Building and testing an automotive platform -

how Automotive Grade Linux is built and tested

Embedded Linux Conference Europe 2016

Jan-Simon Möller

Release Manager, AGL , The Linux Foundation

(jsmoeller@linuxfoundation.org, DL9PF @IRC and elsewhere)

LINUX FOUNDATION

AUTOMOTIVE GRADE LINUX

AGL - what ?

- Automotive Grade Linux is a Linux Distribution
- It is based on the
  Yocto Project/Openembedded
- Platform for multiple device profiles
  (IVI, telematics, …)

AGL - what ?

- Open Source and Code First
- Multiple Architectures:
  - x86 (e.g. Intel Minnowboard)
  - ARM 32 (e.g. Renesas Porter, TI Vayu, RaspberryPI)
  - ARM 64 (e.g. Qualcomm dragonboard 410c, rpi3?)

This talk ...

- why ? ... we do this
- what ? ... tools we use
- how ? ... we combine them
- what ? ... we want to achieve

Why ?
… we do this

Why ?

AGL development …

- AGL development is done in a distributed way
- Developers around the globe contribute
- Code review

Why ?

- Does it build ?
- Does it work ?
  - on board/arch A ?
  - on board/arch B ?
  - on board/arch C ?

    …
  - on board/arch <n> ?

Image: public domain

Why ?

- This must be a common problem!
  - just see how many talks during ELCE we have ;)
  - multiple solutions – good!
  - different use-cases
- Here is what we use! (Well, we think it does the trick – ideas/feedback welcome! Curious to hear your ideas !)

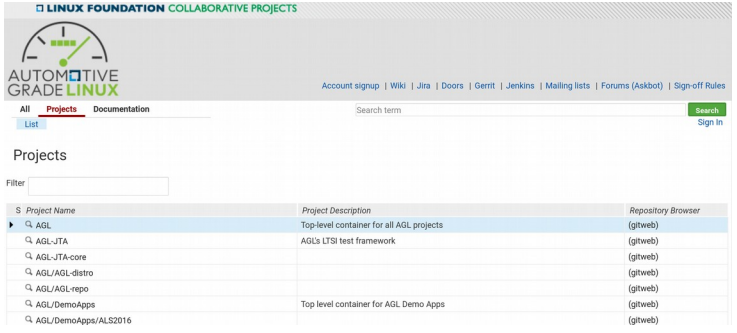
What ?
… tools we use

What ? … tools we use

- SCM + Review: Gerrit (sorry, Greg)
- CI Builds: Jenkins
- Tests on HW: a) AGL-JTA (Fuego)
  b) LAVA
- Data-Postproc … ???

SCM/Code-Review - Gerrit

- https://git.automotivelinux.org
- AGL-related projects in AGL/*
- if we are upstream → /src/*
- to try out code → /staging/*



- we use "repo" to pull down the git repositories

LINUX FOUNDATION

AUTOMOTIVE GRADE LINUX

SCM/Code-Review - Gerrit

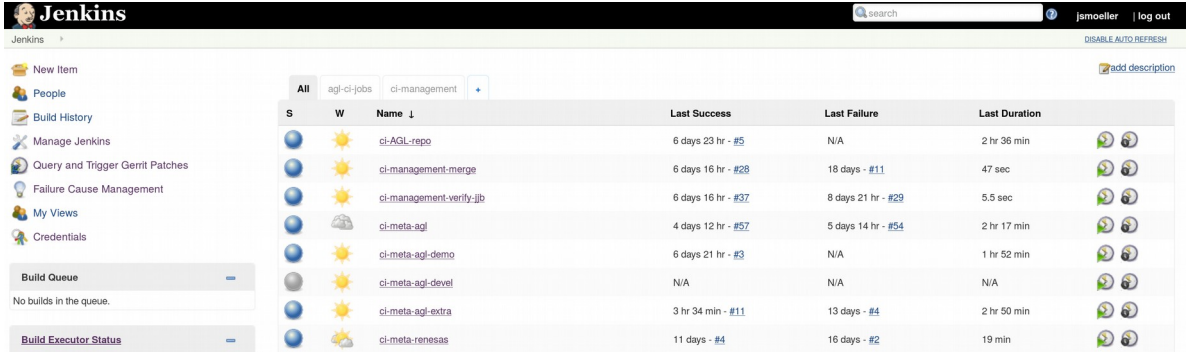
- All code that goes into AGL/* needs to work on all reference and community platforms

  → Test matrix:

  - Renesas Porter
  - Intel Minnowboard
  - Qemux86-64 (emulator)
  - DragonBoard
  - TI Vayu
  - NXP Wandboard, Sabre
  - RPI 2/3
  - ...

