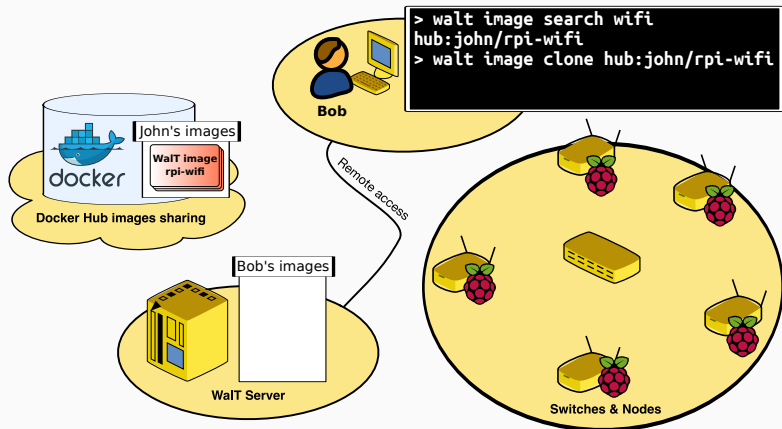
# Sample use case



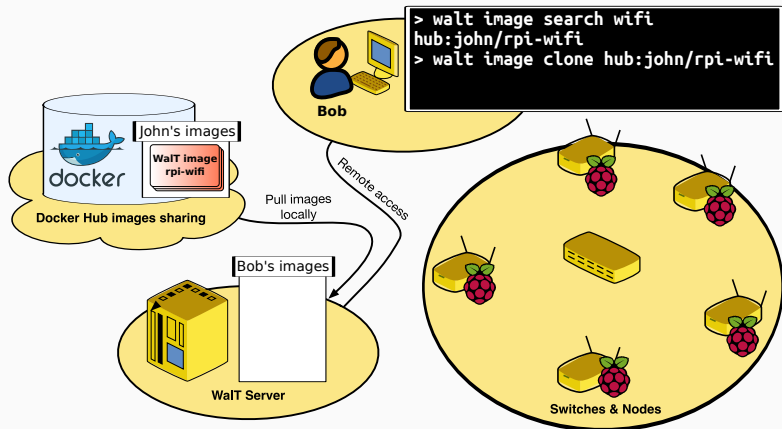
# Sample use case



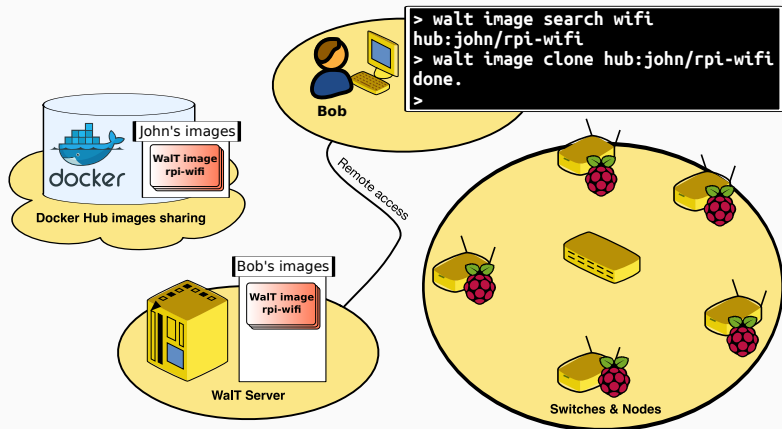
# Sample use case



# Sample use case

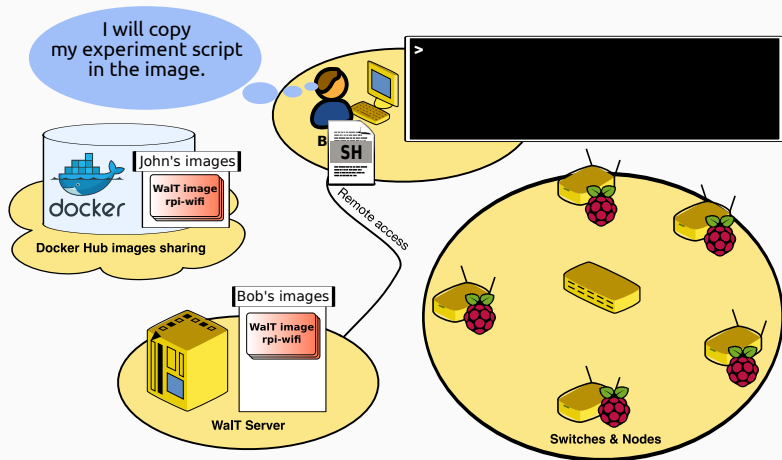


# Sample use case

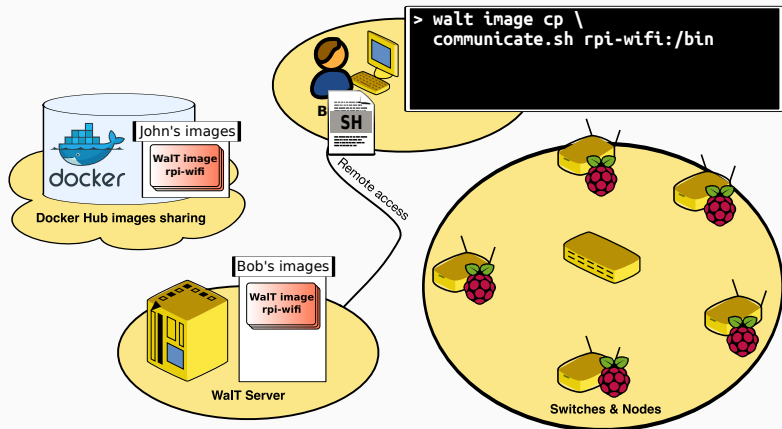




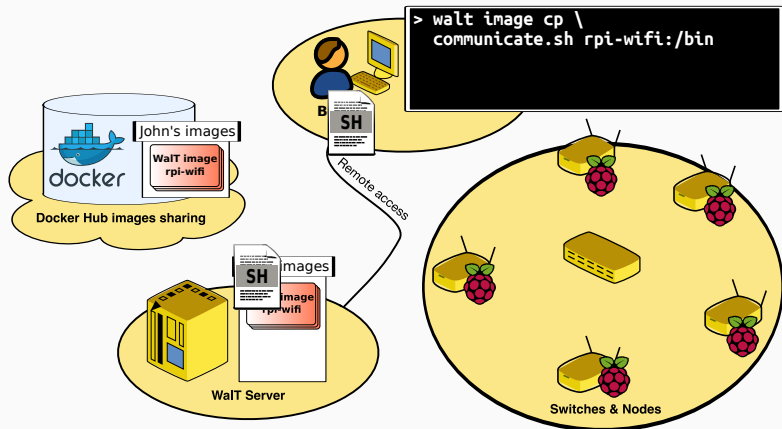
# Sample use case



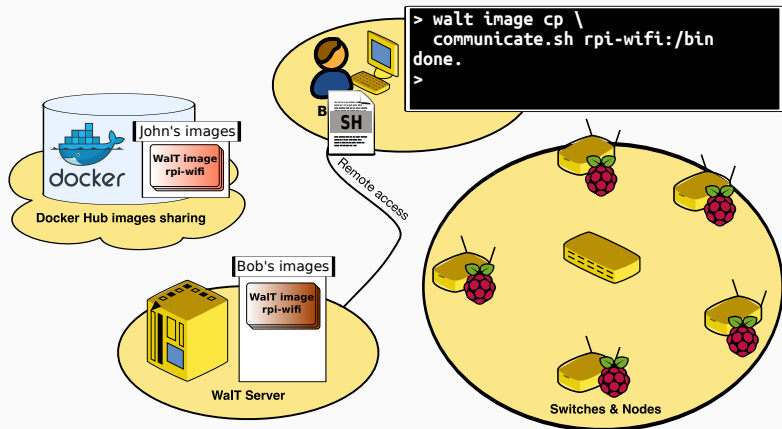
# Sample use case



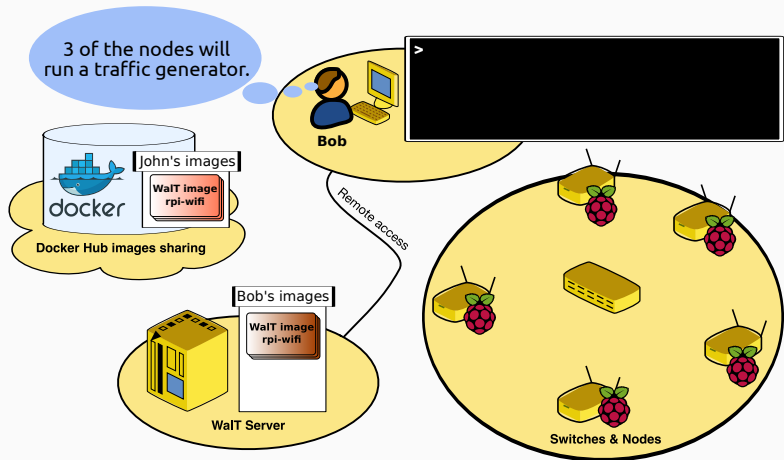
# Sample use case



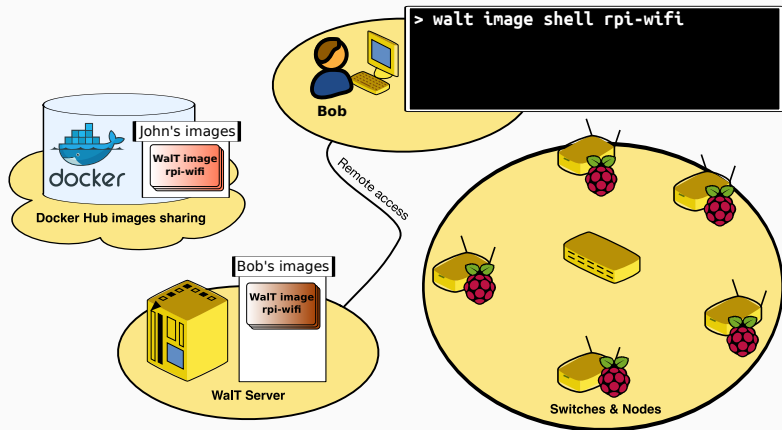
