

Linux Kernel Functional Testing

Rémi Duraffort, [Linaro Ltd](#)
remi.duraffort@linaro.org



Who am I?

- Rémi Duraffort
- Principal Tech Lead at Linaro
- OSS developer since 2007
 - VLC media player
 - v8 js engine
 - PRoot/CARE
 - LAVA, KissCache, lavacli, meta-lava, DummySYS, lavafed, ...
 - tuxrun, tuxsuite cli, ...
- LAVA Architect for 8 years



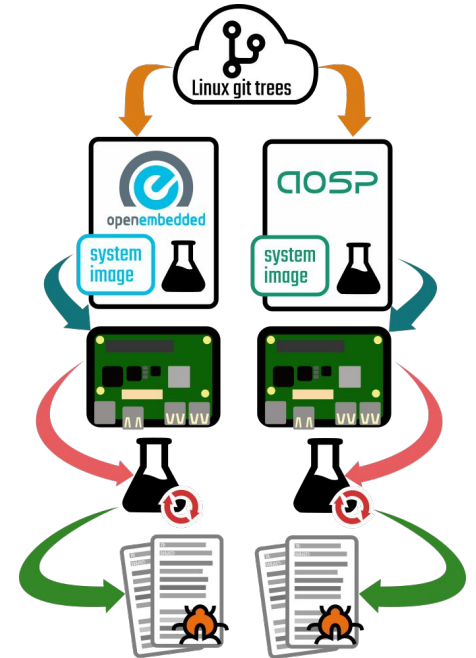
LKFT

Linux Kernel Functional Testing

What is LKFT?

“Improve the Linux kernel quality on the Arm architecture by performing regression testing and reporting on selected Linux kernel branches and the Android Common Kernel (ACK) in real time.”

- Lead by Linaro
- Automated system to build and test a set of linux kernel trees
 - LTS trees
 - mainline
 - next
- 48 hour LTS regression reporting SLA



LKFT 2023 numbers

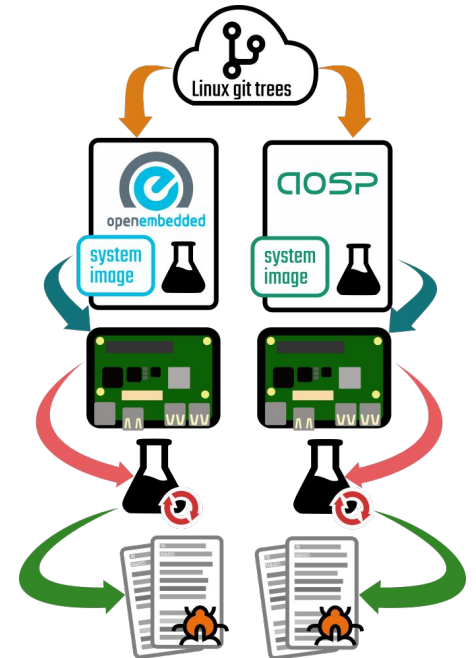
Linux Kernel

- 465 RC
- 2628 revisions
- 1.6M kernels
- 200M tests

Android Common Kernel

- 580M tests (VTS, CTS, ...)