CI Builds - Jenkins

- https://jenkins-new.automotivelinux.org
  - Standard Jenkins
    - + gerrit-trigger plugin (to poll git.automotivelinux.org)
    - + openstack cloud plugin (to start jenkins slaves/minions)
      - + slaves run off identical base-images
  - CI-jobs created with Jenkins-Job-Builder (yaml)



LINUX FOUNDATION

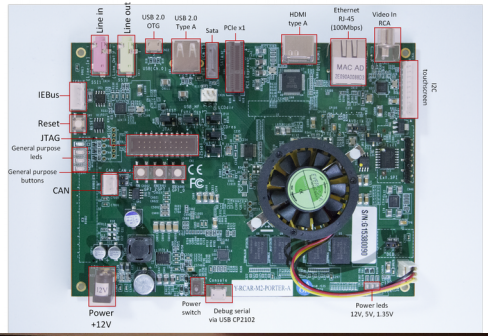
AUTOMOTIVE GRADE LINUX

CI Builds - Jenkins

- A successful build will vote "Verified +1"
- A failed build will vote "Verified -1"

  in Gerrit
- You need to define the success/failure criteria
  - Starts with "it builds" (yay!)
  - Ends with "it boots, runs, passes all tests, updates cleanly, communicates with X and shuts-down properly"

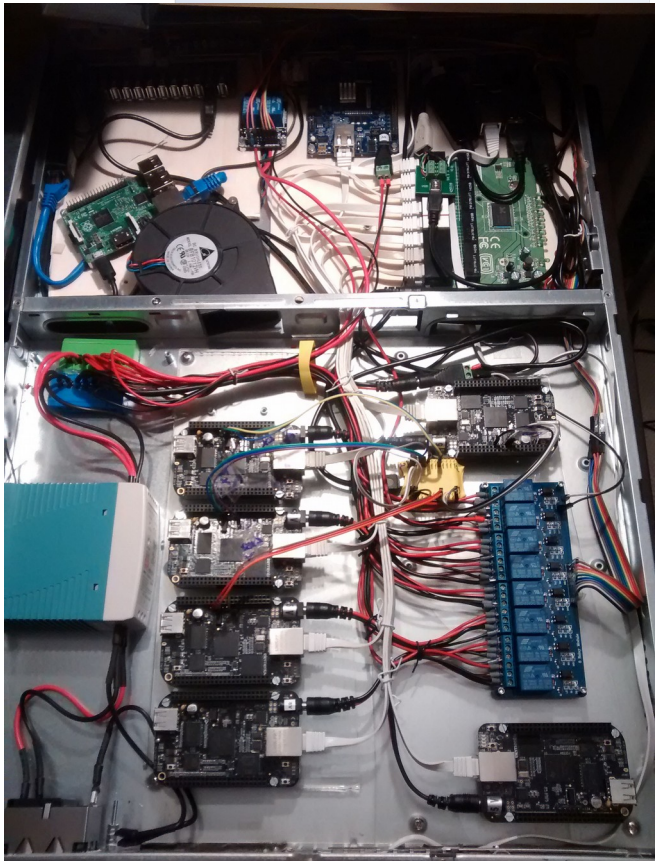
Tests on HW



Tests on HW are hard !?

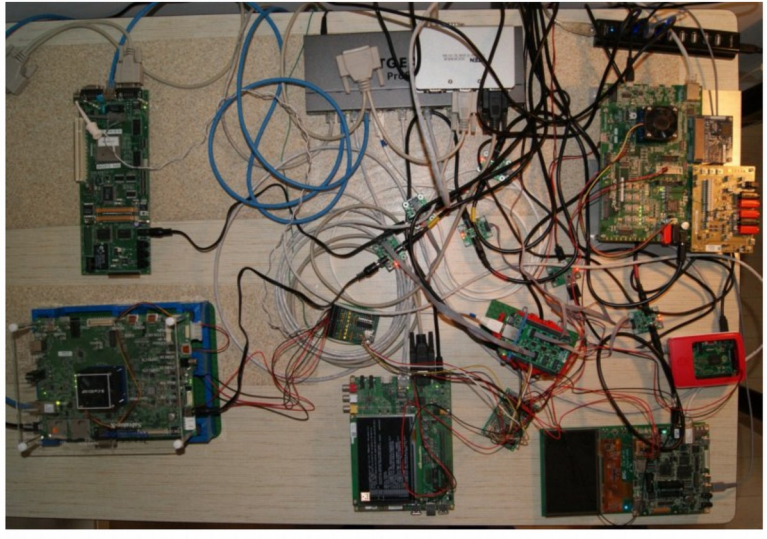
- You need the HW
- You need it on your desk/in your lab
- You need to deploy firmware/filesystems
- You need to reboot the board
- You need to initiate the test
- You need to collect the results
- You need to interpret the results