# Sample use case



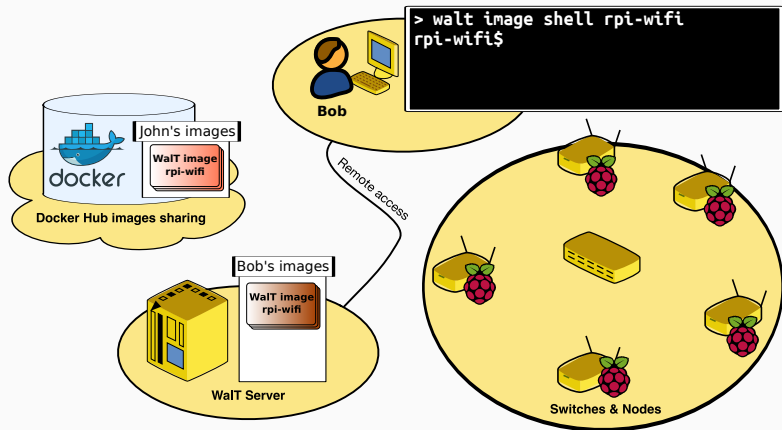
# Sample use case



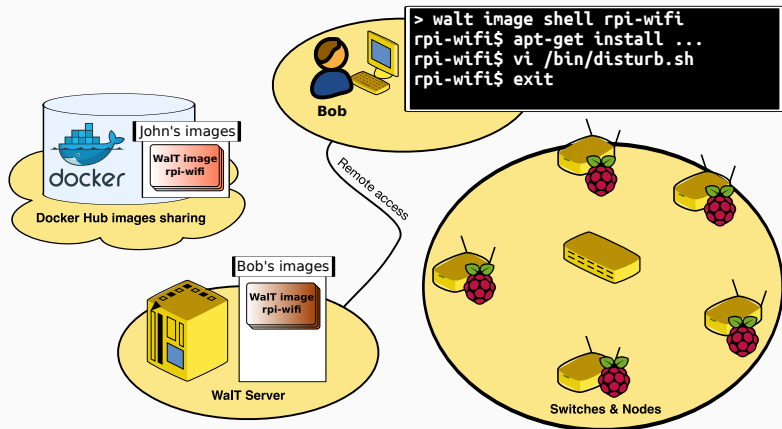
# Sample use case



# Sample use case

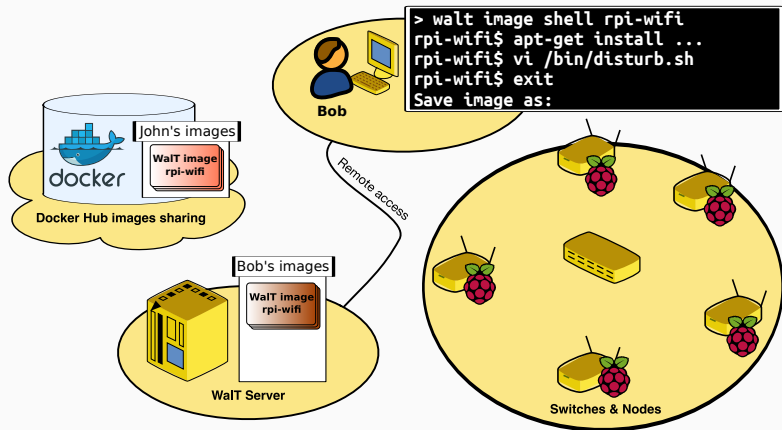


# Sample use case

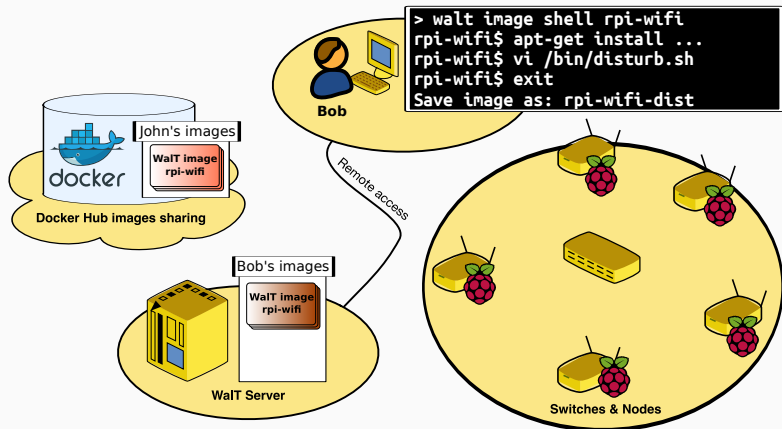




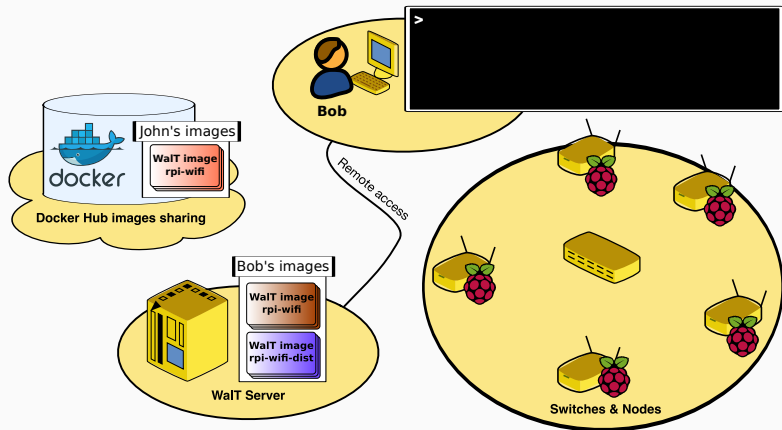
# Sample use case



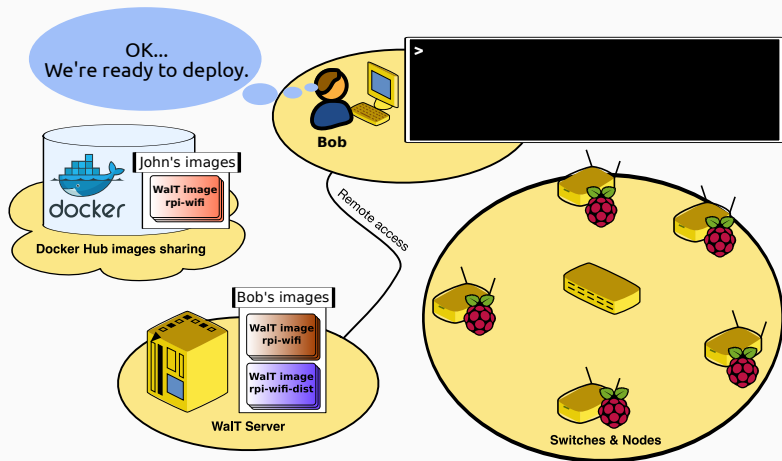
# Sample use case



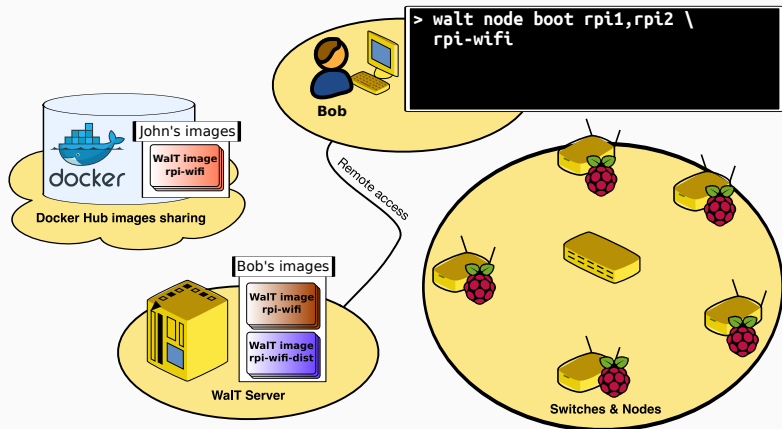
# Sample use case



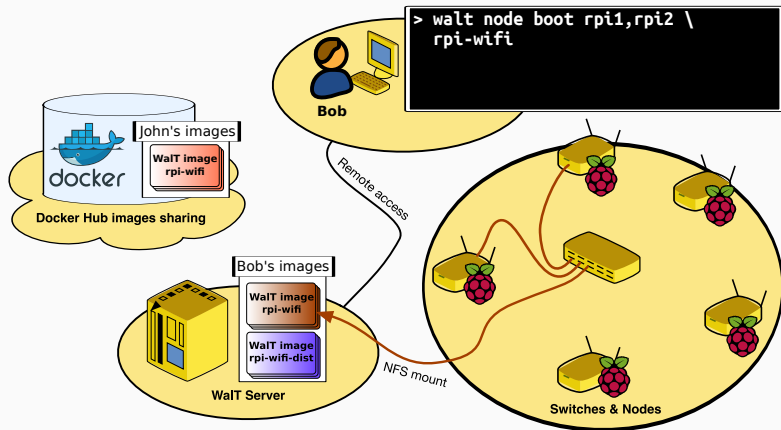
