

WalT: Reproducible Experiments on Networks and Distributed Systems

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

SLICES-FR summer school, Lyon, July 2025

Agenda

Introduction

Sample use case

Architecture

Client tools

News, On-going work

What is WALT software?

Walt software, in short

- Purposes:
 1. Build and Manage a **network testbed**
 2. Ease and automate experiments.
- We started to design WALT in **2012**.

Walt Uses¹

- in the **Industry**,



- for **Research**,



- for **Education**.

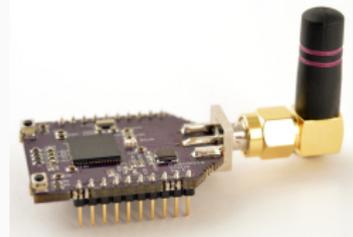


¹WALT is free software, so there may be other uses we are not aware of.

What about alternate platforms?

Limits of existing platforms

1. **Large scale public testbeds** (e.g. IoT-Lab):
 - Limited hardware choice
 - Static environment & network topology
 - No physical access (for debugging)
 - Booking system (limited experiment time)
2. **Handmade platform** “on your own desk”:
 - Does not scale
 - Is hardly reproducible



To overcome the limits of each category,
WalT was designed **halfway between these 2 categories**.

WalT project goals

We provide:

- Open source code
- Documentation: <https://walt.readthedocs.io>
- Support: walt-users@univ-grenoble-alpes.fr

We do **not** provide:

- A publicly available platform
(each team, lab, etc. should set up its own platform)

The *reproducible platform* concept

WaT is a *reproducible platform*¹

- Any researcher can build its own WaT platform
- ... or several platforms of various scales (e.g., one for debugging, one for experiments)
- One may **challenge reproducibility** by running an experiment on several walt platforms.

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

Agenda

Introduction

Sample use case

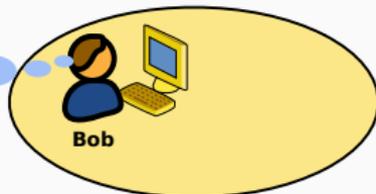
Architecture

Client tools

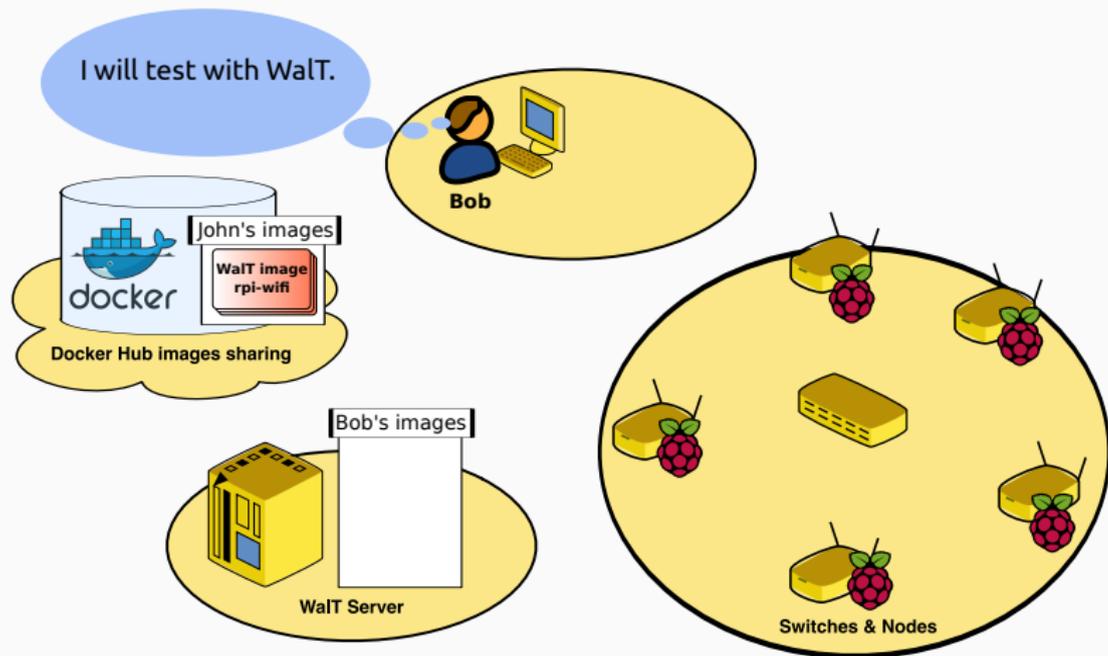
News, On-going work

Sample use case

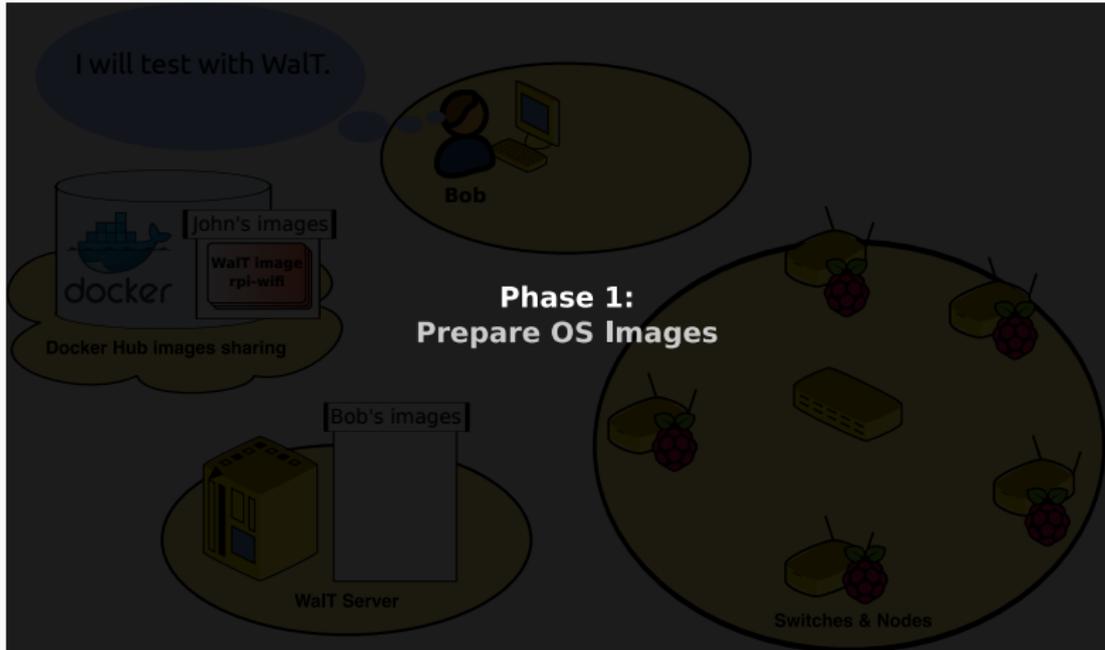
Is this still working
with interference?