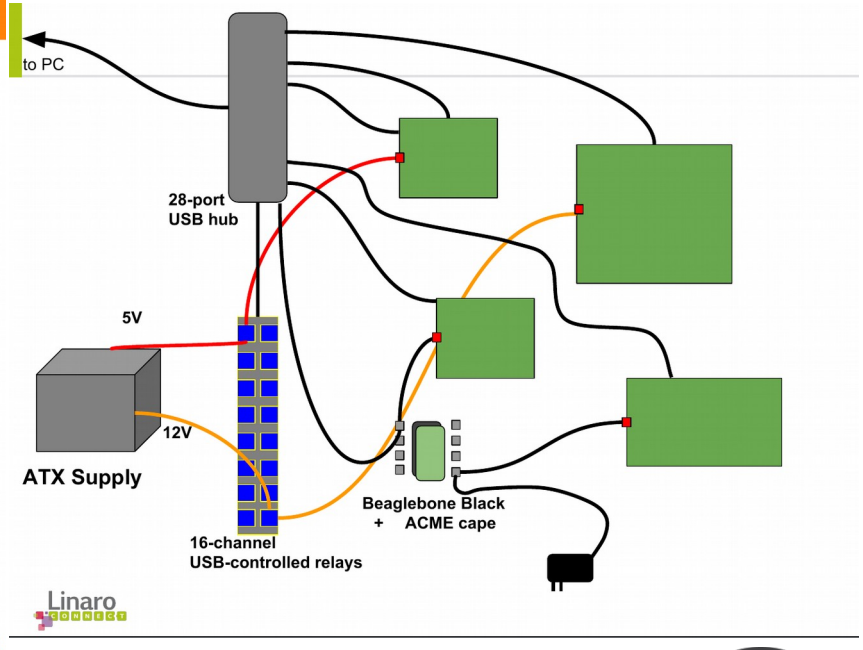
… rinse & repeat



Lab setups … (from ELCE slides)

Board Farm Bird View





Documenting our Lab setup

- WIP document for LAVA:

  http://bit.ly/lavasetup
- Doc for AGL-JTA/Fuego:

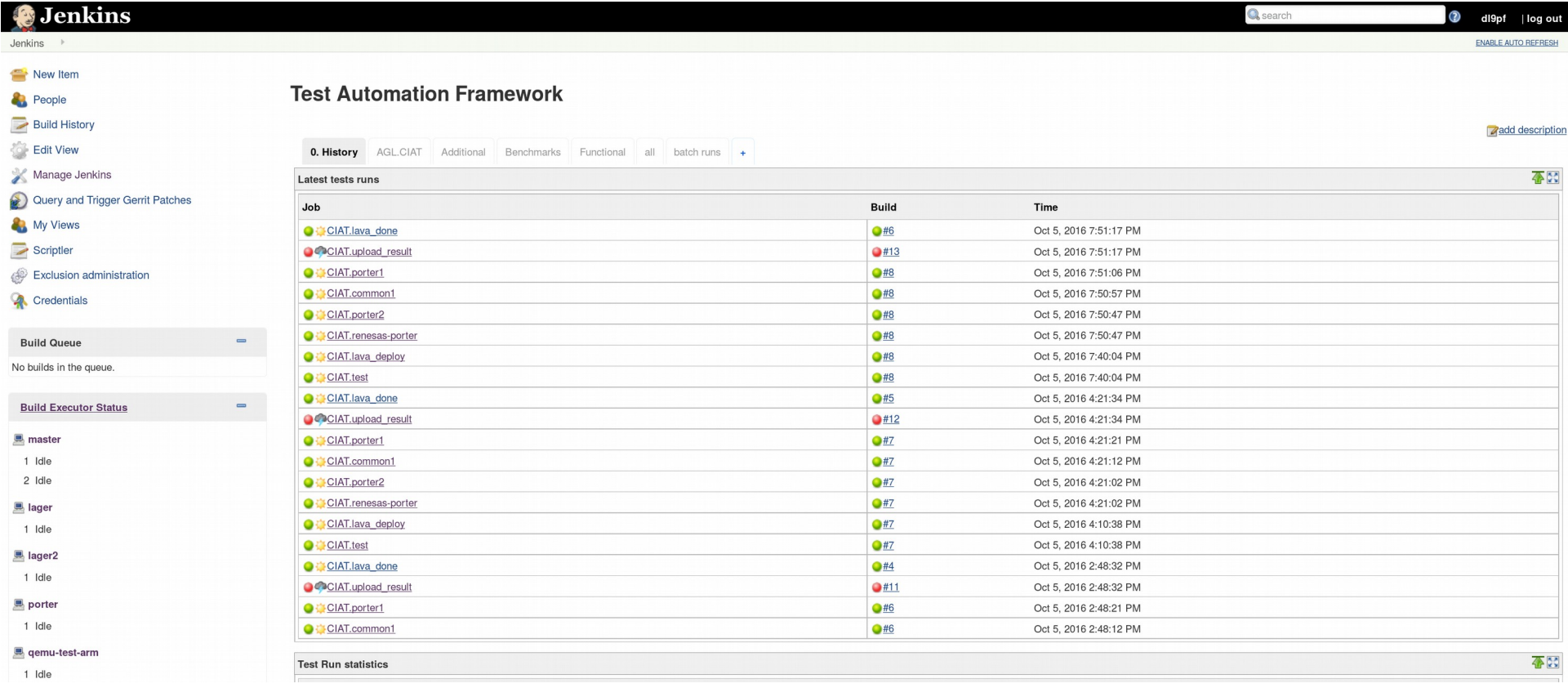
  https://git.automotivelinux.org/gerrit/gitweb?p=AGL-JTA.git;a=tree;f=docs
- Wiki page in AGL wiki:
  - https://wiki.automotivelinux.org/agl-testframework

Tests and frameworks ...

- AGL-JTA (modified/patched Fuego)
  - https://git.automotivelinux.org/gerrit/gitweb?p=AGL-JTA.git
  - jta.automotivelinux.org (Live instance - WIP)
  - runs tests on target boards and collects results
    - results end up right now in a git repo

  ☺ large set of Tests, postproc capabilities

  ☹ Installation, modification, board local (pwr/ssh)



Tests and frameworks ...

- LAVA
  - https://validation.linaro.org
  - https://porter.automotivelinux.org/scheduler/alljobs
  - board farm management + test executor
  - grabs board from pool, pwr & boot & test

    ☺ multiple boards per type, remote lab (WIP),

    ☺ runs even on RPI2/3 with PiFACE ! (2 DUT)

    ☹ first setup little hard, doc for satellite labs

All Jobs

All Jobs | Active Jobs | Queued Jobs | Pipeline Jobs | Healthcheck Jobs

Show 25 entries Search

ID	Status	Priority	Device	Description	Submitter	Submit Time	End Time	Duration
163	Complete	Medium	porter-01	lava-boot job for jta-agl	jta-agl	Oct. 5, 2016, 7:40 p.m.	Oct. 5, 2016, 7:51 p.m.	0:10:58.318669
162	Complete	Medium	porter-01	lava-boot job for jta-agl	jta-agl	Oct. 5, 2016, 4:10 p.m.	Oct. 5, 2016, 4:21 p.m.	0:10:45.334469
161	Complete	Medium	porter-01	lava-boot job for jta-agl	jta-agl	Oct. 5, 2016, 2:37 p.m.	Oct. 5, 2016, 2:48 p.m.	0:10:44.438036
160	Complete	Medium	porter-01	lava-boot job for jta-agl	jta-agl	Oct. 5, 2016, 1:40 p.m.	Oct. 5, 2016, 1:51 p.m.	0:11:08.490490
159	Complete	Medium	porter-01	lava-boot job for jta-agl	jta-agl	Oct. 5, 2016, 8:44 a.m.	Oct. 5, 2016, 8:55 a.m.	0:11:00.376715
158	Complete	Medium	porter-01	lava-boot job for jta-agl	jta-agl	Oct. 4, 2016, 9:32 p.m.	Oct. 4, 2016, 9:44 p.m.	0:10:56.294506
157	Canceled	Medium	porter-01	lava-boot job for jta-agl	jta-agl	Oct. 4, 2016, 9:11 p.m.	Oct. 4, 2016, 9:33 p.m.	0:18:15.838088
156	Canceled	Medium	porter-01	lava-boot job for jta-agl	jta-agl	Oct. 4, 2016, 8:50 p.m.	Oct. 4, 2016, 9:15 p.m.	0:24:15.917332
155	Complete	Medium	porter-01	lava-boot job for jta-agl	jta-agl	Oct. 4, 2016, 4:08 p.m.	Oct. 4, 2016, 4:19 p.m.	0:10:52.630930
154	Complete	Medium	porter-01	lava-boot job for jta-agl	jta-agl	Oct. 4, 2016, 1:13 p.m.	Oct. 4, 2016, 1:24 p.m.	0:10:46.719346
153	Complete	Medium	porter-01	lava-boot job for jta-agl	jta-agl	Oct. 4, 2016, 12:04 p.m.	Oct. 4, 2016, 12:15 p.m.	0:10:55.134659

