Introducing The “Lab in a Box” Concept

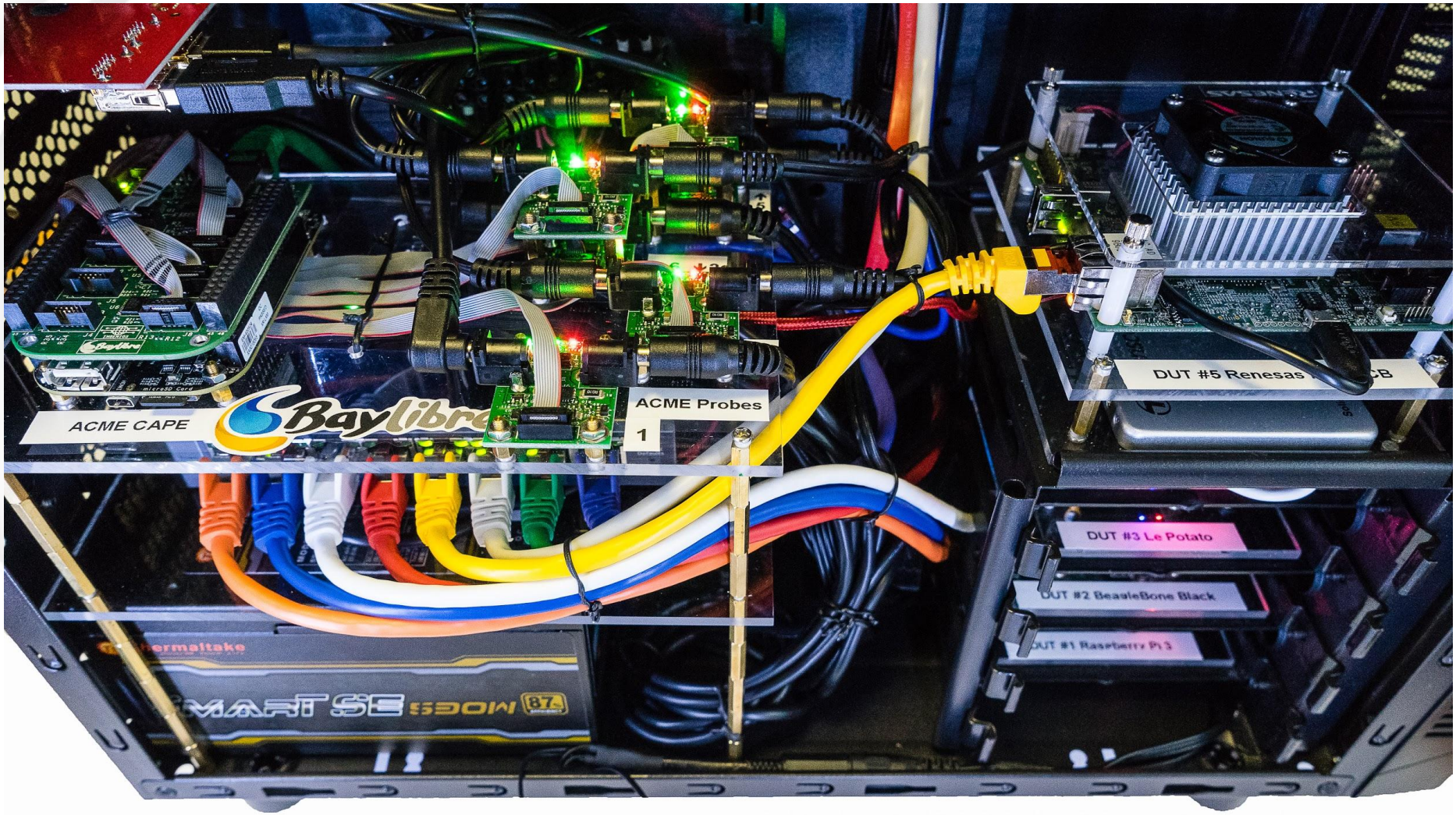
ELC Portland, March 2018

Patrick Titiano - Kevin Hilman, Baylibre.

About us

- Baylibre
  - Embedded Linux Consultancy, Engineering Services
  - 25 senior engineers, coming from the semiconductor world
  - HW and SW products: from concept to manufacturing
  - Upstream Linux kernel development and maintenance
  - Founding developers and active contributors to kernelCI.org project

Teaser: this is... LAVA box...