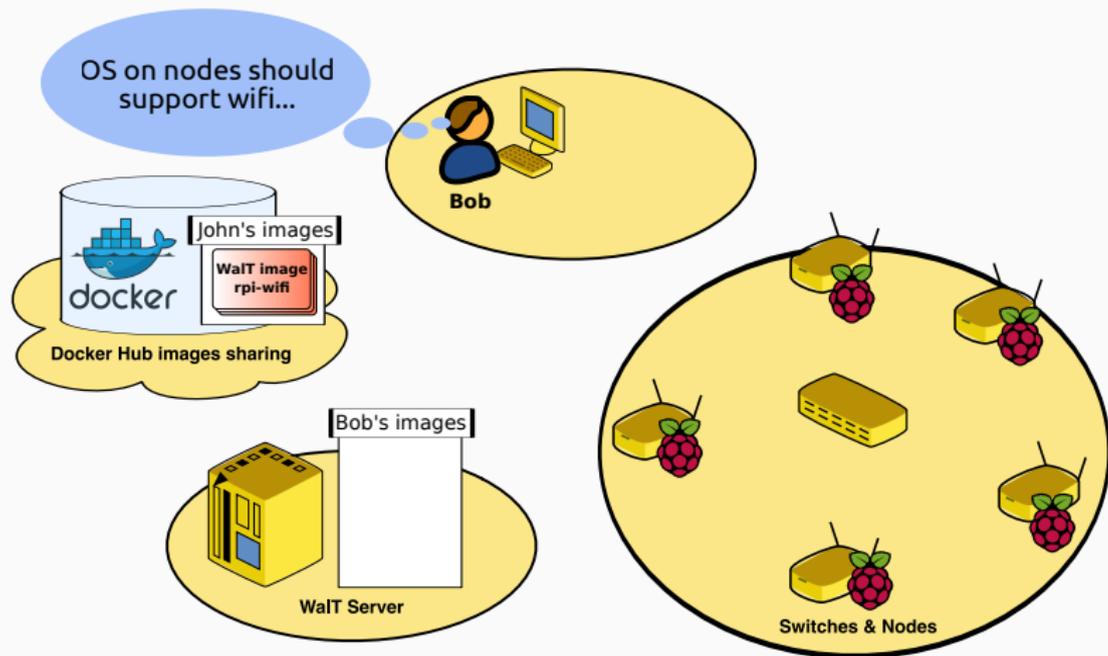
Sample use case



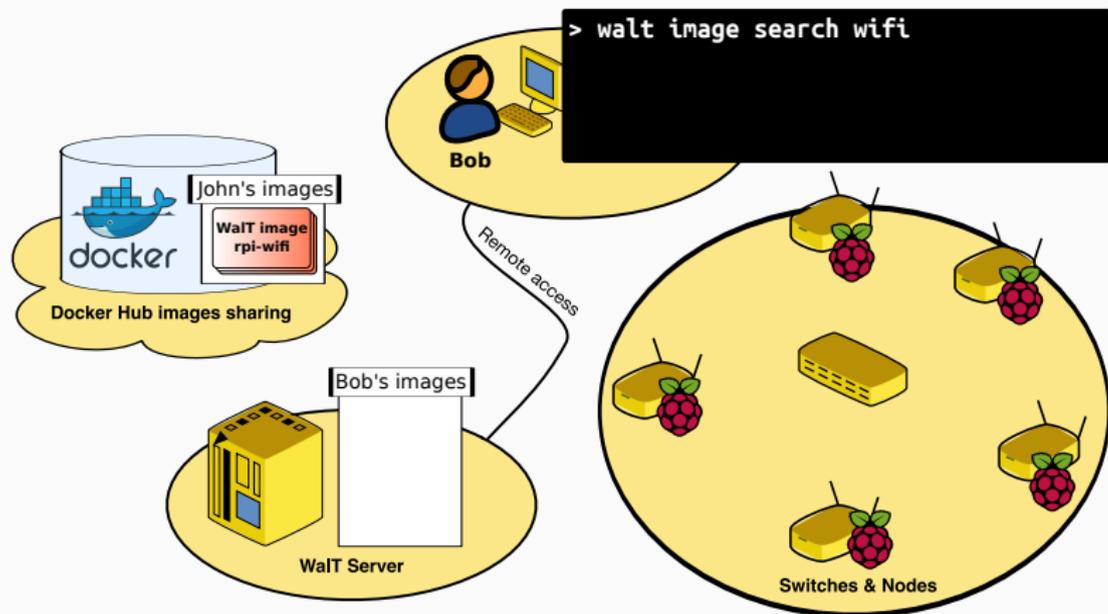
Sample use case



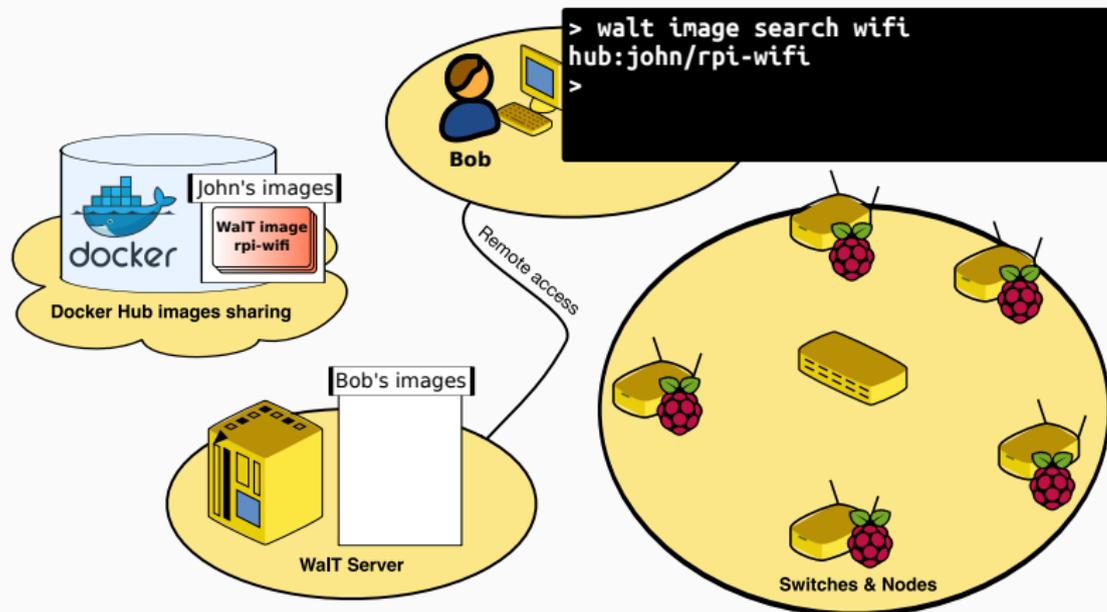
Sample use case



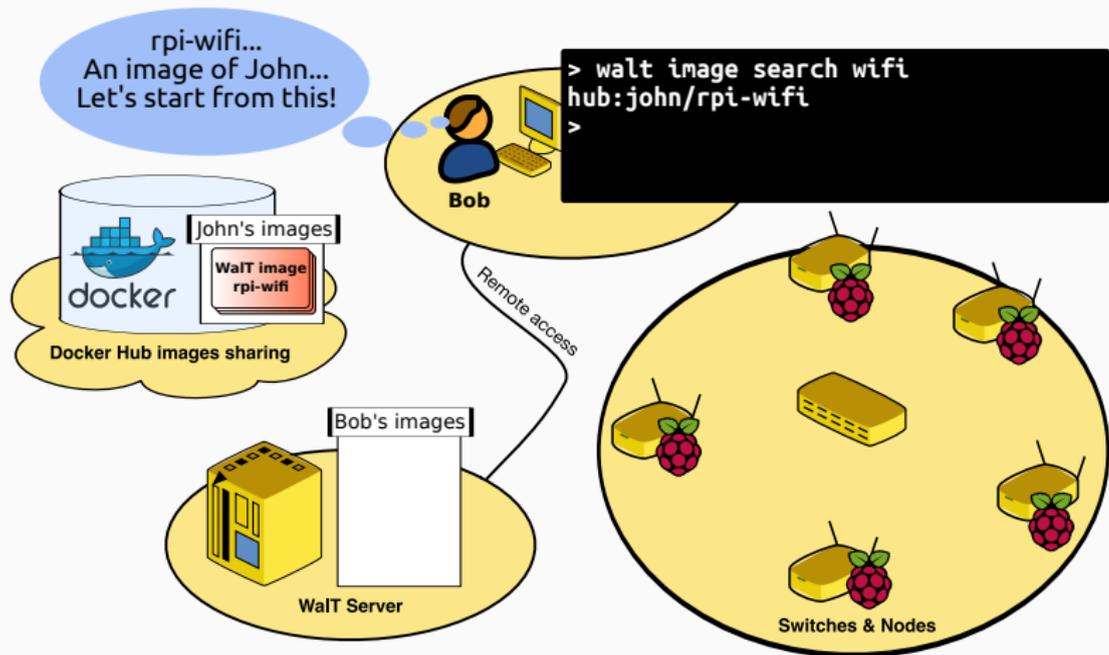
Sample use case



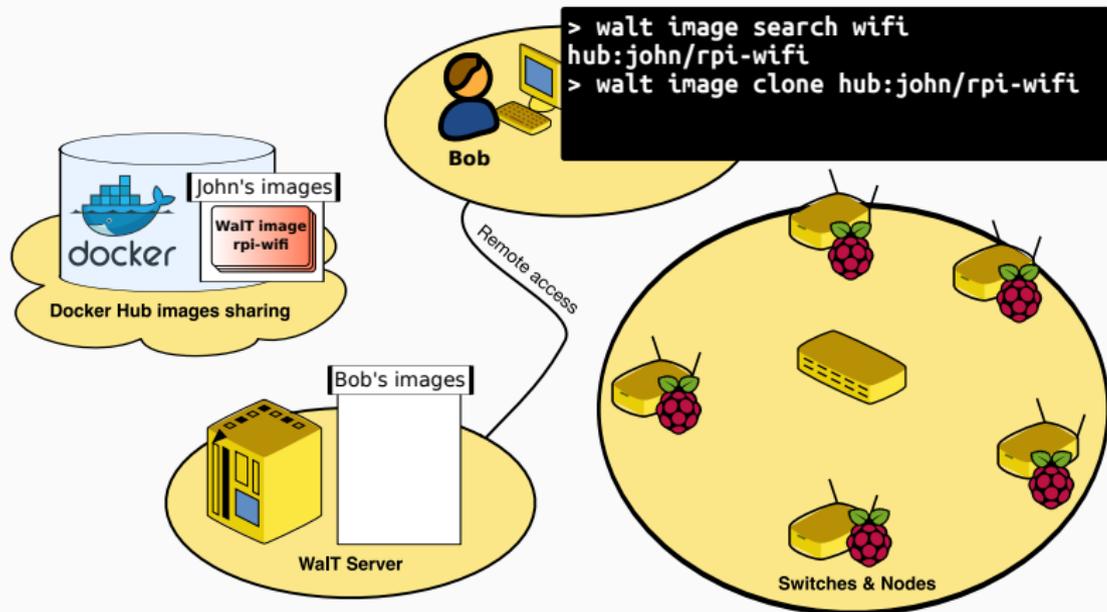
Sample use case



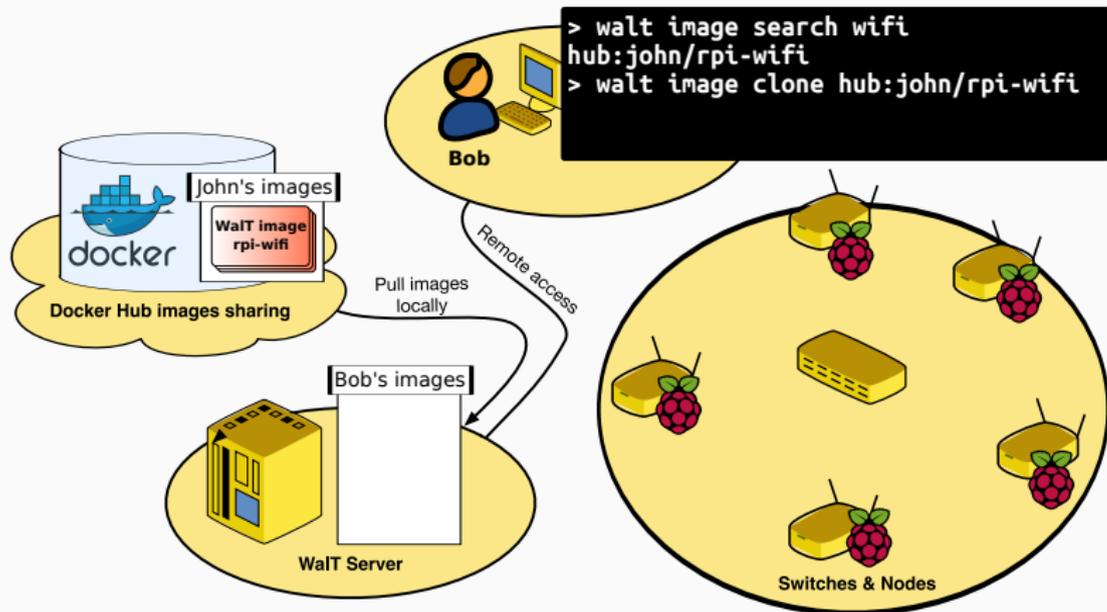
Sample use case



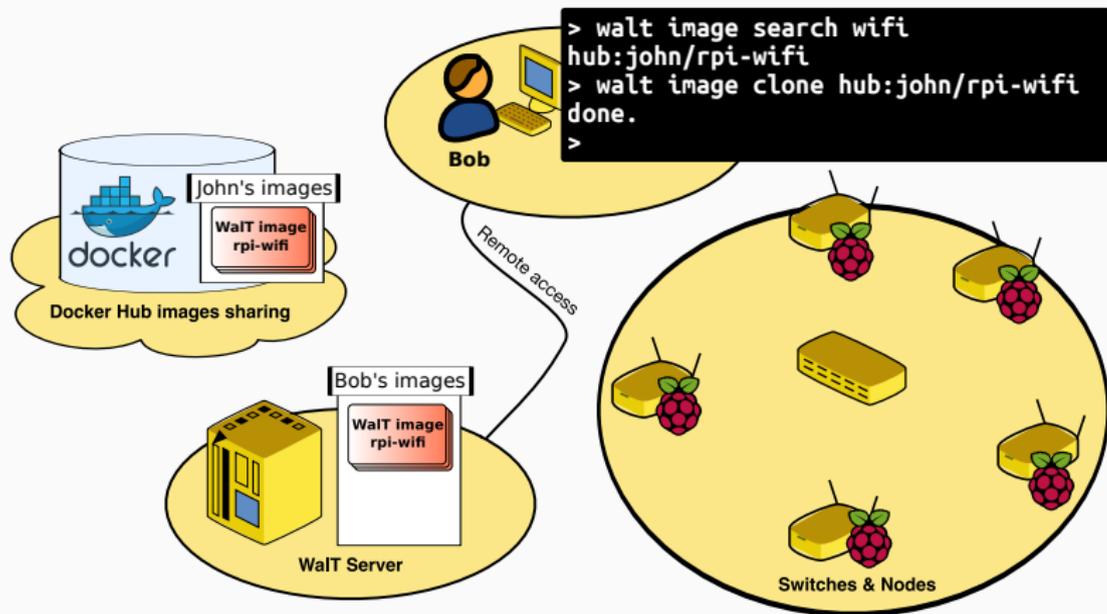
Sample use case



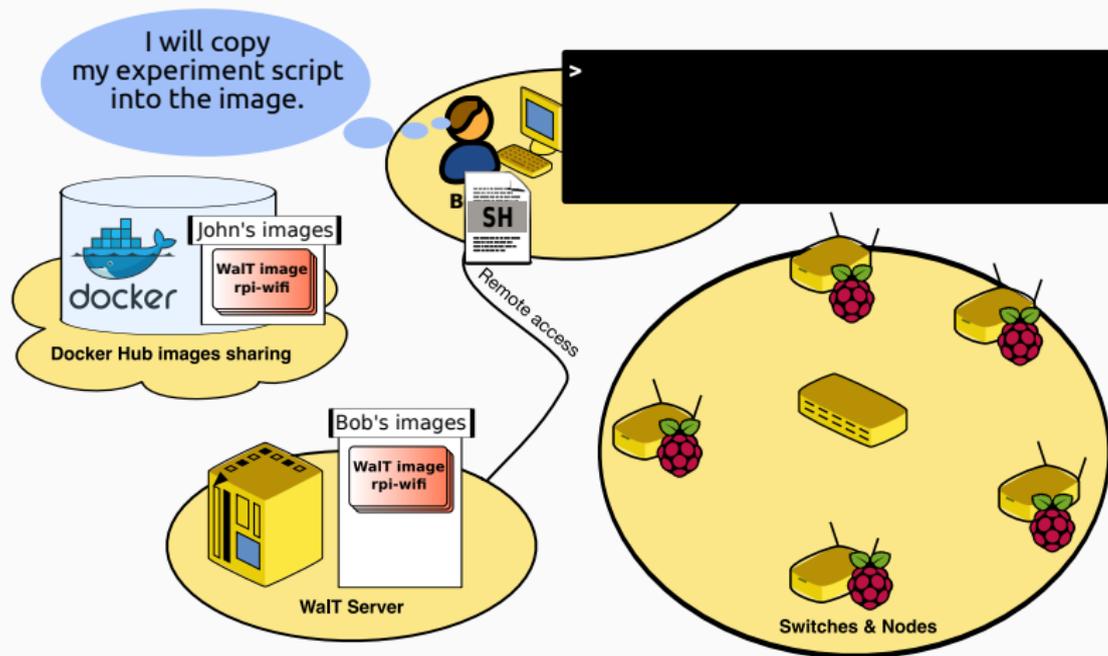
Sample use case



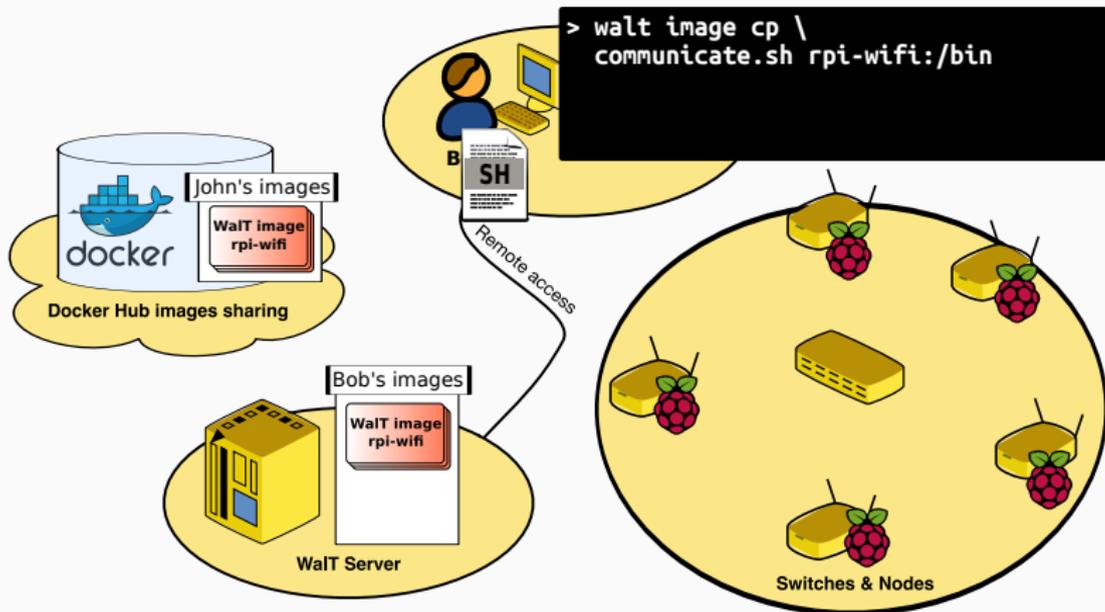
Sample use case



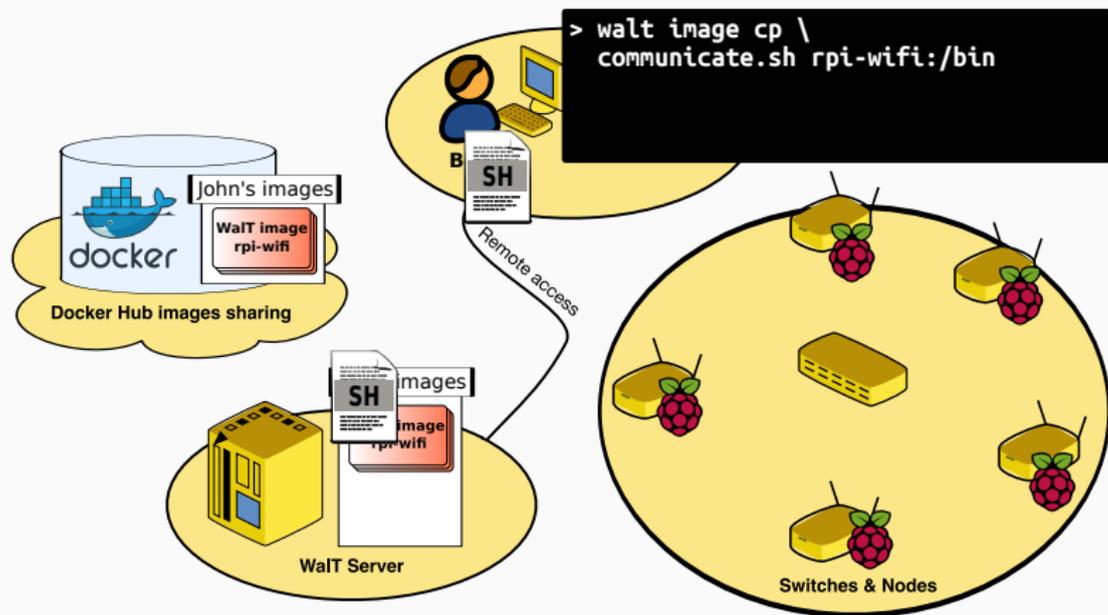
Sample use case