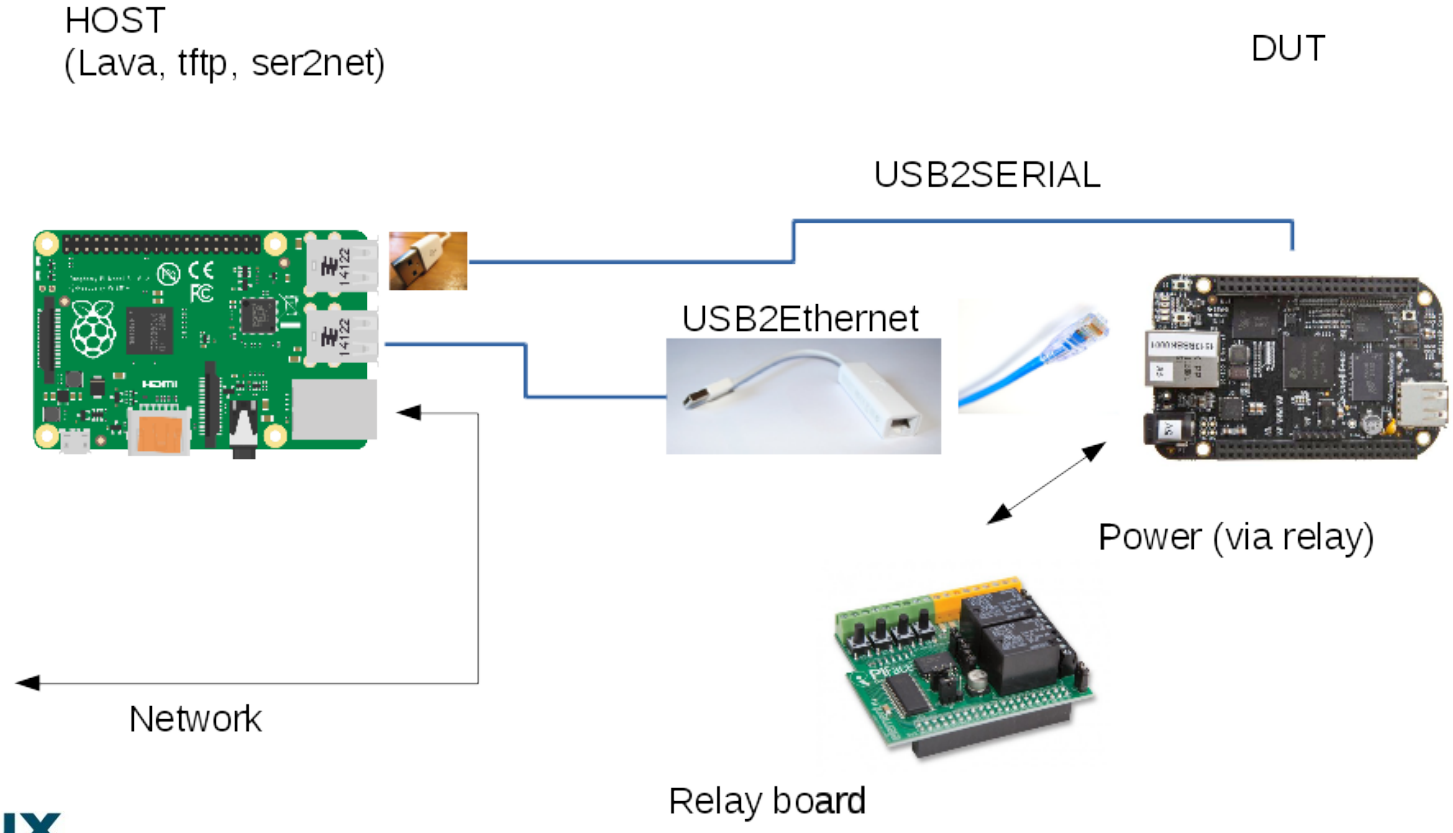
Documenting our Lab setup

- WIP document for LAVA:

  http://bit.ly/lavasetup
- Doc for AGL-JTA/Fuego:

  https://git.automotivelinux.org/gerrit/gitweb?p=AGL-JTA.git;a=tree;f=docs
- Wiki page in AGL wiki:
  - https://wiki.automotivelinux.org/agl-testframework

BOM <= 100 €



Data-Postproc … ???

