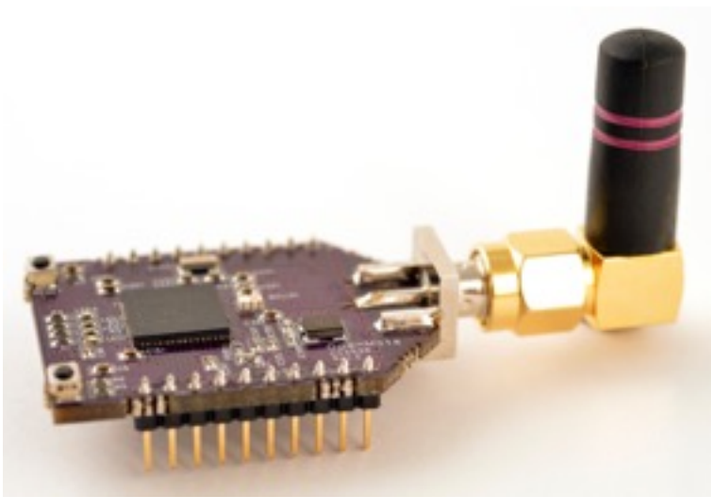
WalT: a Testbed for Reproducible Wireless Networks Experimentations

Pierre Brunisholz, Etienne Dublé, Franck Rousseau
Université Grenoble Alpes
Grenoble INP – CNRS – LIG Lab

Eclipse IoT Day Grenoble 2016
28 April 2016

Motivation for WalT

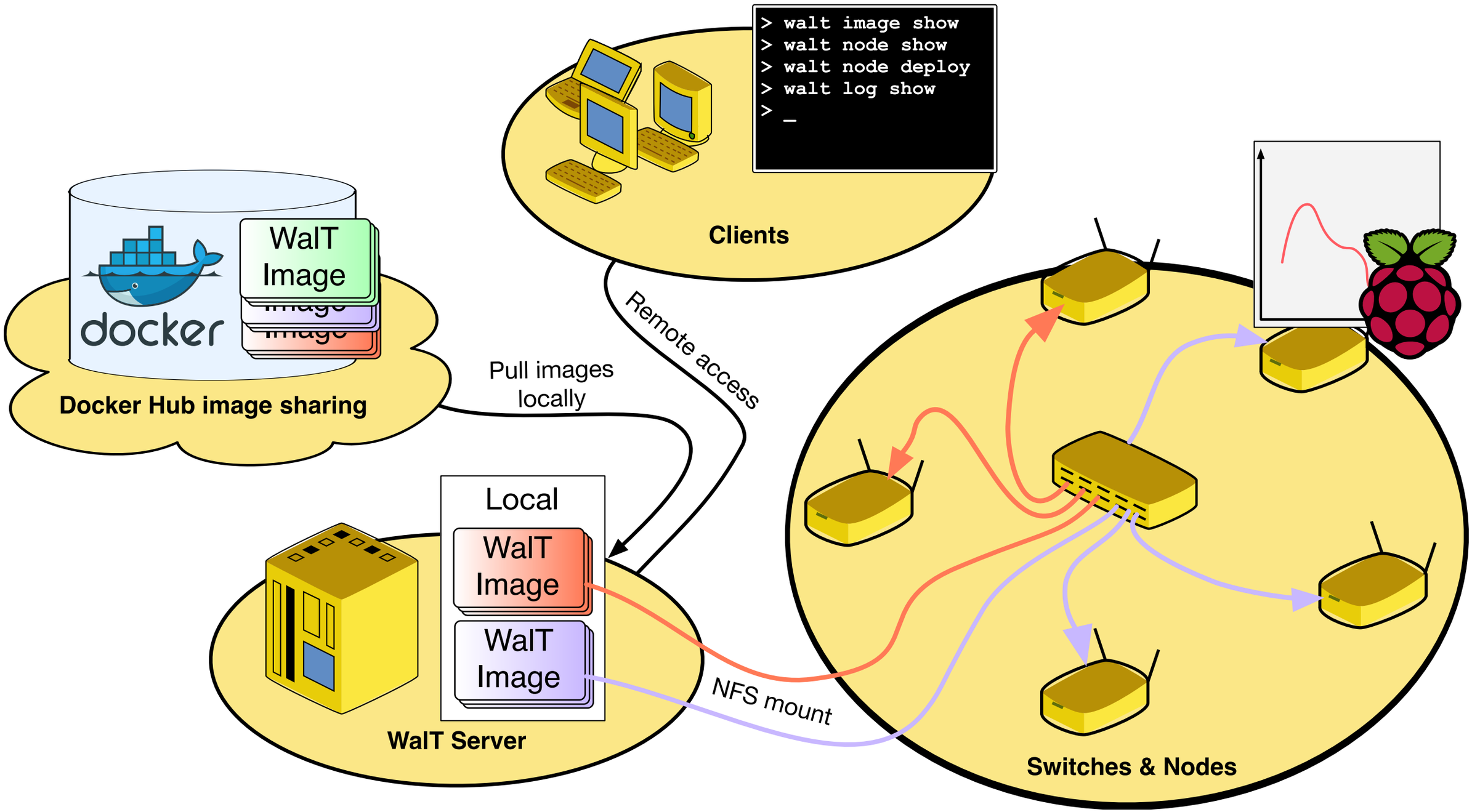
- **Wireless Networks have special needs**
  - Simulations are not enough
  - Spatial diversity is fundamental
- **Large testbeds have their limits**
  - $$$ but limited hardware choice
  - Fixed environment
  - No physical access / debug
  - Reservation system
- **“Desktop” experiments are (very) limited too**
  - Do not (or painly) scale
  - Not easily reproducible