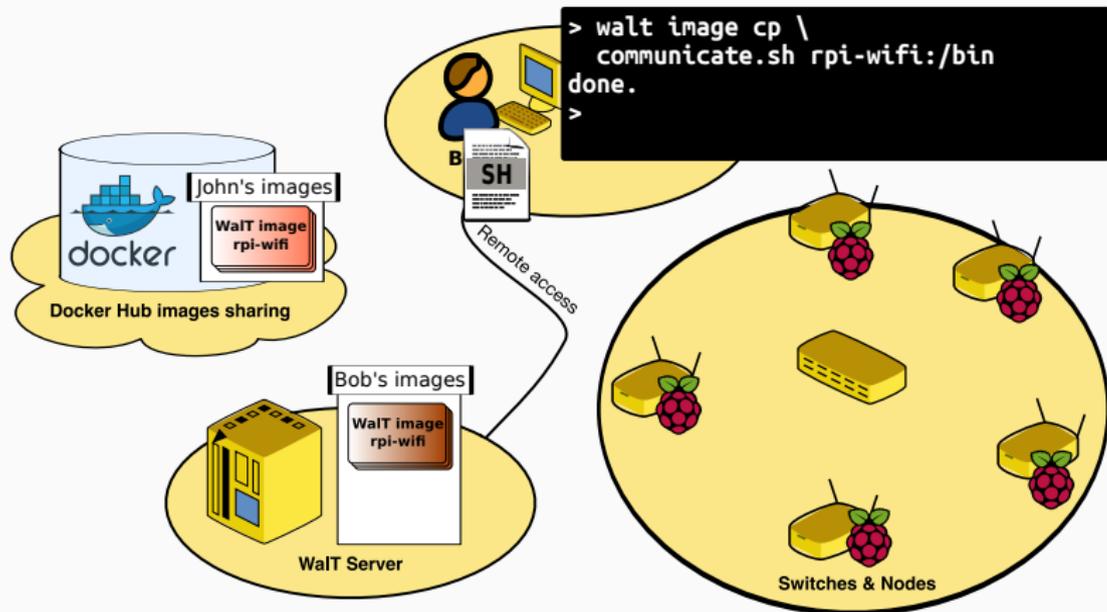
Sample use case



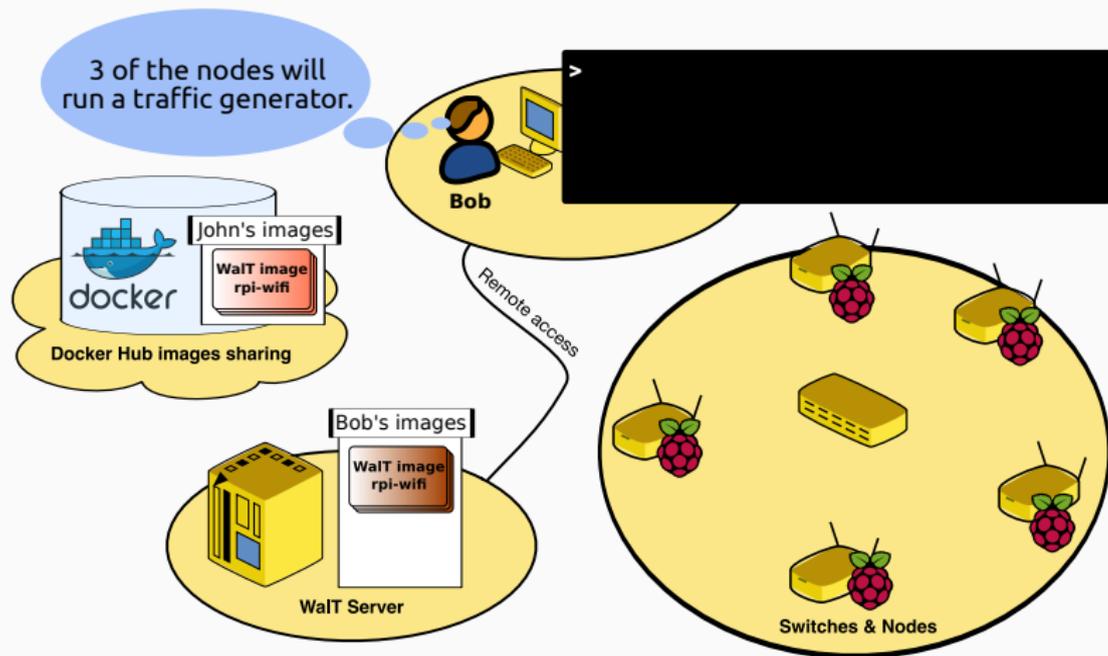
Sample use case



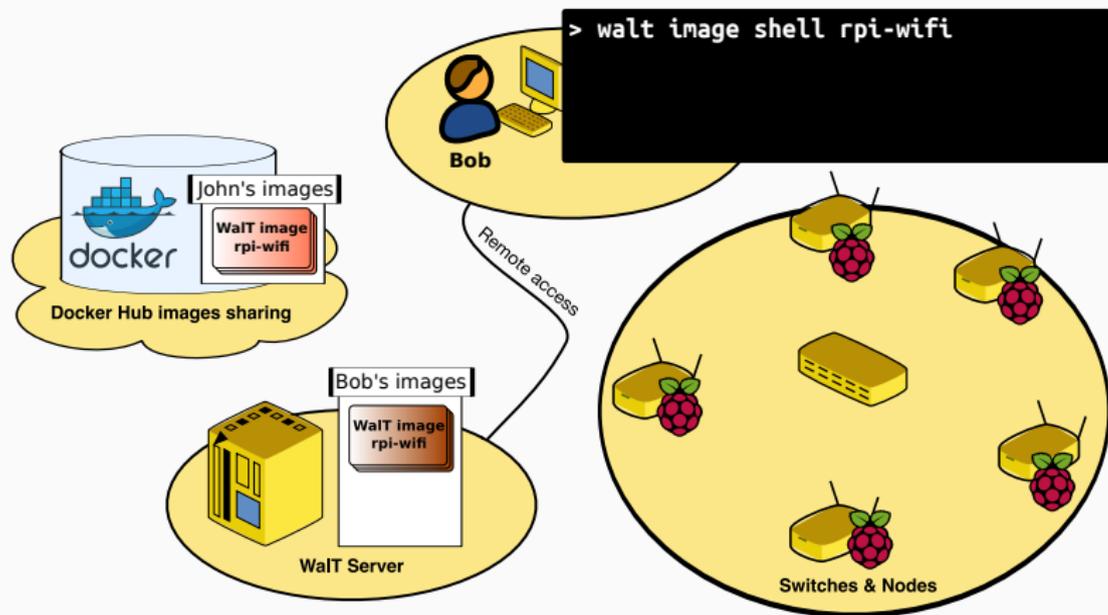
Sample use case



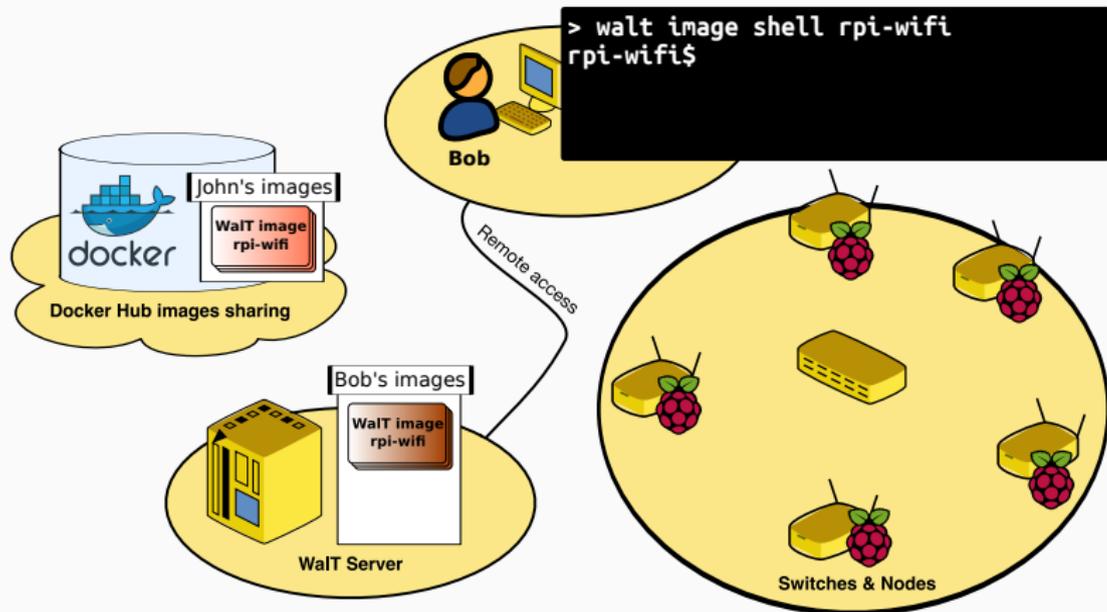
Sample use case



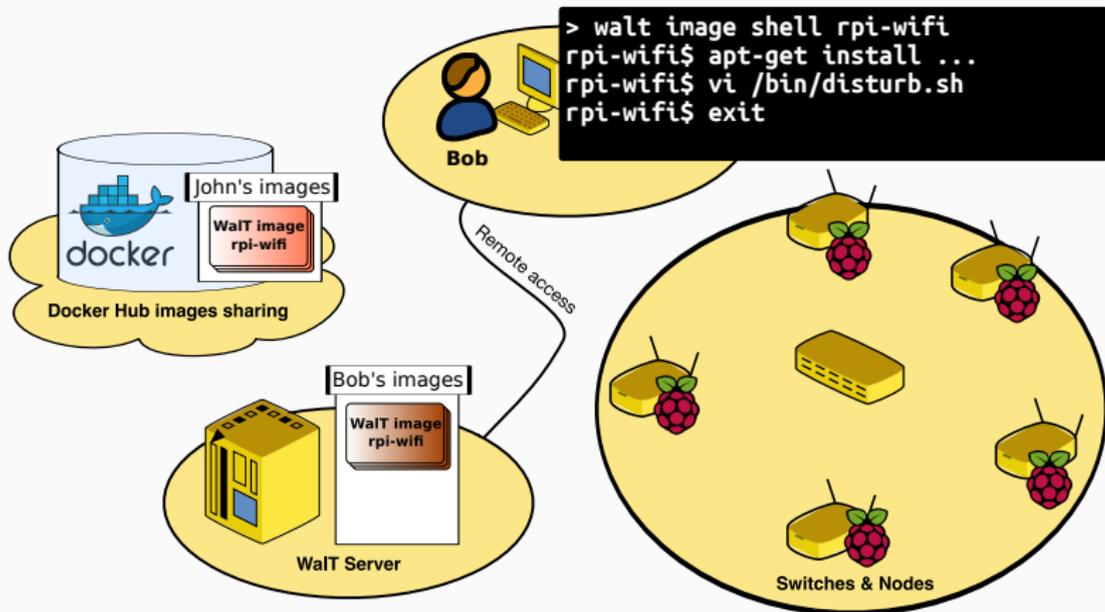
Sample use case



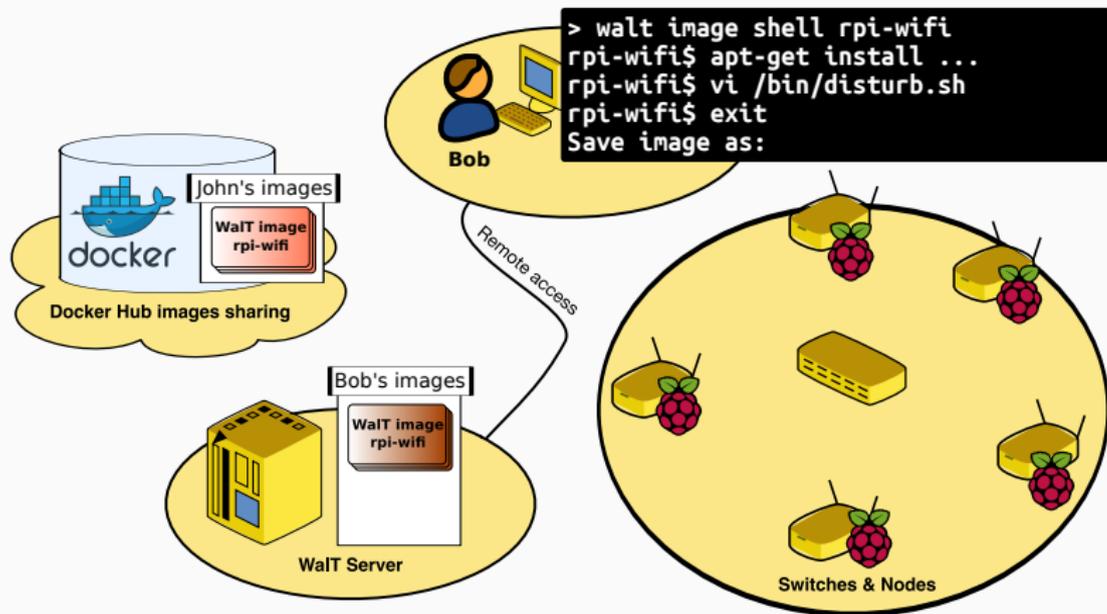
Sample use case



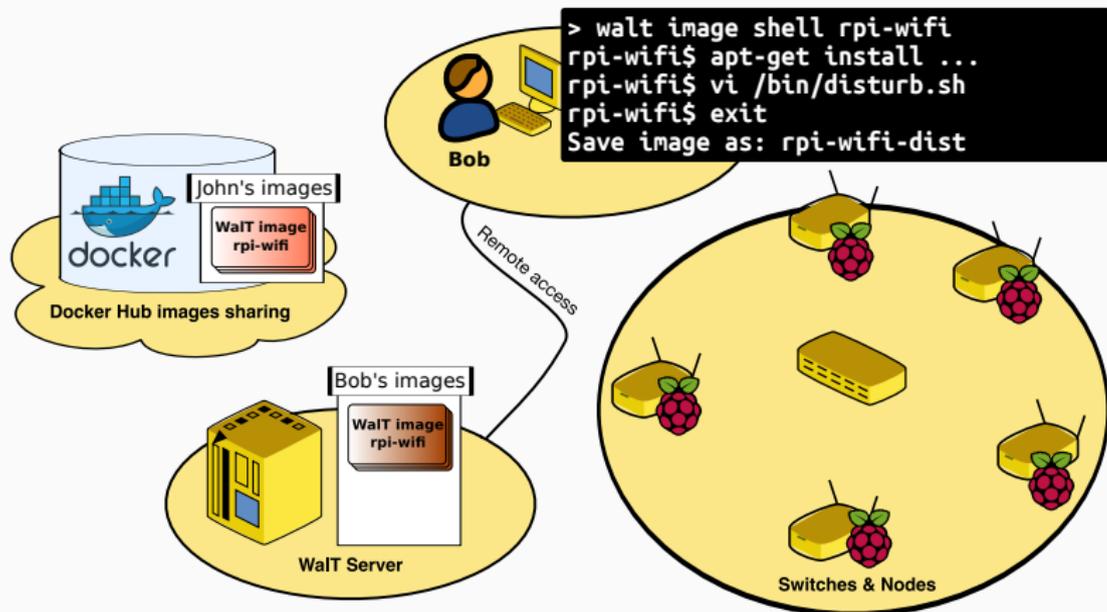
Sample use case



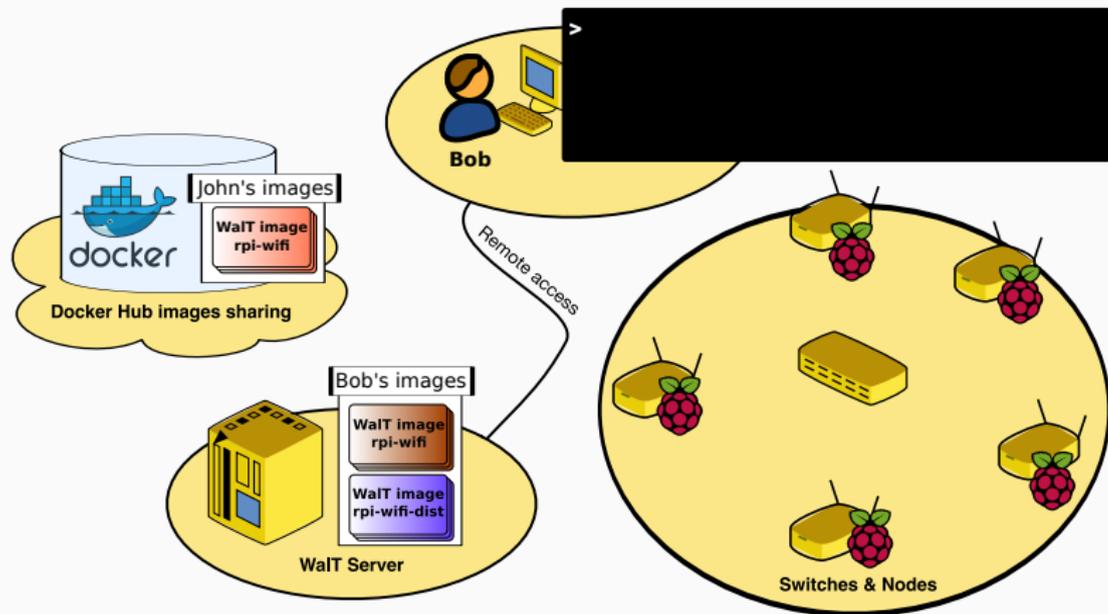
Sample use case



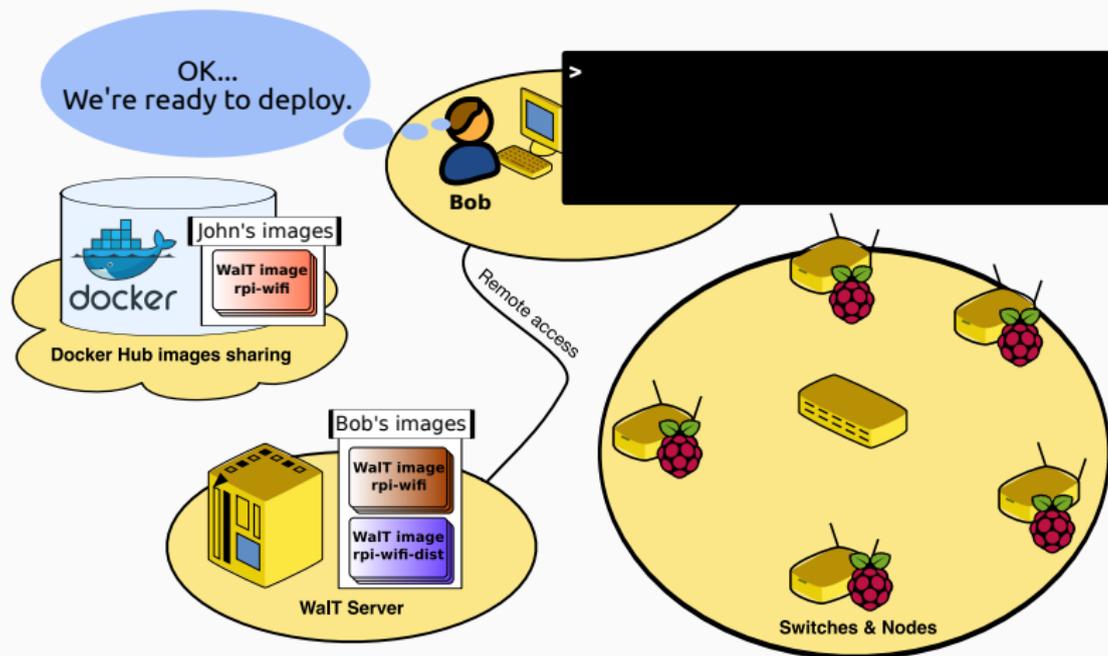
Sample use case



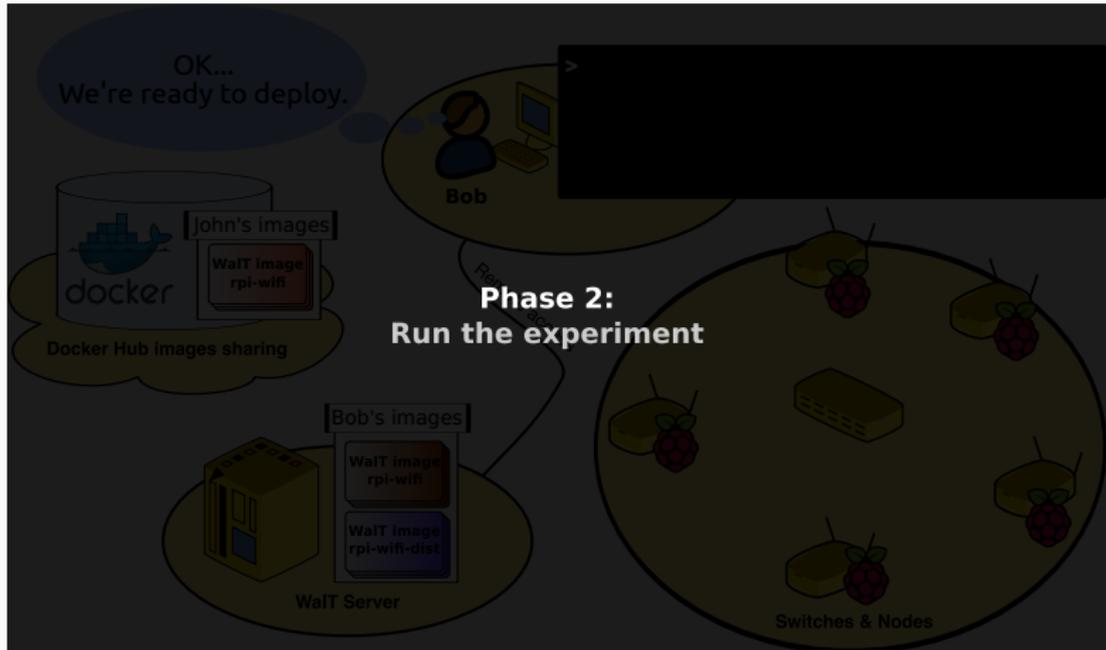
Sample use case