# Sample use case



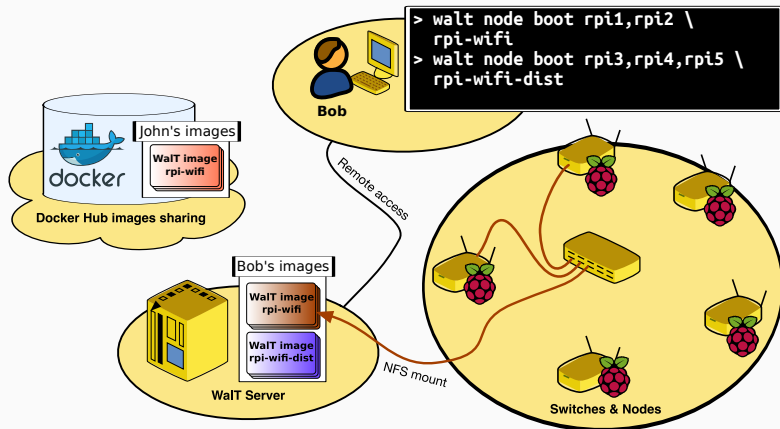
# Sample use case



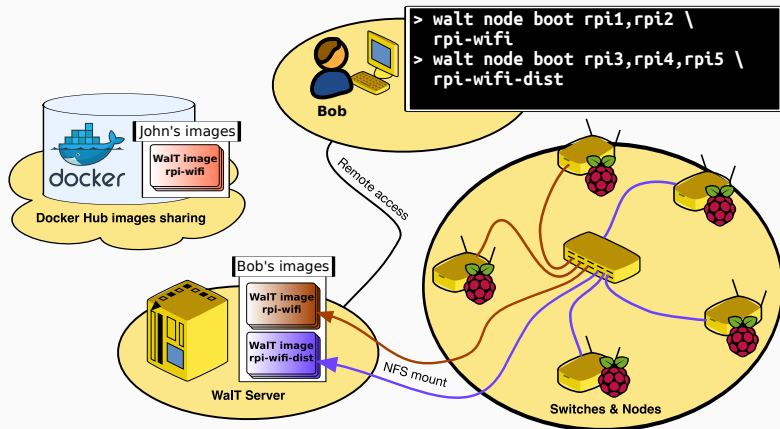
# Sample use case



# Sample use case

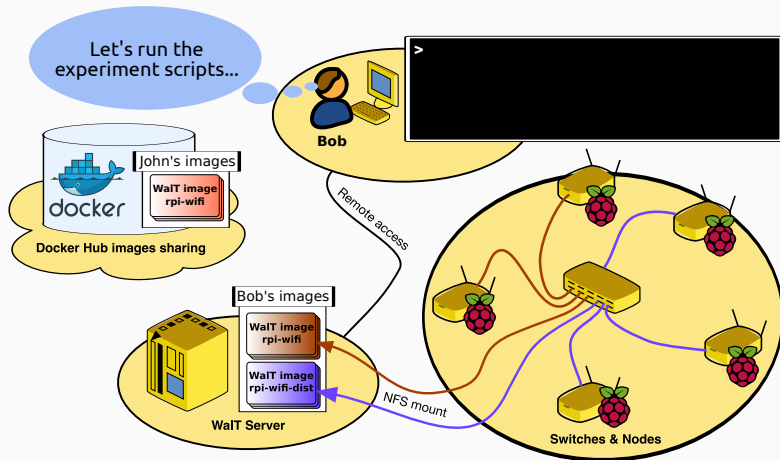


# Sample use case

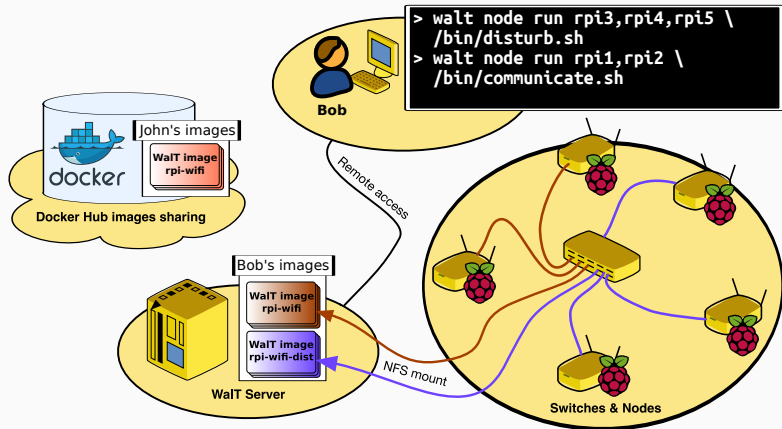




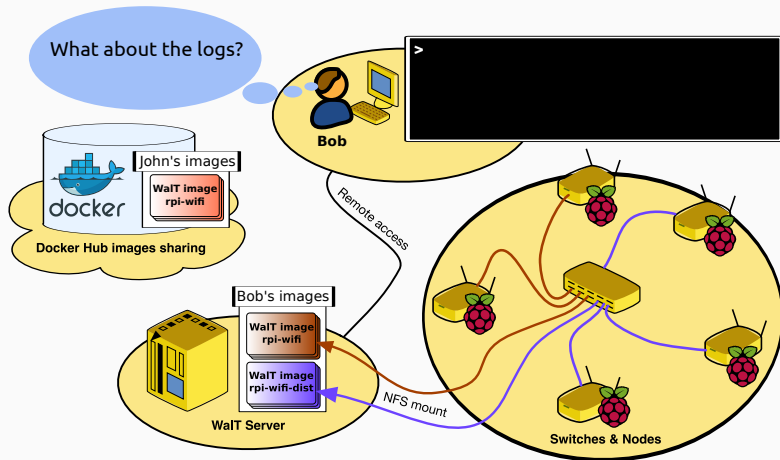
# Sample use case



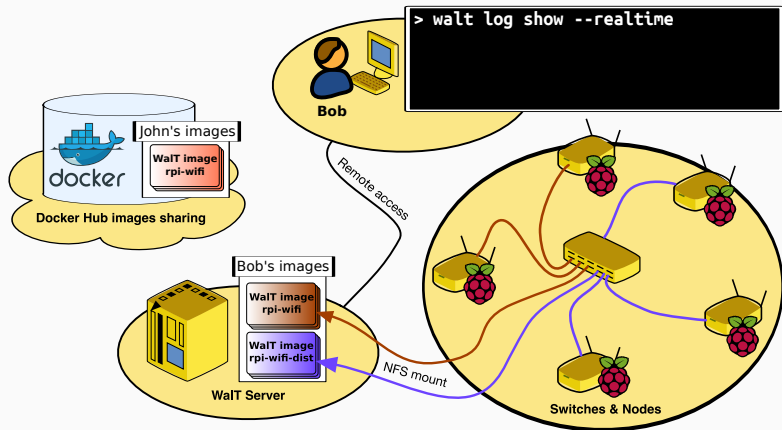
# Sample use case



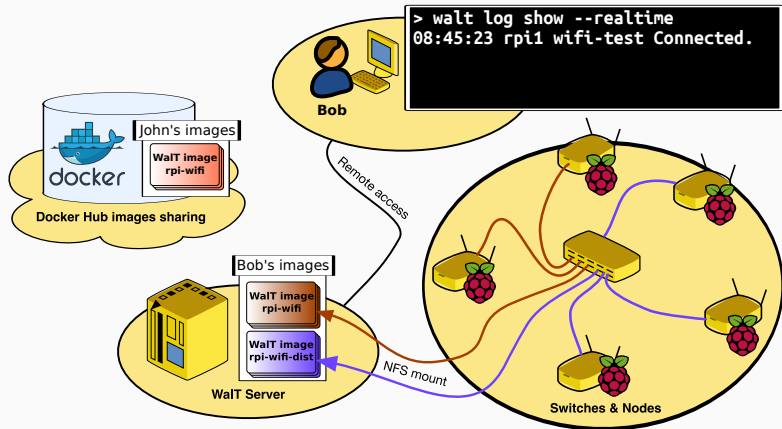
# Sample use case



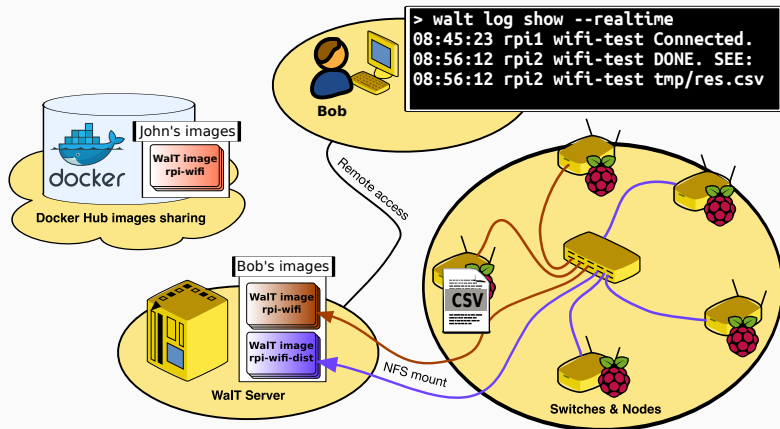
# Sample use case



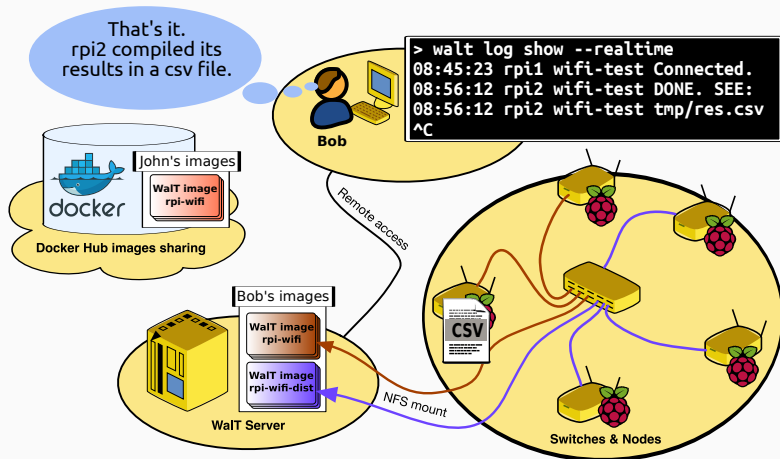