WalT objectives

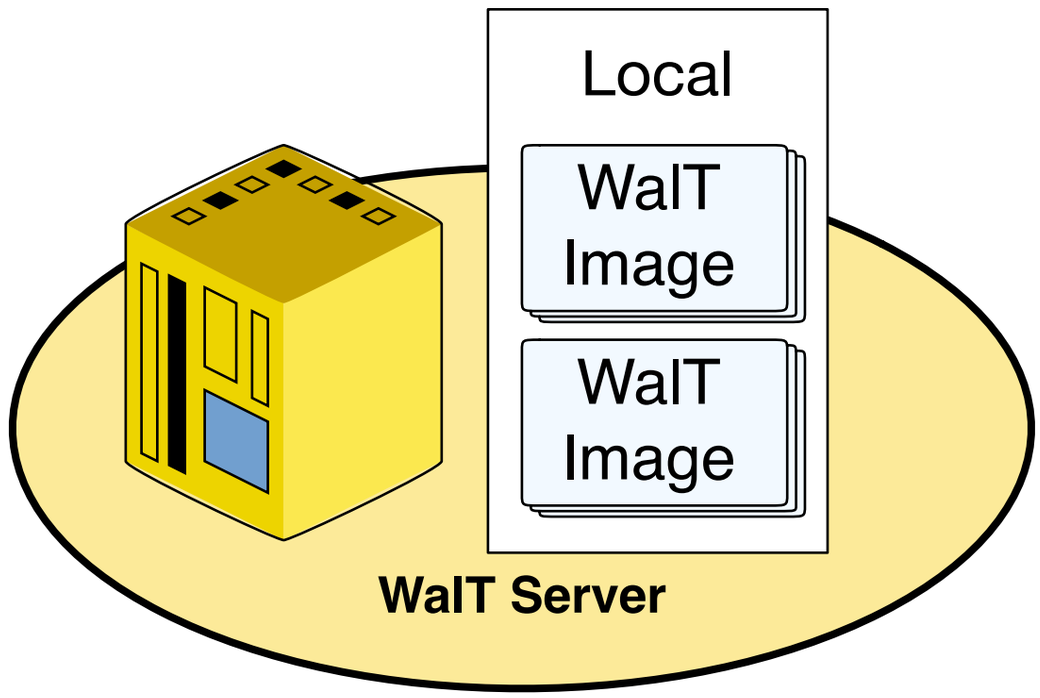
- **Design a lightweight experimentation platform**
  - Cheap to acquire, and to maintain
  - Free and open specifications and code
- **Easily extensible and reproducible**
  - No specialized hardware
  - Painless deployment and setup
  - Easy to add support for new hardware
- **WalT nodes can be the nodes under test or just controllers**
  - Running the code of the experiment
  - Driving other devices for the experiment, eg. sensor nodes