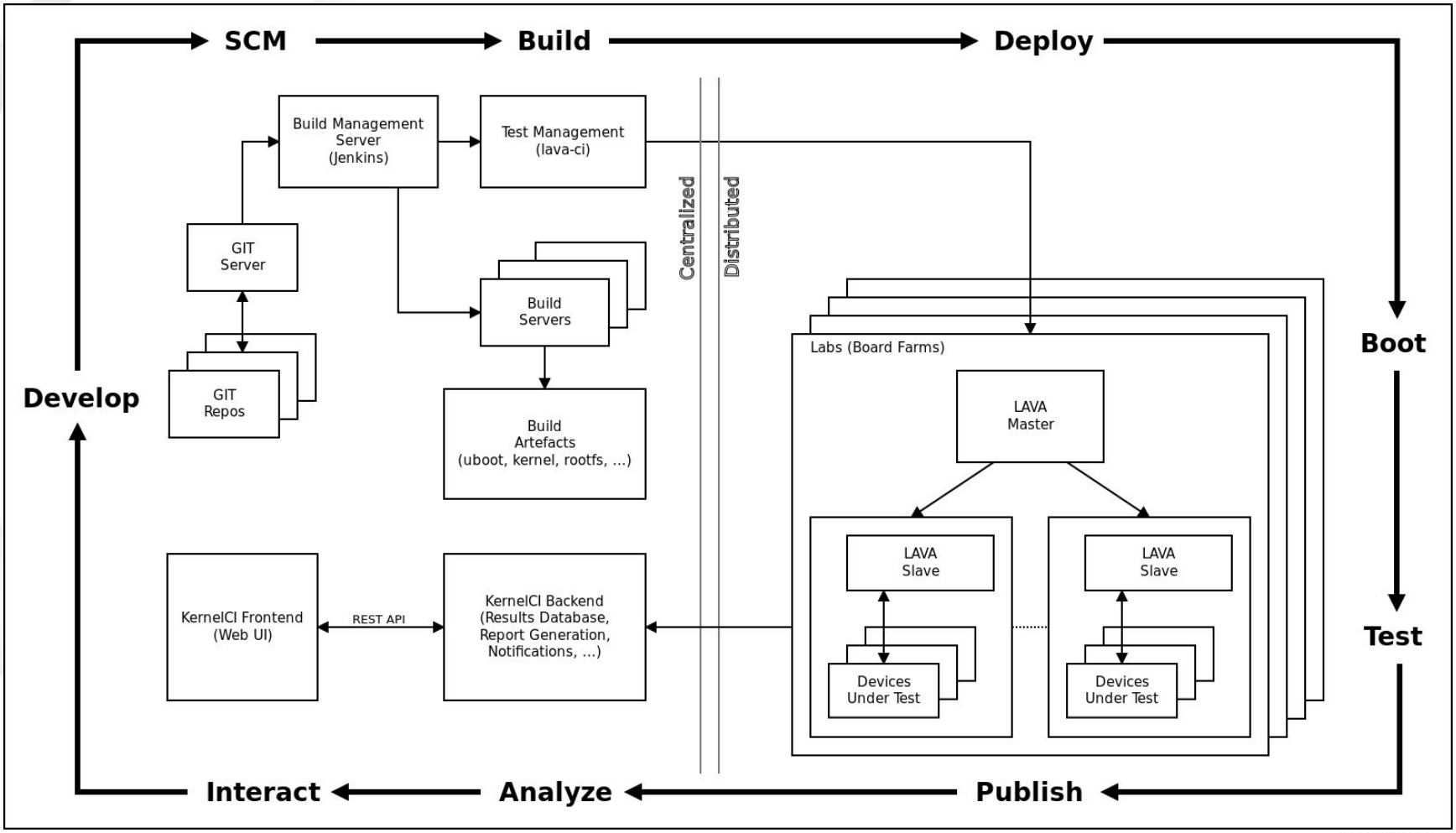
Let’s see how we got there!

KernelCI.org

- **Build & Boot Test Automation System**
  - Focused on **upstream Linux kernel**,
  - Open Source, Community based,
    - (welcomes contributions like HW / Lab / infrastructure / resources)
  - Distributed, leveraging **LAVA**
- Since May 2014 :
  - Performed **3.5M+ boots** on **250+ boards**, across **3 architectures** and **34 SoCs**. (**2500+ boots** per day.)
- Results reported via mailing lists and web site
- Much more likely that kernels will build… and run
  - v3.14: 51 failed configs
  - v4.1: 1 failed config
  - v4.13.y: 0 failed configs

