Open Source Summit Japan 2019

Using Open Source Software to Build an Industrial-grade Embedded Linux Platform from Scratch

SZ Lin (林上智)

Embedded Linux Development Center,
Software R&D Engineer

07/19, 2019

MOXA®
Reliable Networks ▲ Sincere Service

About Me

SZ LIN (林上智)

- Software Engineer at Moxa
- Cybersecurity Fundamentals Specialist
  - ISA/ IEC 62443
- Debian Developer
- Blog - https://szlin.me

Industrial Embedded Linux Platforms

Application

Smart Rail | Smart Grid | Smart Oil Field | Smart Transportation | Smart Factory | Smart Marine

Edge Connectivity

Serial Connectivity

I/O Connectivity

Video Connectivity

Industrial Computing

Embedded Computers

Network Infrastructure

Industrial Ethernet

Industrial Wireless LAN

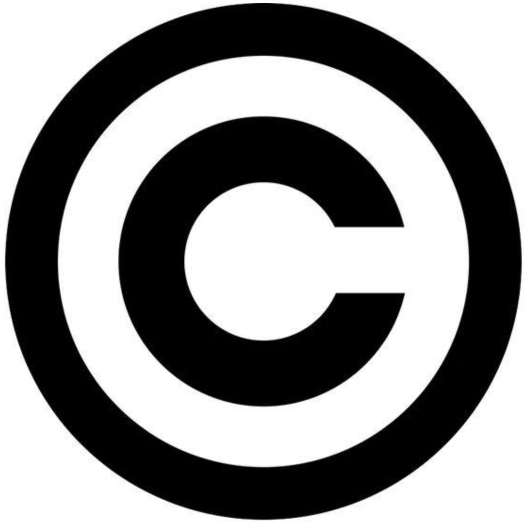
Industrial Routers

Device

MOXA

Before Using Open Source Software

Something You Should Know



Copyright

Copyright is a legal right, that grants the creator of an original work exclusive rights to determine whether, and under what conditions, this original work may be used by others

src: https://en.wikipedia.org/wiki/Copyright

Patent

A patent gives its owner the right to exclude others from making, using, selling, and importing an invention for a limited period of time, usually twenty years.

src: https://en.wikipedia.org/wiki/Patent

Copyright

Identify key recommended processes for effective open source management [1].

Patent

It is a shared defensive patent pool with the mission to protect Linux [2].

Processes, Tooling and Support

OpenChain

Trust between entities in the supply chain

The OpenChain Project builds trust in open source by making open source license compliance simpler and more consistent

SPDX [3]

Trust for software packages

Software Package Data Exchange (SPDX) is a file format used to document information on the software licenses under which a given piece of computer software is distributed.

FOSSology [4]

Free scanning technology

FOSSology is a open source license compliance software system and toolkit



Industrial/ Harsh Environments

Including smart rail, smart grid, intelligent transportation, factory automation, oil & gas, marine, and more

