

KernelCI for Embedded: Enhancing Reliability through Scalable Test Automation

Paweł Wieczorek

OSS EU 2025



COLLABORA

Open First

Agenda

- What is KernelCI?
- Current status
- User stories
- Refining use cases
- Next steps
- Summary



What is KernelCI?

Simple CI loop

- Plan
- Build
- Test
- Evaluate
- Repeat

Various aspects to consider

- Monitoring changes in kernel trees
- Scheduling access to resources
- Processing and archiving results
- Maintaining dependencies

Hide implementation details

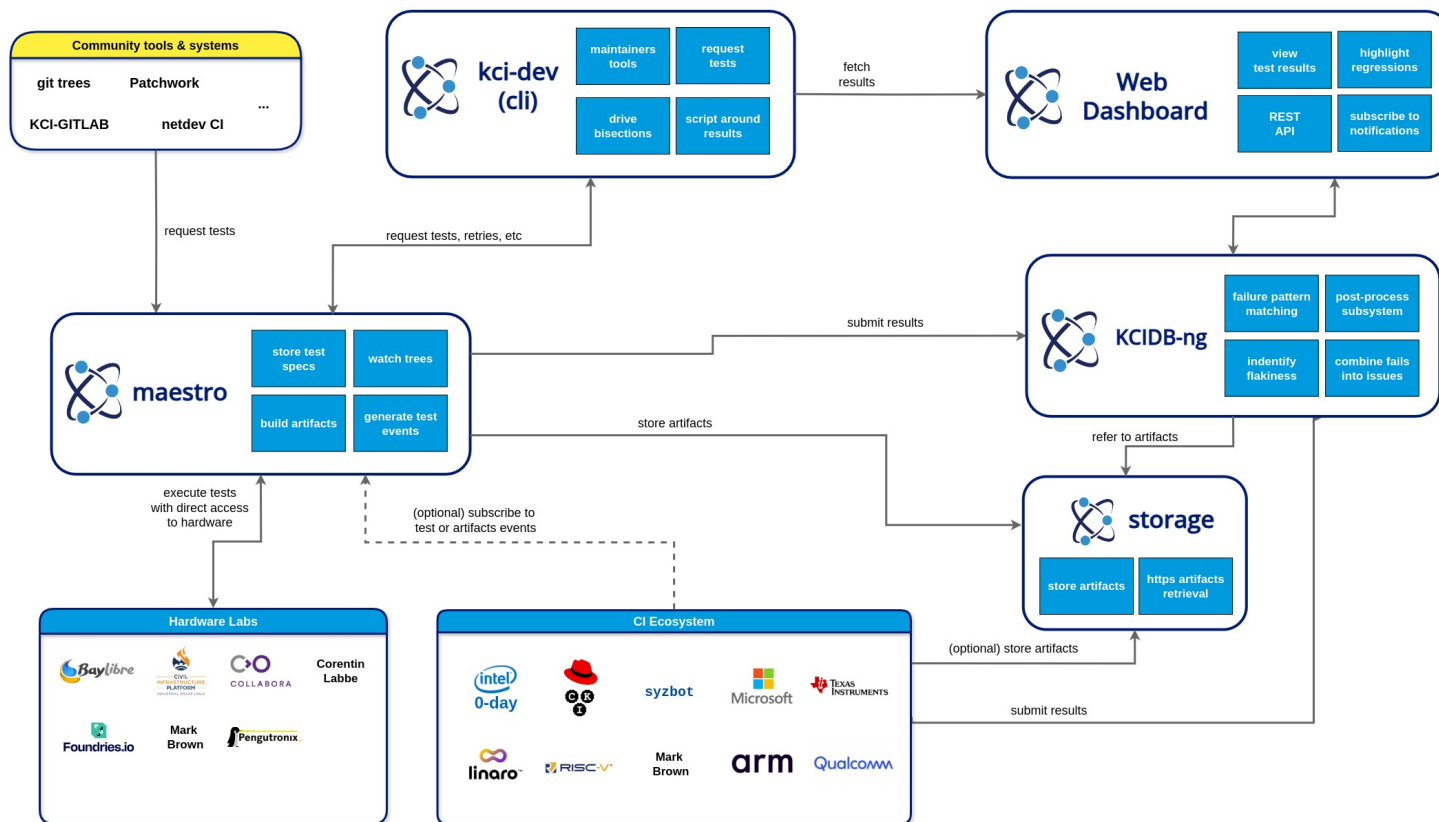
- No need to expose task complexity
- Simplify selected aspects by finding similarities between existing systems
- Provide a platform for both technical collaboration and community engagement