- Investigating
  - In fuego (AGL-JTA) ??
  - Other mechanism ??
  - What data to track at all ??

  → You need to define your key indicators

Feedback to Developers

- In our case – right in gerrit:

Code
Review
(human)

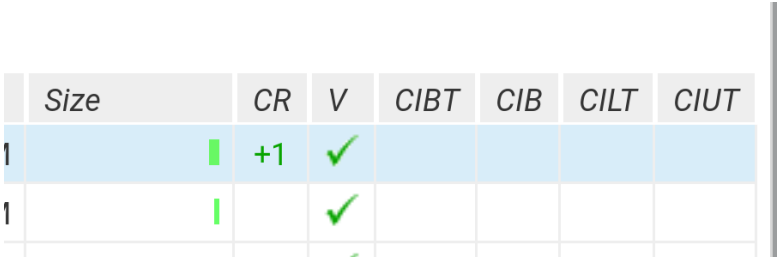
Verified
(CI complete)

CI-**I**mage-**B**uild
("It builds")

CI-**I**mage-**B**oot-**T**est
("It boots on HW")

CI-**I**mage-**L**TSI-**T**est
("The tests pass")

CI-**I**mage-**U**I-**T**est
("The UI tests pass")

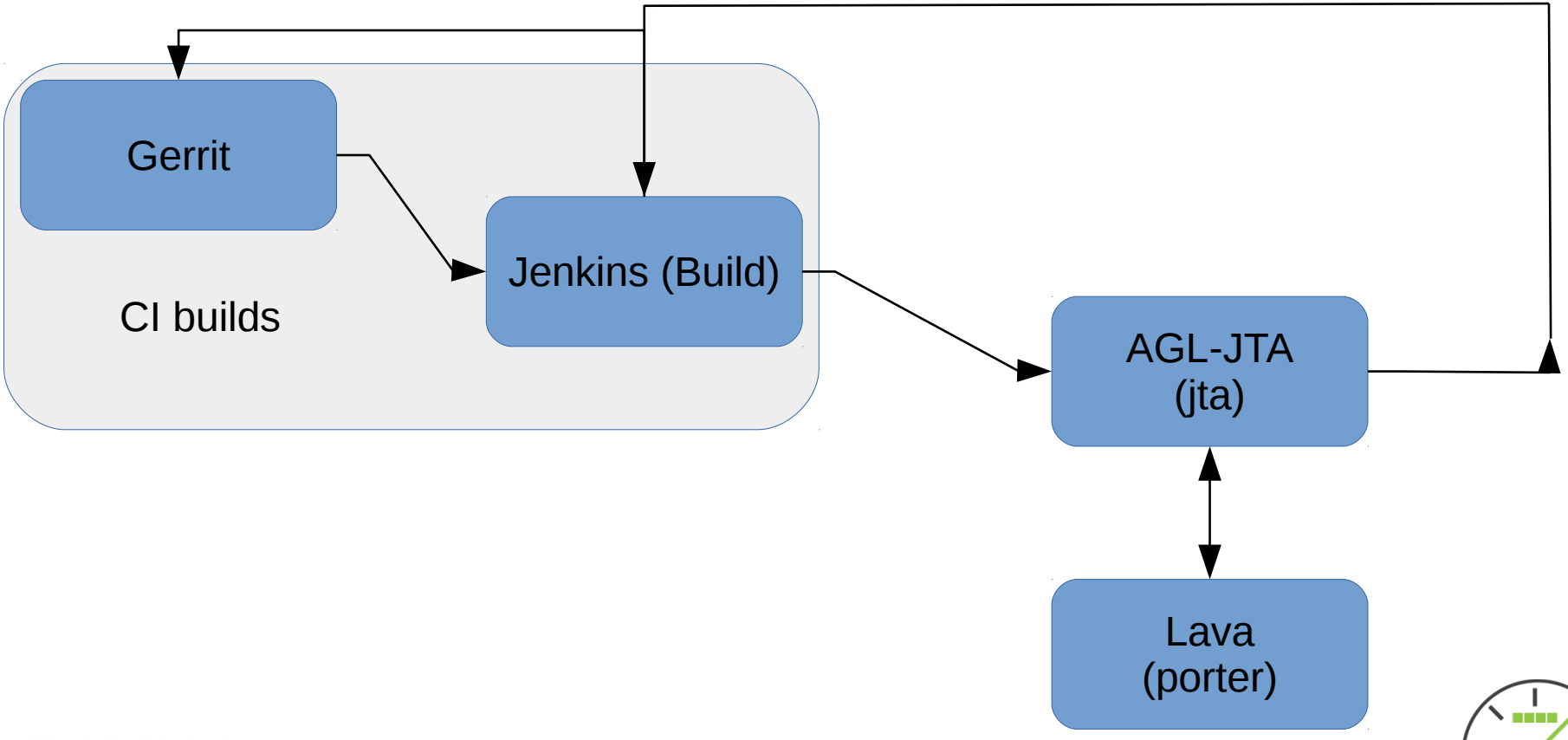


LINUX FOUNDATION

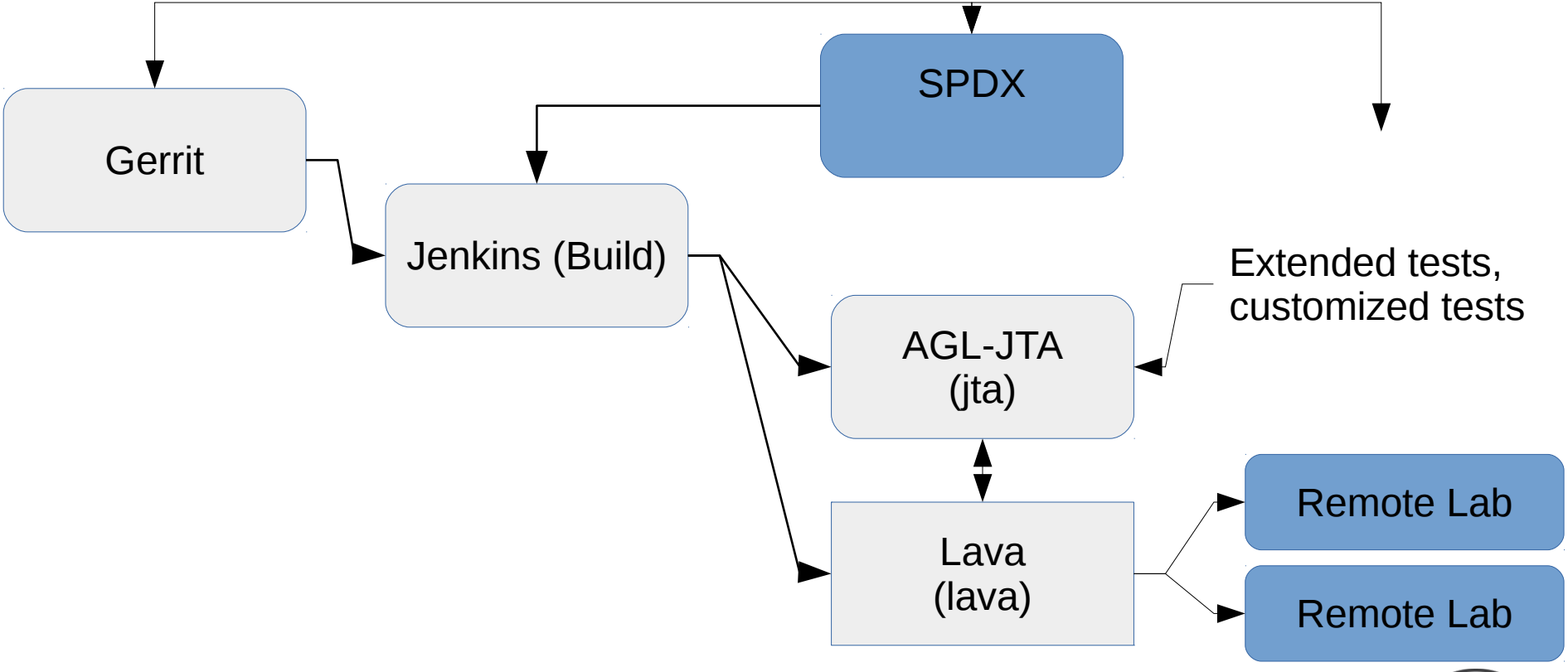
AUTOMOTIVE GRADE LINUX

How ? … we combine them

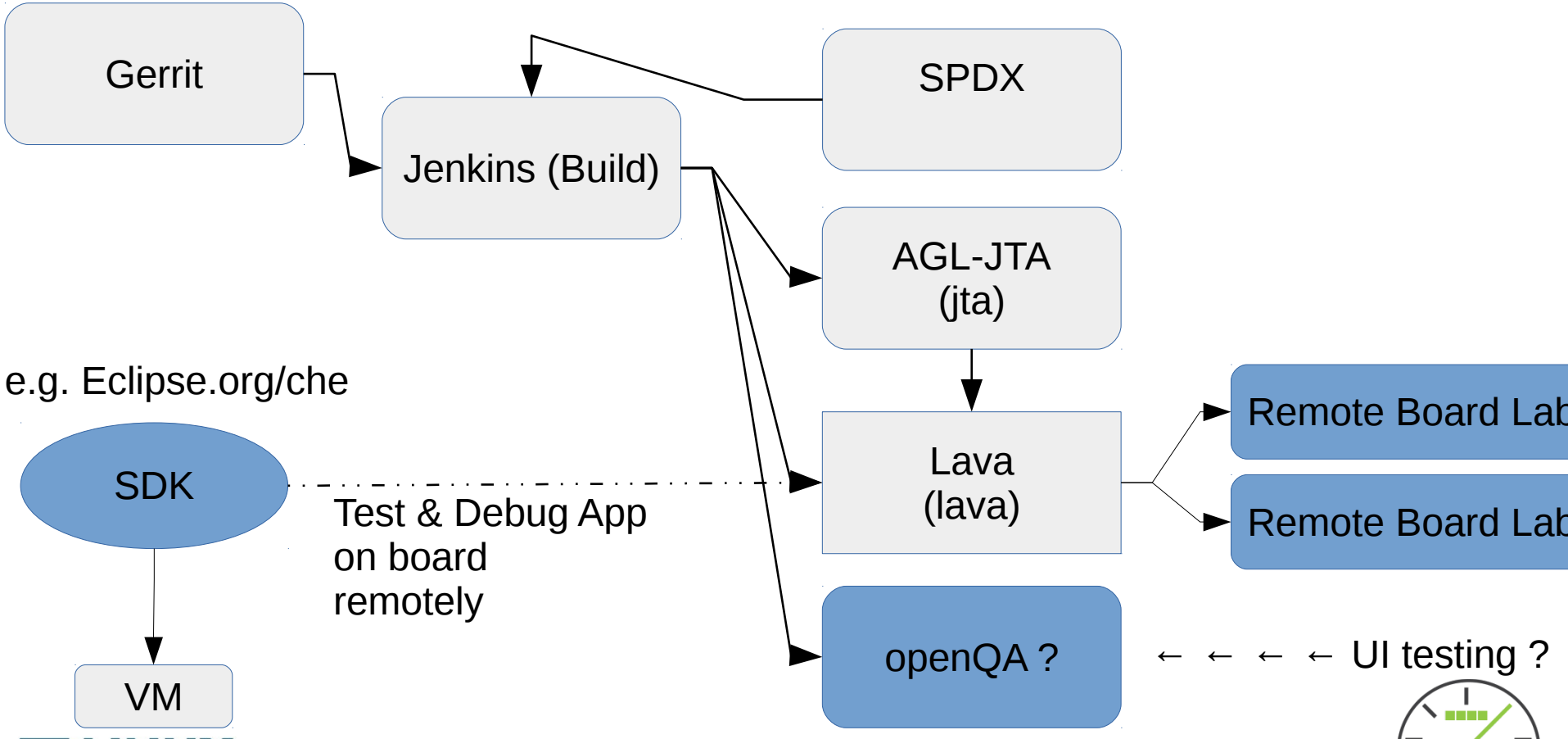
Present



Plan (short/mid)



Plan (mid/long)



LINUX FOUNDATION

AUTOMOTIVE GRADE LINUX

What ? … we want to achieve

What ? ... we want to achieve (Vision)

- Stable and tested platform to build-upon
  - wide range of devices
- Fast development through 'instant' feedback
  - developers work remotely, not all boards available
- Easy development through direct test on hw
  - remote testing capabilities in combination with SDK

Q/A ?!

THANK YOU