* Source: https://kernelci.org/, https://elinux.org/images/b/b9/Brown.pdf

KernelCI Loop

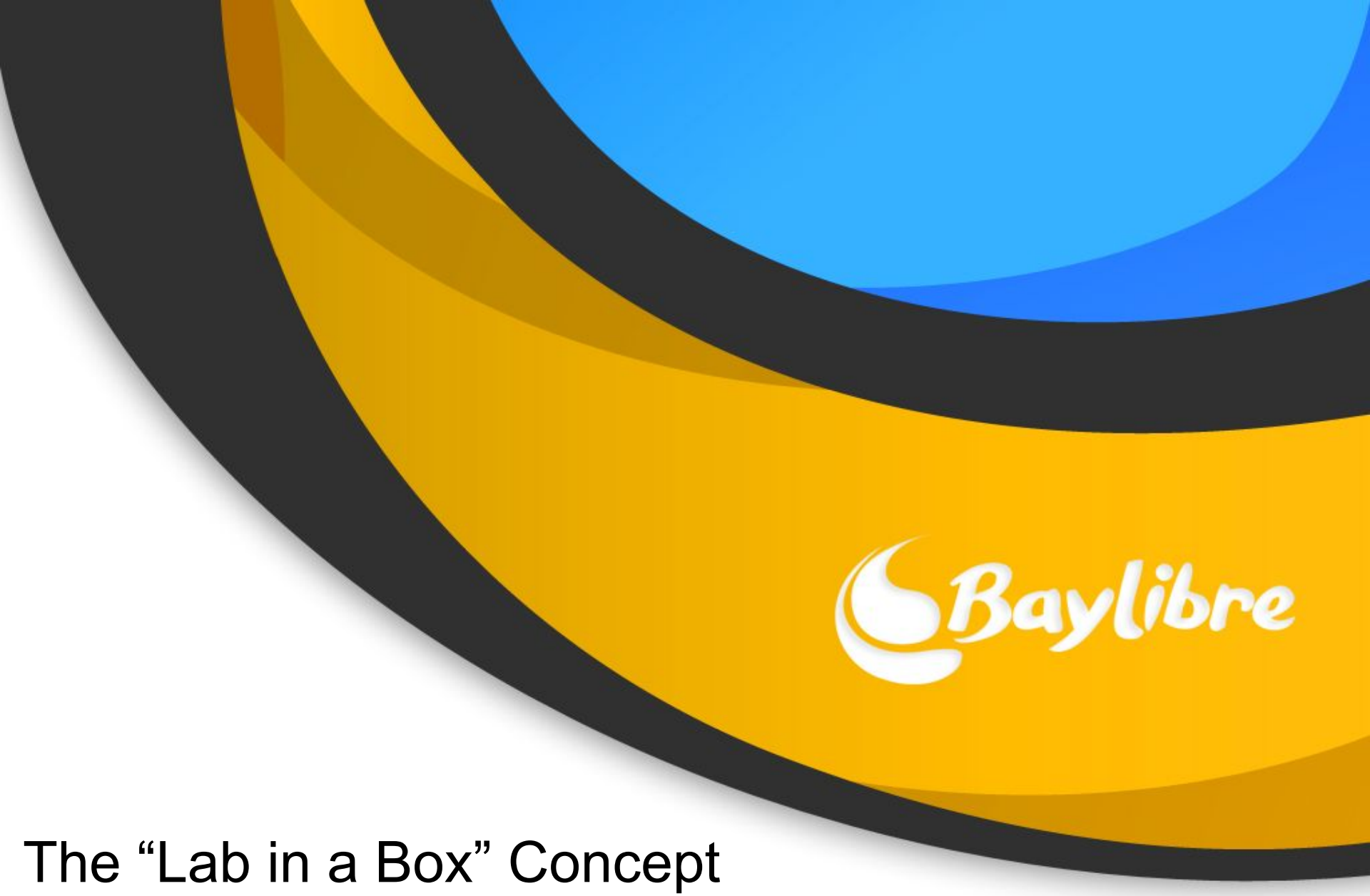


KernelCI Loop

AGL CI Loop

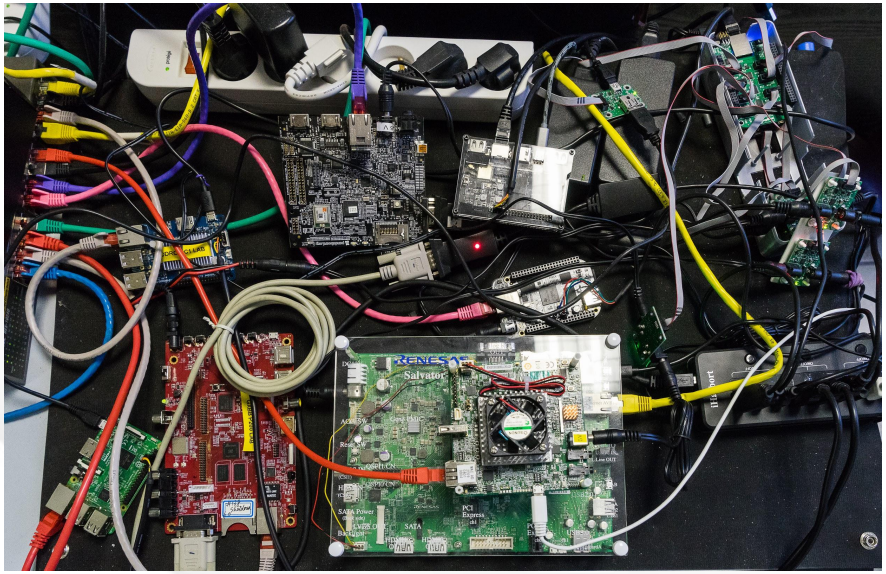
- Funded by Linux Foundation AGL Initiative
  - As part of the CIAT Group (Continuous Integration / Automated Testing)
- Leverages LAVA and kernelCI
- BayLibre updating and extending KernelCI to
