Advanced Test Harness Infrastructure for Validating ARM and FPGA-based Systems

Raus Stefan

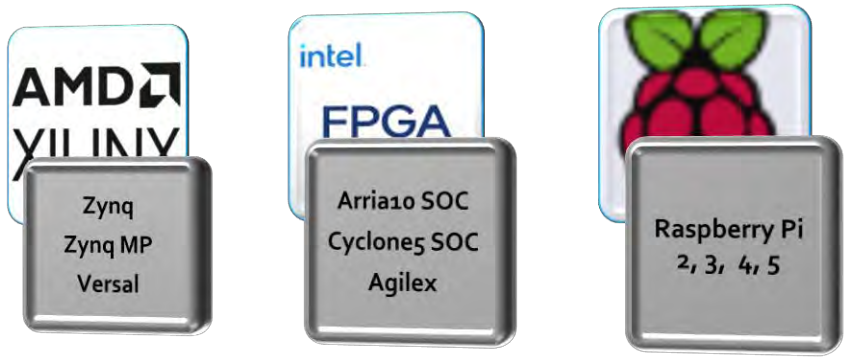
The Need


Automated Tests
Integration
Performance Testing
Regression Testing
Manual Testing
Smoke Testing
Security Testing
Quality
Coverage
Continuous Testing
Production
Unit-level Tests


IS MY CODE BAD?
NO, THE TESTS ARE WRONG
imgflip.com

Analog Devices Kuiper Linux

- Free, open-source Embedded Linux Distribution
- Based on Debian, but customized ADI products, including:
  - 1300+ Linux Device drivers for Analog Devices products
  - Pre-built boot files for 140+ FPGA-based platforms and 30+ RPi-based designs
  - Development applications and tools, software libraries and a variety of examples for quickly ramping up projects

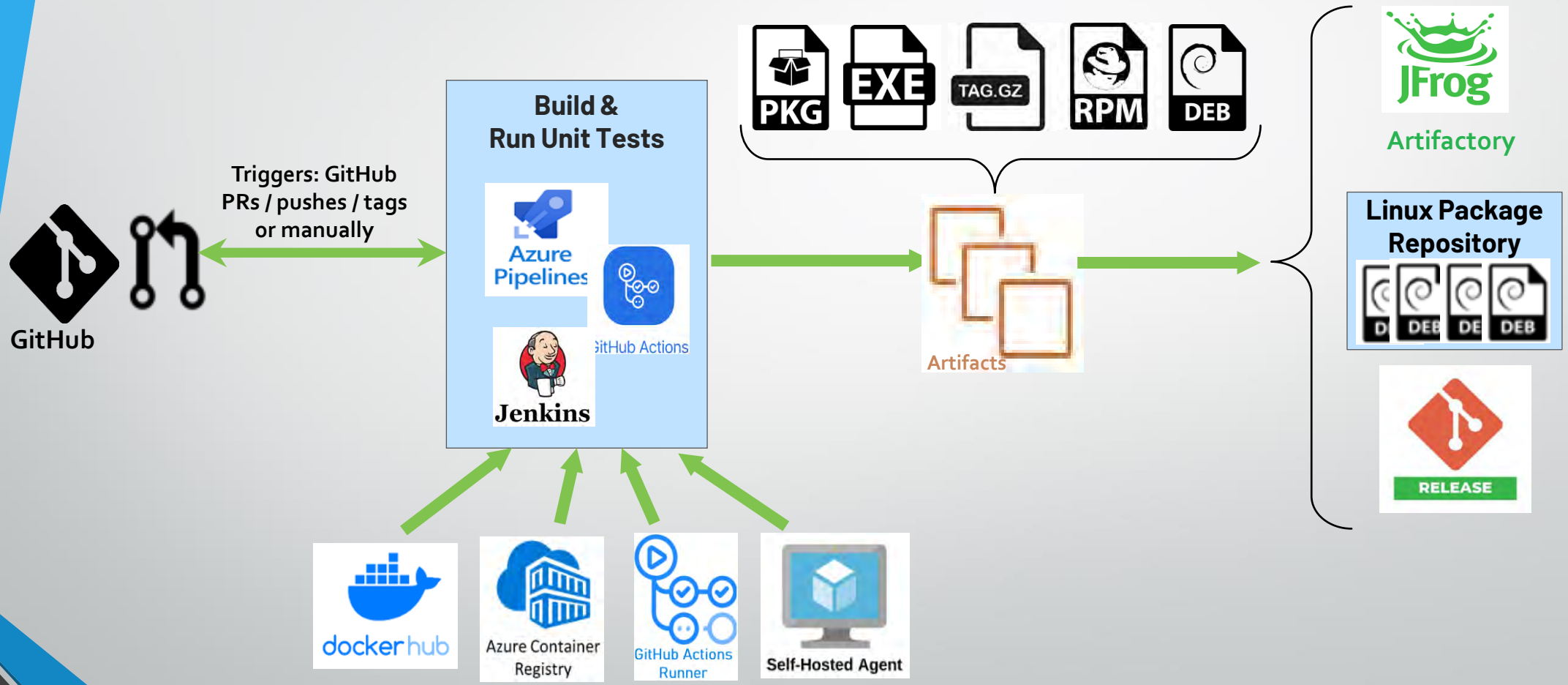


GitHub Repo: https://github.com/analogdevicesinc/adi-kuiper-gen/tree/staging/kuiper2.0