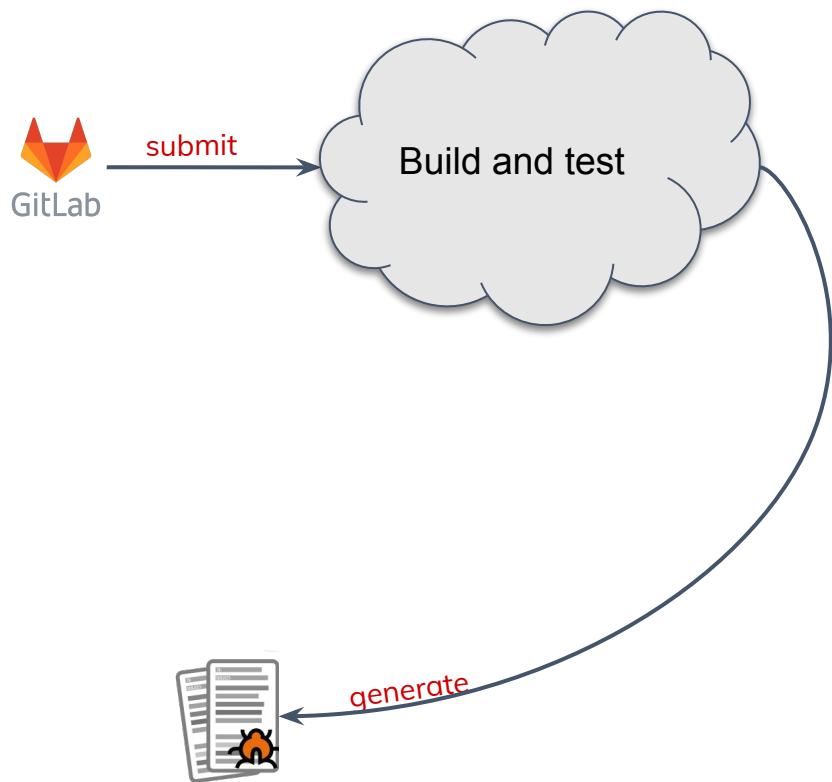
Only 3 engineers



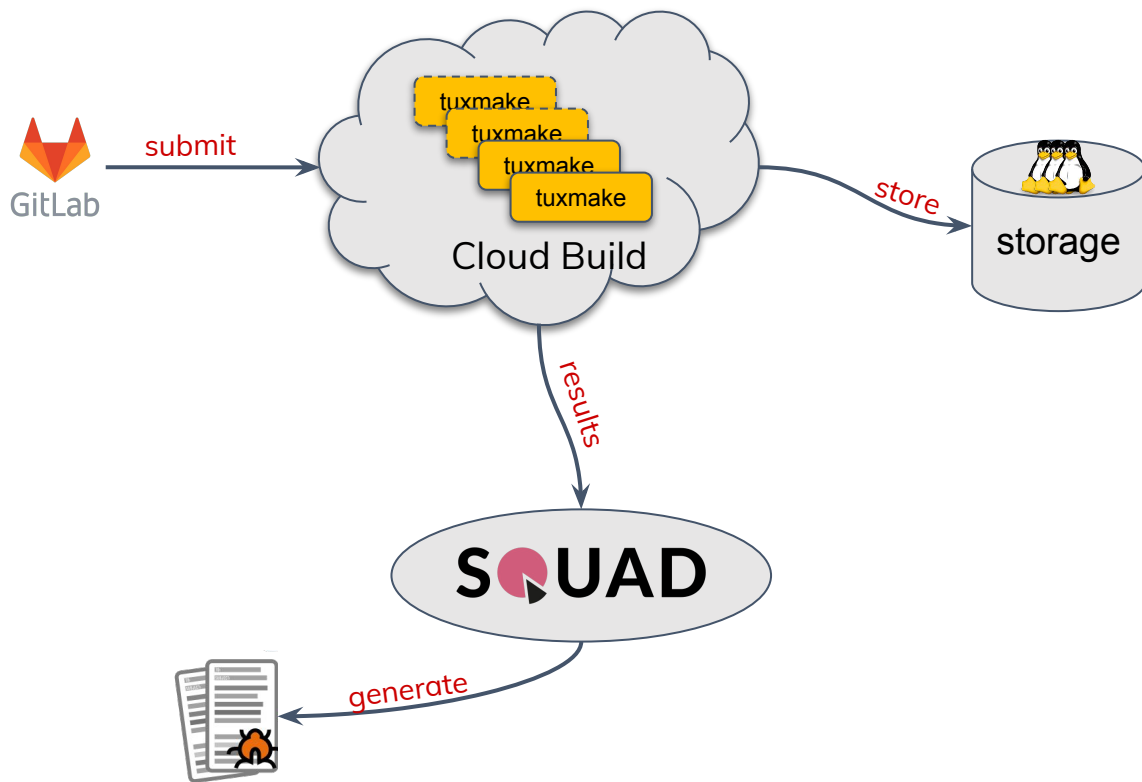
LKFT Architecture

How to build and test so many kernels?