NEMA TS2

EN50155

IEC 61850-3

DNV·GL



MOXA

Target Application

Longevity + Stability + Security

Performance

Real-time

Resource Limited

Safety

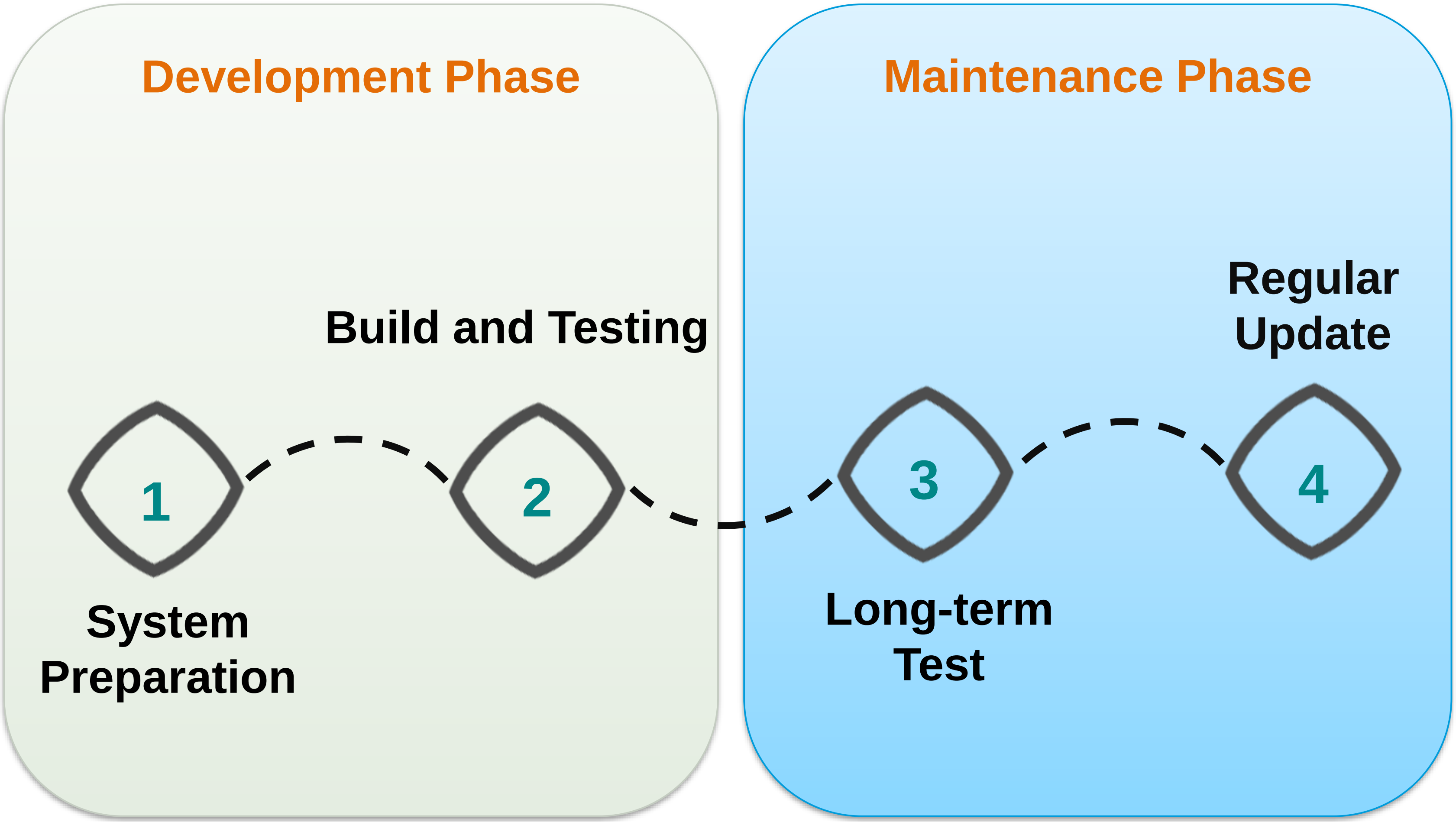
Target Application

Hardware

Software

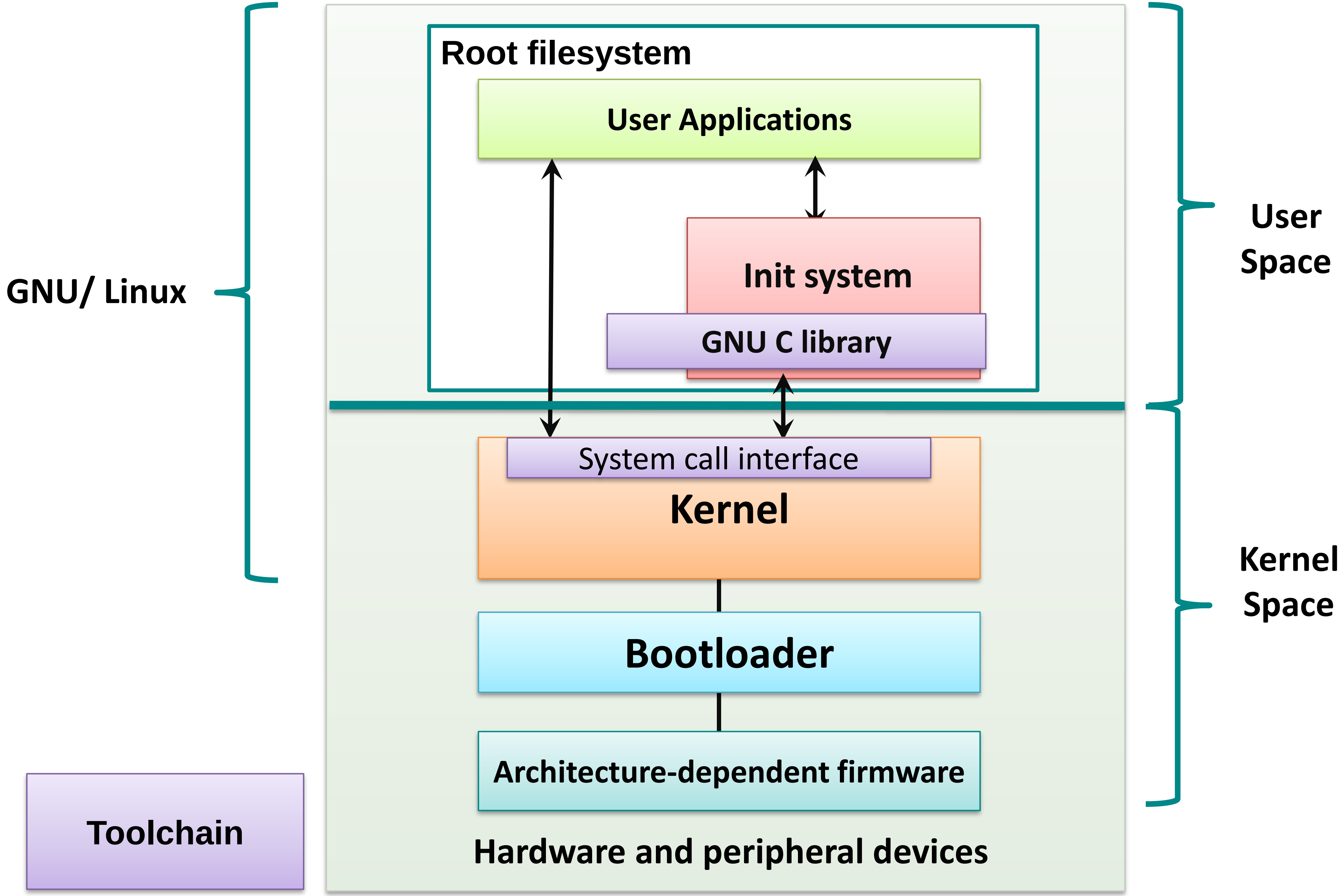
Linux

Lifecycle of Industrial-grade Embedded Linux Platform



Development Phase

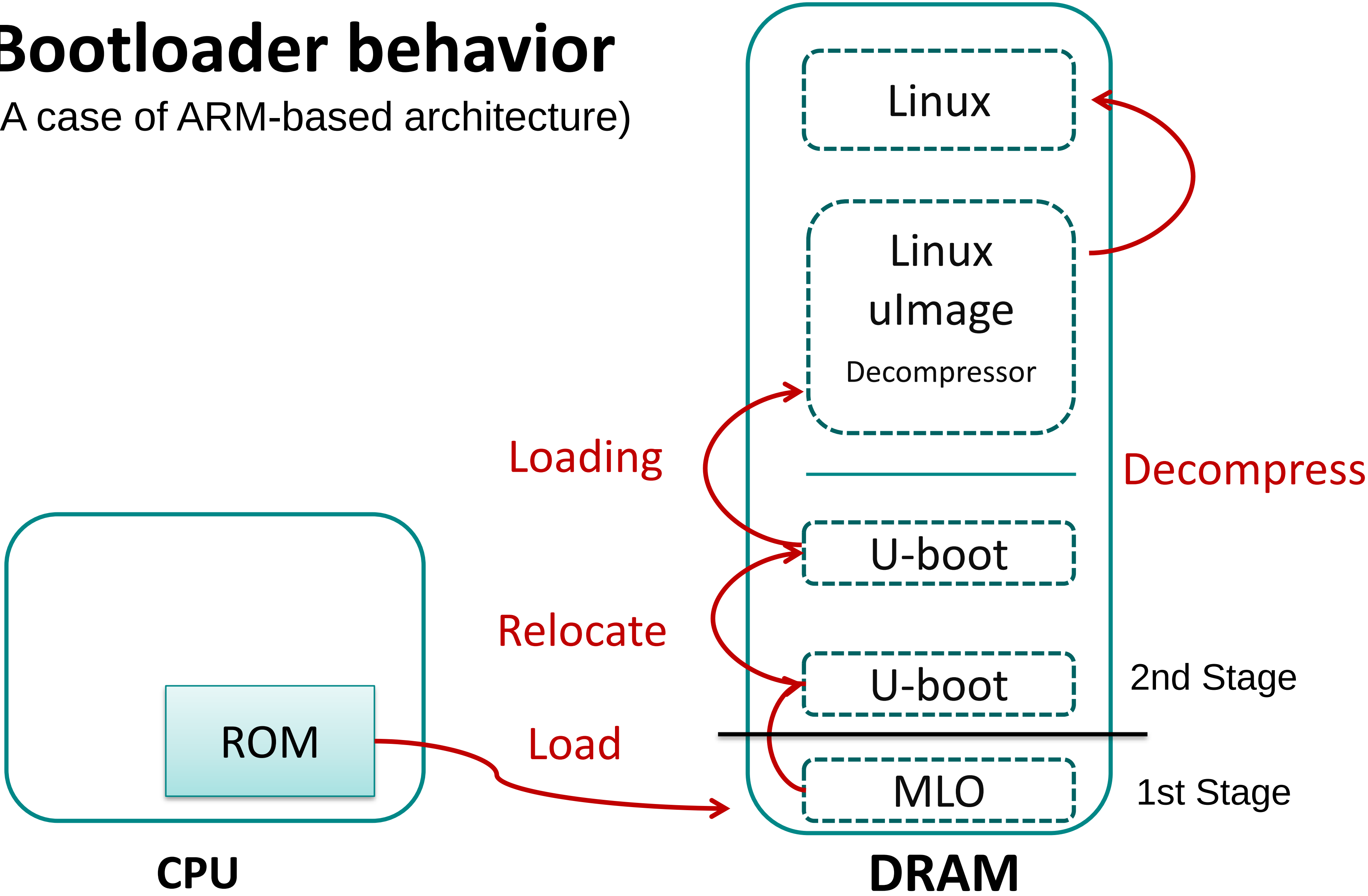
Design and development according to application


GNU/ Linux
Root filesystem
User Applications
Init system
GNU C library
System call interface
Kernel
Bootloader
Architecture-dependent firmware
Hardware and peripheral devices
Toolchain
User Space
Kernel Space

Choose Proper Bootloader

Bootloader behavior

(A case of ARM-based architecture)



Category	License	Supported Platforms	Supported UEFI	Maintainer
Das U-Boot [5]	GPL-2+	68k, ARM, Blackfin, MicroBlaze, MIPS, Nios, SuperH, PPC, RISC-V, x86 (on top of Coreboot)	Y	DENX Software Engineering
Coreboot [6]	GPL-2	IA-32, x86-64, ARMv7, ARMv8, MIPS, RISC-V, POWER8	Y	coreboot.org
GRUB	GPL-3	IA-32, x86-64, IA-64, ARM, PowerPC, MIPS and SPARC	Y	GNU Project
rEFInd [9]	GNU GPLv3, Modified BSD License (original program), additional components released under various licenses	x86, x86-64, or ARM64	Y	Roderick W. Smith

Kernel Space

Choose Proper Kernel

Based on the application requirement

Linux Kernel Comparison Table

Category	Latest version	Target Application	Maintainer
Linux kernel	5.2	• Performance • Resource Limited [12] [13]	Kernel.org
Preempt RT kernel	5.2	• Real-time • Functional safety • Resource Limited	Real Time Linux collaborative project

*Real-time application [14][15]

MOXA

SoC Board Support Package Kernel

- **Kernel version depends on SoC vendors**
  - **Well made but not well maintained**
- **Contain lots of in-house patches**
  - **Errata patches**
  - **Specific feature patches**
  - **…**
- **Different SoC might use different versions of kernel**
- **The lifetime is unsure**

LTS: Long Term Stable Kernel [16]

Longterm release kernels

Version	Maintainer	Released	Projected EOL
4.19	Greg Kroah-Hartman	2018-10-22	Dec, 2020
4.14	Greg Kroah-Hartman	2017-11-12	Jan, 2020
4.9	Greg Kroah-Hartman	2016-12-11	Jan, 2023
4.4	Greg Kroah-Hartman	2016-01-10	Feb, 2022
3.16	Ben Hutchings	2014-08-03	Apr, 2020

Extend software uptime for stable kernel

- Only accept bug fixes and security fixes

img: https://www.kernel.org/category/releases.html

LTSI: Long Term Support Initiative [17]

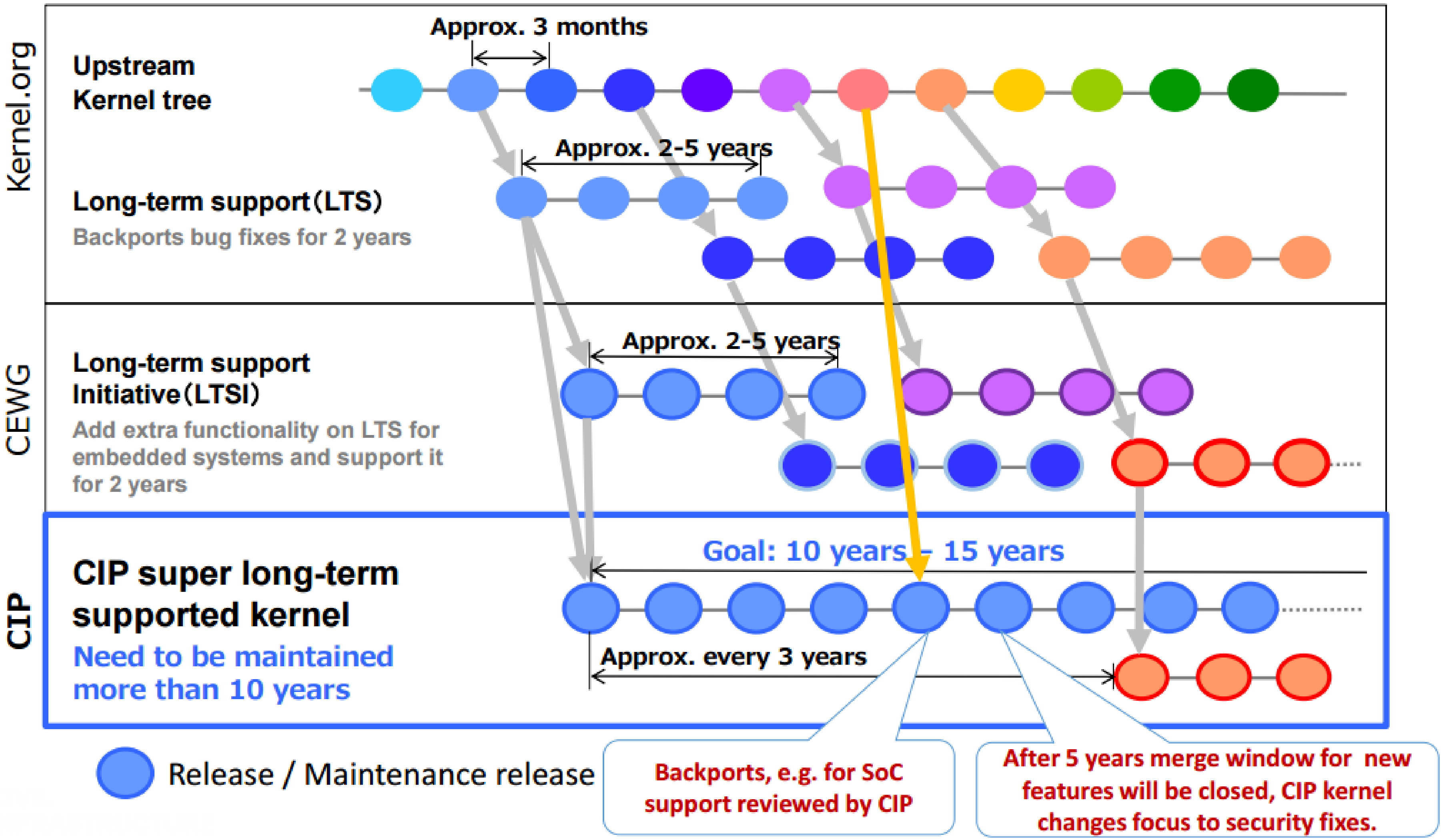
- **Linux Foundation collaborative project**
  - **Based on LTS**
  - **Add another chance to include further patches on top of LTS**
  - **Auto Test framework**
  - **Same lifetime with LTS (yearly release and 2 years life time)**

CIP (Civil Infrastructure Platform) [19]

- **Linux Foundation collaborative project**
  - **Support kernel and core package**
  - **Auto Test framework**
  - **Maintenance period**
    - 10 years and more (10-20 years)

CIVIL INFRASTRUCTURE PLATFORM

MOXA



img: https://wiki.linuxfoundation.org/civilinfrastructureplatform/cipconferences

Linux Kernel Source Comparison Table

Version	Maintenance Period (years)	Features	Latest Version	Supported Realtime kernel	Maintainer
SoC BSP kernel	?	Bug fixes	?	N	SoC vendor kernel team
LTS kernel	2 ~ ?	• Bug fixes • Security fixes	4.19	N	Kernel.org
LTSI kernel	2 ~ ?	• Bug fixes • Security fixes • Specific features • New features	4.14	N	LTSI
CIP kernel	10 +	• Bug fixes • Security fixes • Specific features • New features	4.19	Y	CIP

Longevity + Stability + Security

Mutually Exclusive ?