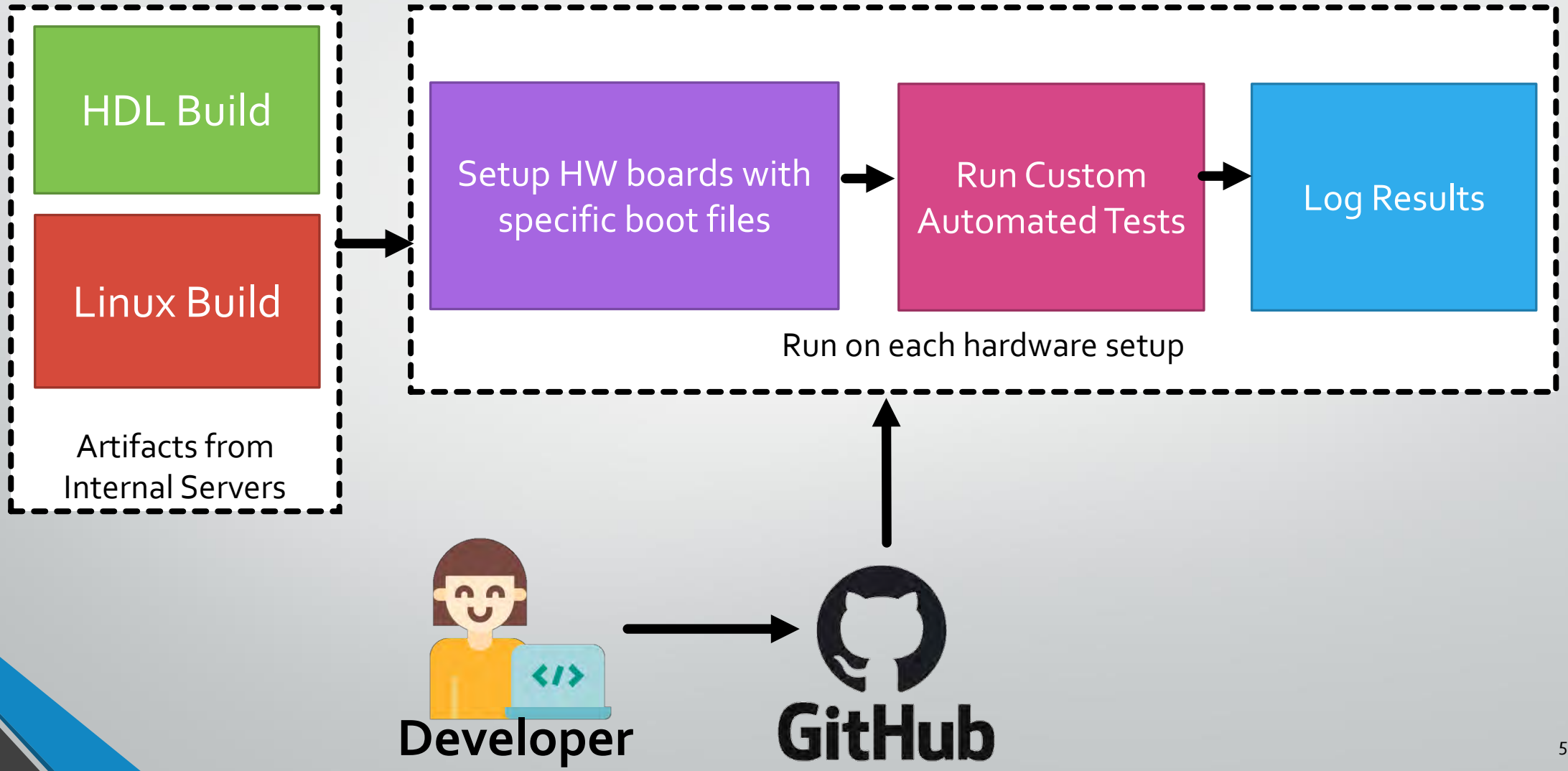
Kuiper Overview: https://www.analog.com/en/resources/evaluation-hardware-and-software/embedded-development-software/kuiper-linux.html

Conf42.Devops – Optimizing Kuiper Linux release: "Refining the Release Strategy of a Custom Linux Distro"