COLLABORA

Current status

Testing landscape



COLLABORA

Open First

Web Dashboard

**KernelCI**

Trees

Hardware

Documentation

Send us Feedback

Trees

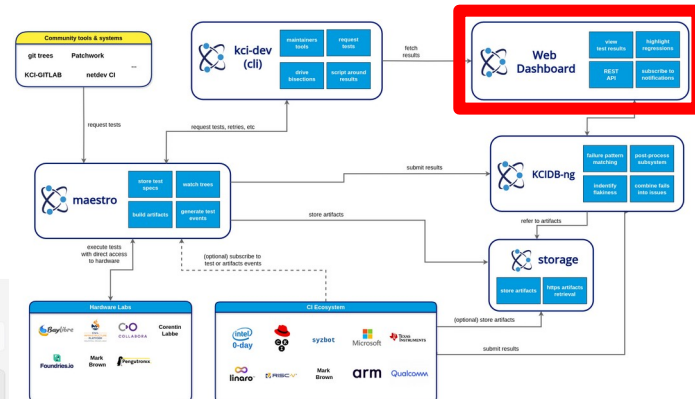
Origin maestro

stable

This is an ongoing project.
Please report bugs and suggestions to kernelci-webdashboard@groups.io.

Last 7 Days Showing: 1 - 10 of 10 Trees Items per page: 10

Tree	Branch	Commit/Tag	Date	Build Status	Boot Status	Test Status
stable	linux-6.6.y	v6.6.74	1/23/2025	22 0 0	171 18 19	385 207 182
stable-rc	linux-5.10.y	1659f4d4b85352fb8da1f9ea86034bed1ef885f1	1/21/2025	29 1 0	521 26 38	2456 485 240
stable-rc	linux-5.15.y	a38aec37d68a477d59deca3dad2b2108c482c033	1/22/2025	28 1 0	469 71 34	3869 754 225
stable-rc	linux-5.4.y	78d73ae31c4fcb1657d86088848a83e4cd996709	1/21/2025	25 4 0	452 4 50	173 303 200
stable-rc	linux-6.1.y	c5148ca733b386401ef46ed4ec93a2f4a078e187	1/22/2025	28 0 0	197 17 38	3501 430 161
stable-rc	linux-6.12.y	0bde21f273434bad90411f296c22f02c62c07b2e	1/22/2025	30 0 0	713 49 17	7476 6209 708
stable-rc	linux-6.6.y	429148729681ff93db022c19a17ce00dff9c04f9	1/21/2025	31 0 0	610 39 27	6976 4581 423
stable-rt	v5.10-rt	v5.10.233-rt125	1/23/2025	17 1 0	91 24 27	1349 6 250
stable-rt	v5.15-rt	v5.15.177-rt83	1/24/2025	19 0 0	101 30 24	1666 7 261
stable-rt	v6.6-rt	v6.6.74-rt48	1/27/2025	19 0 0	161 24 19	2589 74 438