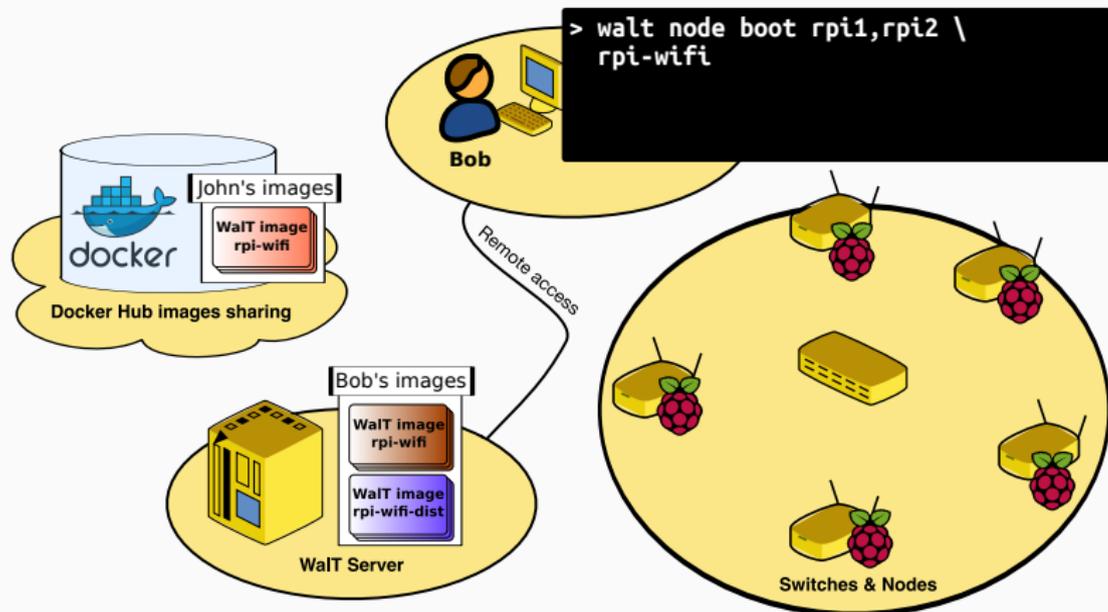
Sample use case



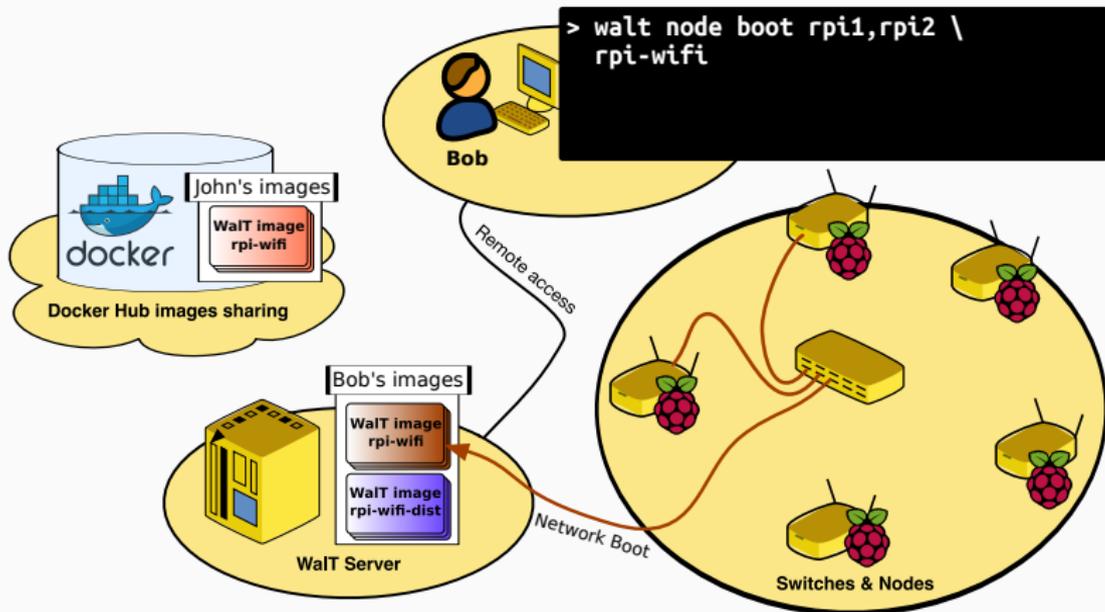
Sample use case



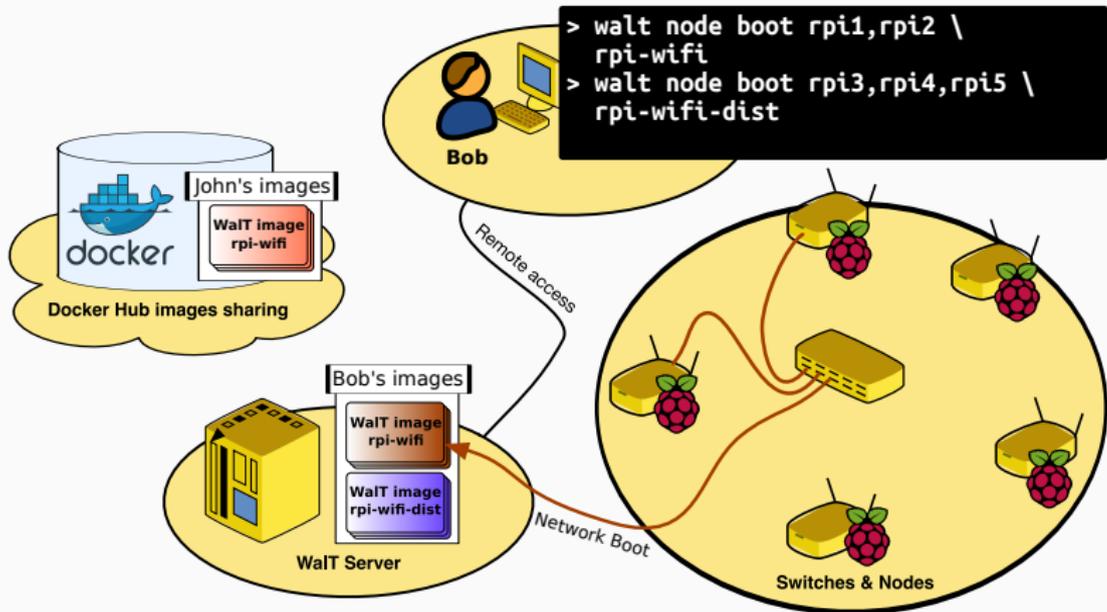
Sample use case



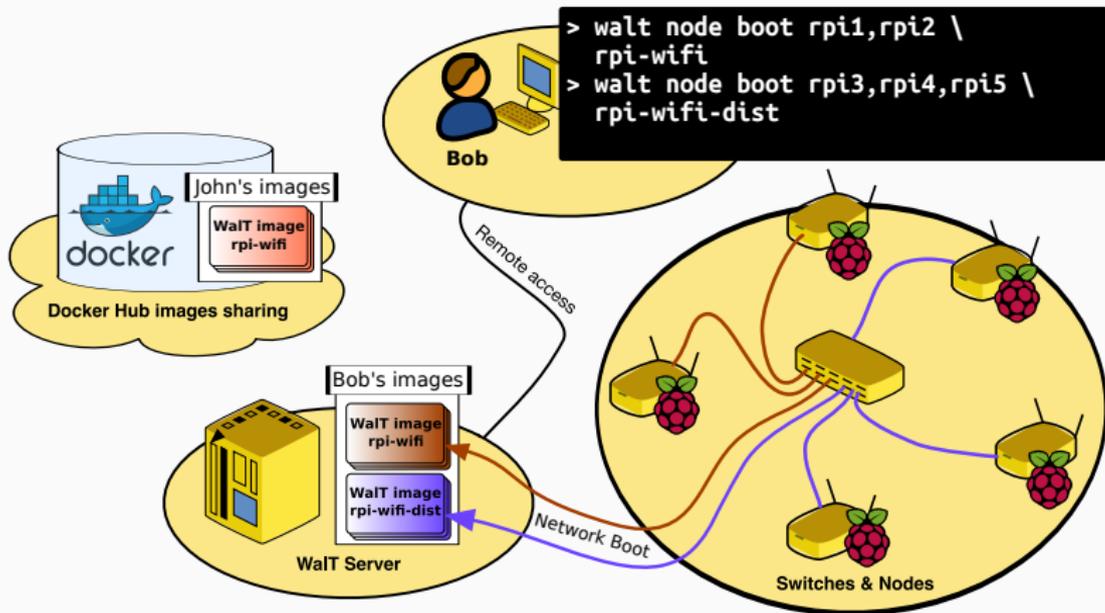
Sample use case



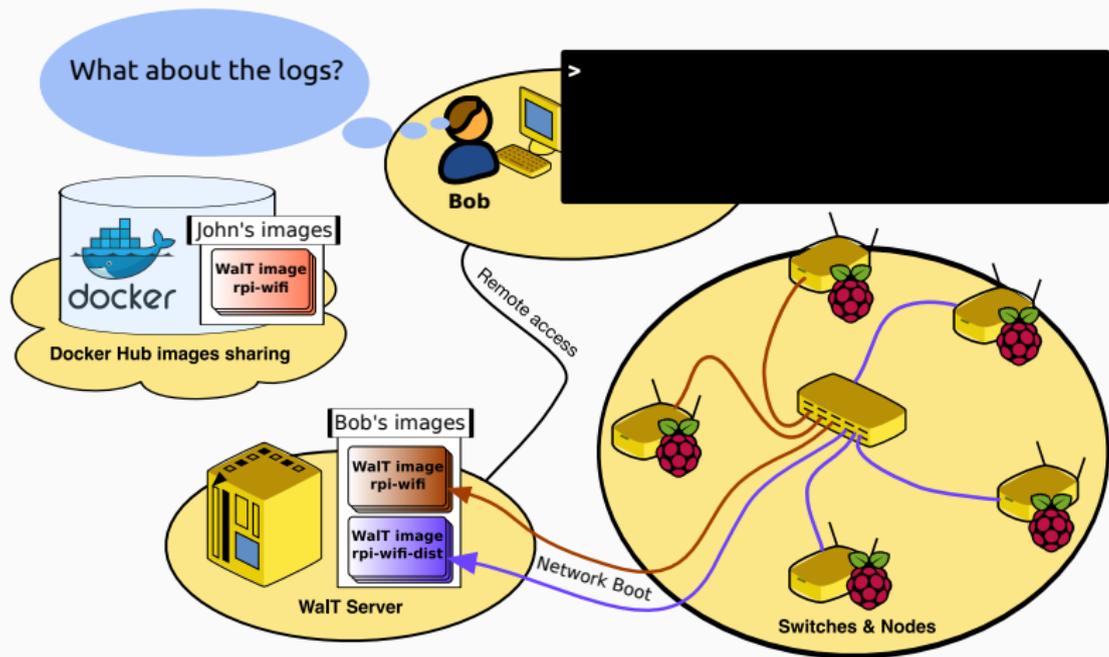
Sample use case



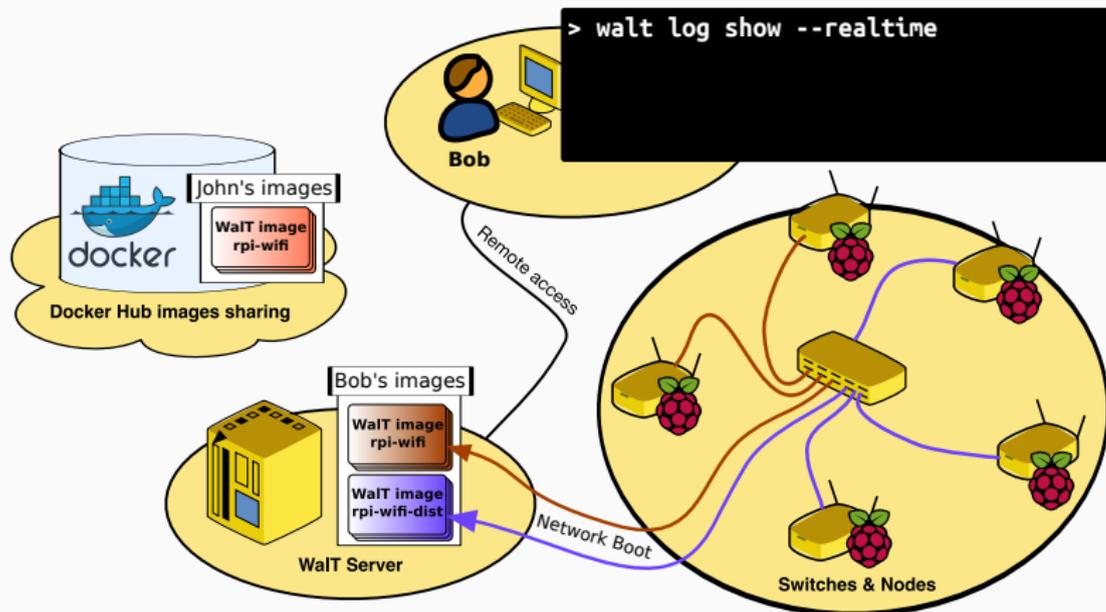
Sample use case



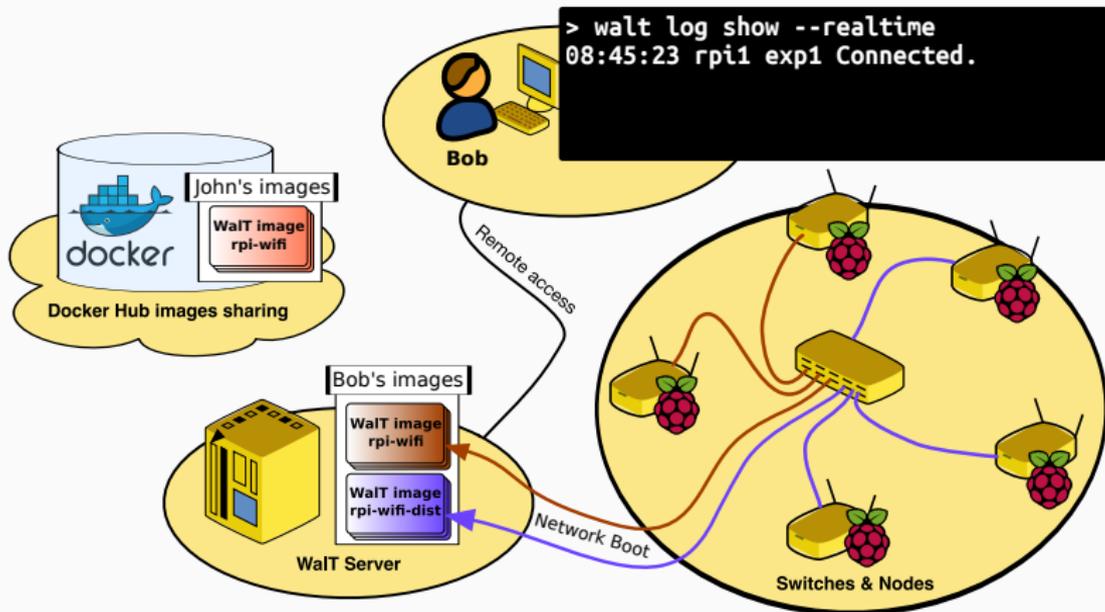
Sample use case



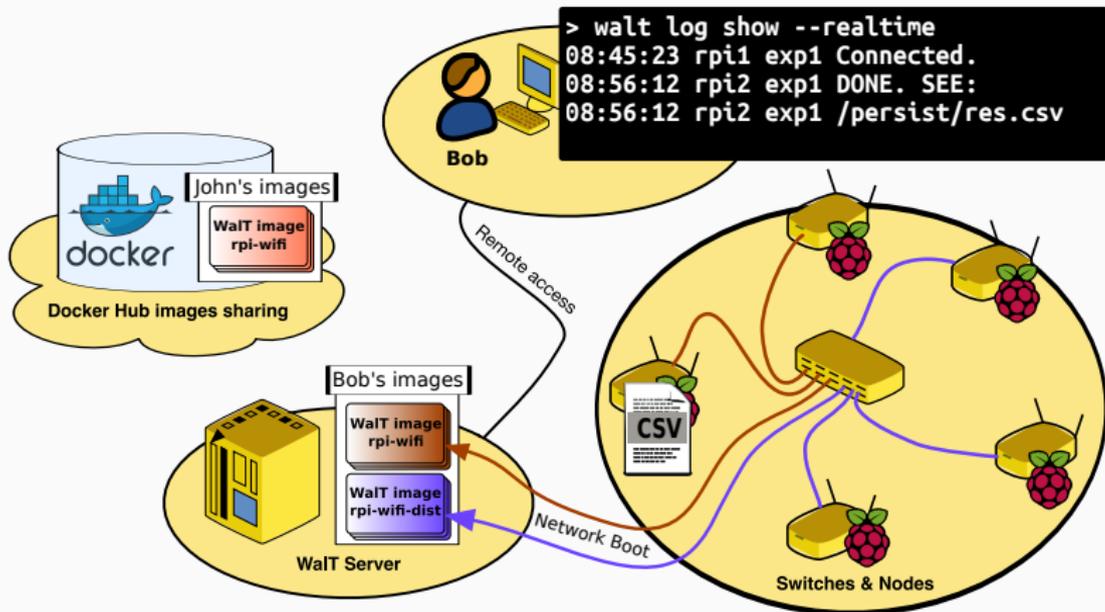
Sample use case



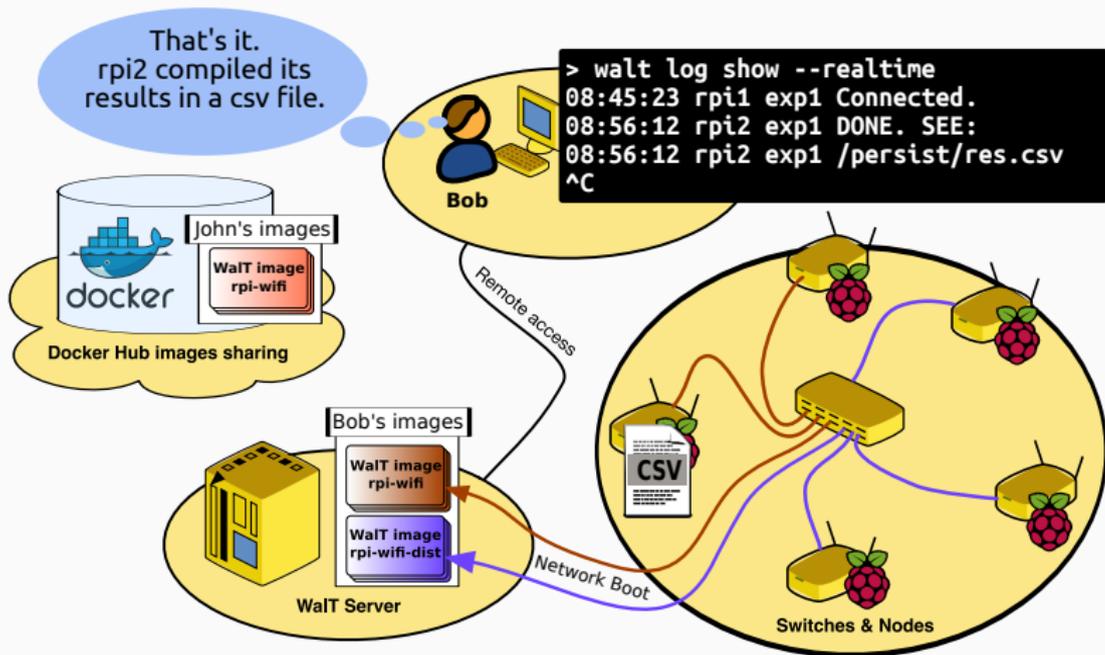
Sample use case



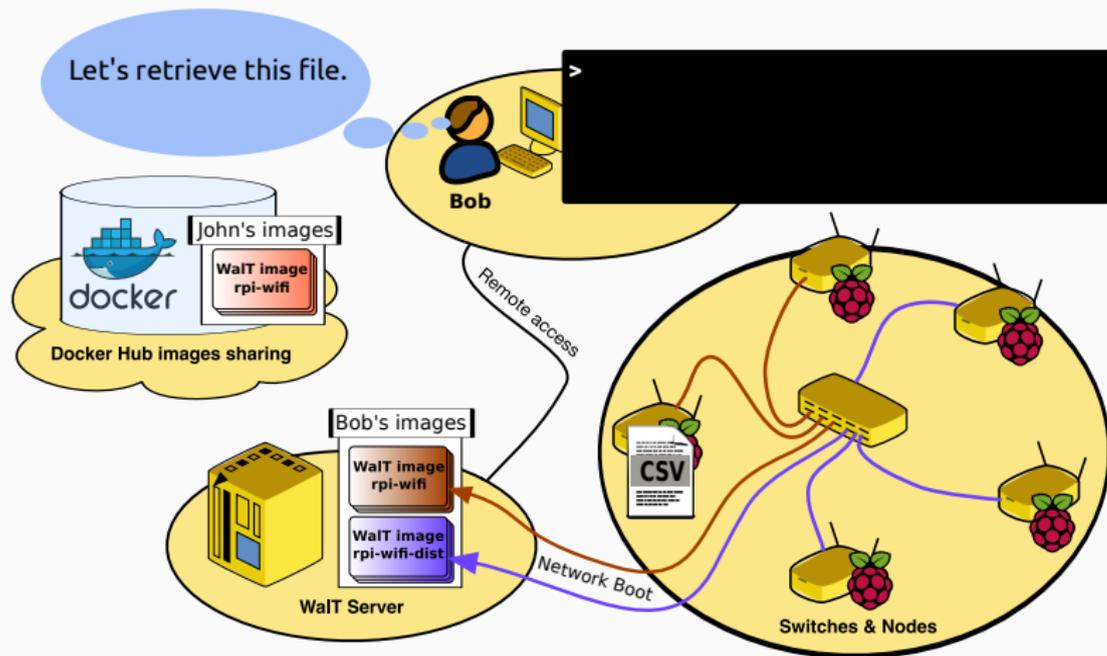
Sample use case



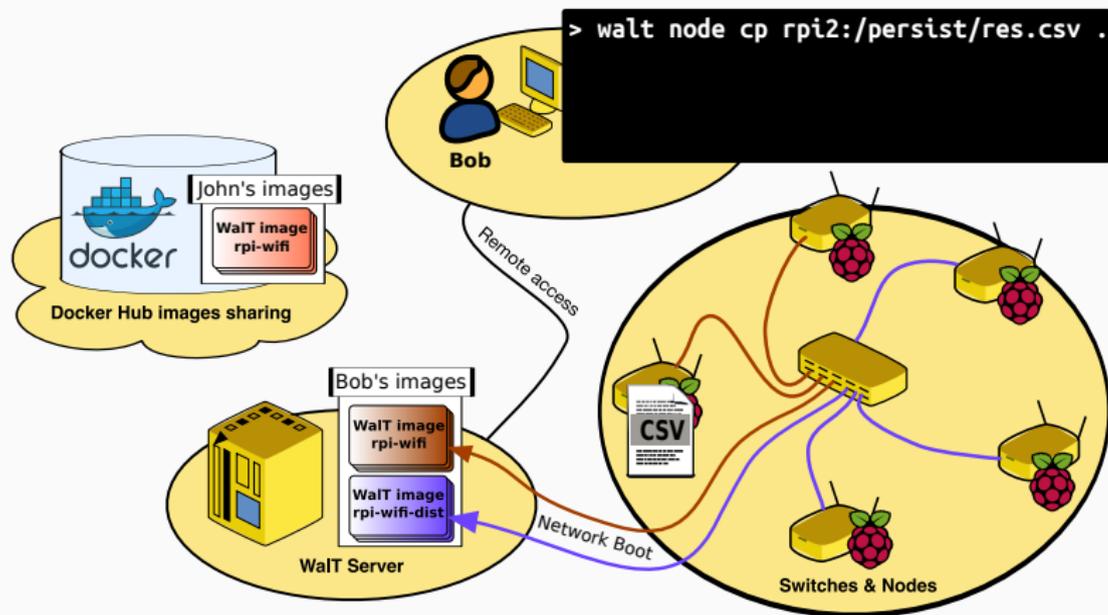