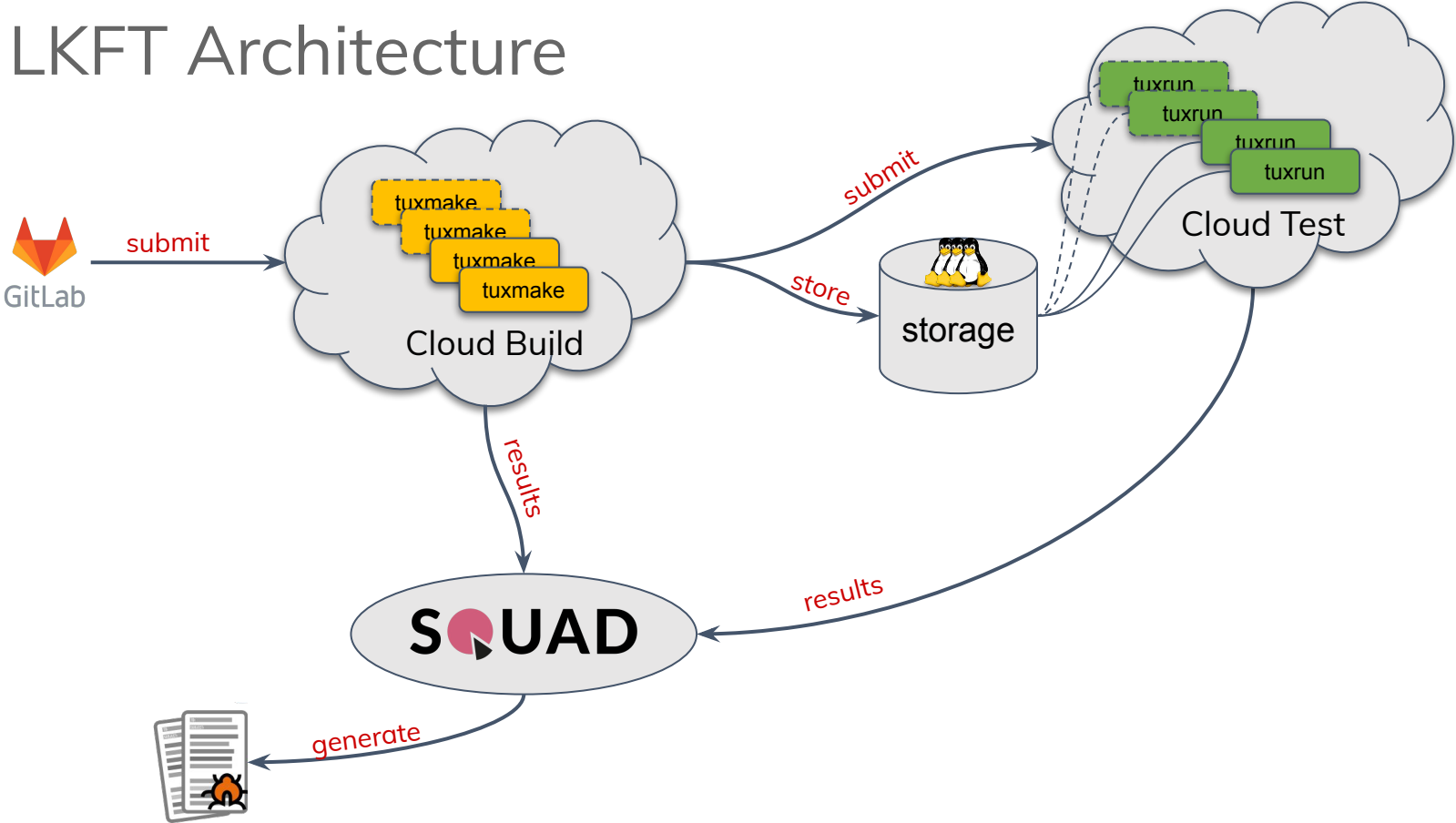
LKFT architecture



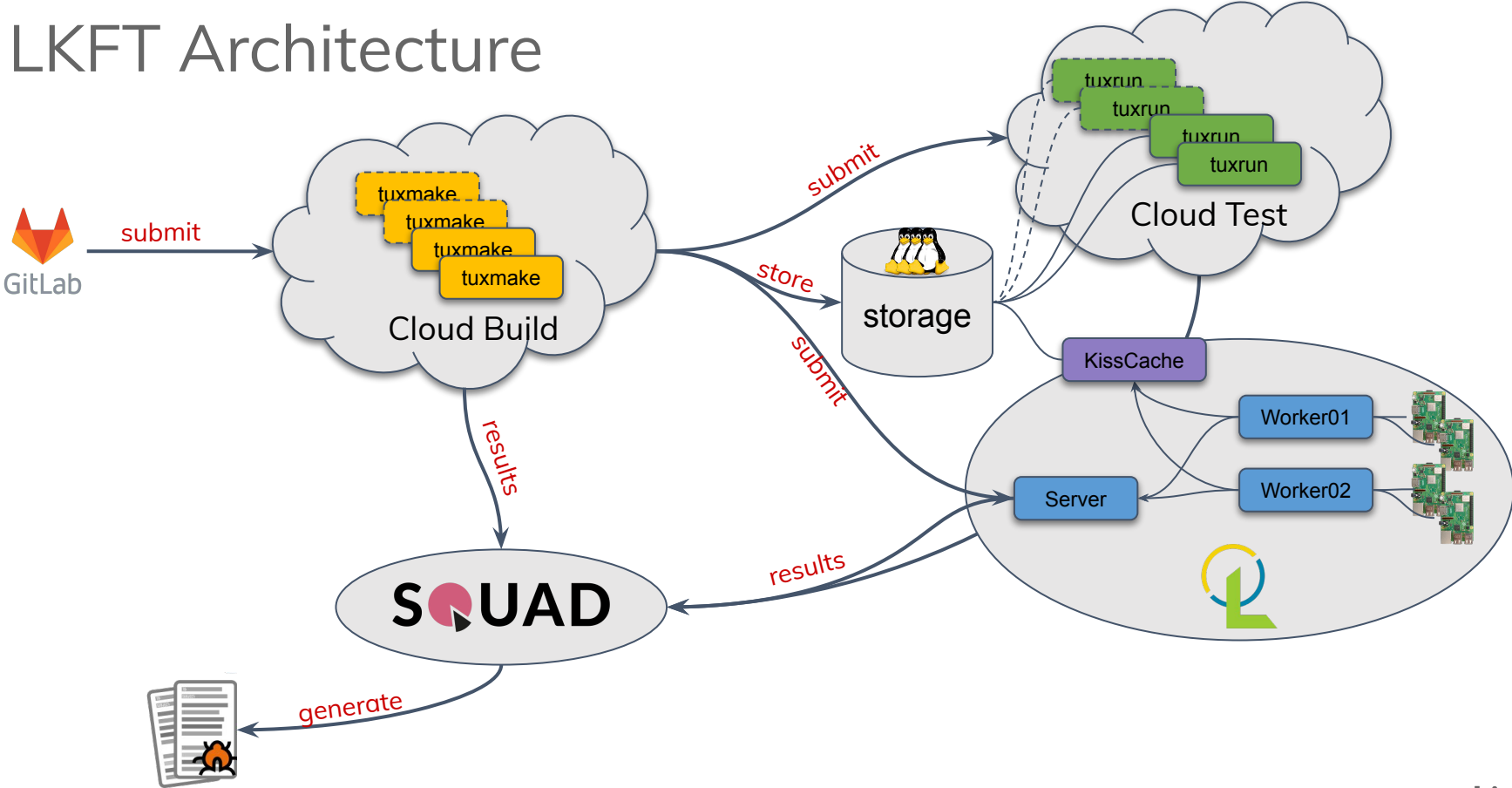
LKFT Architecture



LKFT Architecture



LKFT Architecture



Building

TuxMake

- OSS cli application
 - **portable** and **repeatable** Linux kernel builds
 - Containerized builds
 - <https://tuxmake.org>
- Multiple **toolchains**
 - gcc-8/9/10/11/12
 - clang-10/11/12/13/14/15/android/nightly
- Multiple **target-archs**
 - arm64/armv5/armv7
 - i386, x86_64
 - mips, powerpc, riscv
 - arc, hexagon, openrisc, parisc, s390, sh, sparcs, um



```
tuxmake --runtime podman --target-arch x86_64 --toolchain gcc-12 --kconfig defconfig
```

- Tuxsuite SaaS runs TuxMake at scale (5k builds in parallel) in the cloud

TuxMake explained

```
tuxmake --runtime podman --target-arch x86_64 --toolchain gcc-12 --kconfig defconfig
```

1. Pull the right container image
 - a. [docker.io/tuxmake/x86_64_gcc-12:latest...](https://hub.docker.com/r/tuxmake/x86_64_gcc-12)
2. Create a unique build directory
 - a. `~/.cache/tuxmake/builds/XXX/build`
3. Start the container with bindings
 - a. Sources from CWD
 - b. Build directory
4. Invoke make
 - a. `make --silent --keep-going --jobs=16 O=~/.cache/tuxmake/builds/XXX/build ARCH=x86_64 SRCARCH=x86 CROSS_COMPILE=x86_64-linux-gnu- defconfig`
5. ...
6. Move artefacts in `~/.cache/tuxmake/builds/XXX`