Performance	**Real-time**	**Resource Limited**	**Safety**

Multiple Kernel In Single Platform

To fulfill multiple user scenarios

FIT (Flattened Image Tree)
(A case of ARM-based architecture)

- Tree data structure
- Handle multiple types of image
  - kernel : kernel image
  - fdt : dtb file
  - ramdisk : root file system
- Image hashing
  - md5
  - sha1
- Image signing
- Each node in configurations has their image configuration in booting stage

```
/dts-v1/;

/ {
        description = "Image file for the LS1043A Linux Kernel";
        #address-cells = <1>;

        images {
                kernel@1 {
                        description = "ARM64 Linux kernel";
                        data = /incbin/("./arch/arm64/boot/Image.gz");
                        type = "kernel";
                        arch = "arm64";
                        os = "linux";
                        compression = "gzip";
                        load = <0x80080000>;
                        entry = <0x80080000>;
                };
                fdt@1 {
                        description = "Flattened Device Tree blob";
                        data = /incbin/("./arch/arm64/boot/dts/freescale/fsl-ls1043a-rdb.dtb");
                        type = "flat_dt";
                        arch = "arm64";
                        compression = "none";
                        load = <0x90000000>;
                };
        };
        configurations {
                default = "config@1";
                config@1 {
                        description = "Boot Linux kernel";
                        kernel = "kernel@1";
                        fdt = "fdt@1";
                };
        };
};
```

More info.:
http://git.denx.de/?p=u-boot.git;a=blob_plain;f=doc/uImage.FIT/source_file_format.txt;hb=HEAD

User Space

ELISA: Safety-Critical Systems [20]

- **Linux Foundation collaborative project**
  - **Build and certify Linux-based safety-critical applications**
  - **Define and maintain a common set of tools and processes**
    - SIL2LinuxMP [21] project and the Linux Foundation's Real-Time Linux project
  - **IEC 61508**

Choose Proper
C Library and Toolchain

C Library and Toolchain Comparison Table

Category	License	Features	Target Application	Maintainer User
glibc [25]	LGPL 2.1	• Stable ABI • Backward compatibility • Fully symbol versioning • Stack smashing protection/ heap corruption detection • Profiling	• Performance • Security	GNU
uClibc-ng [26]	LGPL 2.1	• No-MMU architecture support • Tiny size	• Resource Limited	uclibc-ng.org
Musl [28]	MIT	• Stable ABI • Backward compatibility • Stack smashing protection/ heap corruption detection	• Resource Limited • Security	musl-libc.org

Other option [93]

* Be aware of year 2038 problem [29]

MOXA

Year 2038 Problem [92]

Binary : 01111111 11111111 11111111 11110000

Decimal : 2147483632

Date : 2038-01-19 03:13:52 (UTC)

Date : 2038-01-19 03:13:52 (UTC)

- **The time_t datatype is a data type in the ISO C library and kernel structure defined for storing system time values.**

- **32-bit system can represent dates from**
  - **Dec 13 1901**
  - **Jan 19th 2038**

- **It causes integer overflowing on 03:14:08 UTC 19 January 2038**

MOXA

Init System

Init System Comparison Table

Category	License	C Library	User	Note
busybox	GPL 2.0	uClinux-ng Glibc musl	ProteanOS PiBox	Resource-limited application
sysvinit	GPL 2.0+	uClinux-ng glibc musl	Devuan	
systemd	LGPL 2.1+	glibc	Arch, CentOS, CoreOS, Debian, Fedora, Mint, OpenSUSE, Redhat, Ubuntu	Linux only
openrc	2-clause BSD	musl glibc	Gentoo Alpine Linux	
upstart	GPL 2.0	glibc	Chromium OS	Linux only

Choose proper RFS (Root filesystem)

Stable root filesystem

Root filesystem Comparison Table

Category	Maintenance Period (years)	Number of packages	C Library	Security Tracker	CI
Busybox	?	300 ~ 400 applets	• uClibc • glibc	?	?
Yocto	Latest release the previous two releases	It depends on meta-*	• glibc • musl	Y	Y
Buildroot	1	2000+ [42]	• glibc • musl • uClibc-ng	Y	Y
Debian	3 + 2 (i386, amd64, armel, armhf and arm64)	51000+	• glibc • musl	Y	Y

System Development Tools

System Development Tools Comparison Table

Root filesystem	System Development Tools	Toolchain	System Development Tools License
Busybox	Yocto	OE-Core	MIT
Yocto	Yocto	OE-Core	MIT
Buildroot	Buildroot	Buildroot	GPL 2.0+
Debian	ISAR	Debian toolchain	Metadata: MIT Others: GPL 2.0
	ELBE	Debian toolchain	GPL 3.0+
	Yocto Deby (meta-debian)	OE-Core	MIT
	Live-build	Debian toolchain	GPL 3.0+

Why We Choose Debian [49]

debian

Stability
unstable → testing → stable

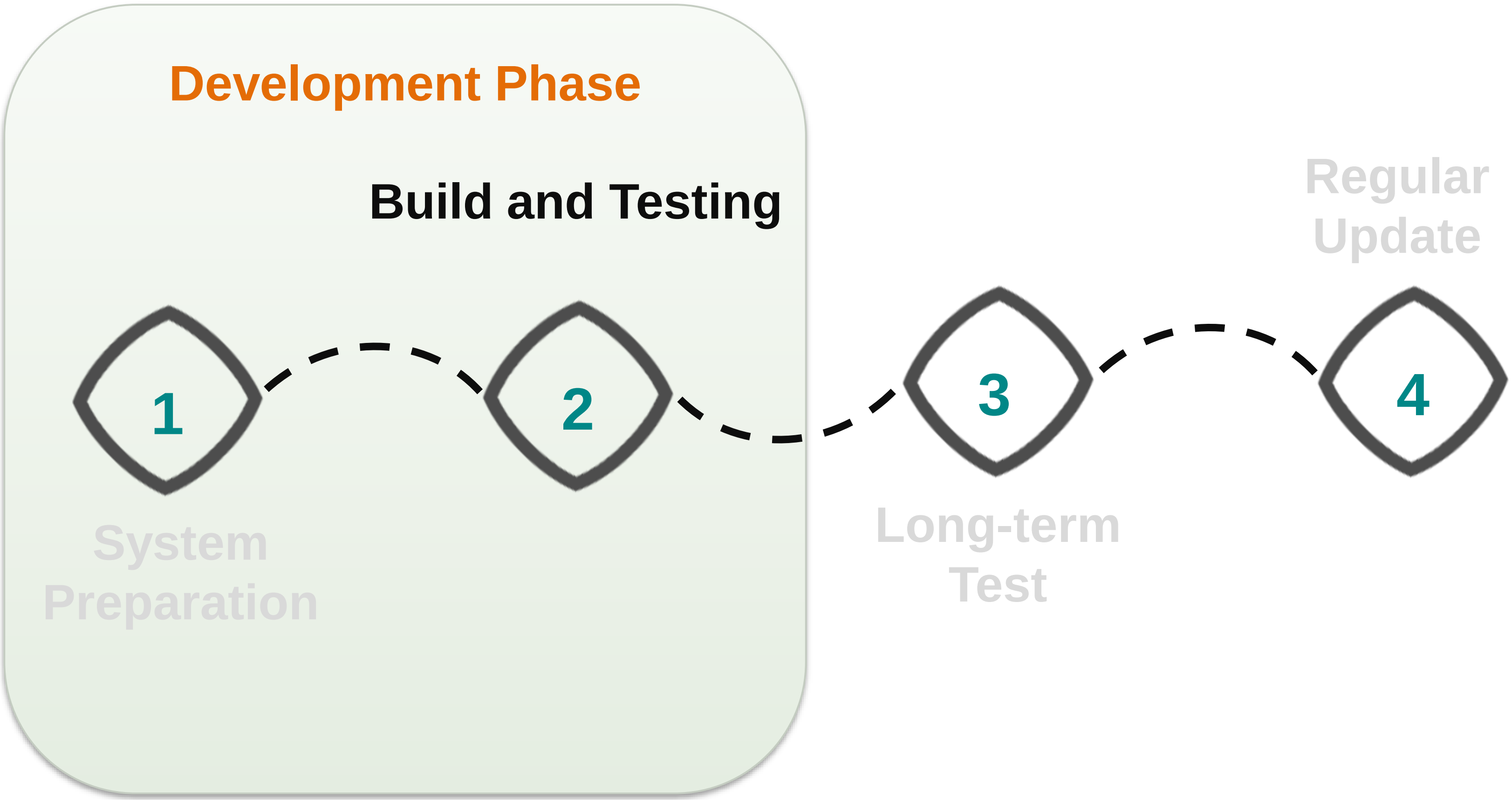
Scalability
Server, Desktop,
Laptop, Embedded devices

Good system security [50]
Everything is open
Usually, fixed packages are uploaded within a few days