Sample use case



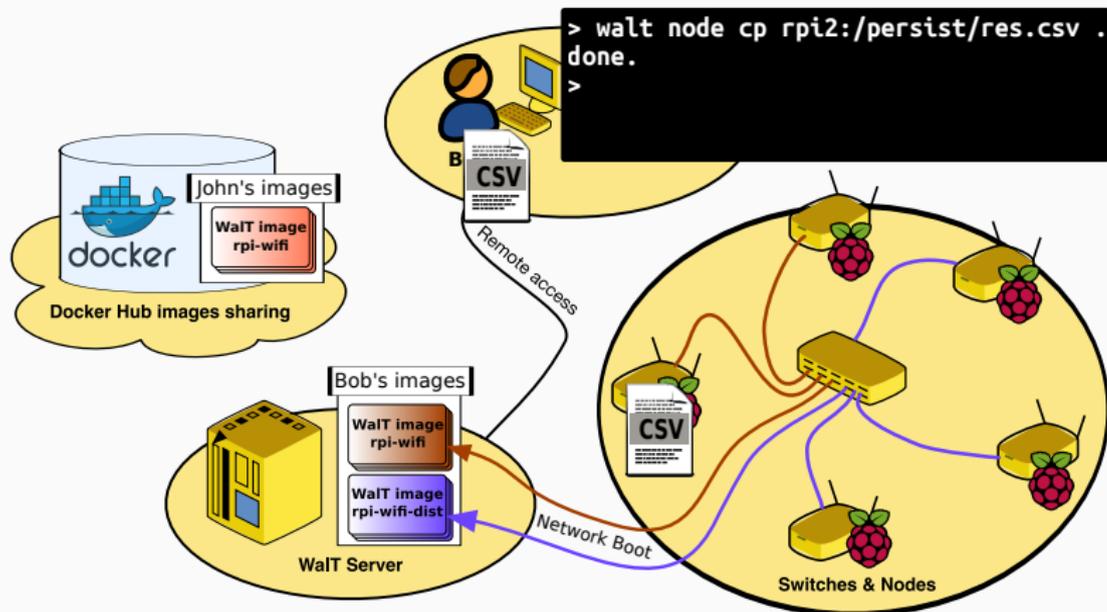
Sample use case



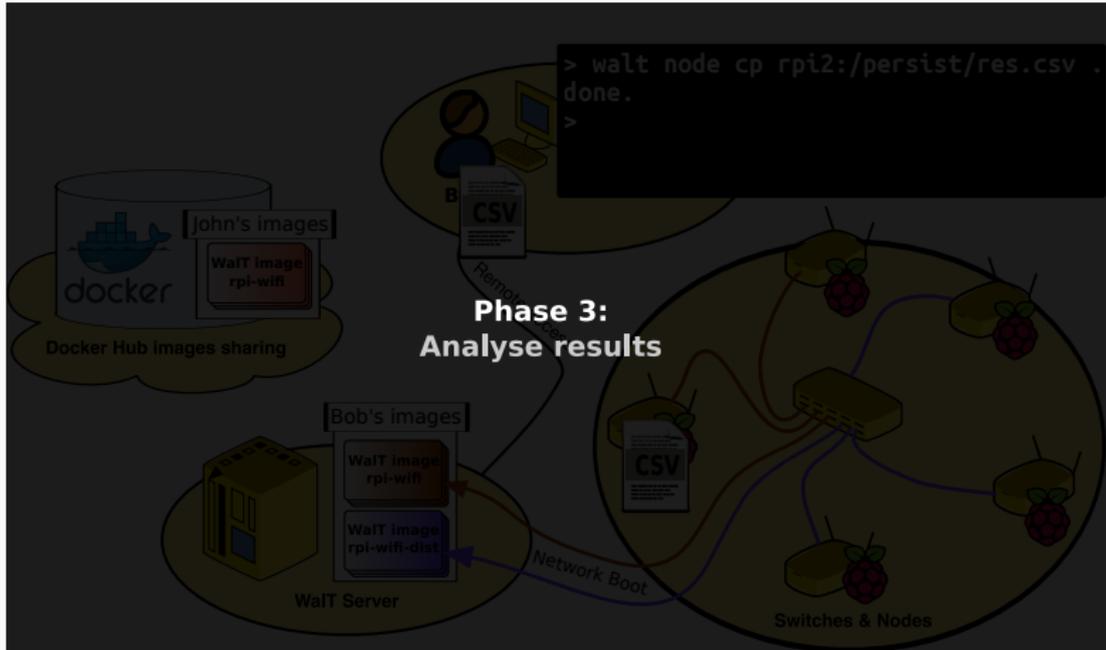
Sample use case



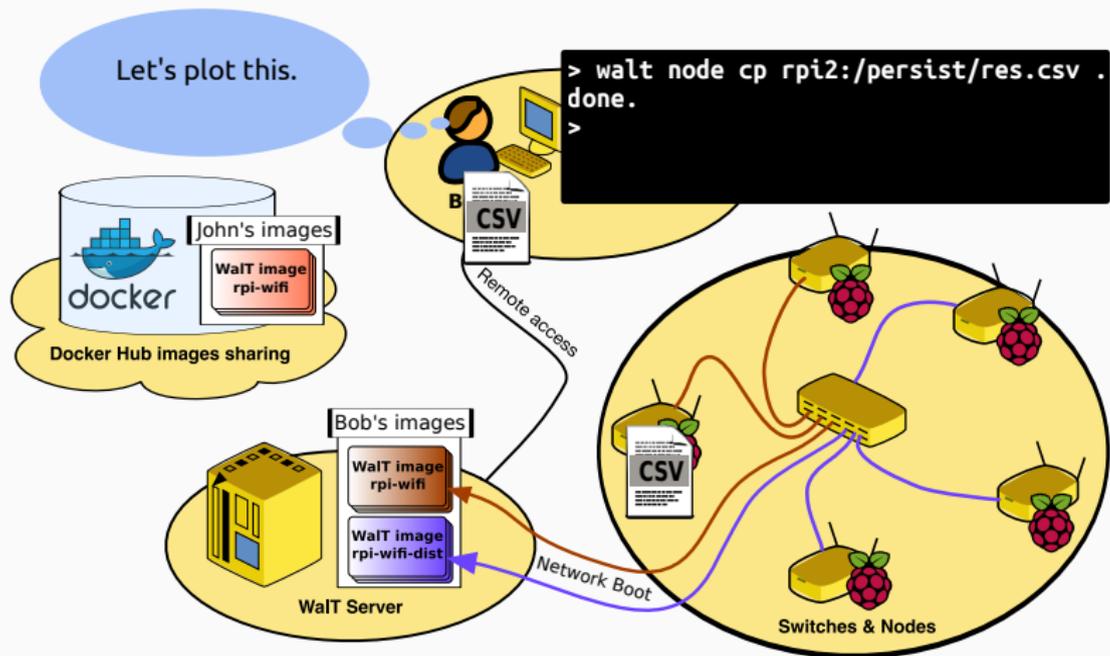
Sample use case



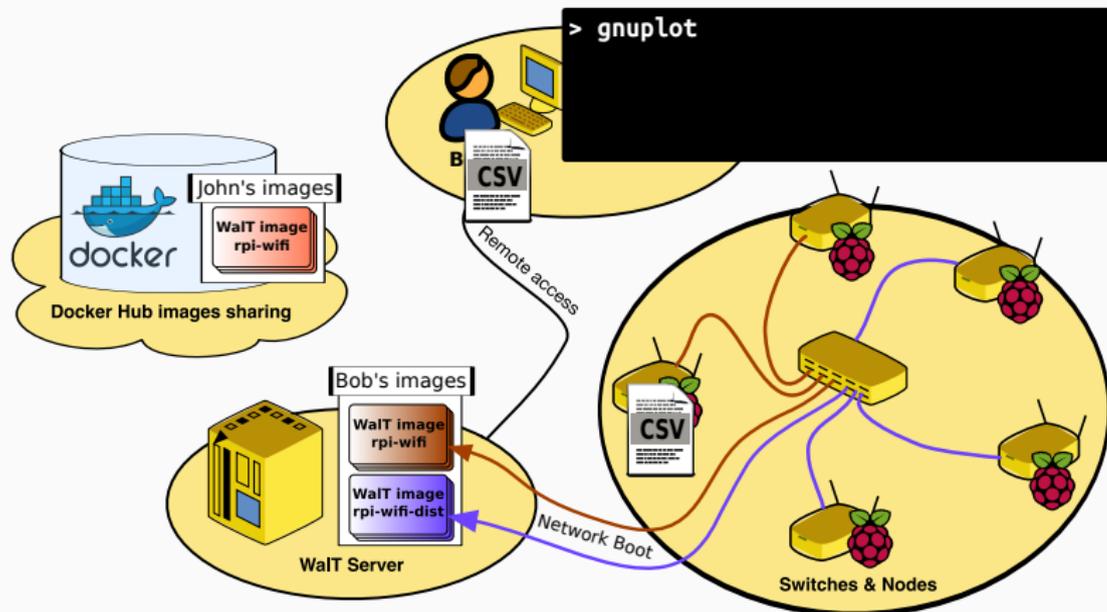
Sample use case



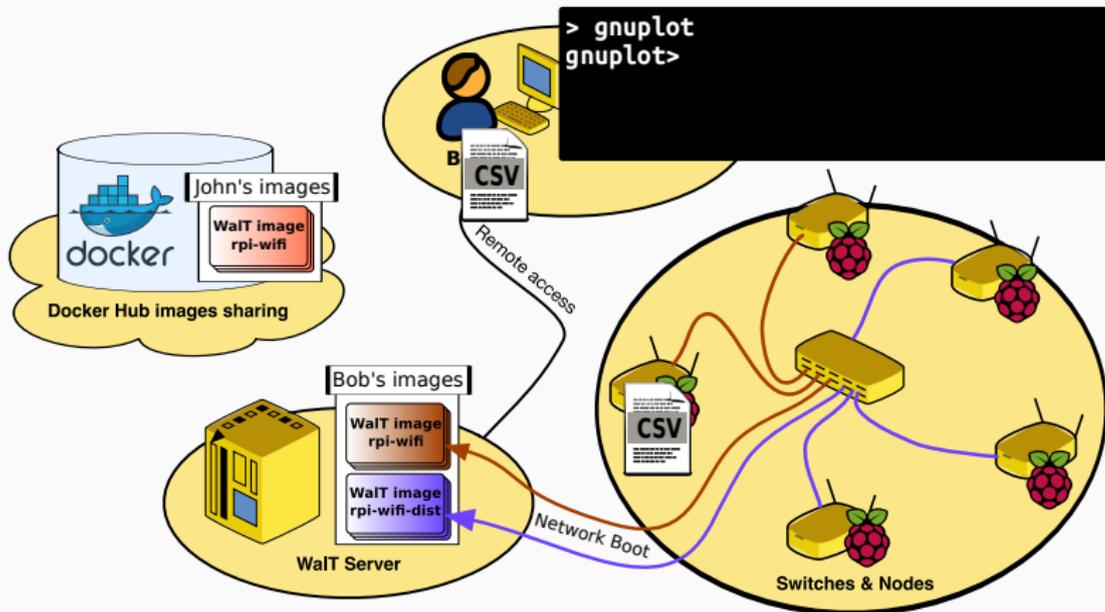
Sample use case



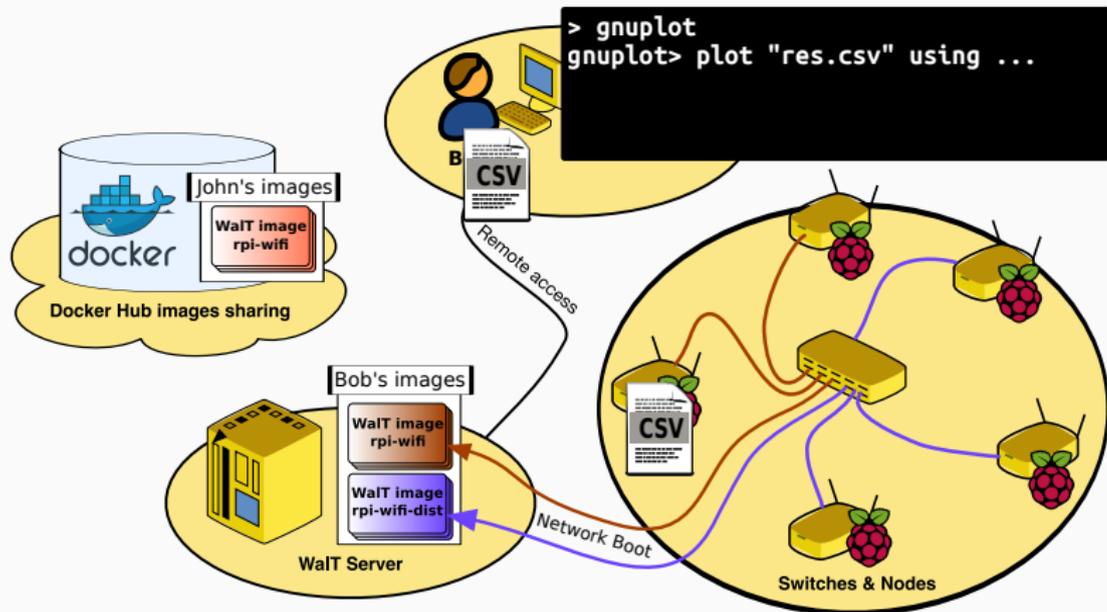
Sample use case



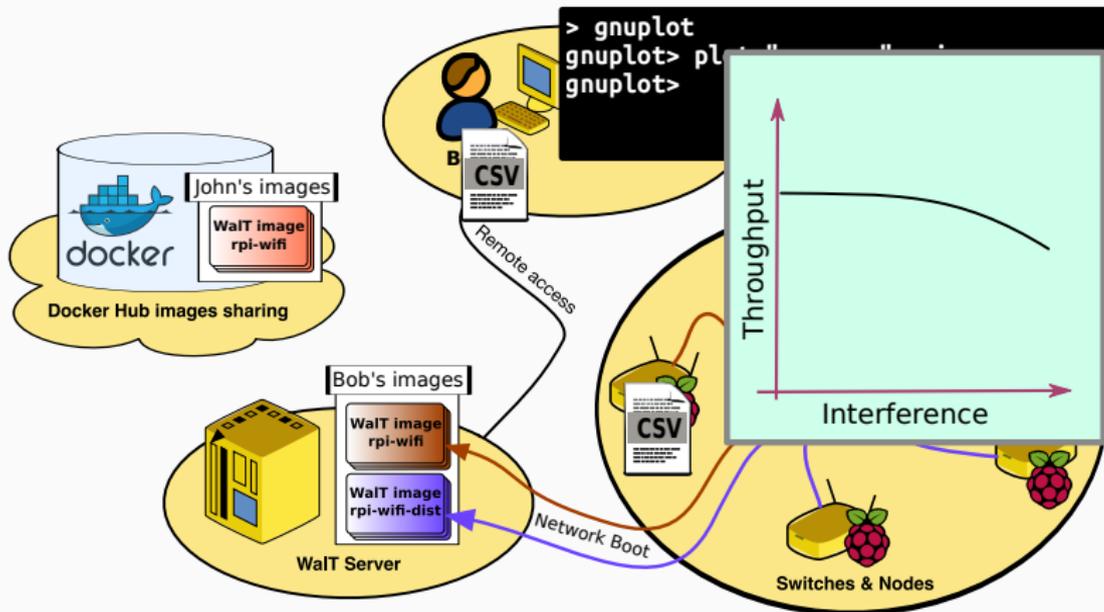
Sample use case



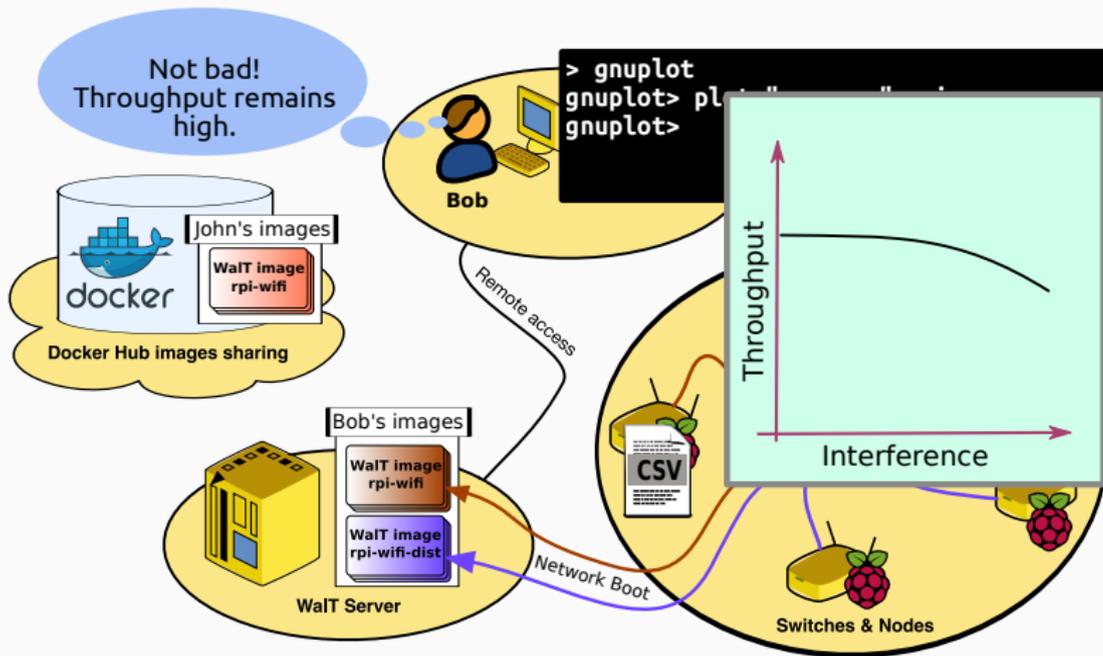
Sample use case



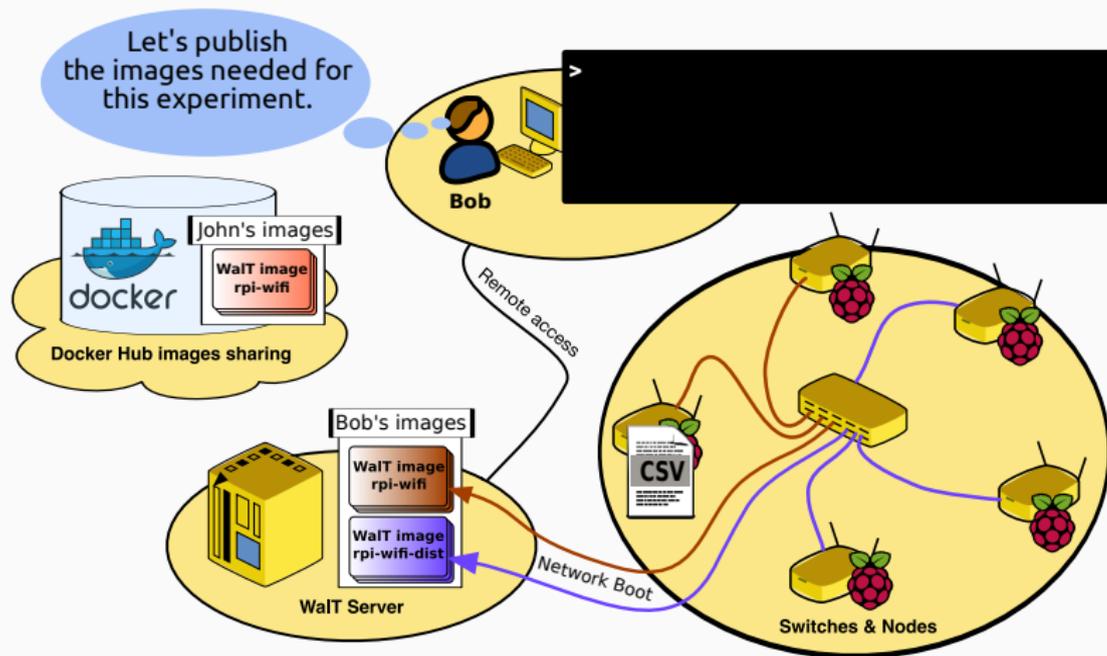
Sample use case



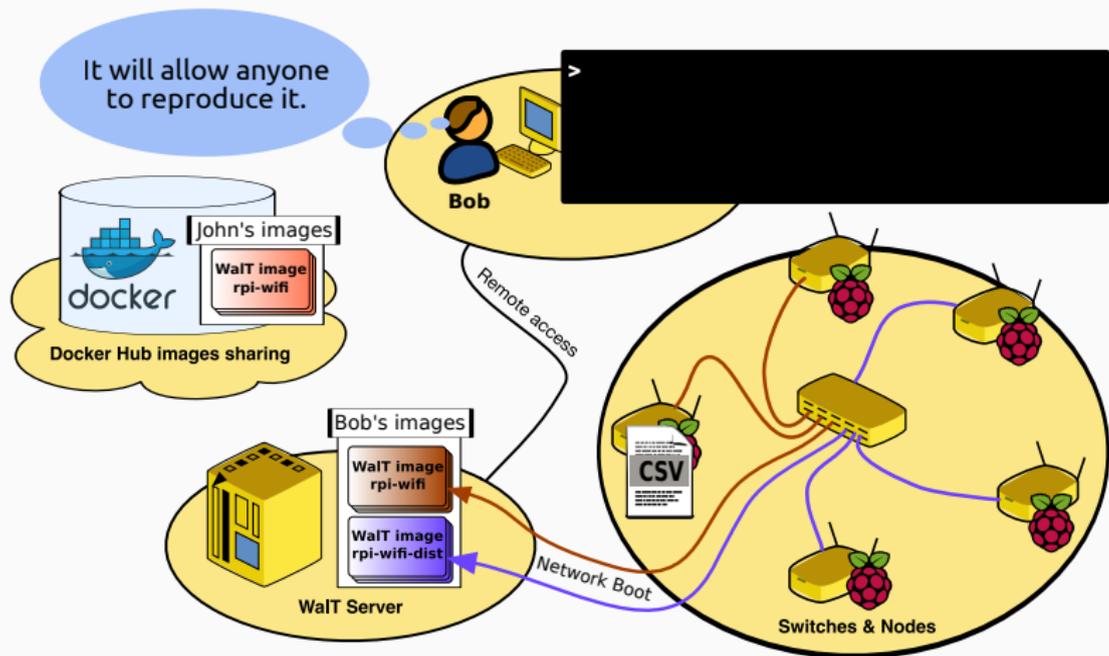