 - a. kernel, headers.tar.xz, modules.tar.xz
 - b. metadata.json

TuxMake containers

- One container for each combination
 - a. Toolchain version X target-architecture
 - i. arm_gcc-11
 - ii. arm_gcc-12
 - iii. ...
- <https://hub.docker.com/u/tuxmake>
 - a. 216 repositories
 - b. Rebuild monthly
 - i. Except for clang nightly
 1. Used by Clang CI pipeline

Testing Virtual devices with TuxRun

TuxRun

- OSS cli application
 - **portable** and **repeatable** kernel tests
 - <https://tuxrun.org>
- Multiple **devices**
 - fvp-aemva (ARMv9.3)
 - fvp-morello
 - qemu-armv5/v7/v7be/64/64be
 - qemu-i386/x86_64
 - qemu-mips32/32el/64/64el, qemu-ppc32/64/64le, qemu-riscv32/64
 - qemu-s390/sh4/sparc
- Multiple **tests**
 - ltp-*, kunit, kselftest, rcutorture, perf, v4l2, libgpiod, libhugtlbfs



```
tuxrun --runtime podman --device qemu-arm64 --kernel Image --rootfs rootfs.ext4.zst
```

- Tuxsuite SaaS runs TuxRun at scale (5k tests in parallel) in the cloud

TuxRun explained

```
tuxrun --runtime podman --device qemu-arm64 --kernel Image --rootfs rootfs.ext4.zst
```

1. Download artefacts
 - a. kernel, dtb, rootfs, modules, ...
 - i. Provide default rootfs for every architecture
 - b. Inject modules into rootfs
2. Start the container with artefacts embedded
3. Run qemu-system-aarch64
4. Parse the output for crashes
5. Run the tests
6. Store results.json