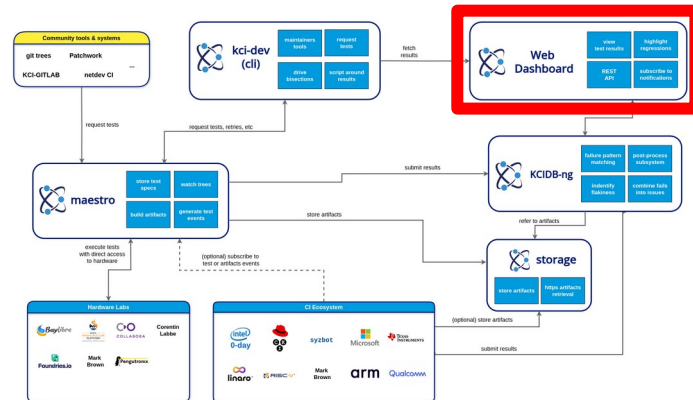
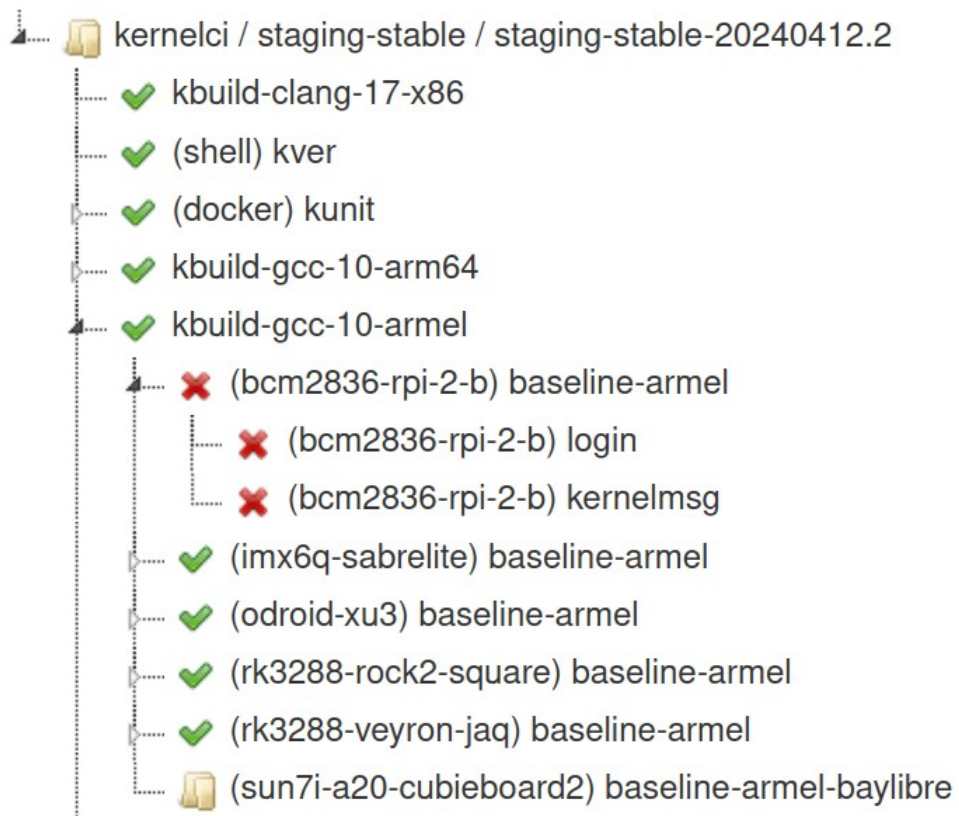
<https://dashboard.kernelci.org>