Sample use case



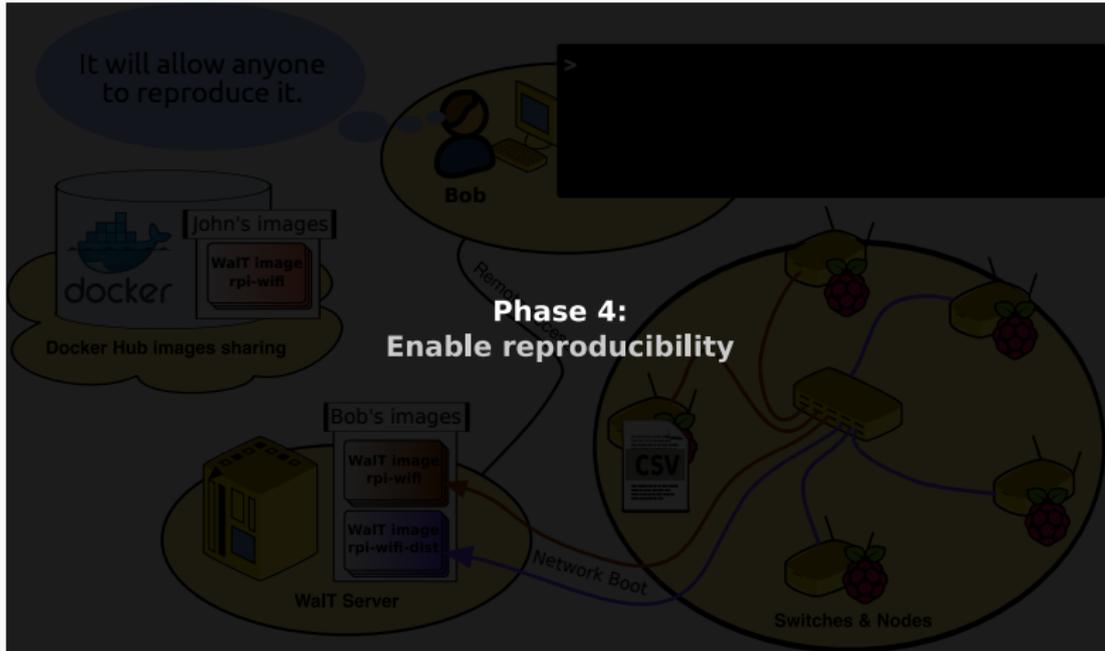
Sample use case



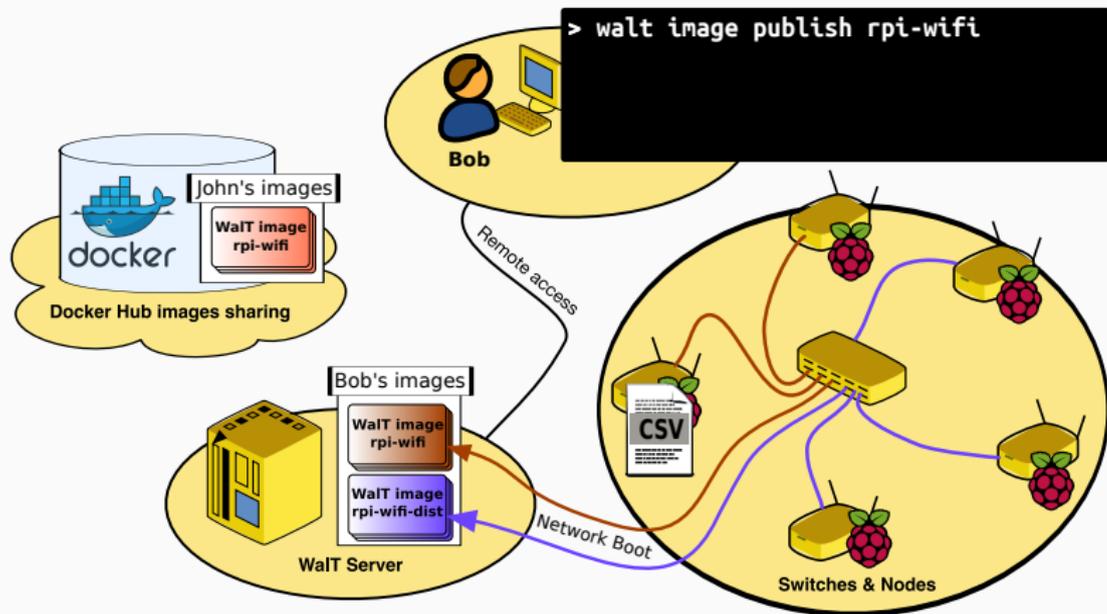
Sample use case



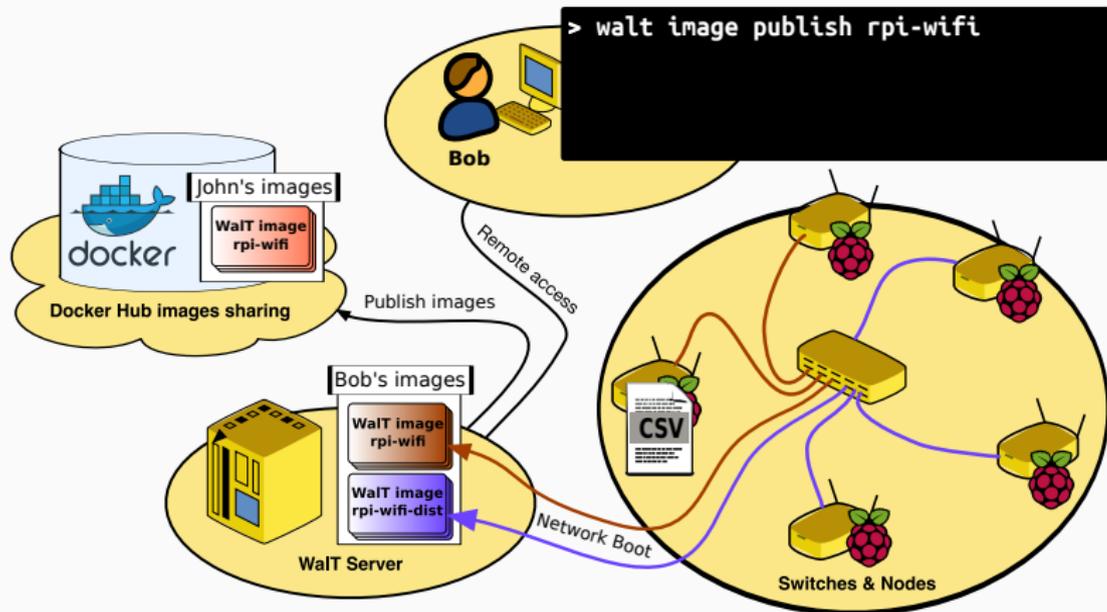
Sample use case



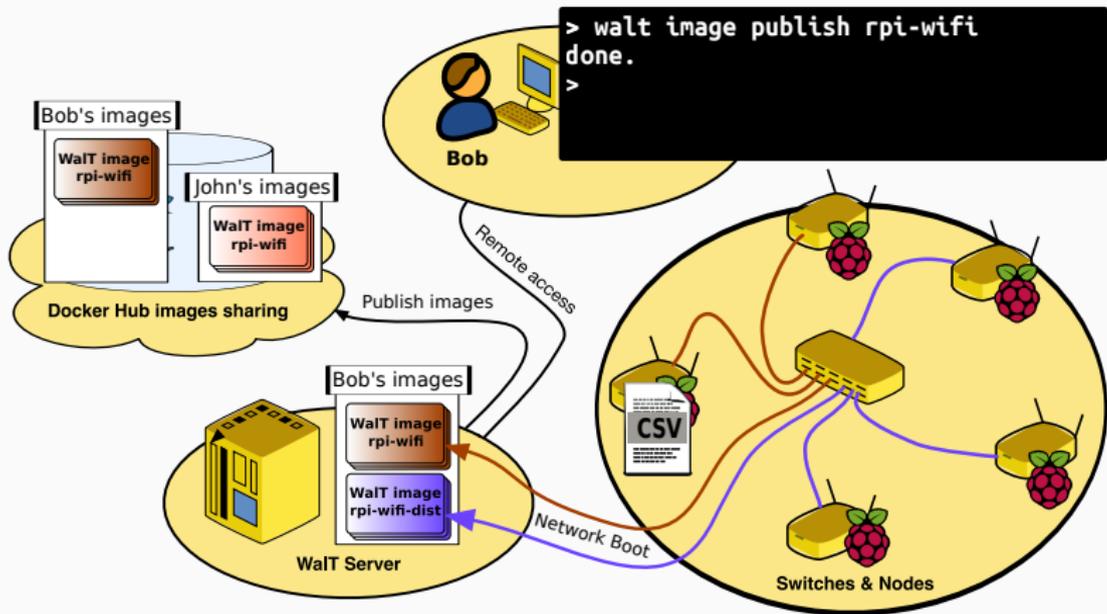
Sample use case



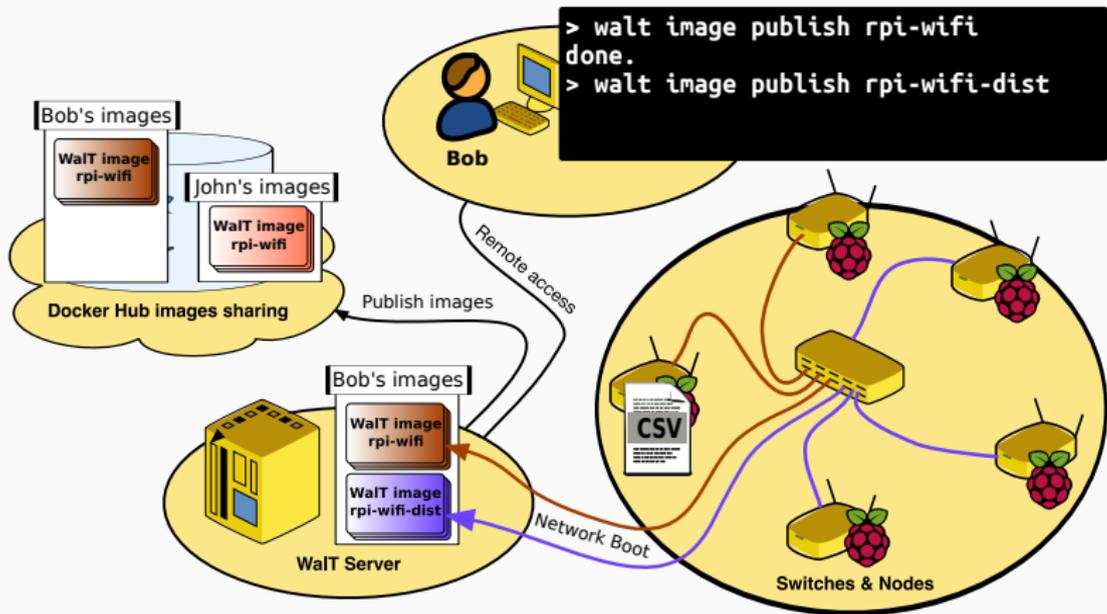
Sample use case



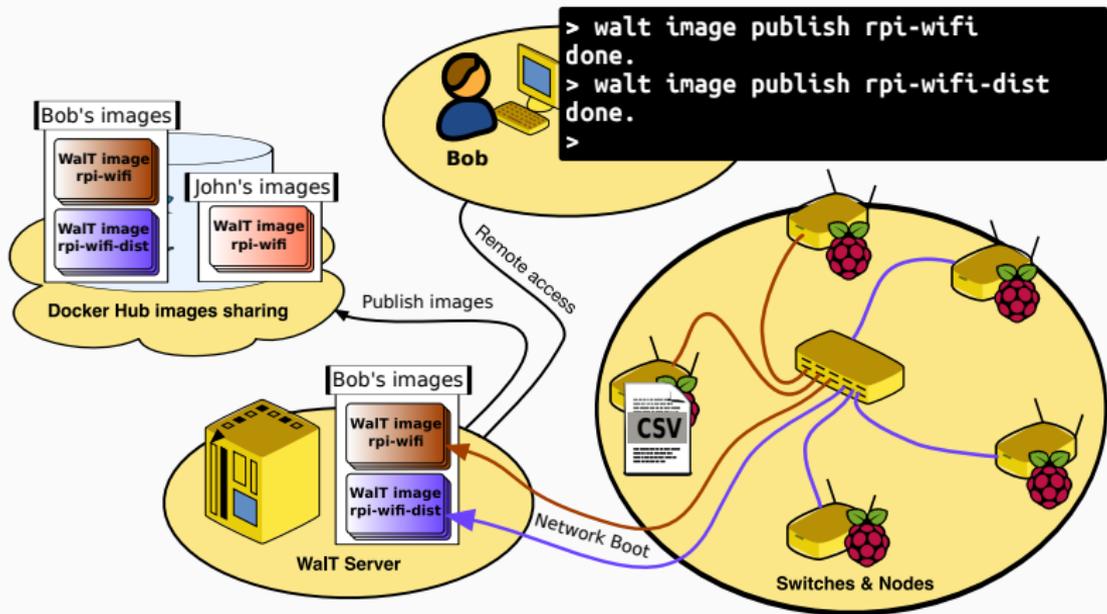
Sample use case



Sample use case



Sample use case



The image shows the words "The End" written in a highly stylized, white cursive script. The letters are thick and have a 3D effect, with dark shadows cast beneath them. The text is centered on a solid, medium-gray rectangular background.