TuxRun rootfs

- Rootfs for multiple architecture are painful to build
- Default rootfs for each architect
 - a. Buildroot based: 19
 - b. Debian based: 19
 - c. Can still use custom ones
- Rebuilt regularly
 - a. Buildroot new releases
 - i. New ltp-testsuite package
 - b. Debian updates
 - c. Tested before deployment
 - i. Recently found multiple issues in qemu 7.2

TuxMake and TuxRun

- Combine TuxMake and TuxRun
- Bisect a run regression
 - Call git bisect
 - Checkout code
 - Cross-compile with tuxmake
 - Cross-run with tuxrun

```
git bisect start next-20230125 e5c645984a3884c92e124717c8c85635ba7a1057
git bisect run tuxmake --runtime podman --target-arch arm64 \
                --toolchain gcc-12 --kconfig defconfig \
                --results-hook "tuxrun --runtime podman --device qemu-arm64 --tuxmake ./ -- 'cd /opt/ltp &&
./runltp -s hugeshmctl01'"
```

Testing Real devices with LAVA

LAVA

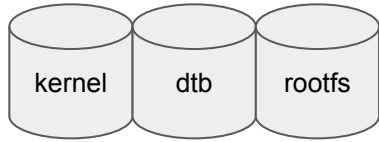
- Linaro **A**utomated **V**alidation **A**rchitecture
- Test execution system: **testing software on real hardware**
 - **Deploy, Boot and Test**
- Usages
 - Boot testing: kernelci
 - System level testing: LKFT
 - Bootloader/firmware testing
- Supports 356 device-types



LAVA

linaro.org/lava

Without LAVA



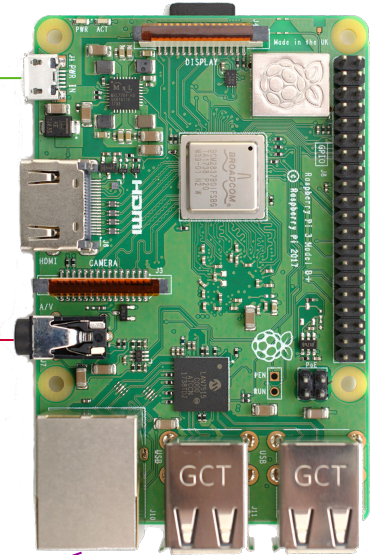
```
zsh % _
```

```
% power on board  
% telnet localhost 2000  
<enter>  
=> dhcp  
=> setenv serverip 10.3.1.1  
=> [...]   
=> bootm 0x01000000 - 0x03f00000  
[...]  
raspberrypi3 login: root  
# run-test.sh  
[...]  
% power off board
```