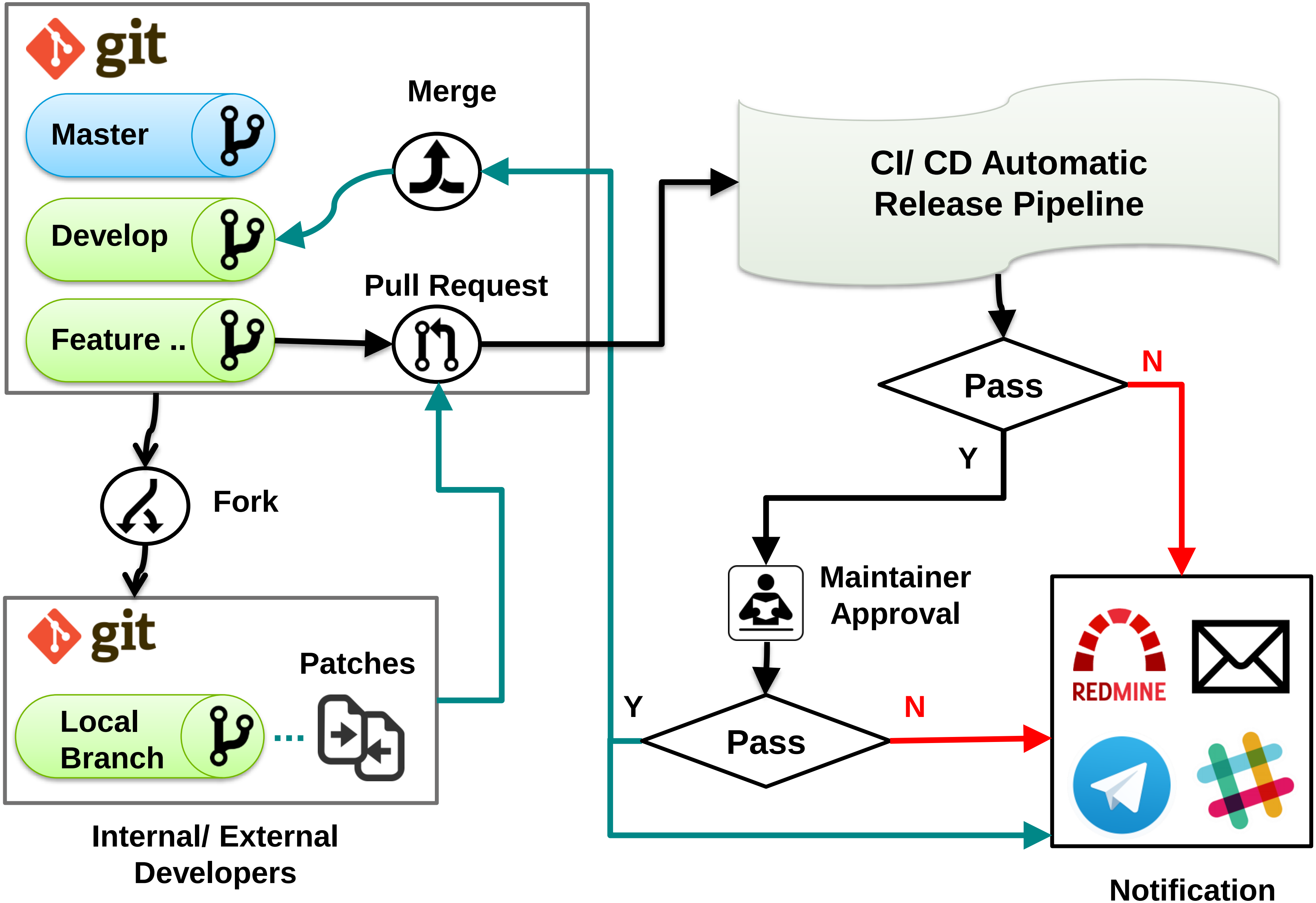
Long term support
5 more years by Debian-LTS project
(i386, amd64, armel, armhf and arm64)

Multiple architectures
alpha, amd64, armel, armhf, aarch64, hppa, i386, ia64, mips, mipsel, powerpc, s390, and spar

Incredible amounts of software
Debian comes with over 51000 different pieces of software with free

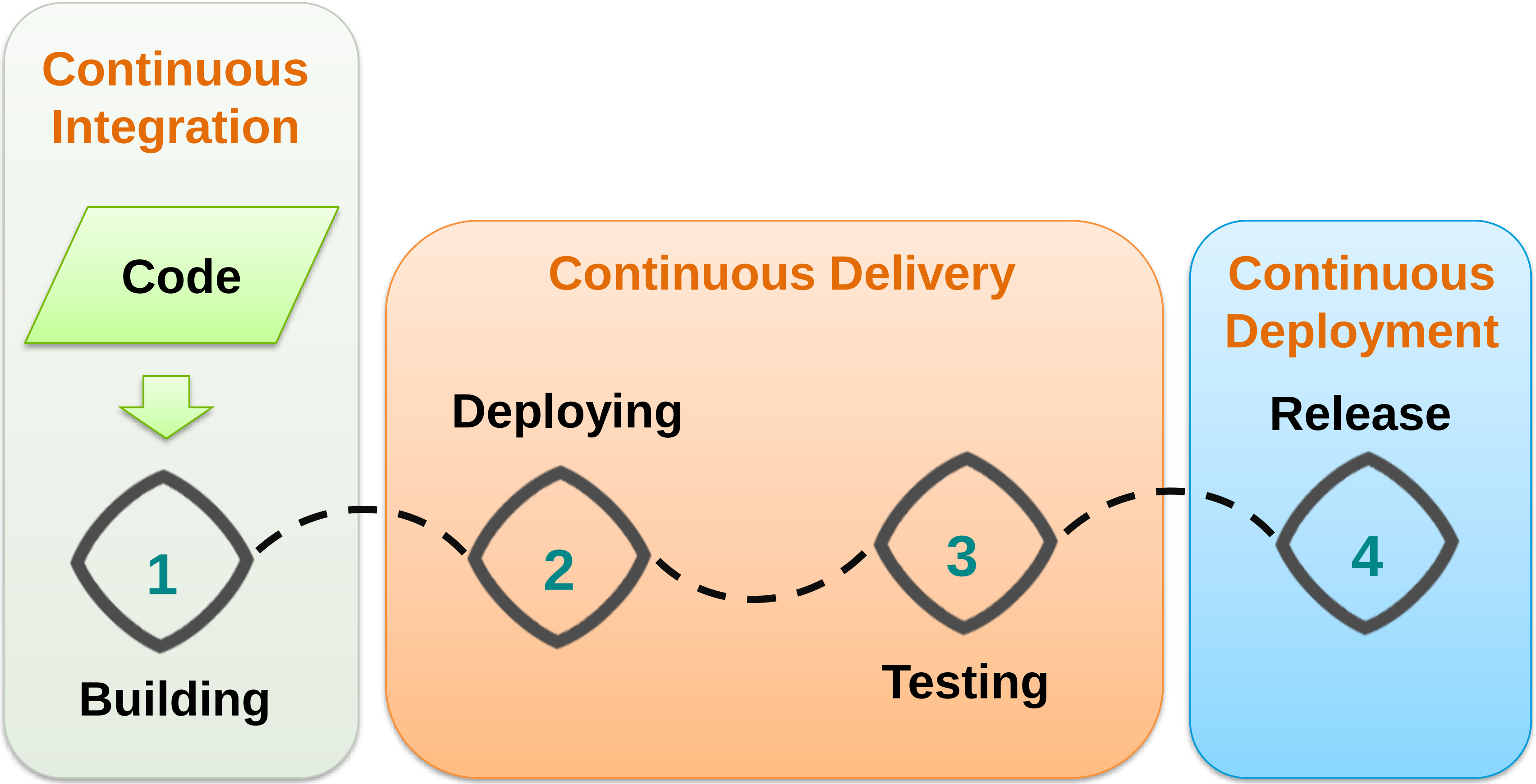


More info: Building, Deploying and Testing an Industrial Linux Platform
Open Source Summit Japan 2017 [51]

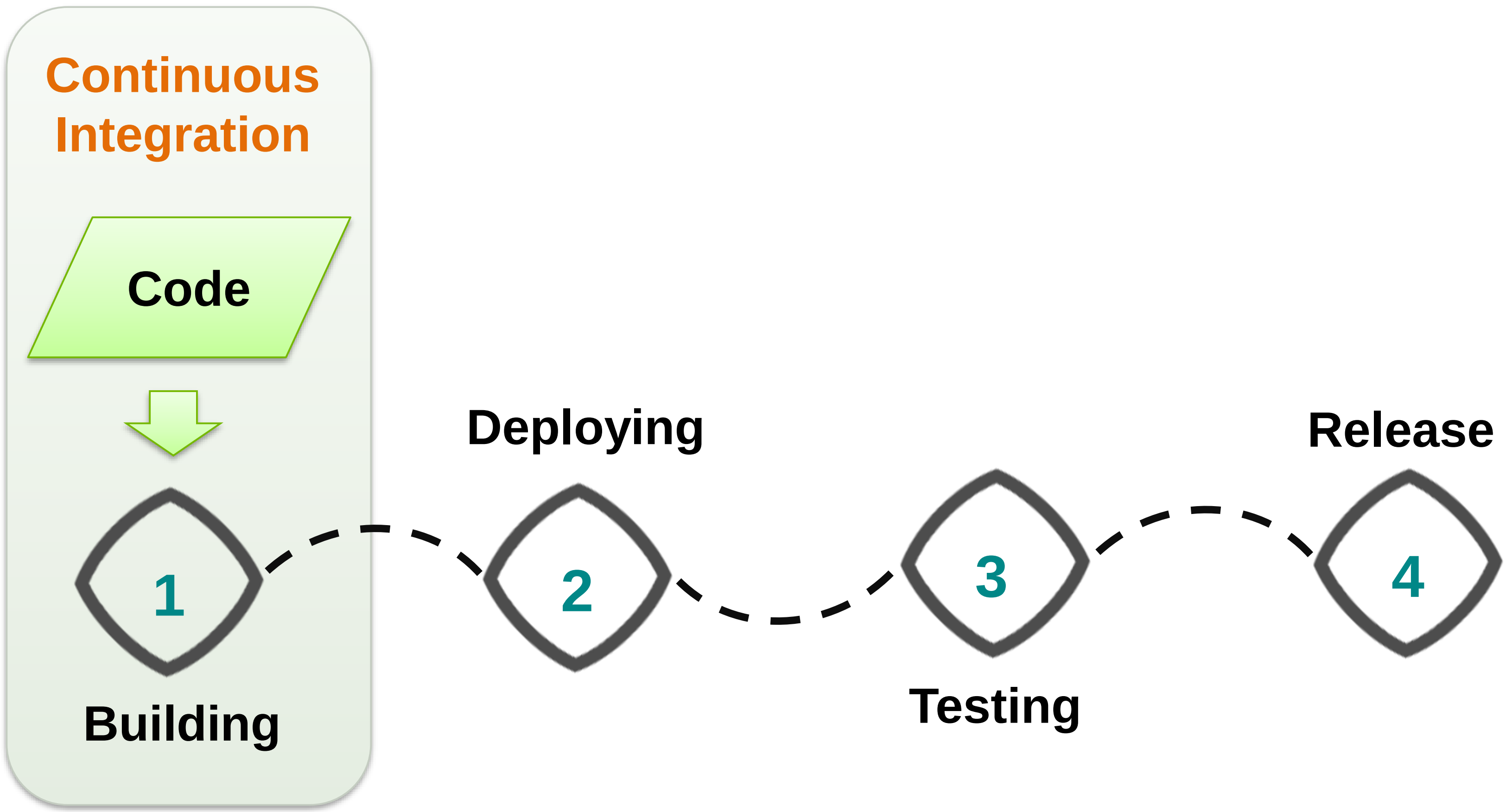

git
Master
Develop
Feature ..
Merge
Pull Request
Fork
git
Patches
Local Branch
Internal/ External Developers
CI/ CD Automatic Release Pipeline
Pass
N
Y
Maintainer Approval
Y
Pass
N
REDMINE
Notification