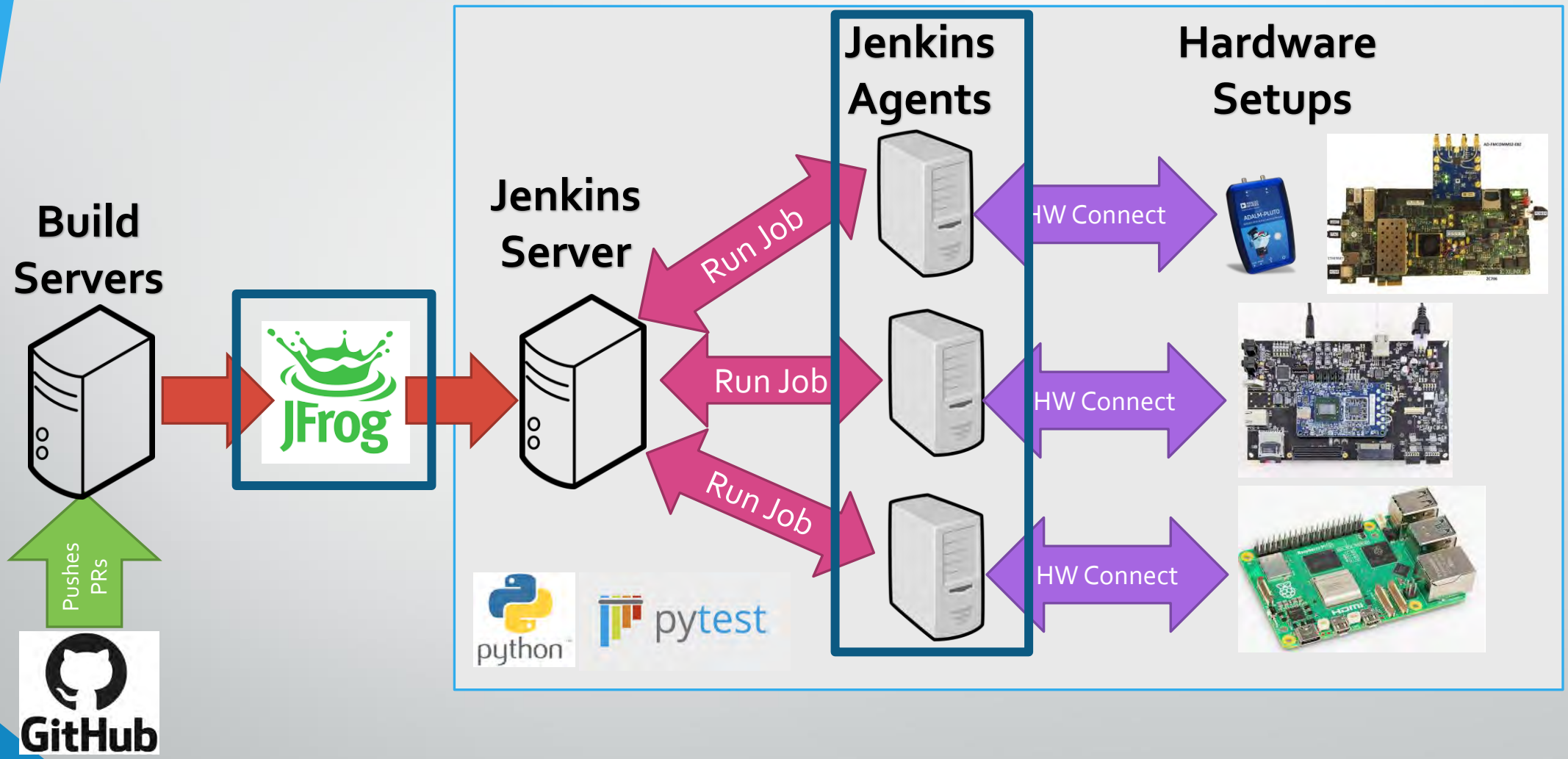
Continuous Integration



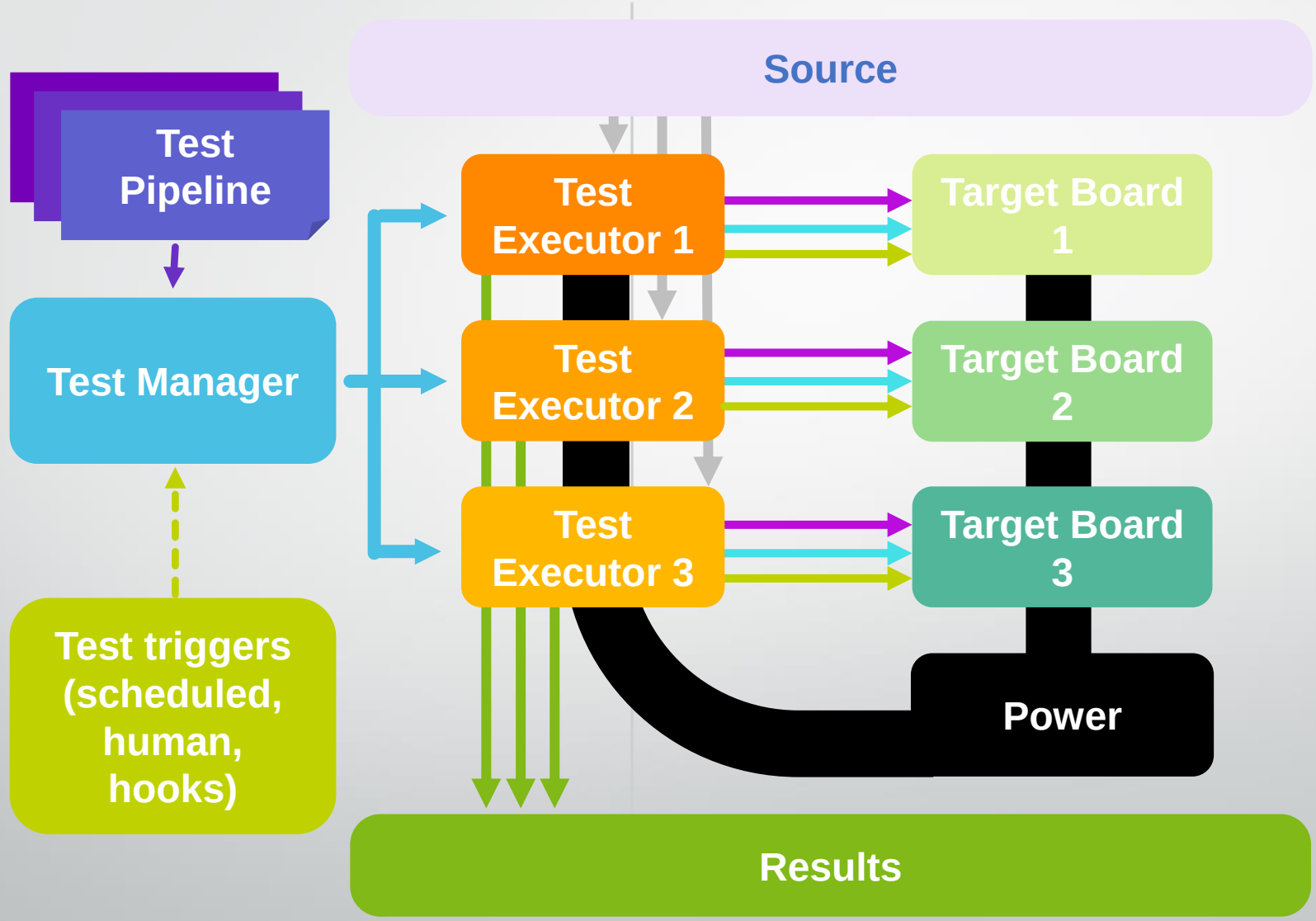
Test Flow


HDL Build
Linux Build
Artifacts from Internal Servers
Setup HW boards with specific boot files
Run Custom Automated Tests
Log Results
Run on each hardware setup
Developer
GitHub

Hardware Test Harness Topology



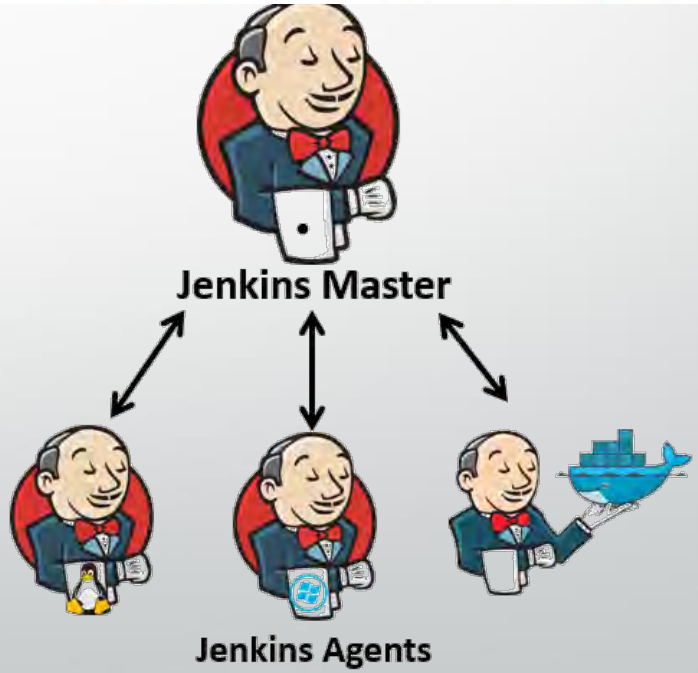
Managing automated tests



Tools Used (1)

Jenkins

- Can be hosted without external servers
- No requirement on source control
- Better cross project flows
- Easy integration with different tools
- Declarative / Scripted groovy pipelines
- Big online community and many plugins
- JSL, DSL, triggering, locking resources



Tools Used (2)