# Sample use case



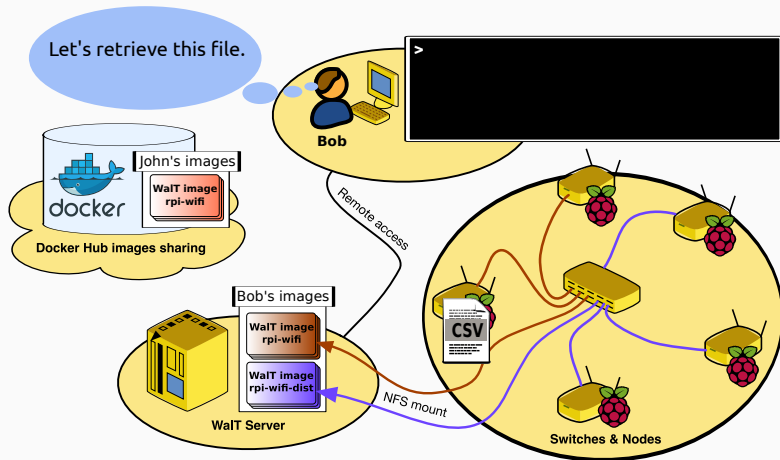
# Sample use case



# Sample use case

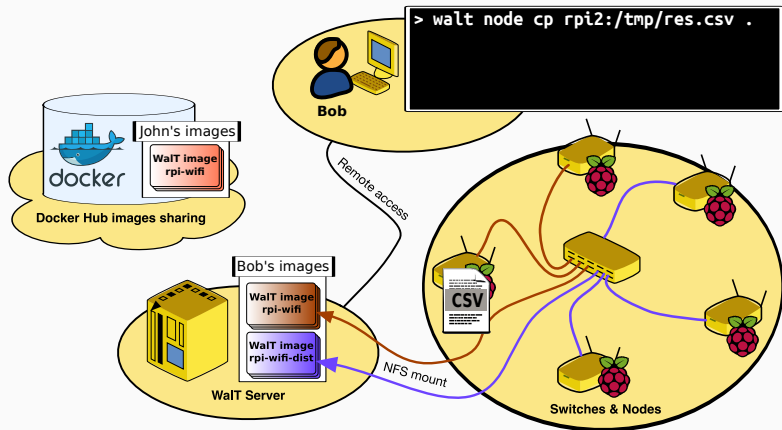


# Sample use case

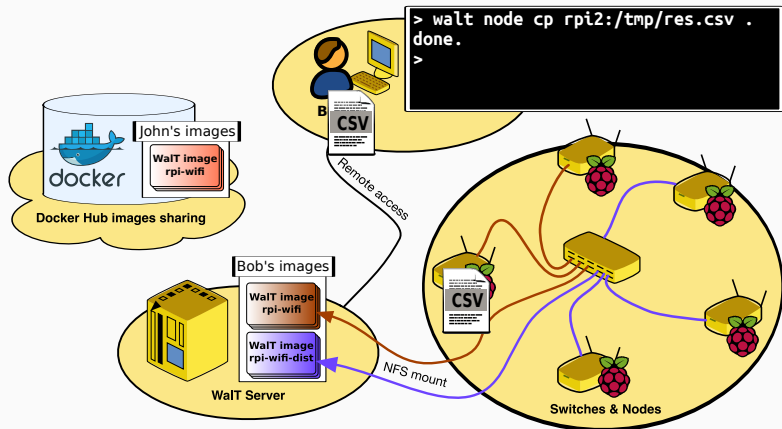




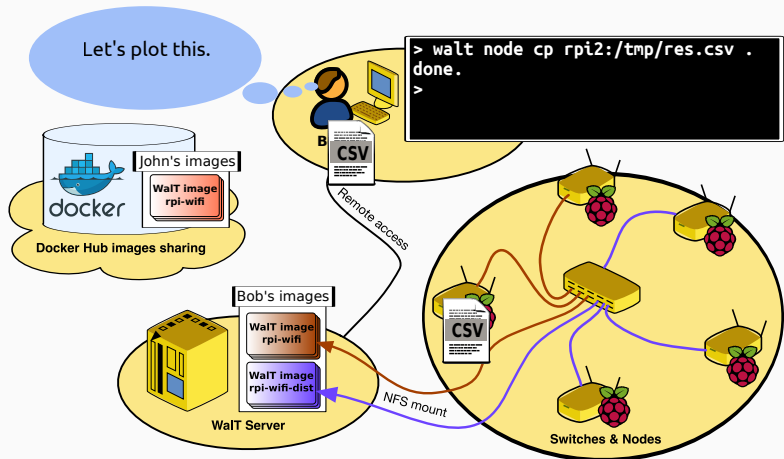
# Sample use case



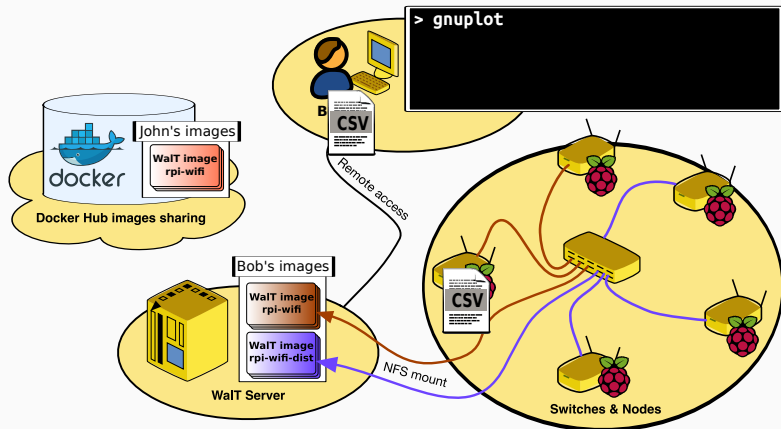
# Sample use case



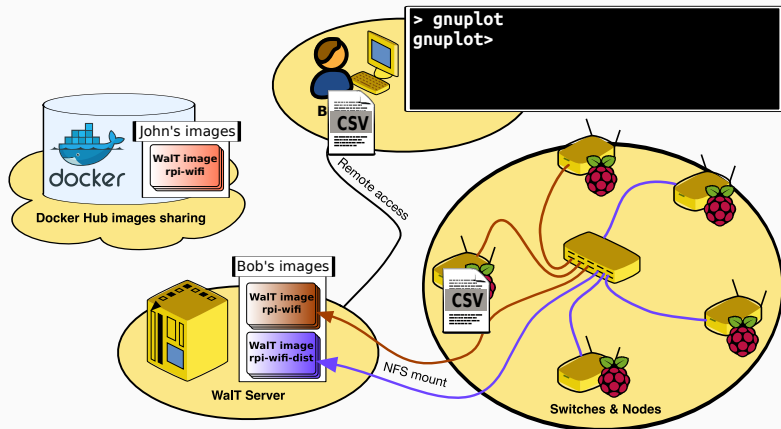
# Sample use case



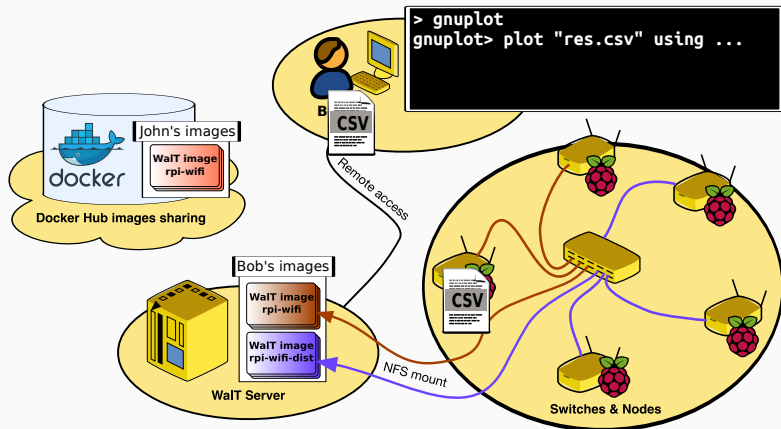
# Sample use case



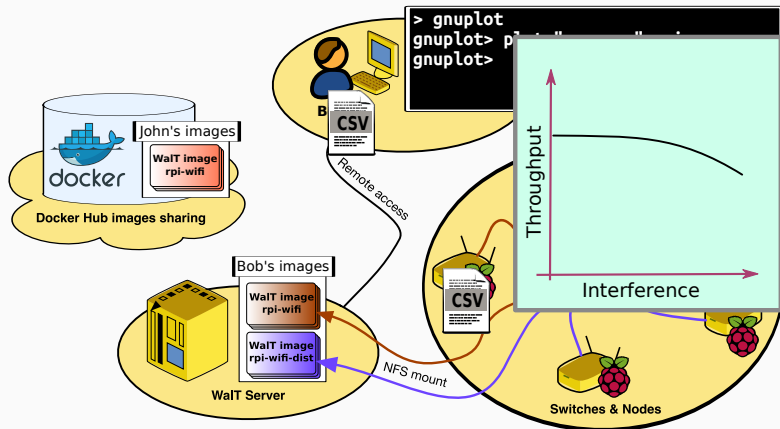