  - Test AGL releases, snapshots and per-commit development
  - Run any kind of test instead of only build and boot
    - Generic test suites,
    - AGL-specific test suites,
    - Automotive-specific test suites,
    - Power & Performance profiling,

* Source: https://wiki.automotivelinux.org/eg-ciat

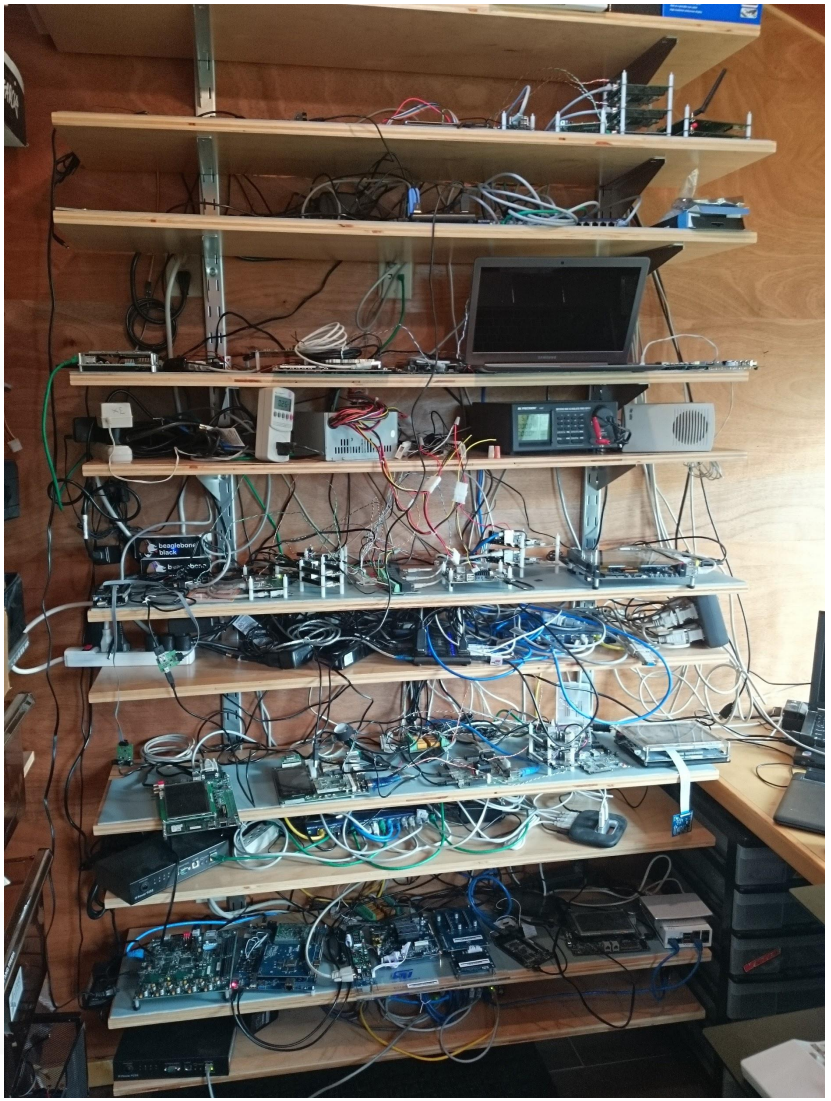


The “Lab in a Box” Concept

Motivations



Time to go pro!