COLLABORA

Open First

Tracking regressions

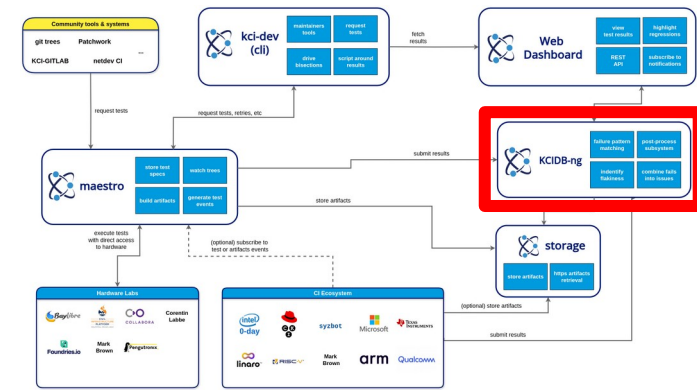


COLLABORA

Open First

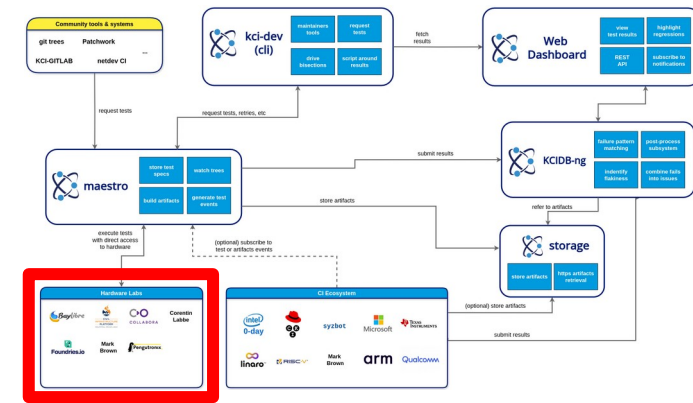
KCIDB-ng

- Compatibility layer with KCIDB
- Focused on data ingestion
- Migration to REST API strongly encouraged!
- <https://github.com/kernelci/kcidb-ng>



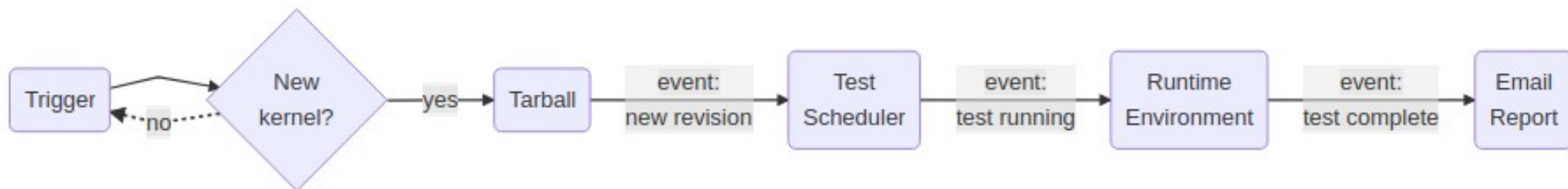
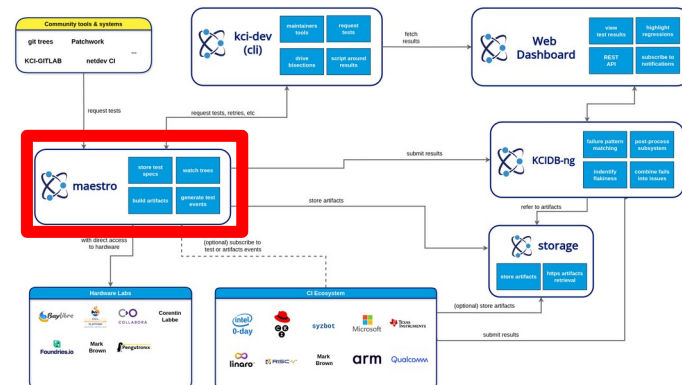
Connecting labs

- Physical/virtual
- Build farms (e.g. Kubernetes)
- LAVA:
 - Baylibre
 - Broonie
 - CIP
 - C. Labbe
 - Collabora
 - Foundries.io
 - Kontron
 - Pengutronix