Nebula

- "in-house" developed tool, written in pure python and not reliant on other tools or shelling out
- Library for managing hardware setups interfaces, such as UART, JTAG, Ethernet, USB, PDU etc
- Designed to handle whole booting process for different hardware platforms in different ways and with different failure modes
- Managing also the situation when hardware setups hang on booting, or binaries to be tested are completely broken

Manager for:

UART	Ethernet	JTAG	PDU	USB	Others

https://nebula-fpga-dev.readthedocs.io

Tools Used (3)

NetBox

- Used for organizing the DUTs, Jenkins agents, PDU outlets, Ethernet switches and other connections in the test harness
- Used to generate **nebula-config.yml** which contains information about DUTs, such as:
  - platform, board connected to it
  - boot files used by DUT from SD Card
  - Ethernet and serial addresses
- It requires maintenance only if there are changes in the hardware setup such as:
  - Ethernet address or connections changes
  - DUTs added / removed / rearranged

https://netboxlabs.com/docs/netbox/en/stable/

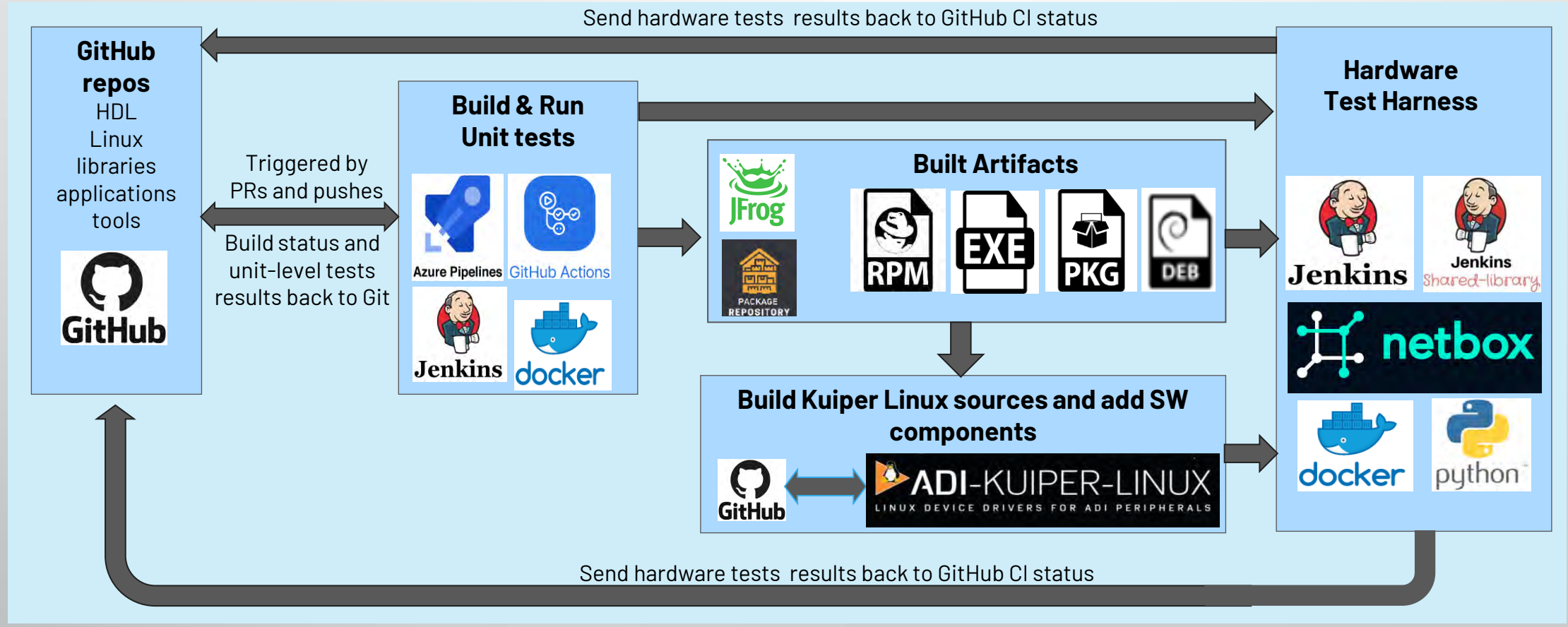
Tools Used (4)

Jenkins Shared Library

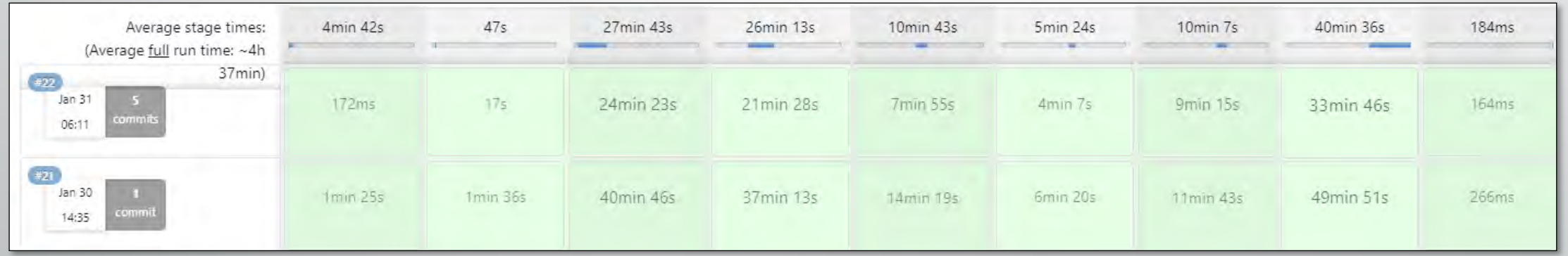
- **JSL** is a groovy library for automating test execution and managing pipelines
- Comprehensive library that handles setting up virtual environments, locking resources, executing tests, and collecting results
- Some examples where JSL is used:
  - Update agents' tools (nebula, docker)
  - Update BOOT files
  - Log test result artifacts