WalT – A lightweight experimentation platform



An easily deployed server

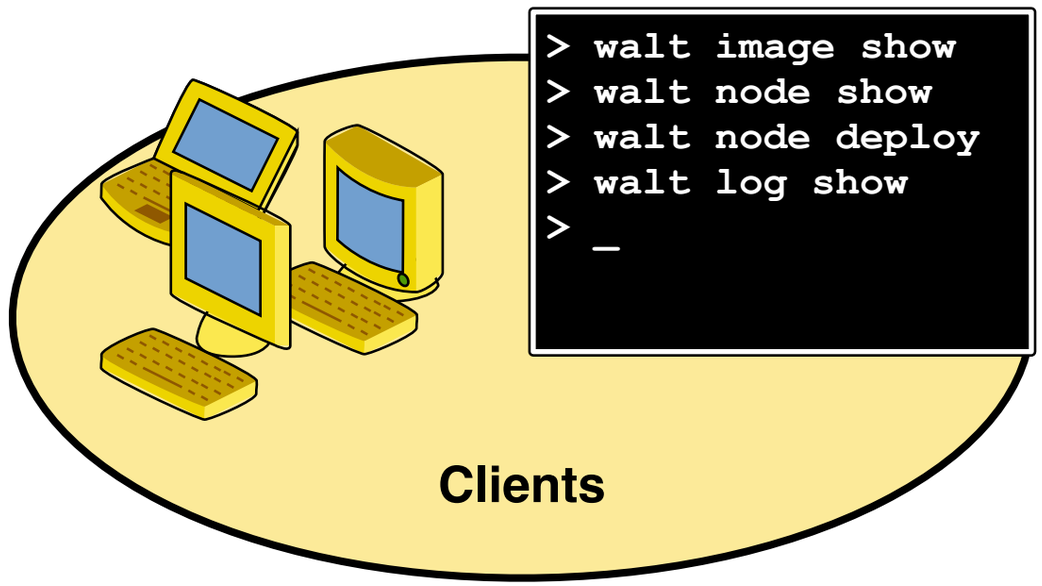
- **Automated installation**
  - bootable image built with `debootstick`
- **Run on any 64 bit computer**
  - from Intel NUC to rack server



`http://walt.forge.imag.fr/setup.html`
`http://walt.forge.imag.fr/diagram.html`

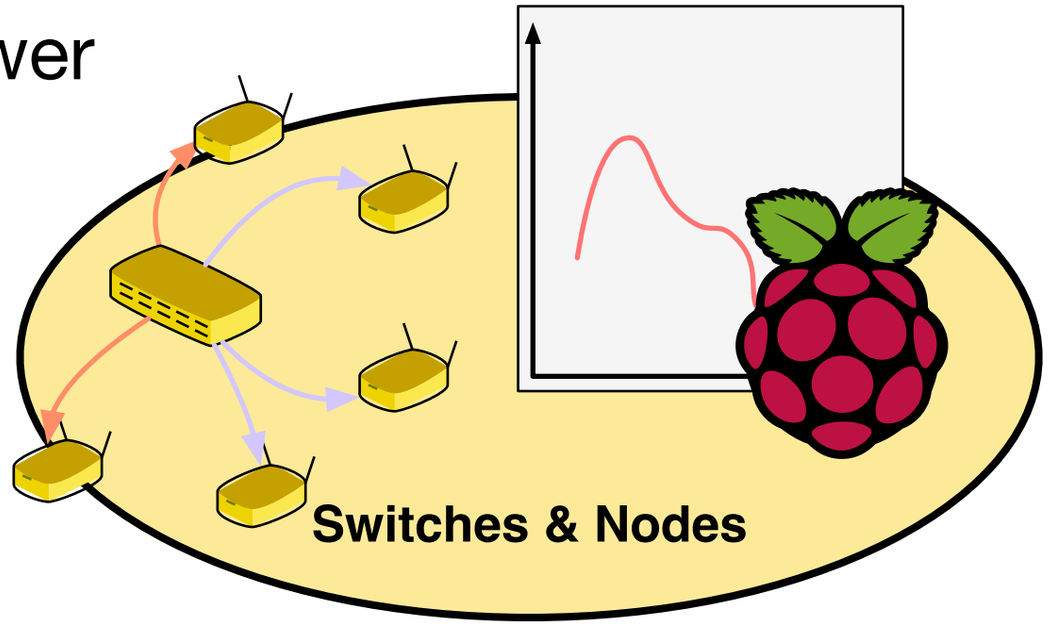
An easy to install (and use) client

- **`walt` command line client**
  - `pip install walt-client`
  - Cross platform (Linux, OSX…)
  - Run experiments in a few commands
- **VizWalT**
  - Visualisation plugin for Cooja