A few alternative ways:

- **walt image build**: OS image build from a Dockerfile
- Inverted workflow (since WALT v10):
 1. **walt node shell**: Apply updates on the node directly
 2. **walt node save**: Save as a new OS image

Agenda

Introduction

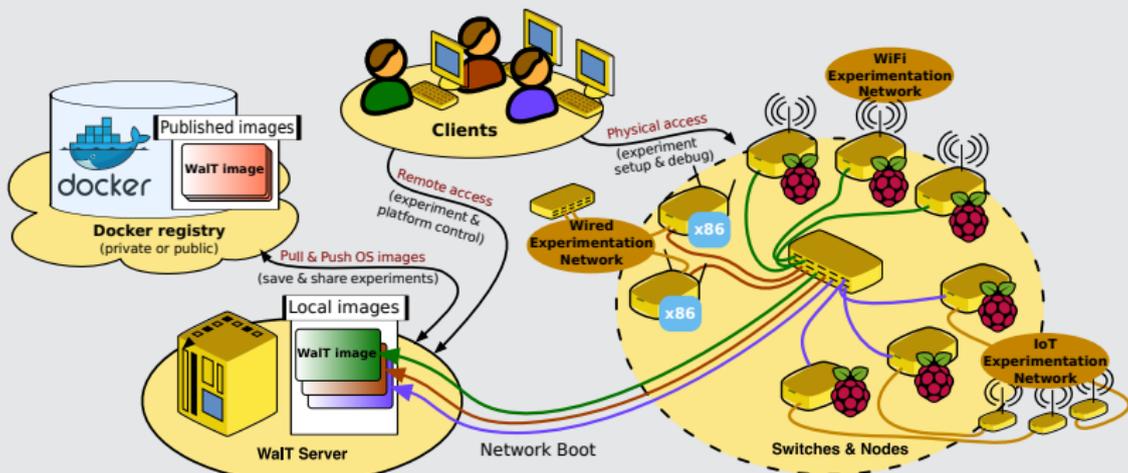
Sample use case

Architecture

Client tools

News, On-going work

Architecture diagram



Other features, not shown here:

- **Virtual nodes**
- **Distant nodes** (VPN-connected, e.g., for **LoRa** experiments)

What's special about WALT OS images & deployment?