Motivations (2)

- Simplify Administration
  - LAVA: nice technology, but difficult to get into it
    - Installation process (now eased with Docker)
    - Device-types
  - USB Serial debug ‘pairing’
  - Ultimately users shouldn’t be aware of the internal technologies to build and run a CI Lab
- Ease duplication / scalability
- Accelerate deployment

Requirements

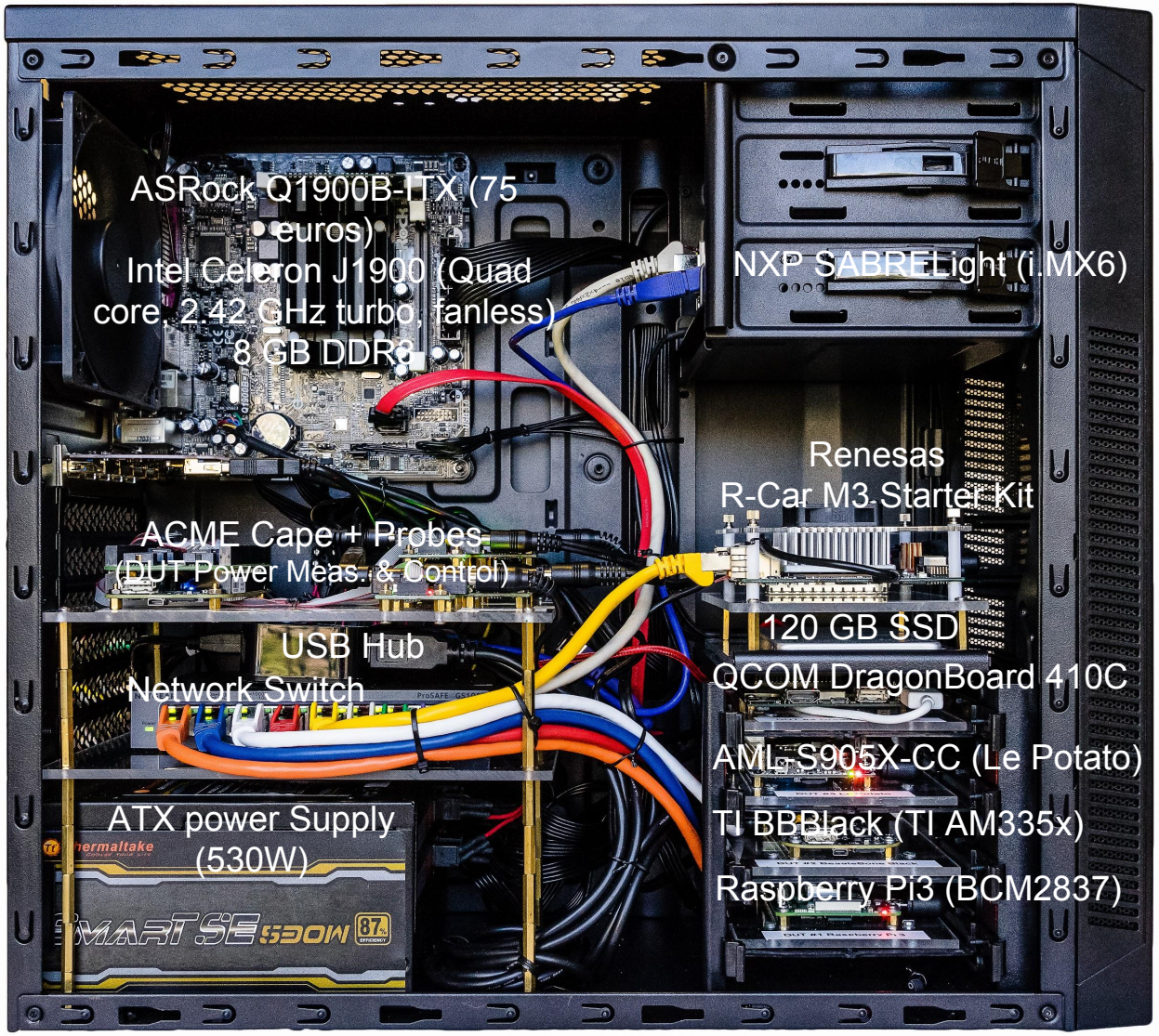
- “All in One” solution, integrating
  - LAVA master and dispatcher, Devices Under Test (DUT), power supplies for all DUT, connectivity / wiring (network, debug ports, power control, etc)
  - Reference & community AGL boards
- Low cost
- Scalable / Reproducible
- Safe / Maintainable
- Easy installation (HW + SW)
  - Pre-installed / pre-configured SW components
  - Administration control panel
- Fits in an apartment (for home workers)
- Documented