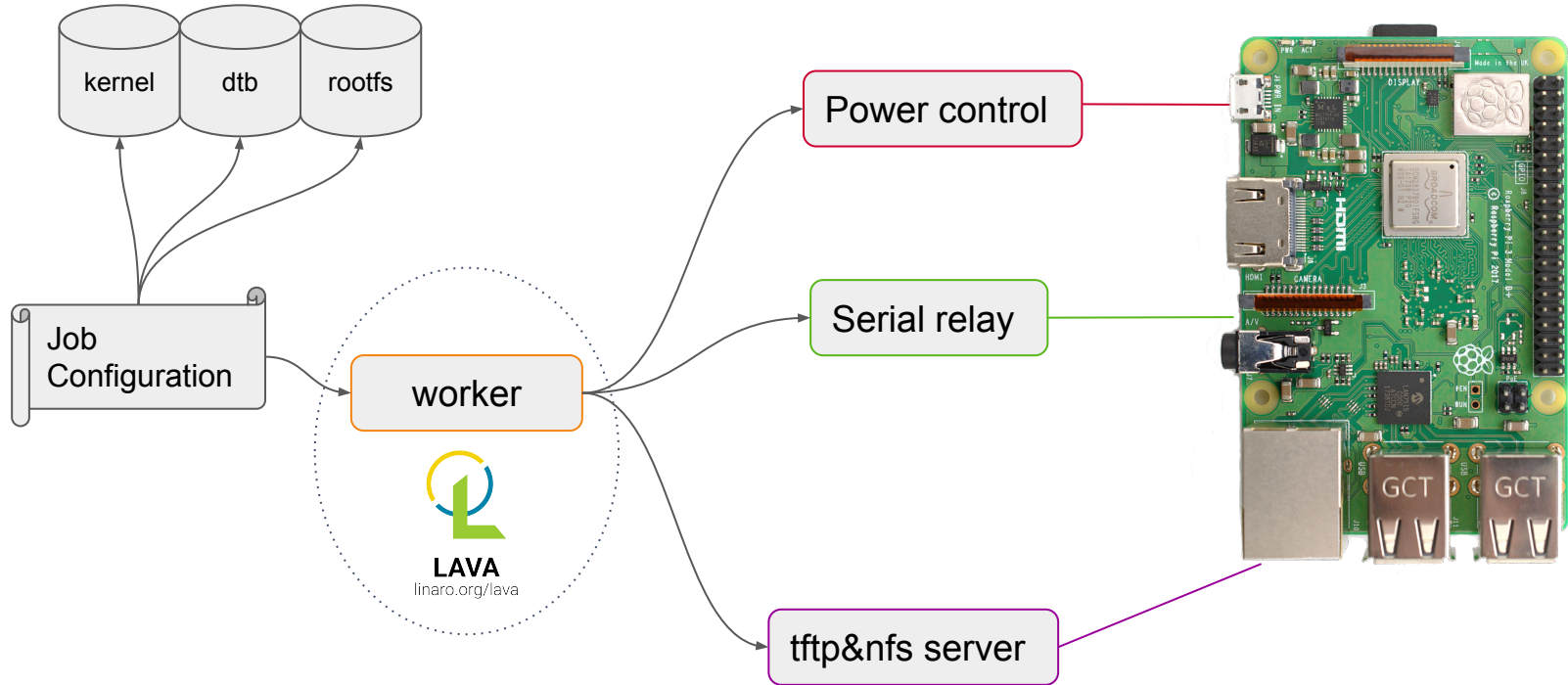
Serial relay

Power control

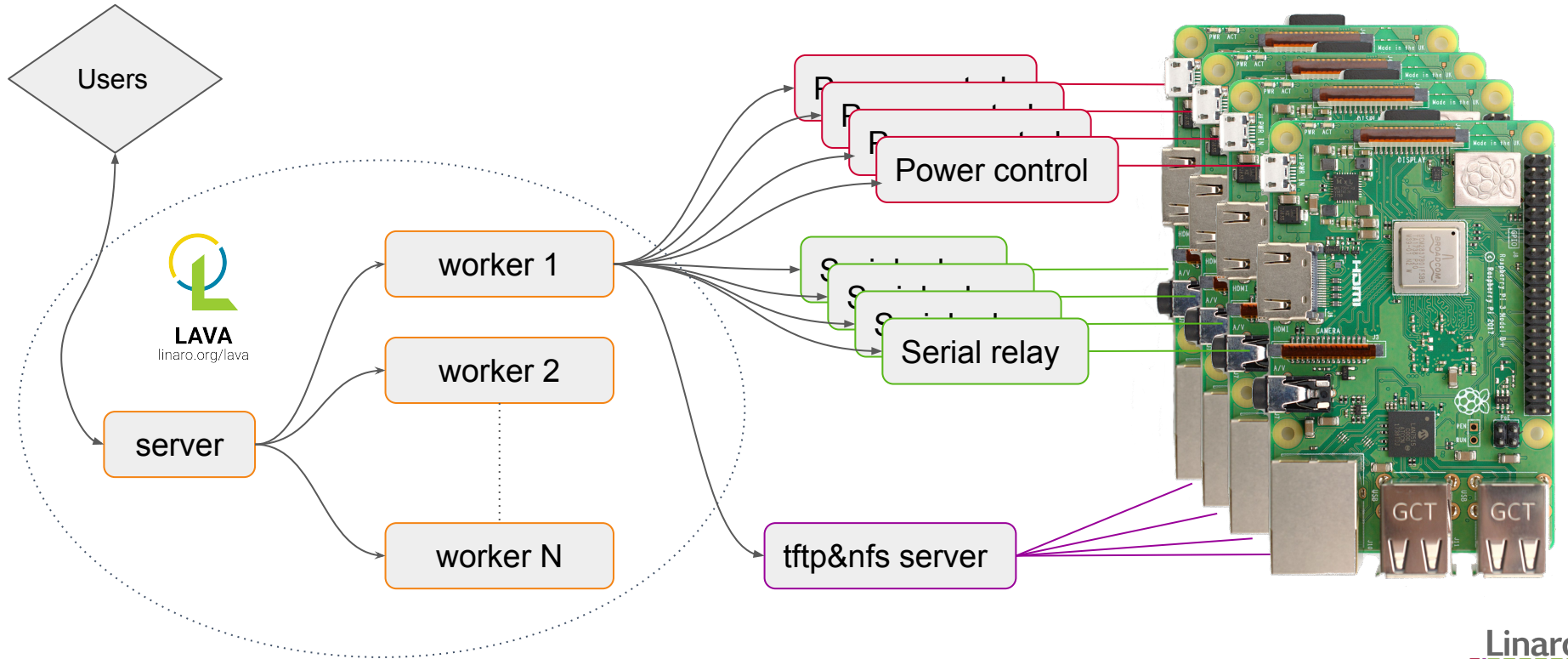
tftp&nfs server



LAVA explained



LAVA explained



Network performances KissCache

KissCache

- LAVA downloads a lot of artifacts
 - Multiple times
 - In parallel (almost exact same time)
- SQUID should fix this?
 - Short answer **NO!**
 - Artefacts are served over **https**
 - Requires to fake SSL certificates
 - Create a wildcard certificate (for every domains)
 - Install on the clients
 - Multiple concurrent downloads of the same artefacts
 - SQUID will download multiple times the same artefacts
 - Cache only when a first download is completed