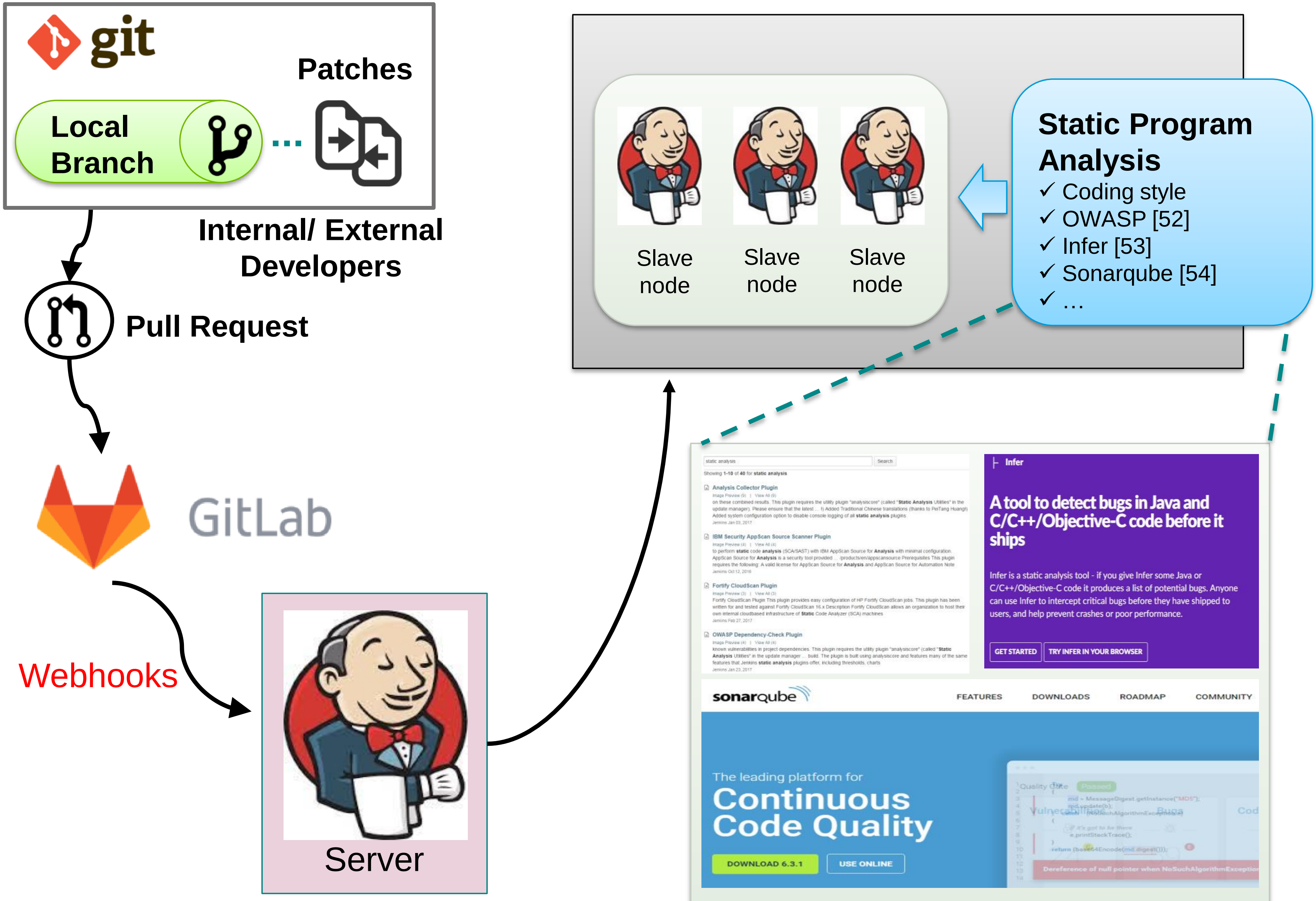
MOXA

CI/ CD Automatic Release Pipeline

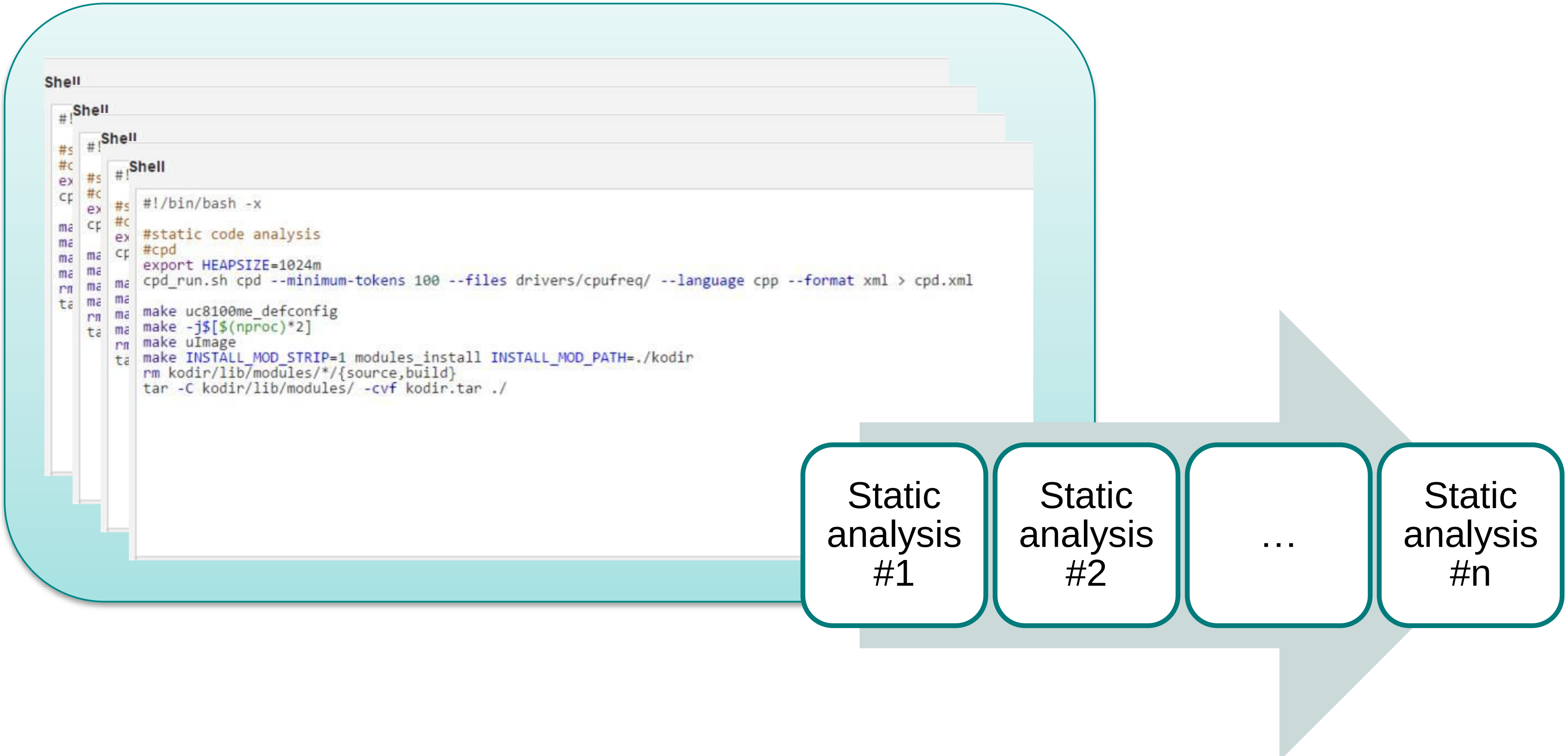


CI/ CD Automatic Release Pipeline




git
Patches
Local Branch
Internal/ External Developers
Pull Request
GitLab
Webhooks
Server
Slave node
Slave node
Slave node
Static Program Analysis
✓ Coding style
✓ OWASP [52]
✓ Infer [53]
✓ Sonarqube [54]
✓ …
Infer
A tool to detect bugs in Java and C/C++/Objective-C code before it ships
Infer is a static analysis tool - if you give Infer some Java or C/C++/Objective-C code it produces a list of potential bugs. Anyone can use Infer to intercept critical bugs before they have shipped to users, and help prevent crashes or poor performance.
GET STARTED
TRY INFER IN YOUR BROWSER
sonarqube
FEATURES
DOWNLOADS
ROADMAP
COMMUNITY
The leading platform for
Continuous Code Quality
DOWNLOAD 6.3.1
USE ONLINE
Analysis Collector Plugin
IBM Security AppScan Source Scanner Plugin
Fortify CloudScan Plugin
OWASP Dependency-Check Plugin