# Sample use case



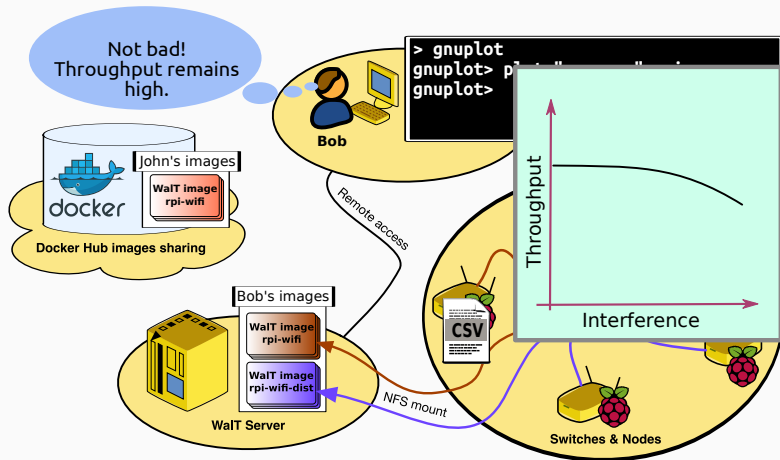
# Sample use case



# Sample use case

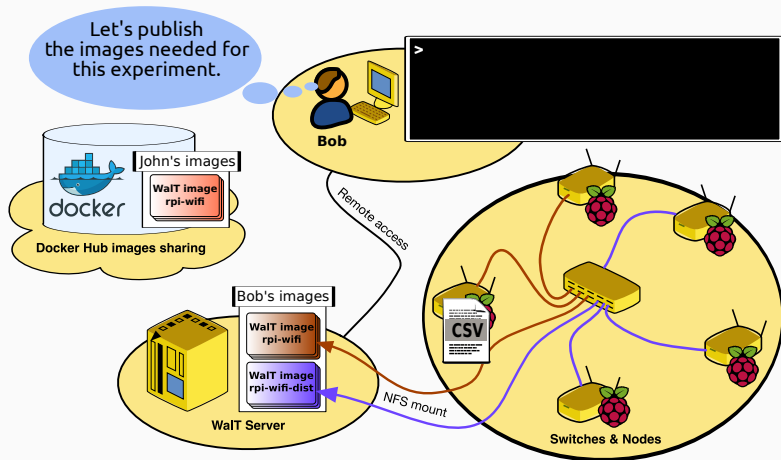


# Sample use case

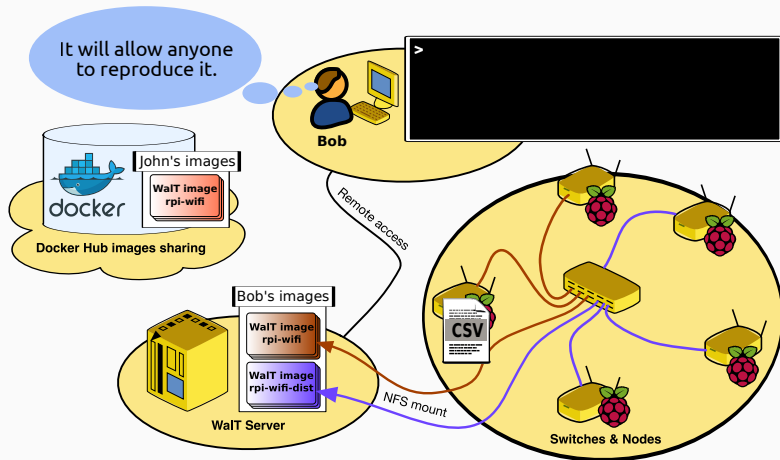




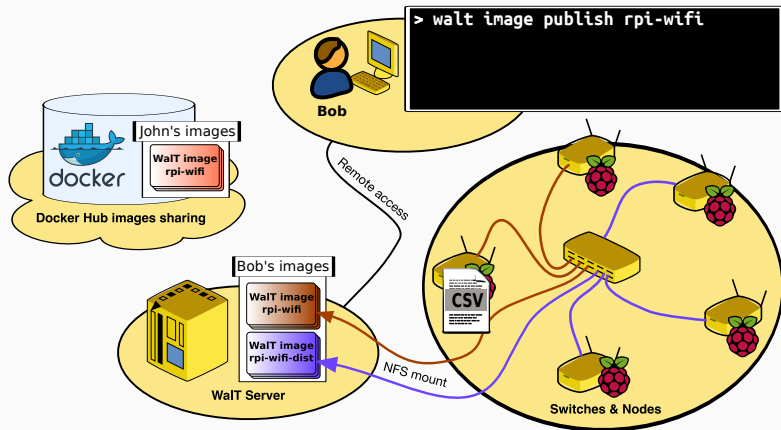
# Sample use case



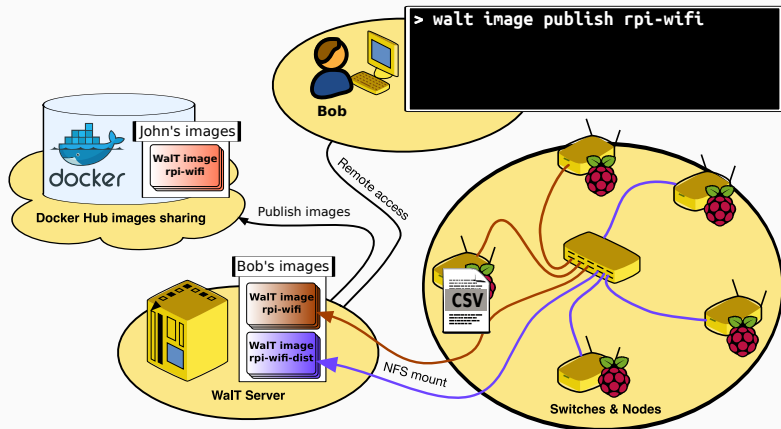
# Sample use case



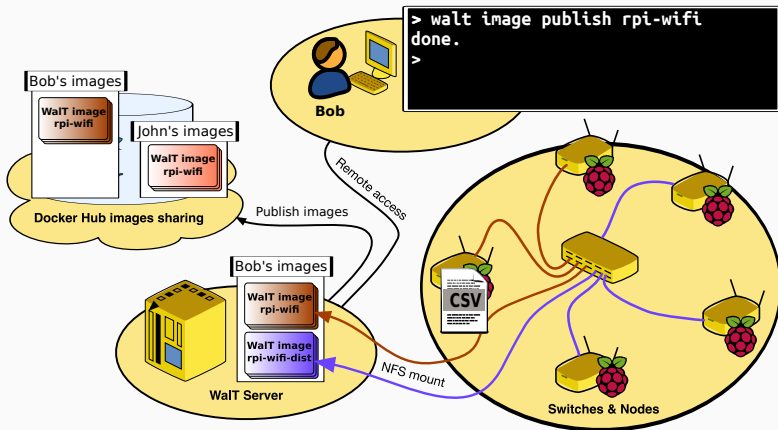
# Sample use case



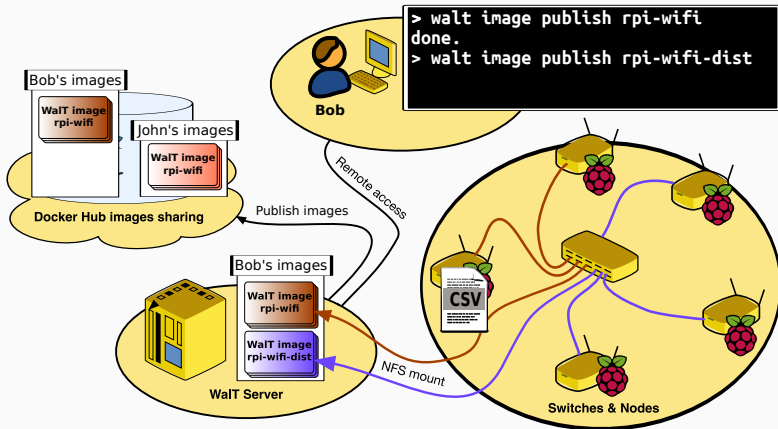
# Sample use case



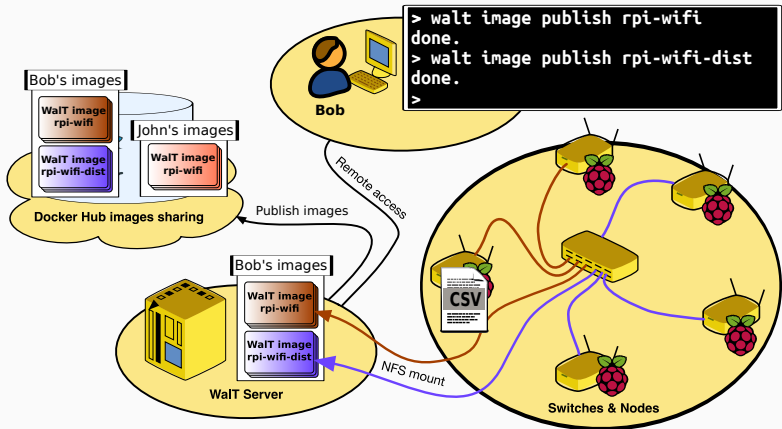