Maestro

- Tree monitoring
- Event processing
- Task scheduling
- Results submission

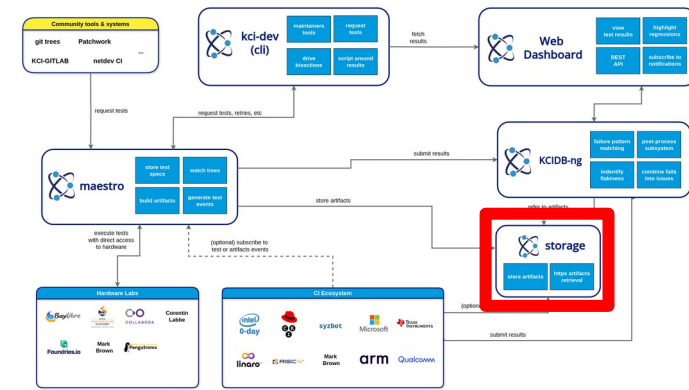


COLLABORA

Open First

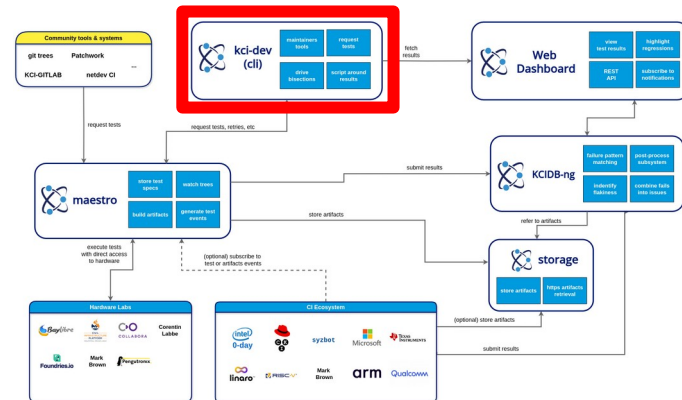
Storage service

- Server for file upload and download
- Multiple backends (e.g. Azure Blob)
- Caching enabled by default
- <https://github.com/kernelci/kernelci-storage>



kci-dev CLI utility

- Stand alone tool to interact with KernelCI
- Custom submissions (test arbitrary commits)
- Results retrieval
- <https://kci.dev>

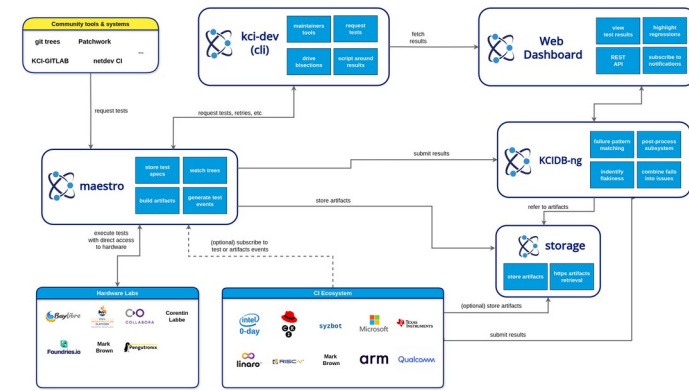


COLLABORA

Open First

Takeaways

- Extensibility
- Improved system scaling
- Support for various lab types
(not only Kubernetes/LAVA)





User stories



Maintainer's PoV

- Prerequisites:
 - Publicly hosted kernel tree
- Expectations:
 - Notifications about anomalies
 - Little to (preferably) no initial setup

Basic abstractions



- Tree and its Build config
 - Pipeline: config/trees.yaml
 - Pipeline: config/trees/mainline.yaml

```
trees:  
  (...)  
  mainline:  
    url: 'https://git.(...)'
```

```
build_configs:  
  mainline:  
    tree: mainline  
    branch: 'master'
```

Basic abstractions - cont.



- Dashboard Notifications

- Dashboard: `backend/data/notifications/subscriptions/mainline.yaml`

```
mainline:
  url: https://git.kernel.org/(...)/torvalds/linux.git
  default_recipients:
    - bot@kernel.org

  reports:
    - default:
      branch: master
      always: True
```

Quick rewards



- Get notified by email about anomalies
- View all the results in the Dashboard
- Simple configuration change request in Maestro



Developer's PoV

- Prerequisites:
 - (Same as for a Maintainer)
 - Optional: Maestro Access token
- Expectations:
 - Finer control over what is built by Maestro
 - Ability to adjust test suites or test cases run in Maestro Pipeline

Intermediate abstractions

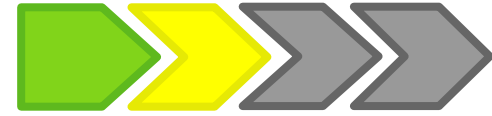


- Jobs and Schedules
- Setting Rules
 - Core: [kernelci/api/helper.py](#)

```
kselftest-dt:  
  template: generic.jinja2  
  kind: job  
  rules:  
    tree:  
      - mainline
```

```
- job: kselftest-dt  
  event: *kbuild-gcc-12-arm64-node-event  
  runtime: *lava-collabora-runtime  
  platforms:  
    - mt8390-genio-700-evk  
    - mt8395-genio-1200-evk
```