Static Testing Cases Management - Jenkins



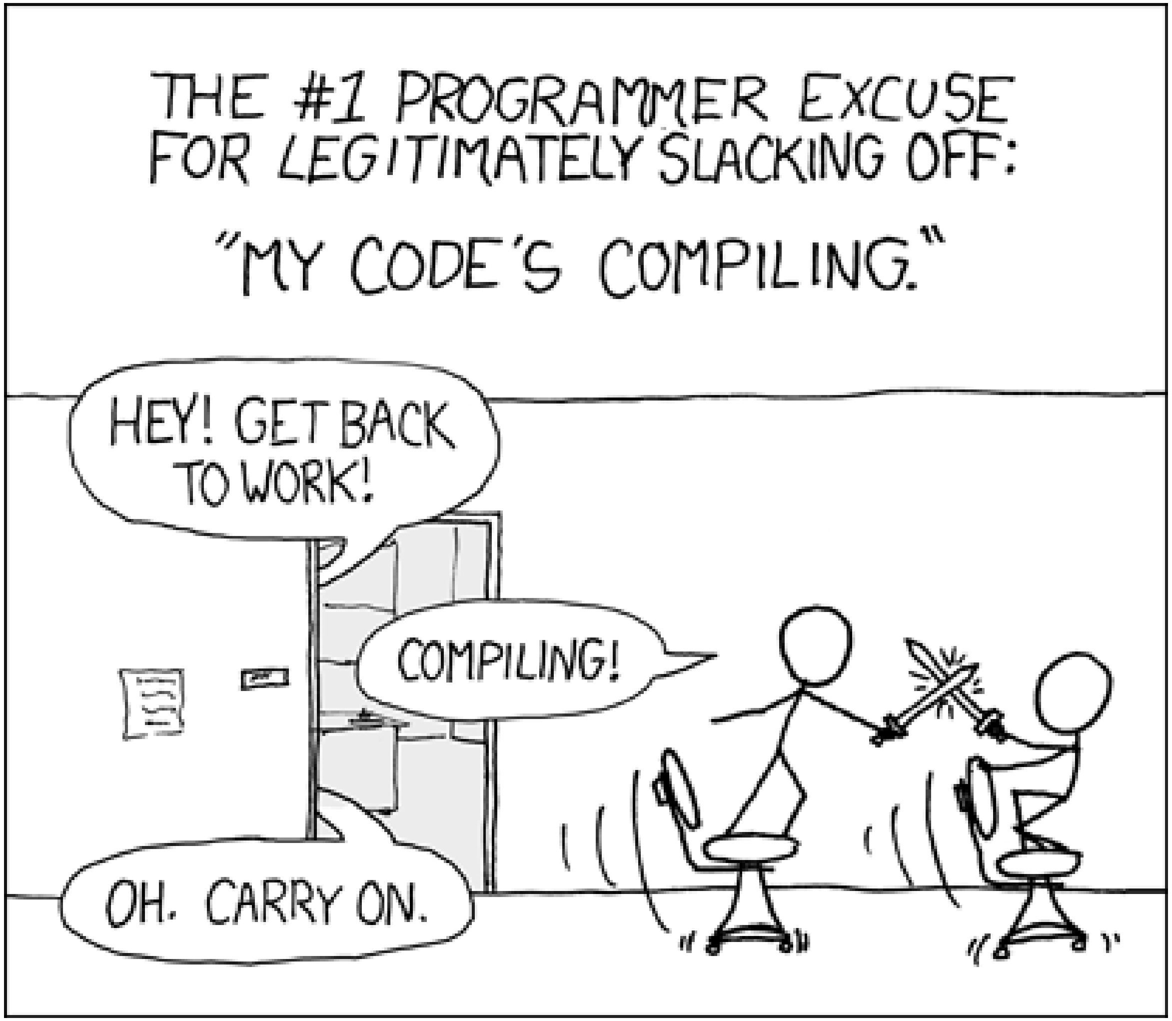
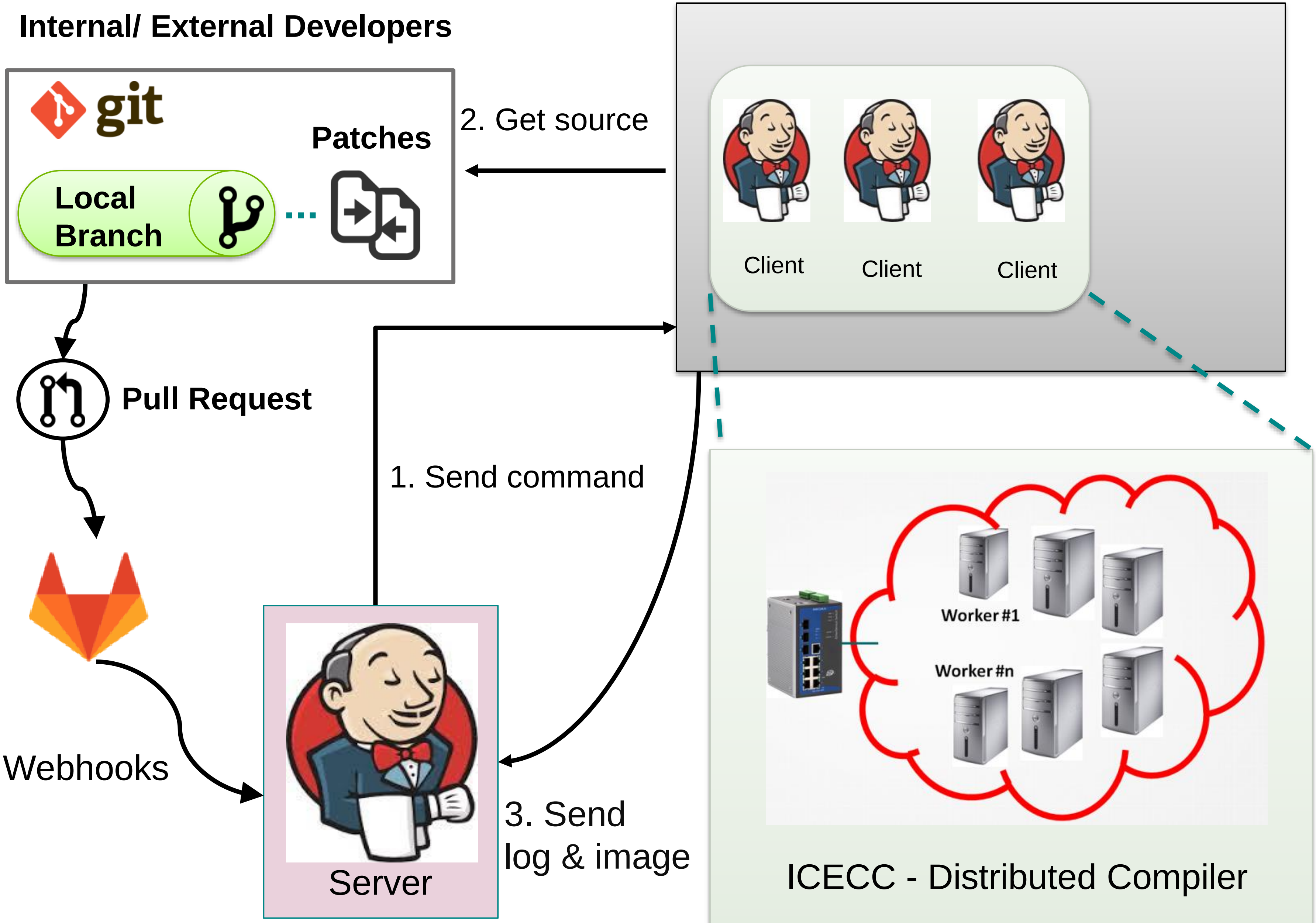


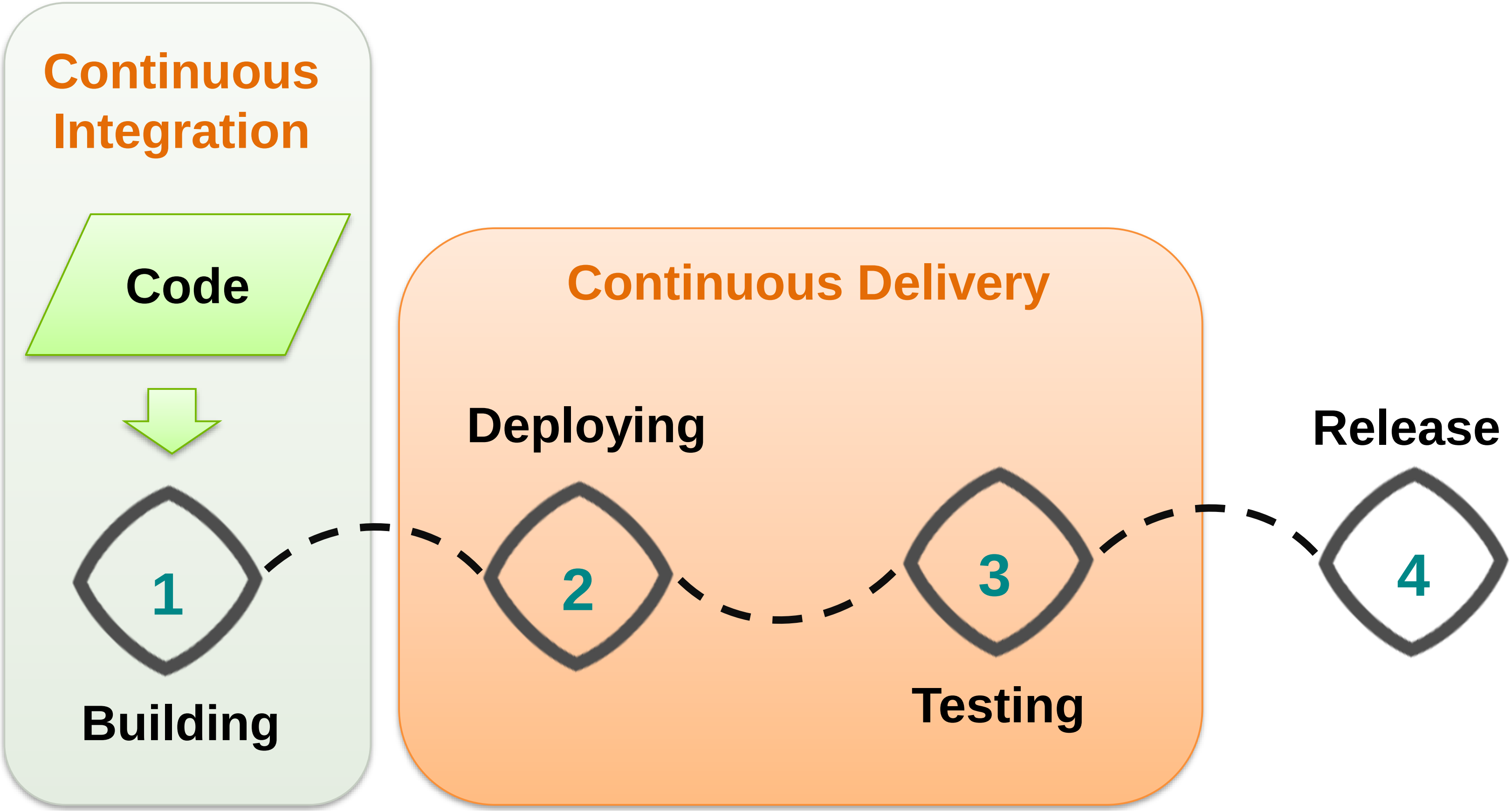
Image: https://c1.staticflickr.com/5/4030/4438139050_04604b4908.jpg

Distributed Compiler

- **Software**
  - **Icecream/ IceCC was created by SUSE based on distcc** [55][56]
    - Improve performance of compile jobs in parallel
    - Add dynamic scheduler of the compilation jobs
    - Support multiple platform
    - Support cross compiling
- **Hardware - for each node**
  - **SSD**
  - **Large capacity memory**
  - **Gigabit LAN**


Internal/ External Developers
git
Patches
Local Branch
2. Get source
Client
Client
Client
Pull Request
1. Send command
Worker #1
Worker #n
Webhooks
3. Send log & image
Server
ICECC - Distributed Compiler

CI/ CD Automatic Release Pipeline



Continuous Delivery – LAVA [57][58]

LAVA 2019.05.post1+stretch Index Contents Page ▾ Contents » Search

Introduction to LAVA

Navigation

Use the navigation bar at the top of each page to quickly navigate between sections of the documentation; Index, Contents, Page and Next.

Index

The Help Index is often the quickest way to find specific sections of the documentation.

Contents

If you are new to LAVA, the Help Contents describes several useful starting points, depending on how you expect to use LAVA.