https://github.com/sdgtt/jenkins-shared-library

Continuous Testing



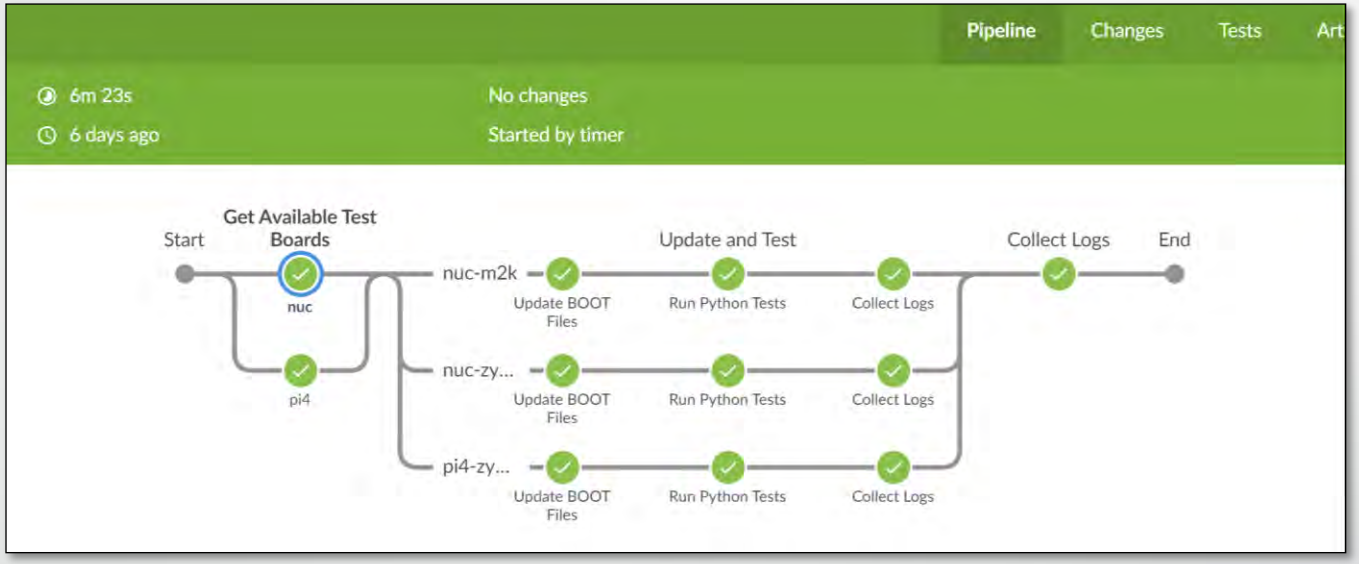
Accessing tests results



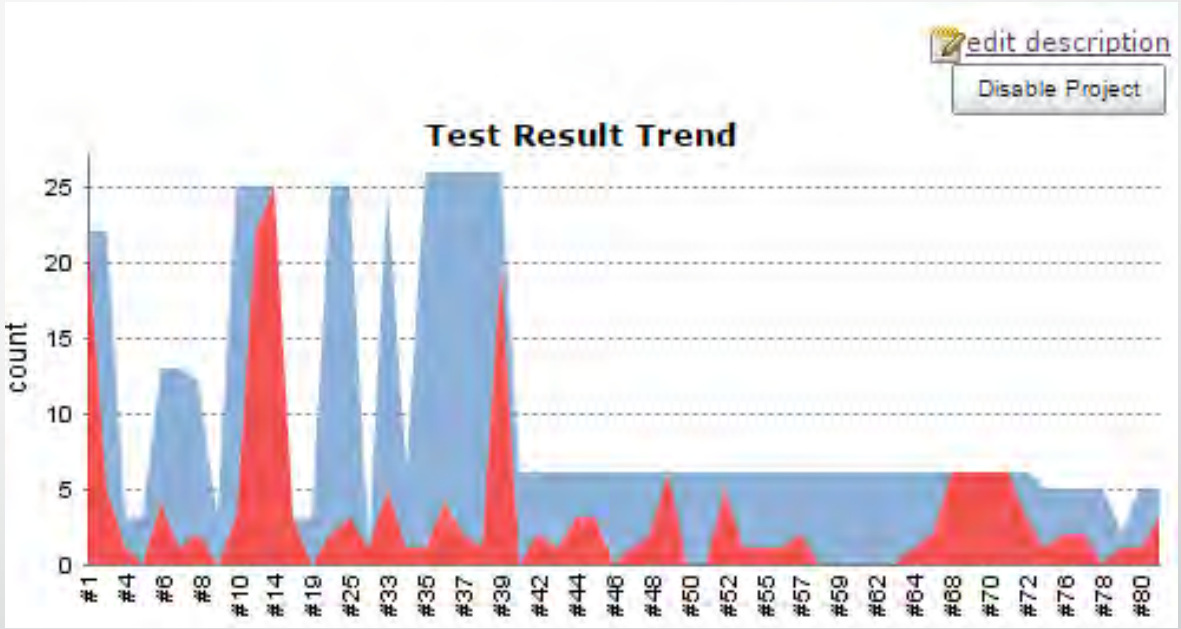
Average stage times: (Average full run time: ~4h 37min)	4min 42s	47s	27min 43s	26min 13s	10min 43s	5min 24s	10min 7s	40min 36s	184ms
#22 Jan 31 06:11 — 5 commits	172ms	17s	24min 23s	21min 28s	7min 55s	4min 7s	9min 15s	33min 46s	164ms
#21 Jan 30 14:35 — 1 commit	1min 25s	1min 36s	40min 46s	37min 13s	14min 19s	6min 20s	11min 43s	49min 51s	266ms