A lightweight infrastructure

- **Managed switches with PoE**
  - Remotely control nodes
  - Nodes do not need external power
  - Easy node reset → power cycle
- **Nodes: RPi B / B+**
- **On-going**
  - UDOO, RPi 2

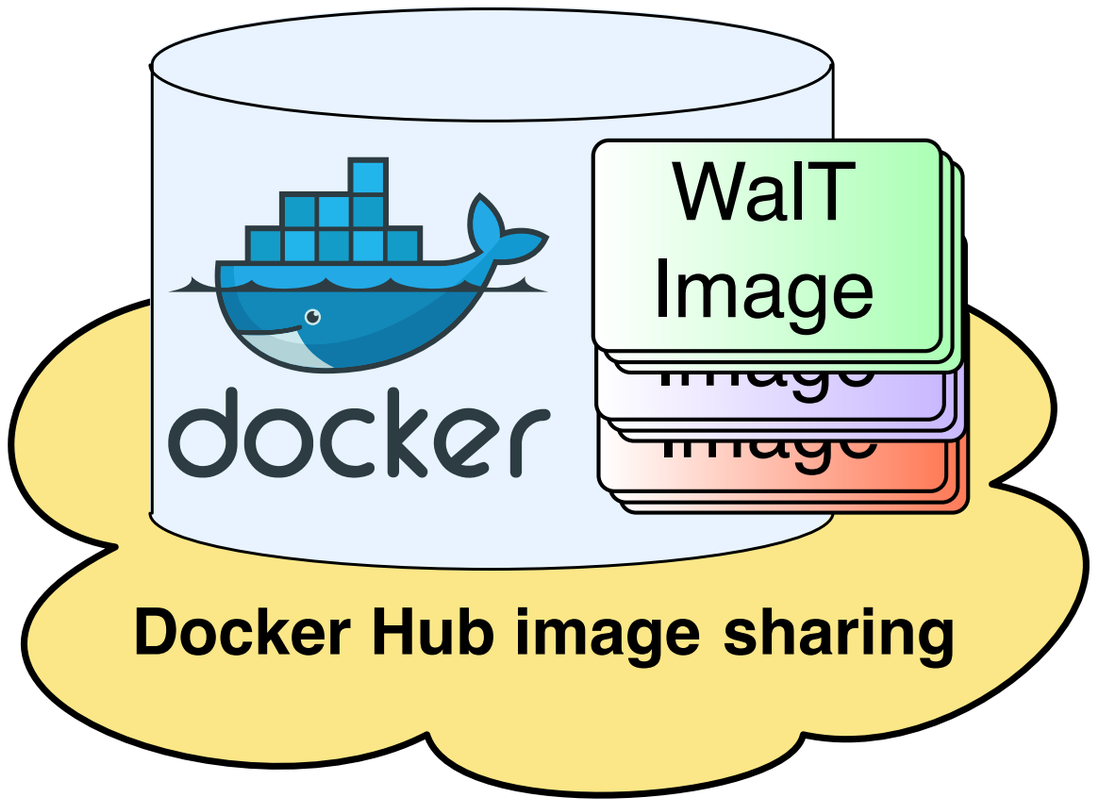


A versatile platform

- **Small size deployments for debugging or mobile demos**
  - Easily flash several WSN nodes
  - Attach oscilloscope or digital analyzer
  - `make test` before commit
  - Continuous integration
- **Larger deployments for experiments**
  - Only need Ethernet for control
  - Deploy across the lab
    - Ethernet sockets on one side
    - Patch bay on the other side

Reproducible experiments powered by docker

- **WalT nodes OSes to packaged as docker images**
  - Easily built or modified
  - Easily shared on the *docker hub*
- **Network boot**
  - Read-only SD cards on nodes
  - Kernel and filesystem on server