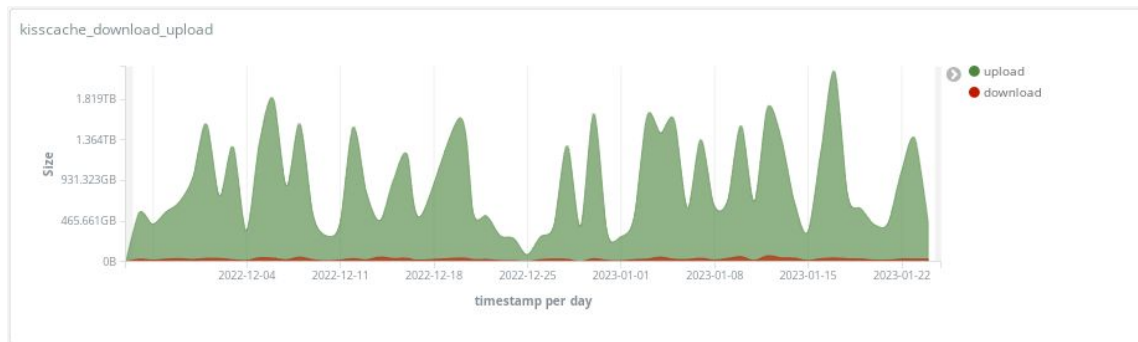


KissCache

- A simple and stupid caching server
 - Cache HTTPS resources
 - Download **once** while streaming to **multiple** clients
 - <https://gitlab.com/Linaro/kisscache>
- Not transparent (prefix based)
 - `https://kisscache/api/v1/fetch/?url=https://example.com/rootfs.ext4.zs`
 - no need for fake SSL certificates
 - Need support in the clients
- Automatic retries on multiple errors
 - 408, 413, 420, 425, 429, 430, 500, 502, 503, 504, 507, 509, 529 and 598
 - Partial download
 - Will use range request to download remaining content

KissCache

- Over 2.5 years
 - 25 TB downloaded (from internet)
 - 1.3 PB served (local network) by KissCache
 - Network usage divided by 52x
 - Improved stability



Storing job results
SQUAD

SQUAD

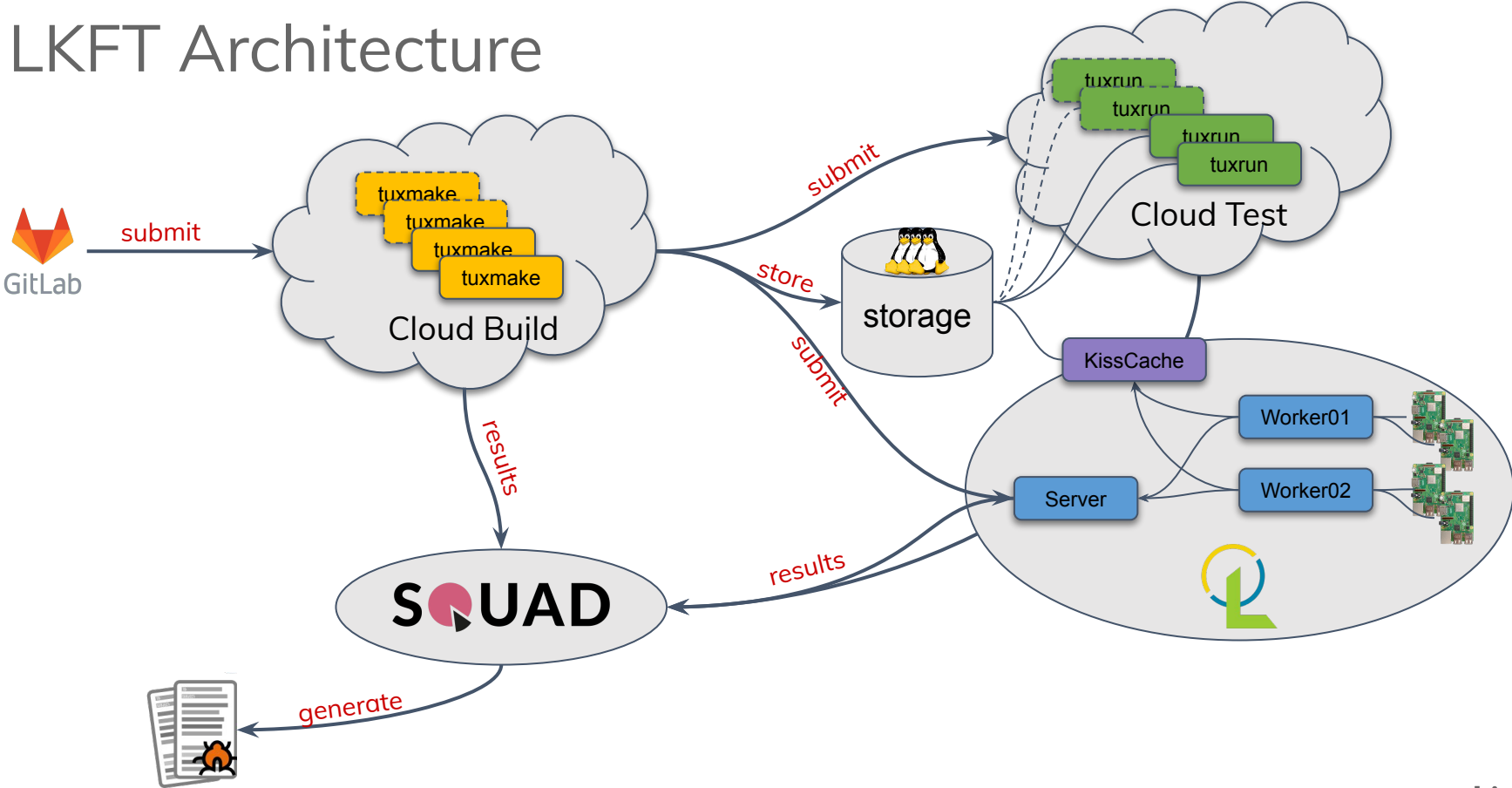
- Software Quality Dashboard aka SQUAD
- A data lake
 - Gather results (builds, tests, measurements, ...)
 - 3.3 billions results
- Create reports
 - Failures, regressions, ...
- Links:
 - <https://qa-reports.linaro.org/>
 - Linaro instance
 - <https://qa-reports.linaro.org/lkft/>
 - LKFT project page