Challenges

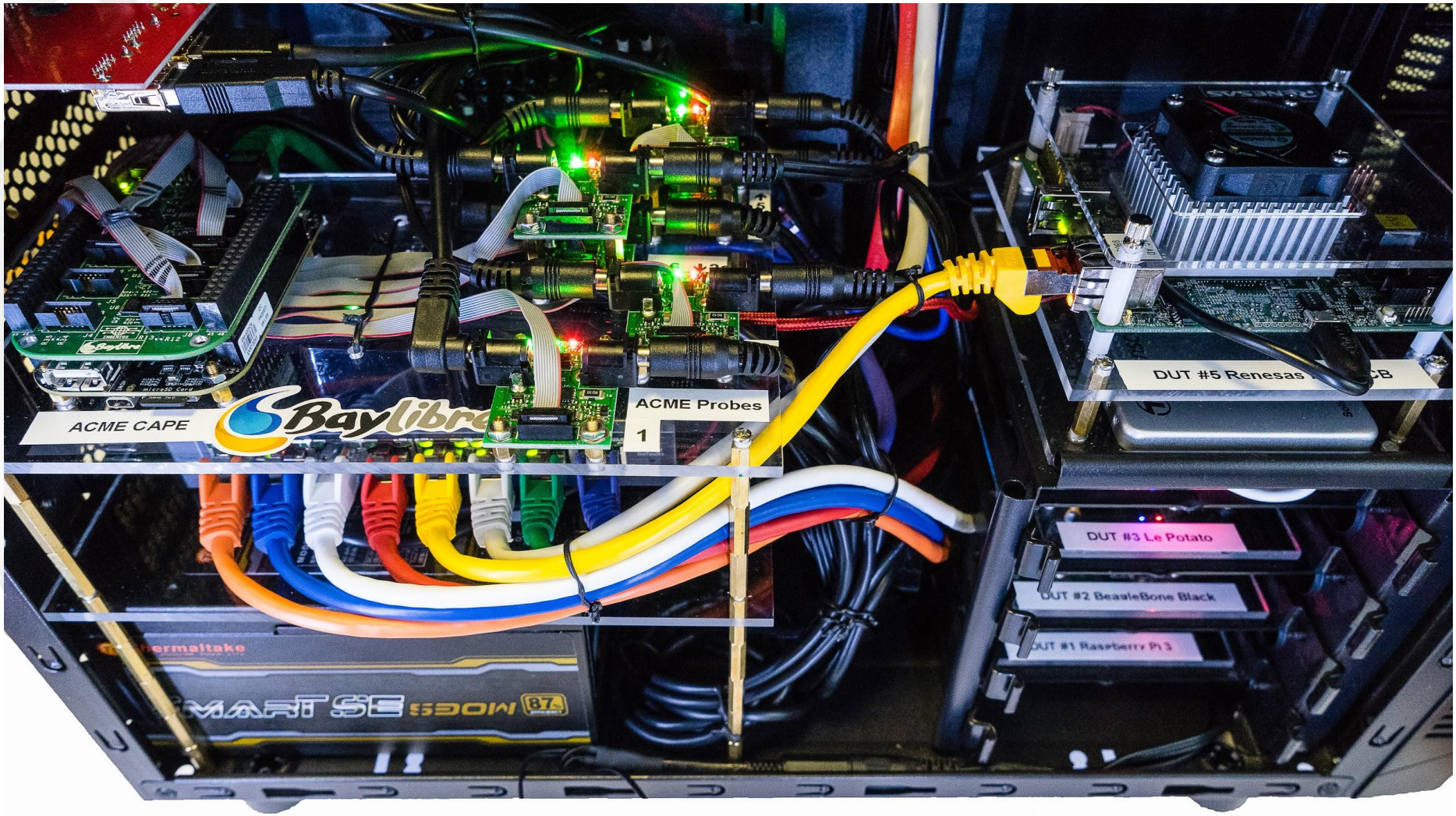
- A lot of stuff to integrate in a single case
  - DUT
    - Custom size
    - Custom connections
  - Power Control unit
  - Lab Wiring
    - Network Switch
    - USB Hub
    - Per DUT
      - Power cable
      - Serial debug cable
      - Ethernet cable
- Maintenance

This is... LAVA box...