Accessing tests results

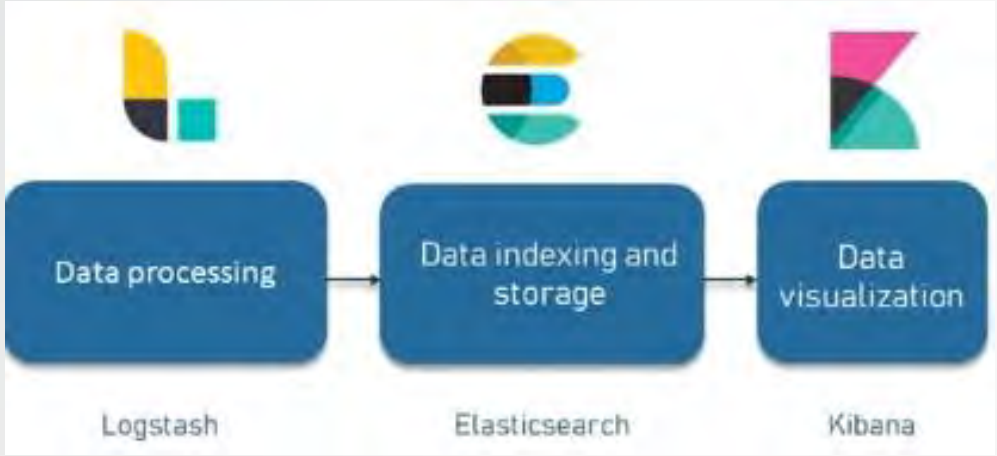
Jenkins Blue Ocean



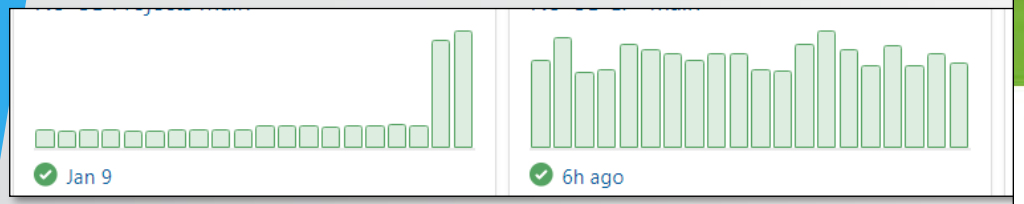
Accessing tests results

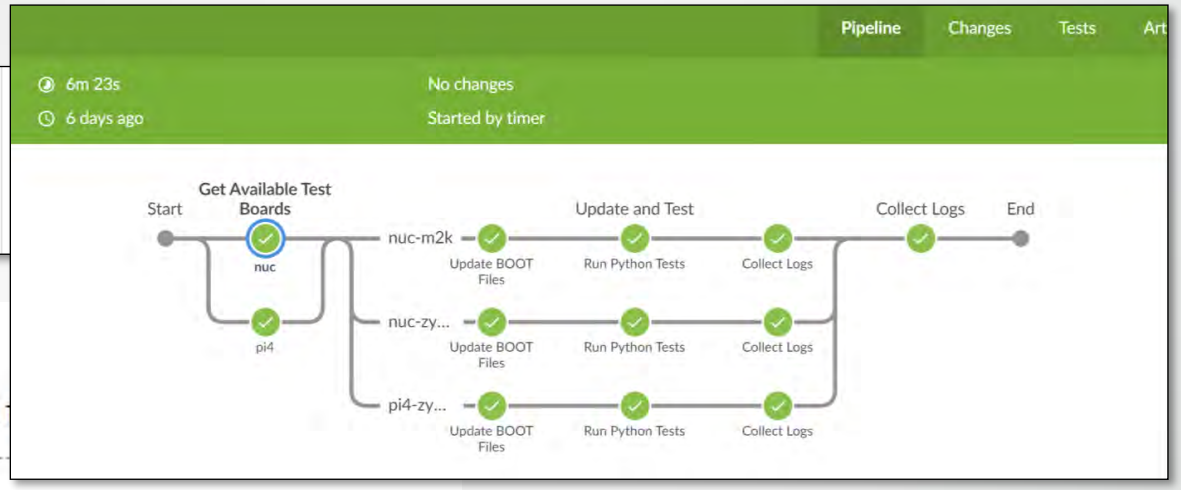


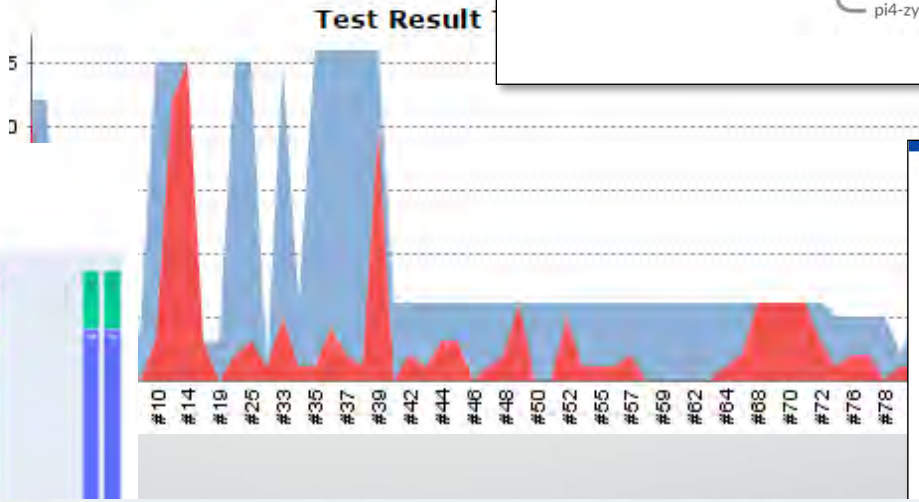
Accessing tests results

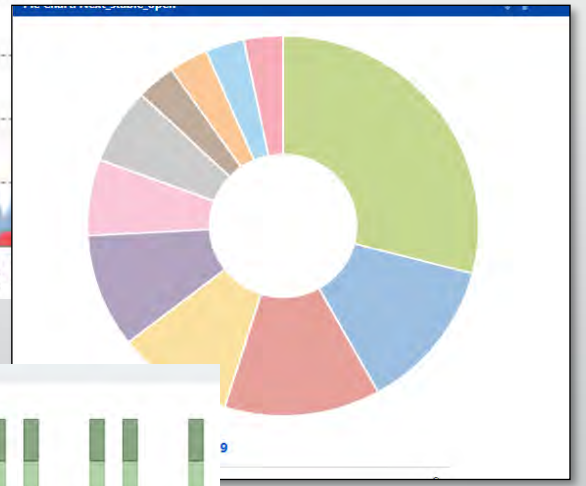


Accessing tests results









Accessing tests results

output.md Raw

Started by upstream project "main/main_HDL_Commit" build number 407

originally caused by: Started by GitHub push by IuliaCMoldovan

Libraries with CRITICAL WARNINGS:

- interfaces/interfaces_ip

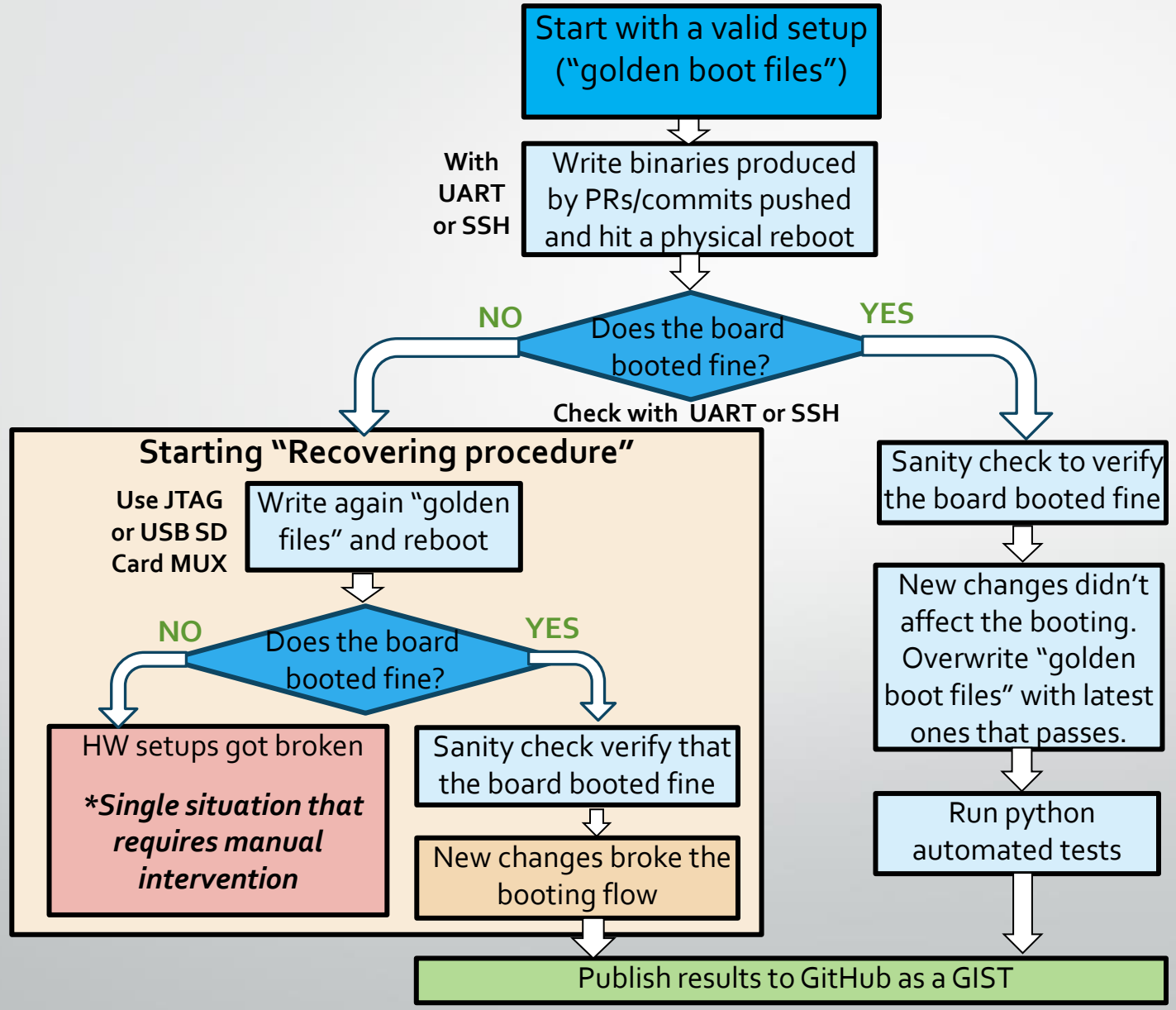
Project	Build number	Status	Hardware Test Results
ad4052_ardz.coraz7s	6	SUCCESS	
cn0577.zed	249	SUCCESS	N/A in boardfarm
ad777x_ardz.zed	247	SUCCESS	
adrv9026.a10soc	156	FAILURE	
ad485x_fmcz.zed	96	SUCCESS	
cn0561.de10nano	261	SUCCESS	
ad9656_fmc.zcu102	251	SUCCESS	
adrv9001.zed	247	SUCCESS	PASSING PASSING
dc2677a.c5soc	257	SUCCESS	N/A in boardfarm
ad719x_asdz.coraz7s	247	SUCCESS	

HW Test Result Summary

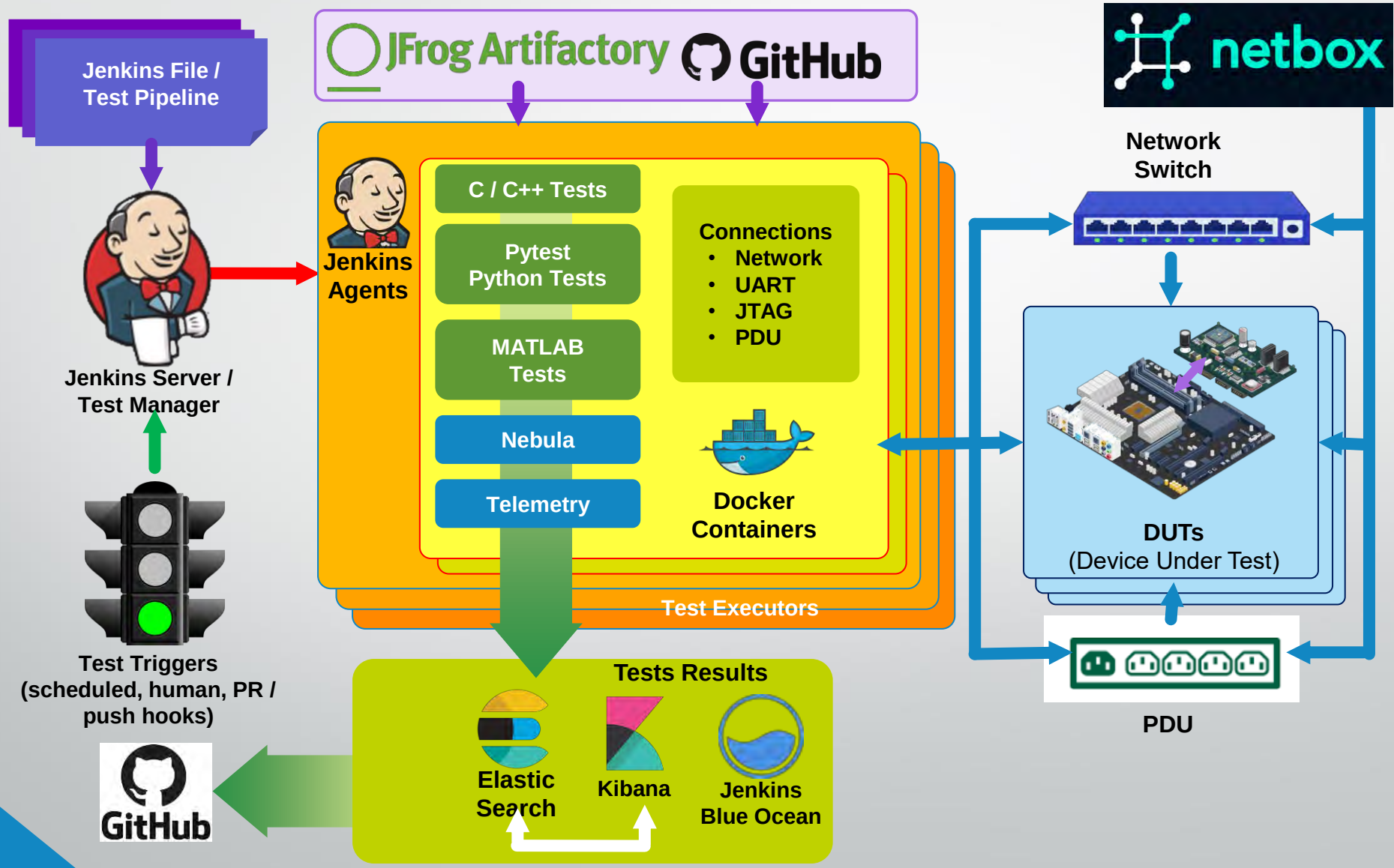
Stage	Result
U-Boot reached?	✔
Linux prompt reached?	✔
IIO Drivers	✔
DMESG	✔
PYADI-IIO Tests	✔