The screenshot displays the SQUAD dashboard for the LKFT (Linux Kernel Functional Testing) project. At the top, the SQUAD logo is visible along with navigation options for Groups, Compare, and API, and a Log in button. The main heading is 'LKFT', with the subtitle 'Linux Kernel Functional Testing'. Below this, a message states 'Displaying only a subset of all projects' with a link to 'Click here to see all projects.' The 'Order by' dropdown is set to 'last updated'. The dashboard lists ten test runs, each with a summary of test results and a link to the full report.

Test Run	Tests	Pass	Skip	Fail	Time
linux-mainline master on OE Linux mainline kernel using OpenEmbedded	168331 tests 338 xfail 72.173	142150 pass	22231 skip	3612 fail	3 minutes ago Jan. 6, 2023, 2:52 a.m.
linux-next master on OE Linux next kernel using OpenEmbedded	183580 tests 337 xfail 95.215	155404 pass	24013 skip	3826 fail	a minute ago Jan. 6, 2023, 2:54 a.m.
linux-stable-rc linux-4.9.y on OE Linux 4.9 LTS kernel using OpenEmbedded	78093 tests 367 xfail 42.433	66992 pass	8290 skip	2444 fail	an hour ago Jan. 6, 2023, 1:24 a.m.
linux-mainline master on OE - sanity Linux mainline kernel using OpenEmbedded - Fast sa...	1003 tests 62.361	989 pass	10 skip	4 fail	3 hours ago Jan. 5, 2023, 11:24 p.m.
linux-stable-rc linux-4.14.y on OE Linux 4.14 LTS kernel using OpenEmbedded	44275 tests 326 xfail 152.133	36326 pass	5831 skip	1792 fail	8 hours ago Jan. 5, 2023, 6:54 p.m.
linux-stable-rc linux-4.9.y on OE - sanity Linux 4.9 LTS kernel using OpenEmbedded - Fast sa...	1137 tests 109.851	1116 pass	8 skip	13 fail	11 hours ago Jan. 5, 2023, 3:30 p.m.
linux-stable-rc linux-5.15.y on OE	181596 tests 361 xfail 58.913	135143 pass	21278 skip	4814 fail	11 hours ago Jan. 5, 2023, 3:28 p.m.
linux-stable-rc linux-4.19.y on OE Linux 4.19 LTS kernel using OpenEmbedded	66622 tests 123 xfail 193.205	42076 pass	21511 skip	2912 fail	11 hours ago Jan. 5, 2023, 3:01 p.m.
linux-stable-rc linux-5.4.y on OE Linux 5.4 stable kernel using OpenEmbedded	58748 tests 337 xfail	40612 pass	15131 skip	2668 fail	15 hours ago Jan. 5, 2023, 11:23 a.m.

The end

LKFT Architecture



Thank you