… unboxed


ASRock Q1900B-ITX (75 euros)
Intel Celeron J1900 (Quad core, 2.42 GHz turbo, fanless)
8 GB DDR3
NXP SABRELight (i.MX6)
Renesas R-Car M3 Starter Kit
ACME Cape + Probes (DUT Power Meas. & Control)
120 GB SSD
USB Hub
Network Switch
QCOM DragonBoard 410C
AML-S905X-CC (Le Potato)
ATX power Supply (530W)
TI BBBlack (TI AM335x)
Raspberry Pi3 (BCM2837)

Welcome inside...



Hardware details: USB connections

Serial consoles

- USB serial cables
- cheaper cables are flaky
- we use FTDI

Power

Many devices power over USB too

(sometimes on the same cable as fastboot !)

USB Misc.

Android: fastboot / adb

USB gadget:

- ethernet
- mass storage

Allows loading without “real” networking

Hardware details: Power

Power Distribution

Standard PC supply: ATX

- +5V
- +12V
- “standby” +5V

Power Switching

Simple, inexpensive

- Relays: GPIO or USB-controlled

Flexible, add measurement

- BayLibre ACME
- BBB cape + 8 channels of power switching and measurement

https://baylibre-acme.github.io/

Hardware details: networking

- 8-port switch inside the lavabox
- All devices on an separate LAN, internal to the lava-slave container
  - Isolated from the office LAN
  - Can integrate any kind of office LAN policy
- LAVA box needs internet access for jobs from kernel CI etc
- But, could also be internal LAN-only for local jobs

Software: LAVA dispatcher (slave)

Manage all connections between boards and “real world”

Services

- DHCP
- TFTP
- NFS
- NBD
- HTTP

Power control

- BBB + ACME
- lavapdu-daemon

Serial consoles

- USB / serial cables (FTDI)
- udev rules
- ser2net / conmux

USB misc.

- fastboot
- gadget: ethernet, mass storage

Containerized using: lava-slave-docker project

https://github.com/kernelci/lava-docker/

Software: LAVA server (master)

Web interface

Job scheduling, priorities

XML-RPC API

Board description

Board description

`device-type`

What all boards of this “type” have in common

- u-boot , fastboot, barebox, etc.
- Load addresses
- Bootloader environment

Can inherit/extend other device-types (e.g. base-uboot)

`device`

Specific to one instance of a board

- Select `device-type`
- How to connect to serial console
- PDU: how to power on/off
- Can override/extend settings from device-type

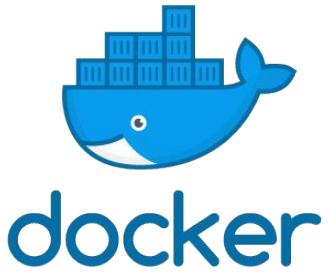
Containerized using: lava-docker project (from kernelCI)
https://github.com/kernelci/lava-docker/

Software: Bringing it together

```
# cat docker-compose.yml
    services:
      lava-master:
        build: {context: lava-master}
        devices: ['/dev/kvm:/dev/kvm']
        hostname: lava-master
        ports: ['10080:80', '1022:22', '5555:5555', '5556:5556']
        restart: always
        stdin_open: true
        tty: true
        volumes: ['/boot:/boot', '/lib/modules:/lib/modules']
      lava-slave:
        build: {context: lava-slave}
        devices: ['/dev:/dev']
        environment: {LAVA_MASTER: lava-master}
        hostname: lab-slave-0
        links: [lava-master]
        ports: ['69:69/udp', '80:80', '55980-56000:55980-56000']
        restart: always
        stdin_open: true
        tty: true
      squid:
        build: {context: squid}
        hostname: squid
        ports: ['3128:3128']
        restart: always
        volumes: ['squid-cache:/var/spool/squid']
    version: '2.0'
```

Multi-container management:

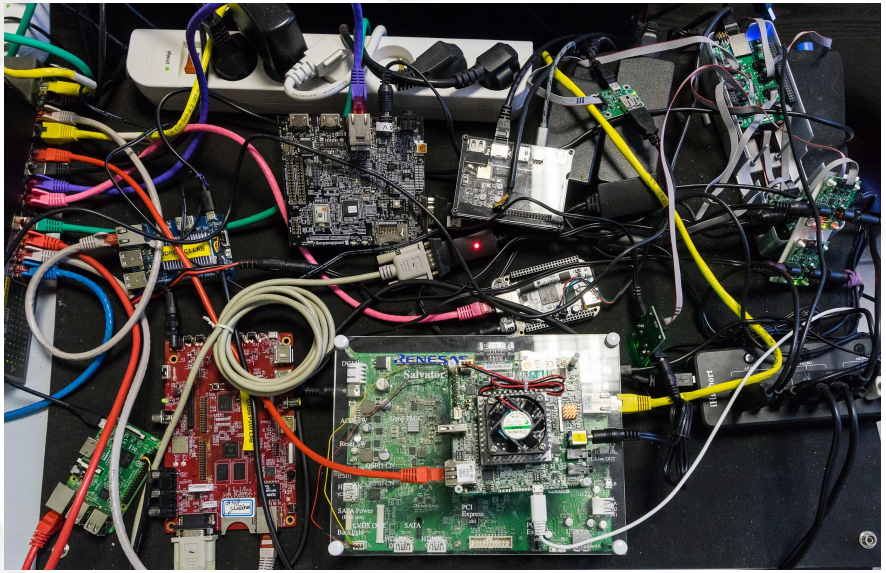
Docker compose

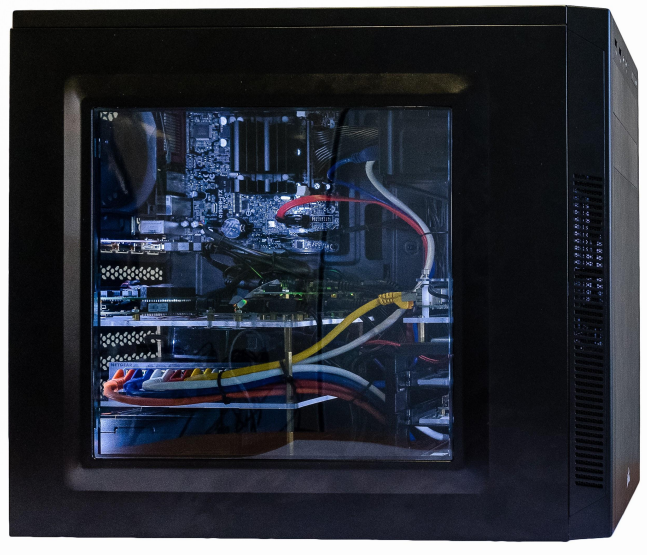


Important notice

- This is one HW implementation of the “Lab in a Box” concept
  - You may select your own components
    - Case, processing unit, power control unit, etc
  - You may decide to not integrate all the boards in the case
    - Large boards with accessories or test equipments
- The “Lab in a Box” SW does not depend on the HW, thanks to
  - LAVA HW abstraction layers,
  - Configuration files
  - Administration control panel

Achievements (1)





Achievements (2)

- Fully functional
- Complete CI LAVA lab integrated in single PC case
- No more wiring or boards laying on a desk / on shelves
- Fits well in our (small) apartments (for home workers)
- Good demonstrator for evangelising CI
- All DUT on drive trays, allowing easy maintenance
- Reasonable BOM cost (400 euros, excl. DUTs)
  - Reduced when recycling PC / USB Hub / Network Switch / ...
- Partially Automated SW installations (still under work)
- Containerized, scalable SW

Limitations

- Tedious (long) to build / Difficult to “mass produce”
- Requires good tinkering (incl. soldering) skills
- Heavily packed
- DUT size limited (2x 5”¼, 5x 3”½, height)
- Supports only +5V and +12V powered DUT
- DUT power consumption must be balanced across ATX connectors
  - Do not exceed 4A per pair of wires
- Using a larger PC case may not allow integrating many more DUT
  - Excessive internal wiring
- No standard “CI” connector
  - Custom wiring for each new DUT

What could be improved?

- Use a more powerful power supply
  - The more powerful the ATX power unit is, the more SATA/Molex connectors (i.e. power rails) we get
- Integration of larger development boards
- Administration control panel
  - Automatic detection and assignment of new devices
- Too complex & expensive for a tiny lab (1 or 2 boards)
- Documentation

What's next?

- "Lab in a Box" was a first experimentation to validate the concept
  - Low-cost,
  - Targeting individuals/groups with only a few boards
- Next:
  - Address "1-board lab" use-case ("LavaBox-mini")
  - Address Professional-grade "Lab in a ~~Box~~ Rack"
  - More SW installation automation
  - More SW administration automation
    - Including administration control panel
  - Connectivity (Wi-Fi / BT)
  - Integrate standard test jobs
  - Documentation
    - Basic docs and quick-start available in github project: https://github.com/kernelci/lava-docker/blob/master/README.md

Q+A

THANK YOU!

Come see inside the LAVAbox at the technical showcase !

(Tuesday evening)