Page indices

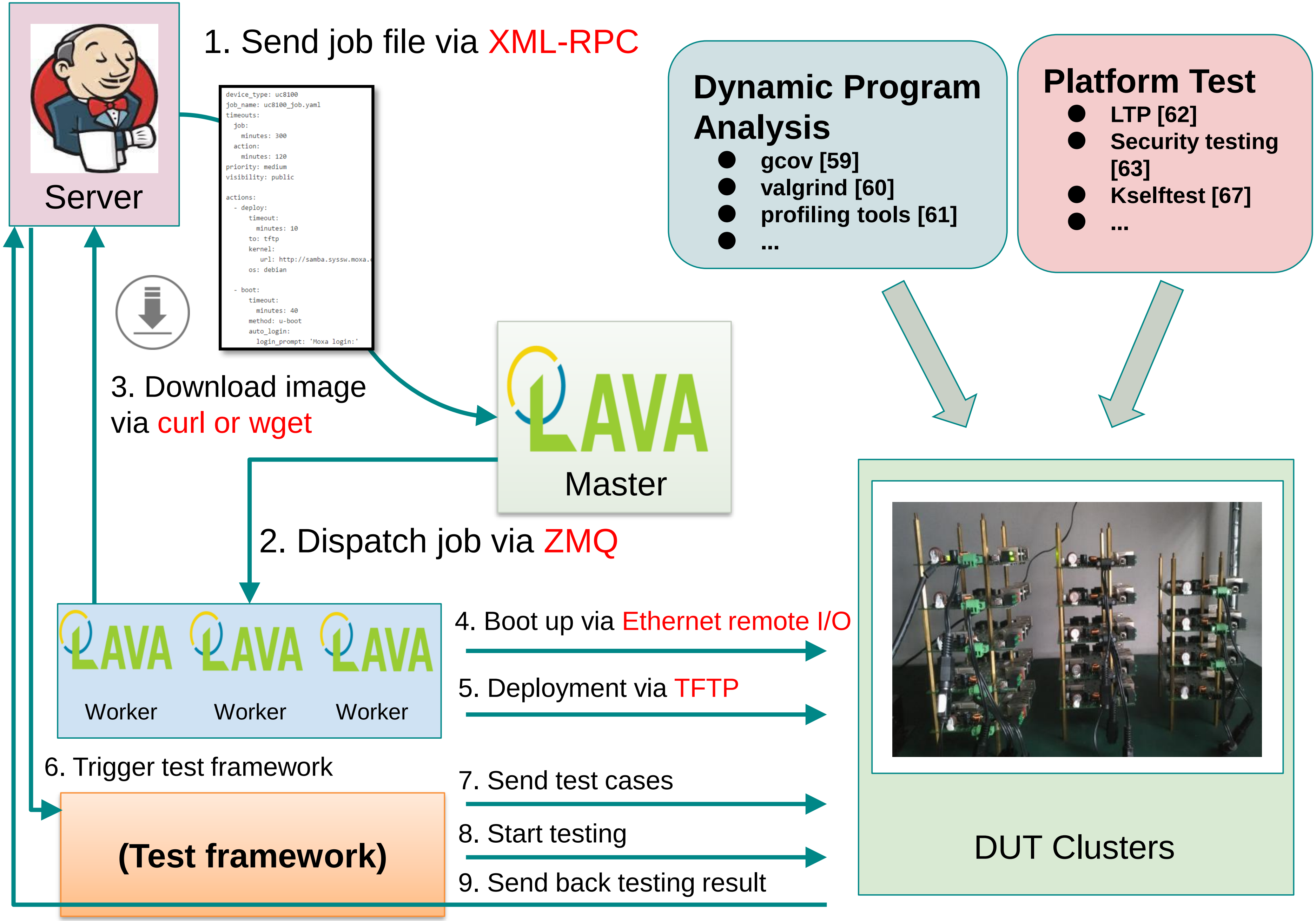
Each page also has a **Page** menu for topics within the page as well as forward and back navigation to lead readers through in a logical manner.

About LAVA V2

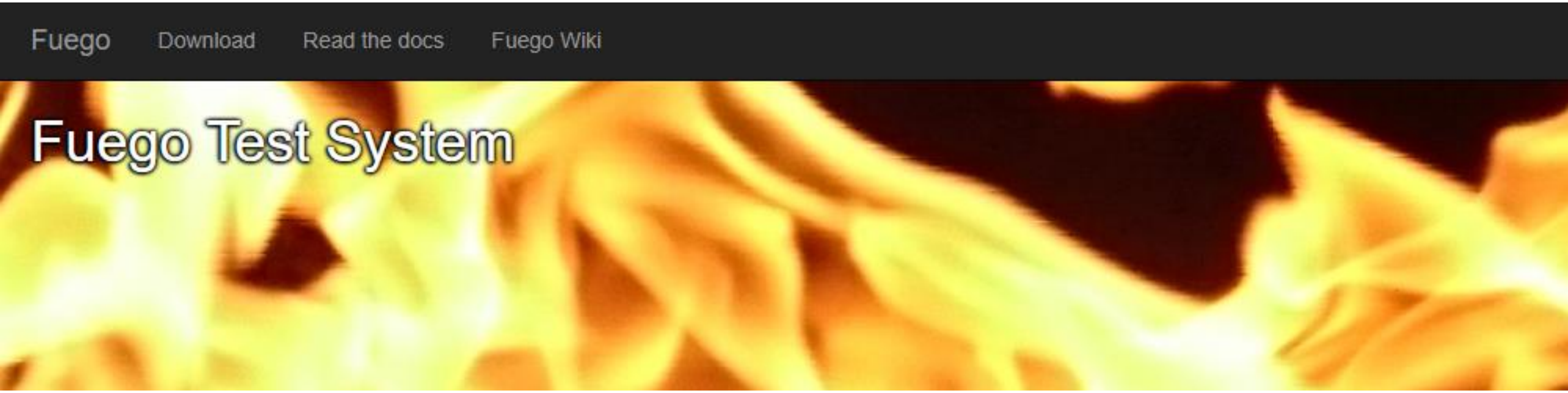
LAVA V2 is the second major version of LAVA. The major user-visible features are:

- The Pipeline model for the dispatcher
- YAML job submissions
- Results
- Queries
- Charts
- Data export APIs

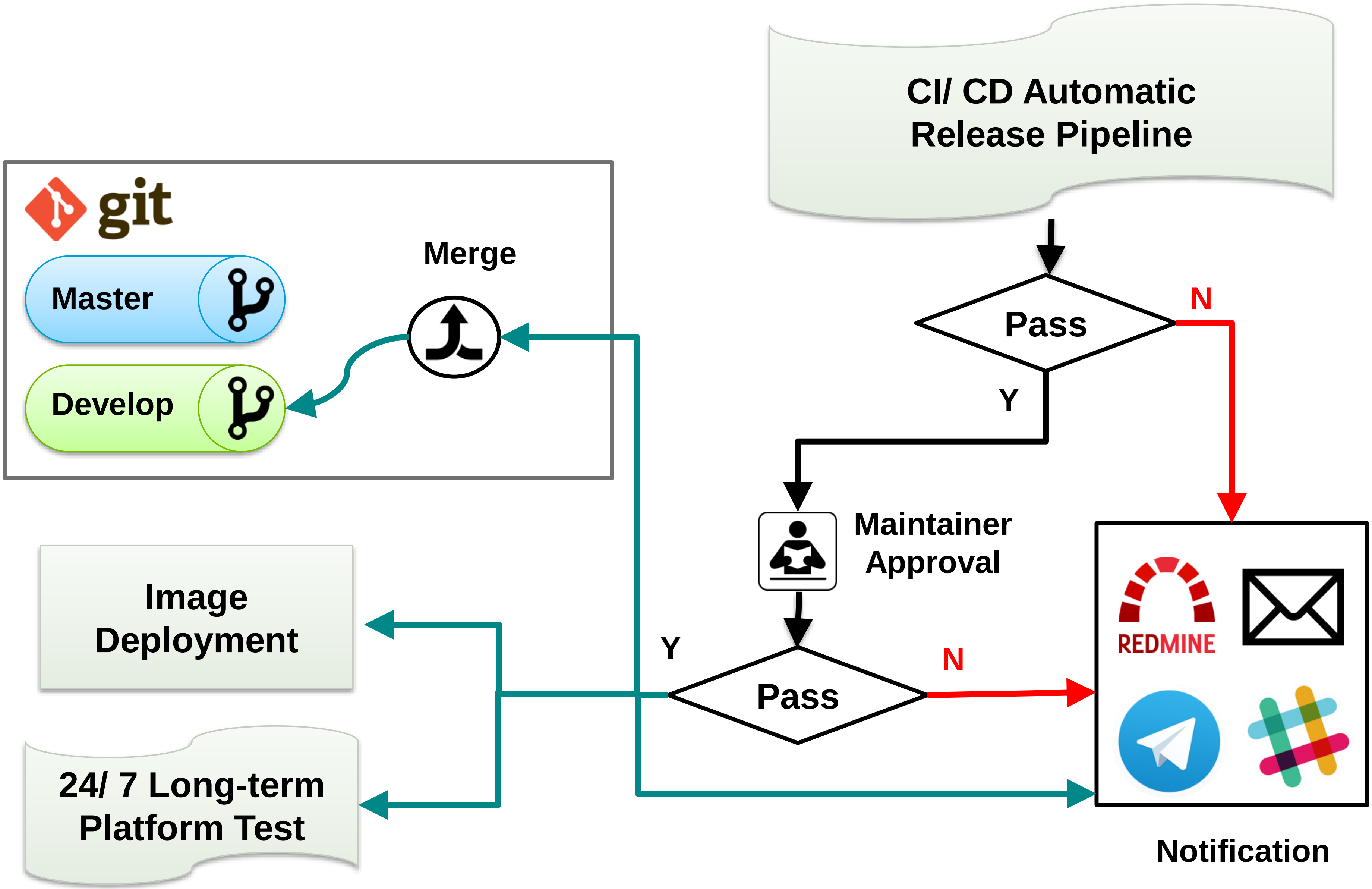
The architecture has been significantly improved since V1, bringing major changes in terms of how a distributed LAVA instance is installed, configured and used for running test jobs.