	Standard containers		Bootable containers ¹	
	docker	podman	bootc	walt
Docker image format	✓	✓	✓	✓
Compatible image tools	✓	✓	✓	✓
Image embeds a full OS	✗	✗	✓	✓
Access node hardware	✗	✗	✓	✓
Access connected devices	✗	✗	✓	✓
Light deployment	✓	✓	✗	✓ ^{2,3}
Nodes are stateless	✗	✗	✗	✓ ^{2,4}

¹This concept is described at: <https://containers.github.io/bootable/>.

²Considering the default boot-mode of WALT nodes (network boot).

³A WALT node only transfers the WALT image files it really needs (5-10%).

⁴The fact nodes are stateless eases reproducibility.

What is a WaT node?



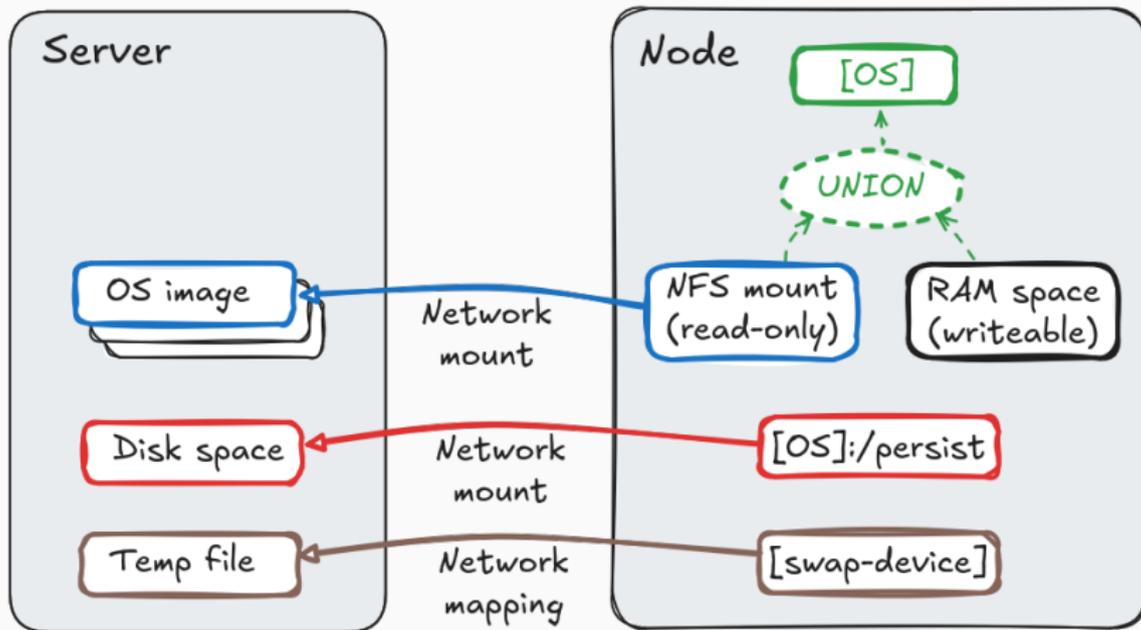
A WALT node can be:

- A **Single-Board-Computer**¹
- An **x86 PC** or laptop
- A **virtual node**

WaT also provides basic functions for devices not supported as a WaT node.

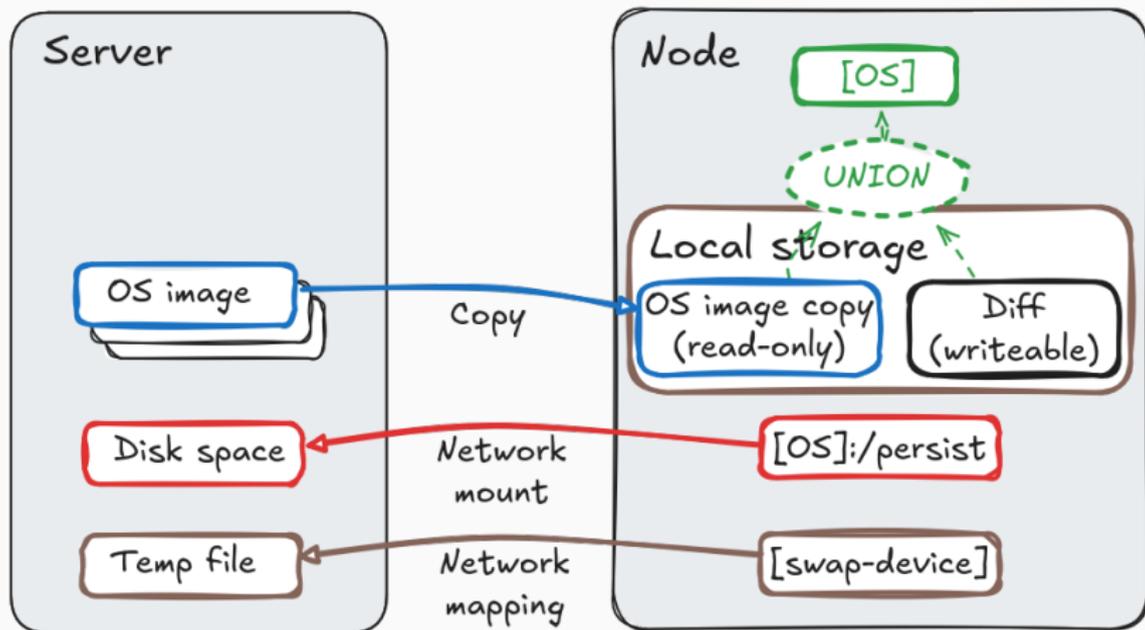
¹We support **Raspberry Pi B/B+/2B/3B/3B+/4B/5B**, Google Coral Dev boards, and **x86 boards**.

Network boot (default boot-mode)



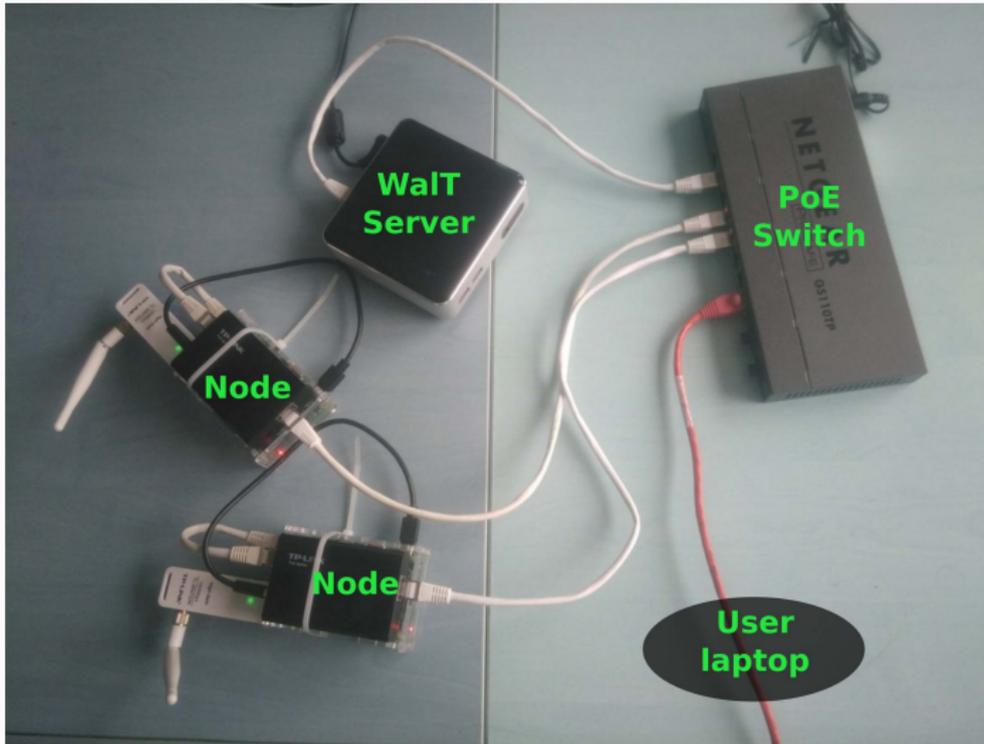
- Network mount: near-zero deployment delay.
- Created and modified files are saved in RAM.
- After each reboot, only the OS image files remain (reproducibility).

Hybrid boot (for specific experiments)



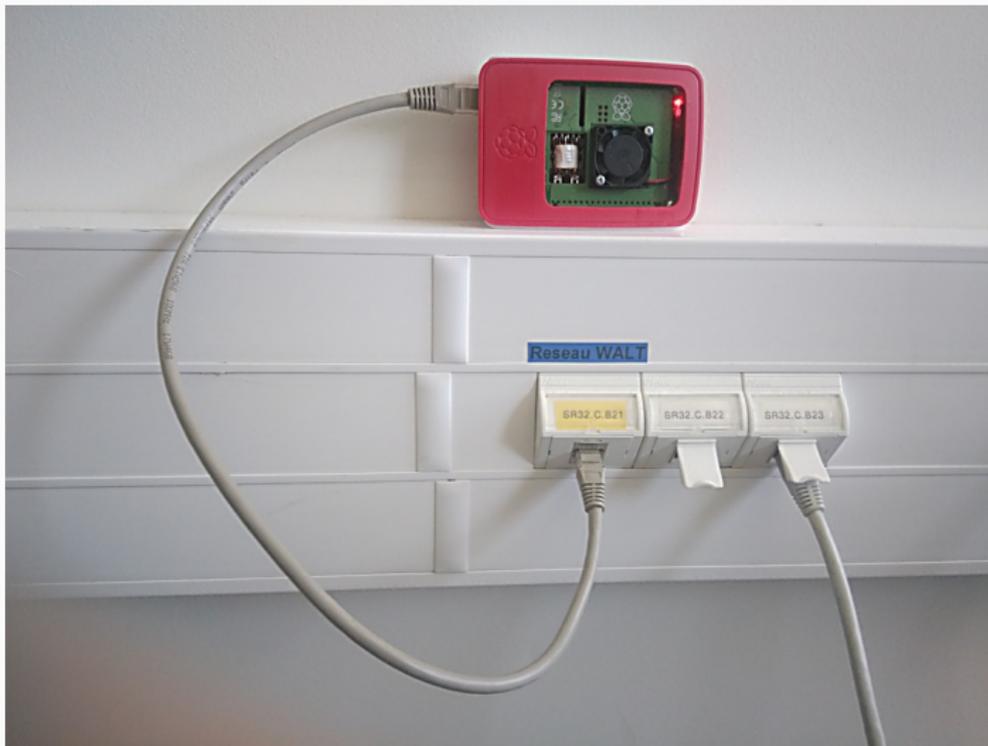
- Copy of the OS image on node's local storage at first boot.
- **hybrid-volatile** variant: Diff cleared at each reboot.
- **hybrid-persistent** variant: Diff preserved.

Sample mobile setup



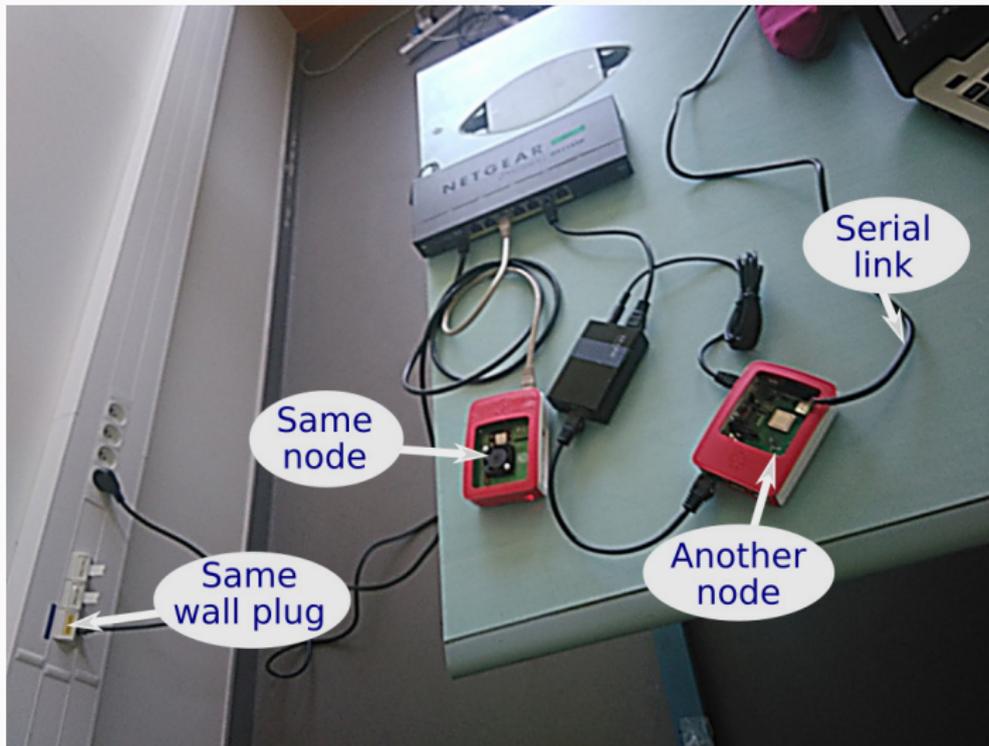
Minimal setup for an ANR Datatweet meeting

WaT versatility shown in two slides (1)



A node of the WaT network in IMAG building

WaT versatility shown in two slides (2)



Reusing the same wall plug for temporary low-level debugging

Agenda

Introduction

Sample use case

Architecture

Client tools

News, On-going work

walt client tool

```
root@ibiza:~# walt
No sub-command given
-----

WALT platform control tool.
Version: 8.2

Usage:
  walt CATEGORY SUBCOMMAND [args...]

Help about a given category or subcommand:
  walt CATEGORY --help
  walt CATEGORY SUBCOMMAND --help

Help about WALT in general:
  walt help show

Categories:
  image      management of WALT-nodes operating system images
  node       management of WALT nodes
  log        management of logs
  device     management of WALT platform devices
  vpn        VPN related sub-commands
  advanced   advanced sub-commands
  help       help sub-commands
root@ibiza:~#
```



- Organised as: `walt <category> <sub-command>`
- Optimised, replies very quickly to simple commands
- Advanced bash & zsh completion
- Includes various help pages, see: `walt help show`

Scripting using the python API

Python scripting

- Allows to automate experiments easily
- Easy to explore, using an interactive python shell
- Entrypoint: `from walt.client import api`
- More info: `walt help show scripting`

Web API (http+json)

- Allows external tools to retrieve walt platform data (e.g., to populate a Grafana dashboard)
- Get data about: nodes, images & logs
- More info: `walt help show web-api`

Agenda

Introduction

Sample use case

Architecture

Client tools

News, On-going work

WALT in the Industry

WALT at Schneider Electric

- Current use: Test Zigbee devices in electrical enclosures.
- 2 testbeds in France (Eybens), 1 testbed in India (Bengaluru)
- Industrial contract LIG + SE since December 2022.



Goal: large scale WALT platform

Nodes scattered **on the roofs of buildings** on the Grenoble campus

- Allows experimenting **long range protocols** (LoRa & Co.)
- **Targets:** Education, Research, Industry
- **Financing:** Labex Persyval (Education) + Carnot LSI
- **Challenges:** Reconfigurable hardware (for long term reuse), robust, autonomous and secured WALT nodes.
- **Status:**
 - Summer 2024: Exploratory work
 - June 30, 2025: WALT v10, with VPN re-design
 - Since february 2025: hardware prototyping

- Funding
 - Permanent staff funded by **CNRS** and **Grenoble INP** (since 2012)
 - Projects & Contracts:
 - Univ. Grenoble Alpes, Grenoble INP / UJF, AGIR WaIT (2013-2014)
 - ANR IRIS (2011)
 - FP7 ICT CALIPSO - Connect All IP-based Smart Objects (2011)
 - ANR DataTweet (2013)
 - Schneider Electric (2022-)
 - WaToR Labex Persyval (2024-2025)
 - WaToR Carnot LSI (2025-)
- Thanks to the many people who have worked with us over the years!

Useful links

Contact us:

`walt-contact@univ-grenoble-alpes.fr`



Web site:

`https://walt-project.liglab.fr`

`https://walt-project.liglab.fr/docs`



Mailing list:

`https://walt-project.liglab.fr/walt-users`

Questions?