Conf42.Devops Presentation:
"Secure Integration of Private Testing Infrastructure with Public GitHub Repositories"

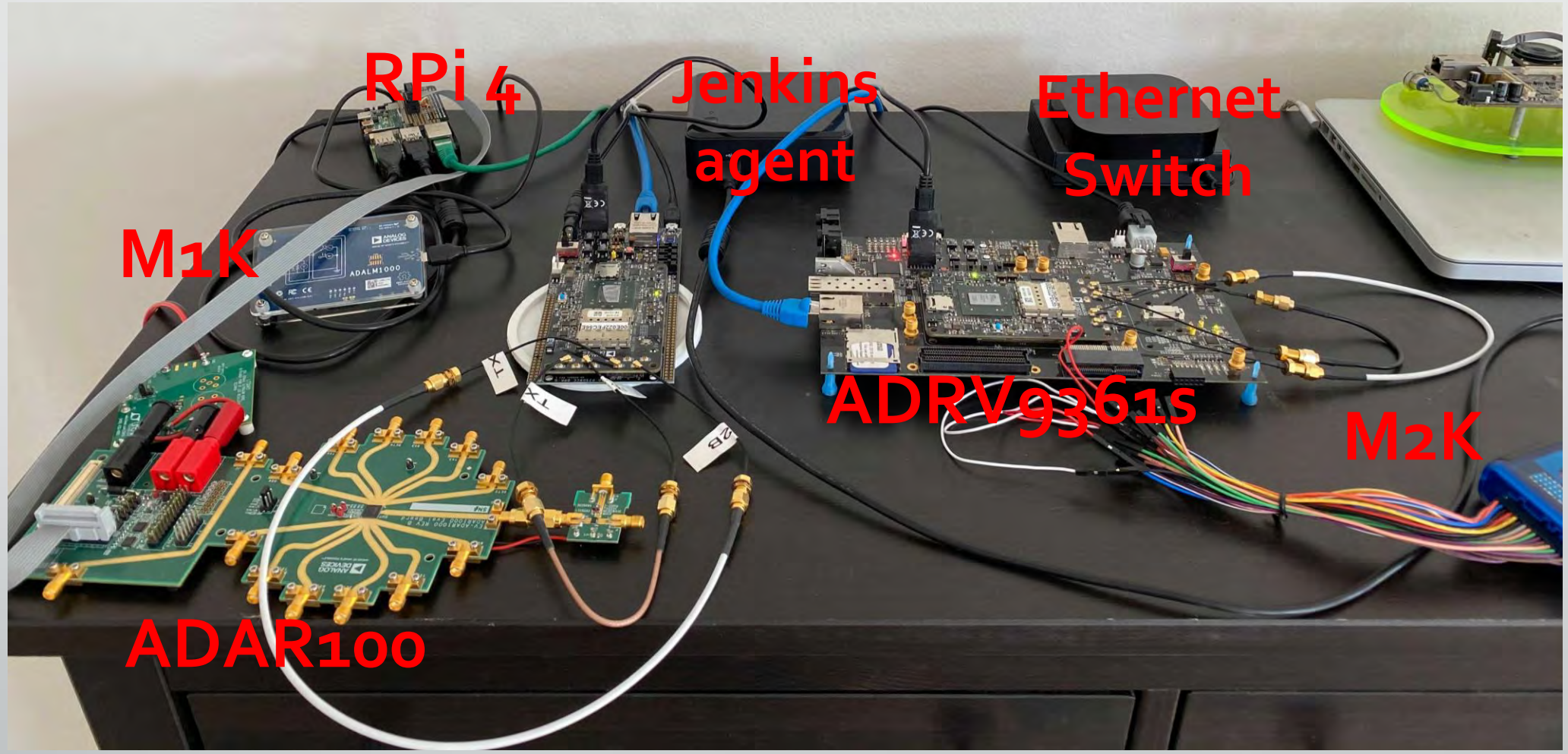
Recovering hardware setups



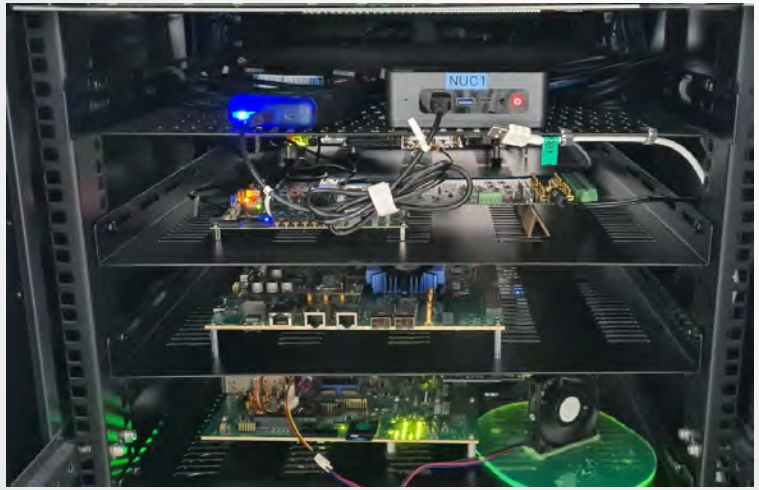
Hardware Test Harness Design



Hardware Test Harness in Real World



Hardware Test Harness in Real World



Conclusions

Hardware Test Harness:

- supports multiple platforms (FPGA- and ARM-based)
- tests can be written in different languages (C/C++, Python etc)
- complex triggering mechanism
- managed queue
- distributed across different physical locations
- robust recovery mechanism
- well-structured results returned in GitHub

Thank you !

Email: stefan.raus@analog.com