# Sample use case



# Sample use case



# Sample use case



The image shows the words "The End" written in a highly stylized, cursive script font. The letters are white with a thick black outline, giving them a three-dimensional, embossed appearance. The text is centered on a solid, medium-gray rectangular background. The overall aesthetic is classic and elegant, reminiscent of vintage movie titles.



# Agenda

Motivation

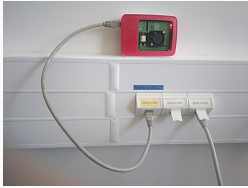
Sample use case

**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**<sup>1</sup> 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)

<sup>1</sup>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)**

## 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 version 5 now has experimental support for distant nodes (cf. LoRa, SigFox).
- These nodes reach WalT network through a VPN  
(+ a level of virtualization to implement network boot procedure)

# The *reproducible platform* concept

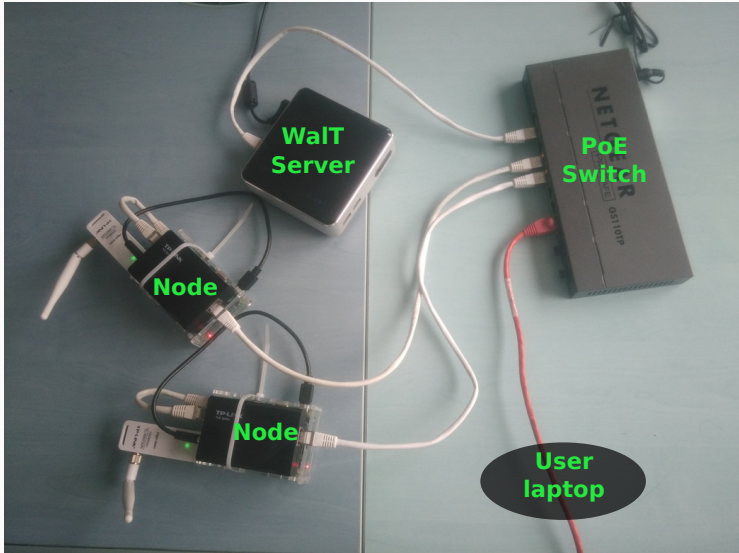
## WalT is a *reproducible platform*<sup>1</sup>