Using kci-dev CLI utility



```
kci-dev checkout
--giturl https://git.(...)
--branch staging-mainline
--commit (...)
--job-filter
    kbuild-gcc-12-arm64+
--platform-filter
    mt8390-genio-700-evk
```

```
kci-dev maestro-results
--nodes
--filter treeid=(...)
--filter kind=test
--field name
--field result
```



Benefits



- No longer limited to the tasks set up by default
- Better control over resource utilization (with Rules)
- Testing kernel trees at a specified point
(not just the most recent change)



QA Engineer's PoV

- Prerequisites:
 - (Same as for a Developer)
 - Custom test suite to be executed in the Pipeline
- Expectations:
 - Specifying build options
 - Customizing build and execution environments

Advanced abstractions



- Core repository:
 - Build configs: [config/core/build-configs.yaml](#)
 - Docker images templates: [config/docker/README.md](#)
 - Rootfs configs and templates: [config/core/rootfs-configs.yaml](#)
- GitHub Workflows
 - Docker images: [.github/workflows/docker_images.yml](#)
 - Rootfs images: [.github/workflows/rootfs.yml](#)

Investments returns



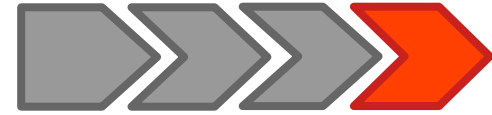
- Full control over tasks execution
- Byproduct:
contributing templates for others to reuse



Lab Operator's PoV

- Prerequisites:
 - Custom HW waiting to be utilized
- Expectations:
 - Ensuring new HW works reliably (using existing test suites)
 - Handing over build and execution environment maintenance

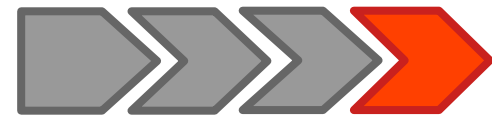
Maestro expertise



- TestJob templates:
 - Core: config/runtime
- Platforms:

```
platforms:  
  mt8390-genio-700-evk:  
    arch: arm64  
    boot_method: u-boot  
    mach: mediatek  
    dtb: dtbs/mediatek/mt8390-genio-700-evk.dtb  
    compatible: ['mediatek,mt8390-evk', 'mediatek,mt8390', (...)]
```

Profits



- Ramping up frequently used test suites at little cost (token exchange)
- Putting HW to a good use
- Keeping control over HW utilization (with job priorities)

Takeaways



- Start small
- Learn by example
- Build domain expertise



Refining use cases

Configuration in `.kernelci.yaml`

- Build configuration kept in-tree
- More control handed over to tree maintainers
- Increased flexibility, shorter time-to-test

Testing out-of-tree changes

- Additional Pipeline API endpoint accepting user input
- More flexible approach than default workflow
- Potential code-forges integration

Takeaways

- Reduce complexity
- Respond to users' requests
- Evolve continuously providing easy-to-use tools



Next steps

Not only managed instances

- Managed
 - <https://api.kernelci.org>
 - <https://staging.kernelci.org:9000>
- Local – guided
 - <https://docs.kernelci.org/components/maestro/api/local-instance>
- Local – semiautomatic
 - <https://github.com/kernelci/kernelci-deploy>

Feedback

- What is missing?
- What might have been duplicated?
- What new hardware could be tested?

Getting in touch

- Send an email to kernelci@lists.linux.dev
- IRC: #kernelci on libera.chat
- Matrix: #kci-general:matrix.org
- Discord: <https://discord.gg/vZVEQ2Qc>
- Workshop: August 28th (announcement)



COLLABORA

Summary

Final thoughts

- The new system is steadily going through a stabilization phase
- Focus on reliable tests results and relevant reports
- It's all about engaging the community in testing



Thank you!



COLLABORA

Open First



We are hiring
col.la/careers



COLLABORA

Open First