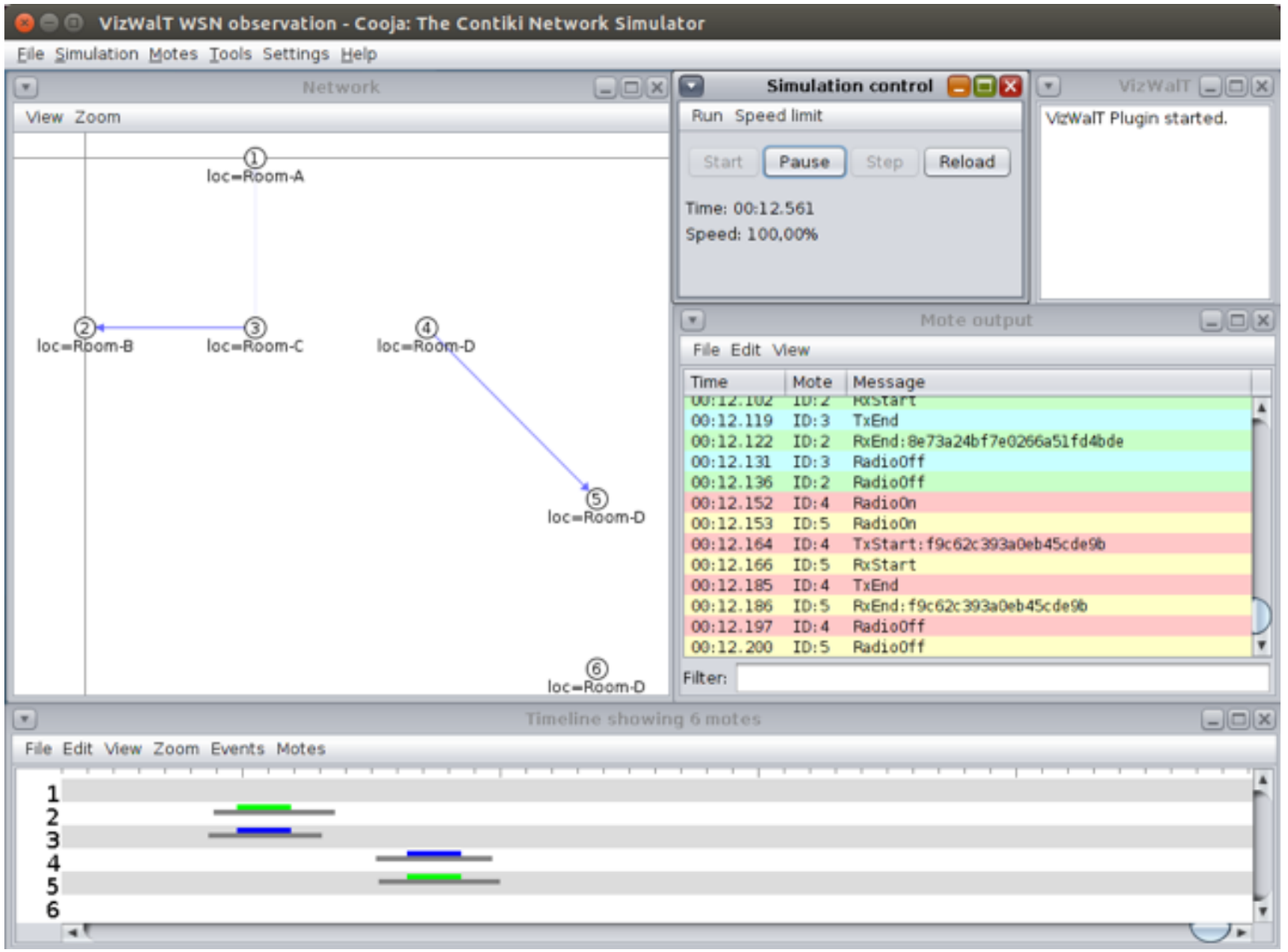


A WalT platform provides

- **Full remote control over nodes**
  - Rebooting, remote shell sessions, deploying OS images
- **Management of OS images**
  - Clone from the docker hub, modify locally, publish images
- **Log management**
  - Means to timestamp, collect, store, and query experiment logs and event traces
- **Automated discovery of the platform topology**
  - Add, remove devices

VizWalT

- **Visualization of traffic in near real-time**
- **Built on Cooja the emulator for Contiki**



Reproduce experiments easily

- **WalT is cheap**
- **WalT is easy to set-up and maintain**
- **Existing experiments run in a snap**

Future work

- **Support new hardware**
- **Fully packaged experiments**
- **Compatibility with FIT/IoT-LAB experiments**

- **Any thing you would like to do**
  - Free and open specification
  - Free and open software

http://walt.forge.imag.fr

Questions?

WalT credits

- **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**
  - Bastien Faure, core WalT
  - Pierre-Henry Frohring, core WalT
  - Jorge Luis Baranguan Castro, VizWalT visualization plugin for Cooma
  - Cosmin Nichifor, WalT synchronization
  - Joao Guilherme Zeni, sensor integration & Contiki instrumentation
  - Matheus Castanho, sensor integration
  - Iacob Juc, sensor integration
  - Liviu Varga, sensor integration