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

---

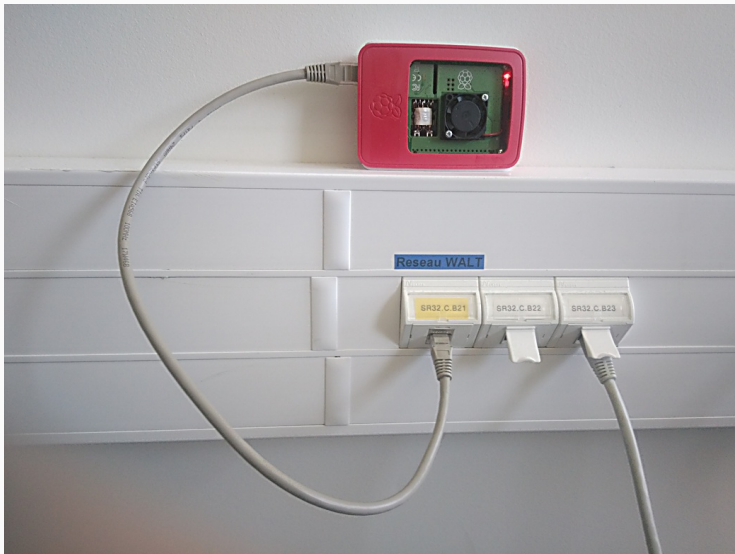
<sup>1</sup>P. Brunisholz, E. Duple, F. Rousseau, and A. Duda, “WalT: A Reproducible Testbed for Reproducible Network Experiments”, in Proc. of the International Workshop on Computer and Networking Experimental Research Using Testbeds, 2016.

# Sample mobile setup



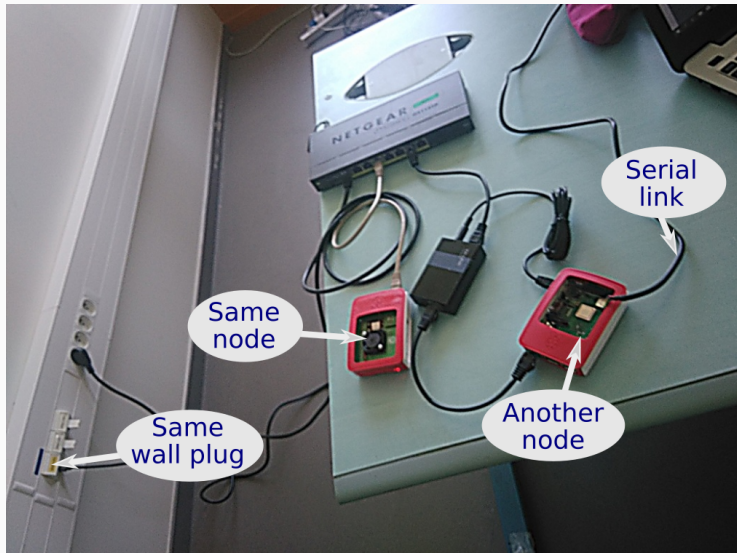
Minimal setup for an ANR Datatweet meeting

## WalT versatility shown in two slides (1)



A node of the WalT network in IMAG building

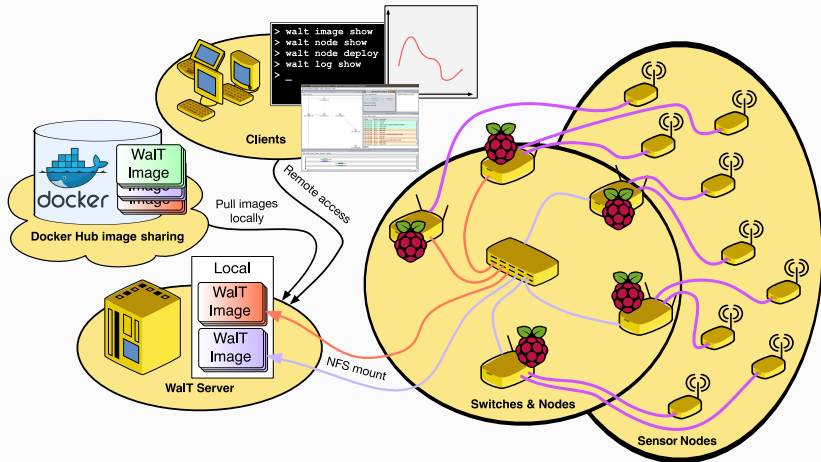
## WaT versatility shown in two slides (2)



Reusing the same wall plug for temporary low-level debugging

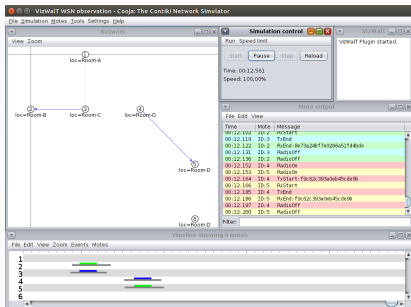


# Architecture for WSNs



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



- 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

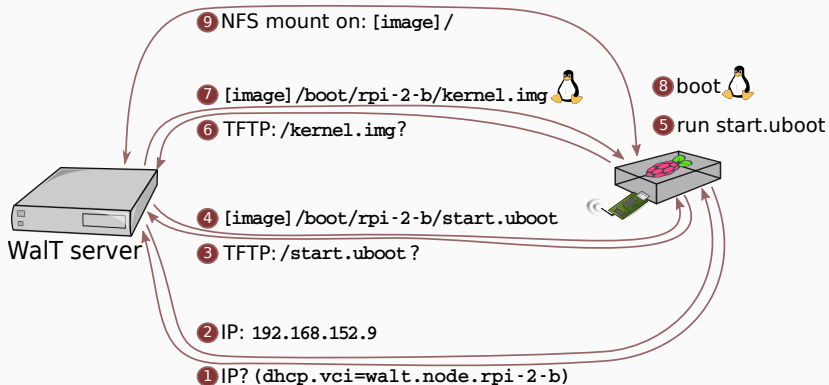
Sample use case

Architecture

WaIT under the hood

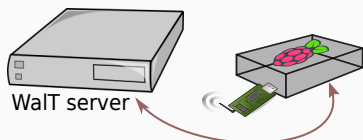
News, On-going work and Perspectives

# Node boot procedure



- **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 transferred as a whole to the node.**

## Node boot procedure (continued)



9 NFS mount on: `[image] /`

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

11 Start waiT-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

## 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**)



## 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

- WALT v5 is available since september 8, 2020
- 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
- The deployment in IMAG building at LIG now has 12 more 3B+ nodes on the ground floor
- We connected our WalT platform at IMAG to our data management platform (<https://sakura-platform.liglab.fr/>)

## 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<sup>1</sup>
- Possible new walt deployments (e.g. at LAAS (Toulouse), at Bu-Ali Sina University (Iran))
- Several planned software improvements<sup>2</sup>

---

<sup>1</sup>with G. Enderlé, É. Jullien of Grenoble University

<sup>2</sup><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](mailto:walt-contact@univ-grenoble-alpes.fr)

# How to contribute

- Setup<sup>1</sup> 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

---

<sup>1</sup>Contact us ([walt-contact@univ-grenoble-alpes.fr](mailto:walt-contact@univ-grenoble-alpes.fr)), we will assist you.

# WaT credits

- Initial funding
  - Univ. Grenoble Alpes, Grenoble INP / UJF, AGIR WaT (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, WaT images design, development, testing & documentation
  - Bastien Faure, Pierre-Henry Frohring, Jorane Congio-Hollard, Cosmin Nichifor, core WaT
  - Jorge Luis Baranguan Castro, VizWaT
  - 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?