Server
1. Send job file via XML-RPC
device_type: uc8100
job_name: uc8100_job.yaml
timeouts:
job:
minutes: 300
action:
minutes: 120
priority: medium
visibility: public
actions:
- deploy:
timeout:
minutes: 10
to: tftp
kernel:
url: http://samba.syssw.moxa.
os: debian
- boot:
timeout:
minutes: 40
method: u-boot
auto_login:
login_prompt: 'Moxa login:'
Dynamic Program Analysis
gcov [59]
valgrind [60]
profiling tools [61]
...
Platform Test
LTP [62]
Security testing [63]
Kselftest [67]
...
3. Download image via curl or wget
LAVA
Master
2. Dispatch job via ZMQ
LAVA
Worker
LAVA
Worker
LAVA
Worker
4. Boot up via Ethernet remote I/O
5. Deployment via TFTP
6. Trigger test framework
(Test framework)
7. Send test cases
8. Start testing
9. Send back testing result
DUT Clusters

Fuego [68][69]

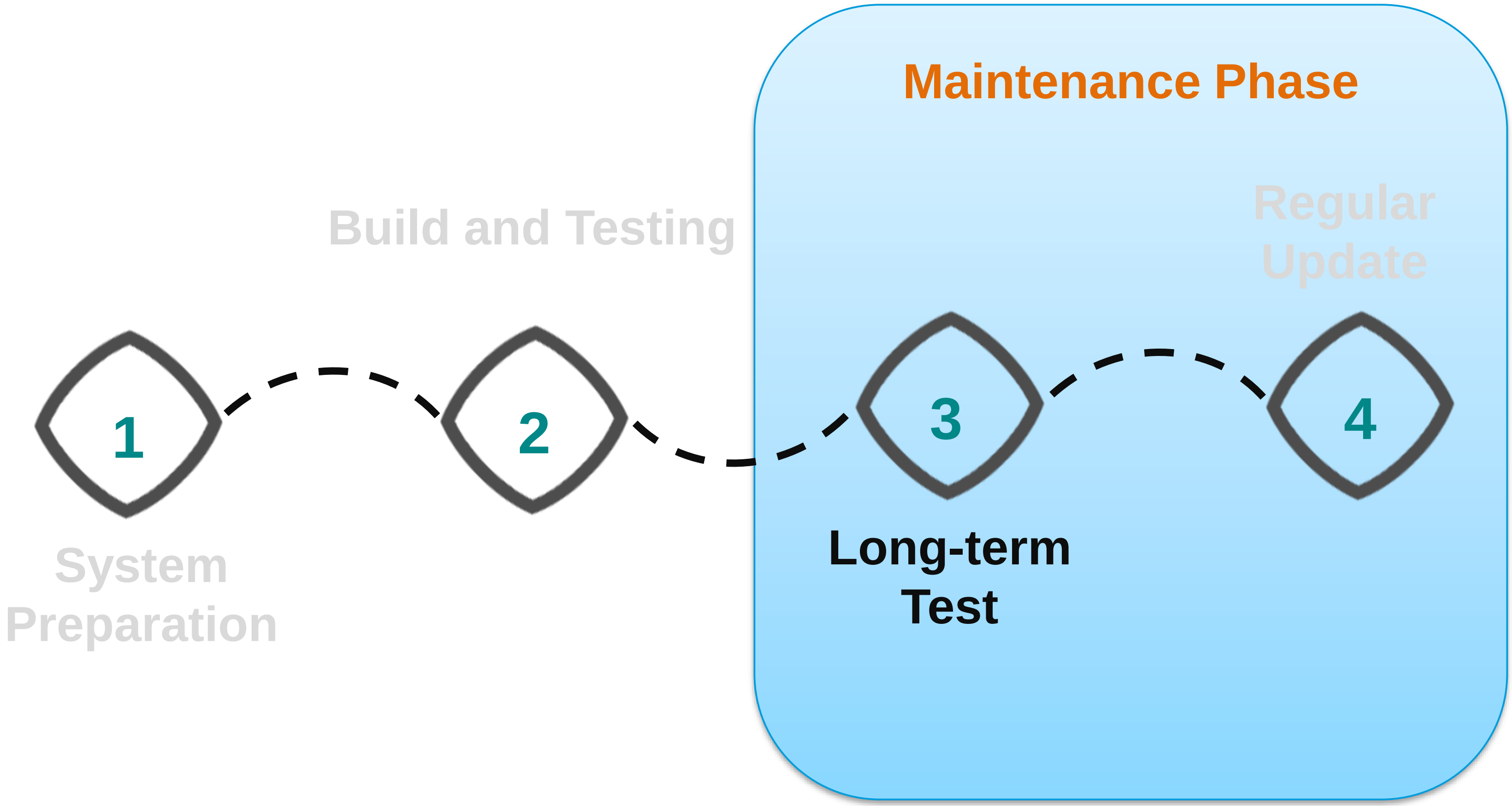


- **Test framework for testing embedded Linux**
  - **Official automated test framework for the LTSI project.**
    - BSD 3-Clause license in default
    - Over 100 pre-packaged tests
  - **Ability for 3rd parties to initiate or schedule tests on our hardware, and the ability to share our test results with others.**

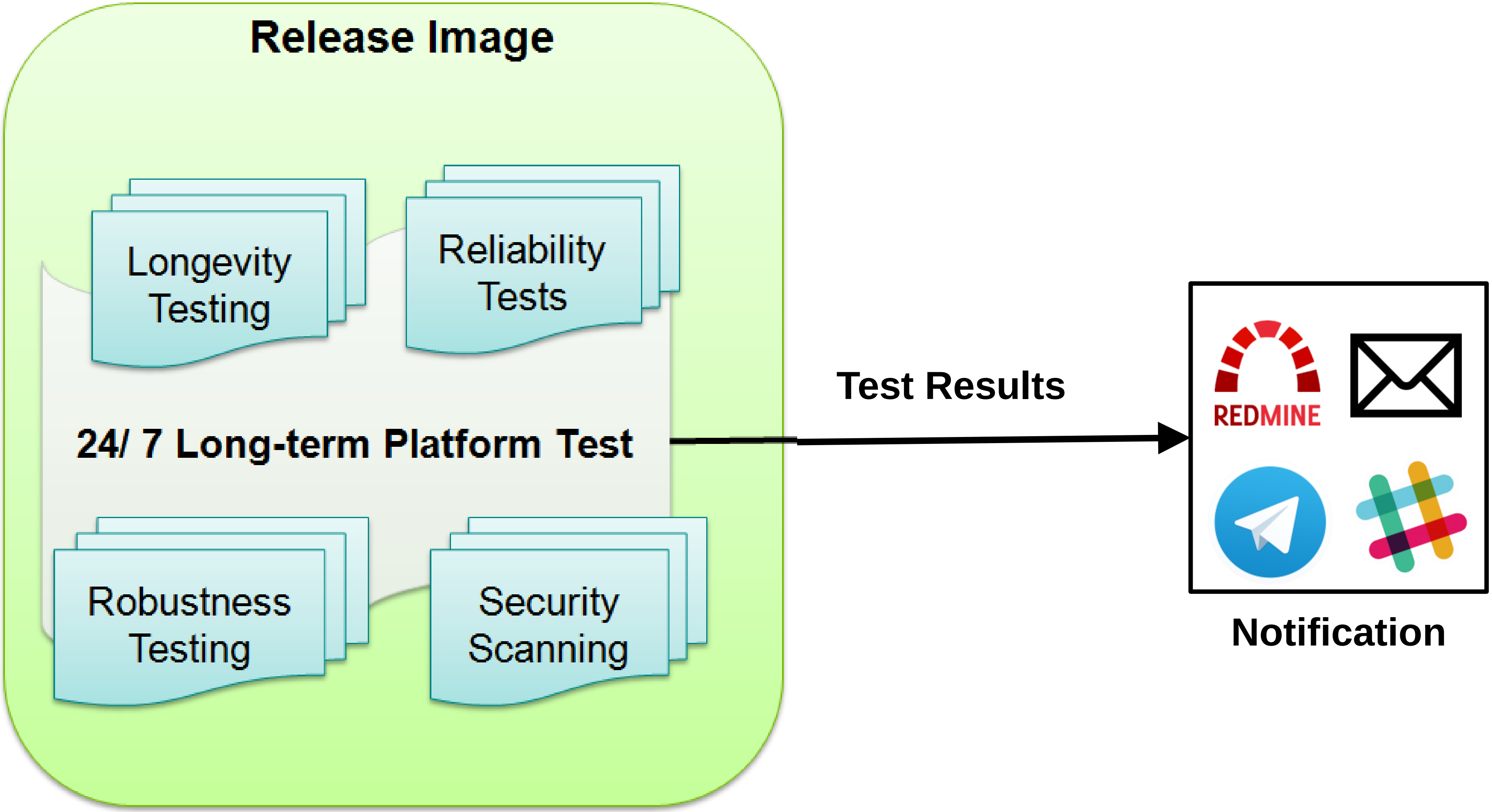

CI/ CD Automatic
Release Pipeline
git
Master
Develop
Merge
Pass
N
Y
Maintainer
Approval
Image
Deployment
Y
Pass
N
REDMINE
24/ 7 Long-term
Platform Test
Notification

Maintenance Phase

Long-term Testing and Regular Update



More info: Building, Deploying and Testing an Industrial Linux Platform
Open Source Summit Japan 2017 [51]



* Test cases are managed by test framework

24/ 7 Long-term Platform Test

Endurance test
Compatibility test
…

Longevity

Long-term support at least **10 years life cycle** with bug fixes, new features and new hardware components

Robustness

Robustness is the ability of a computer system to cope with errors during execution and cope with erroneous input [71]

Reliability

Reliability is enhanced by features that help to avoid, detect and repair hardware faults [72]

Security

Quick response in resolving CVE/ vulnerabilities and attacks in platform

24/ 7 Long-term Platform Test

Fuzz testing
[64][65][66]
…

Longevity

Long-term support at least **10 years life cycle** with bug fixes, new features and new hardware components

Robustness

Robustness is the ability of a computer system to cope with errors during execution and cope with erroneous input [71]

Reliability

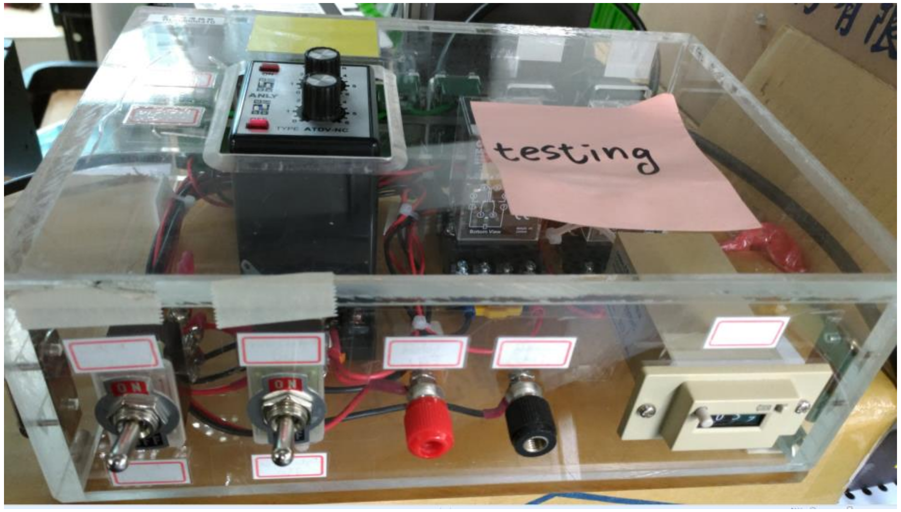
Reliability is enhanced by features that help to avoid, detect and repair hardware faults [72]

Security

Quick response in resolving CVE/ vulnerabilities and attacks in platform

24/ 7 Long-term Platform Test

Power failure test
Reboot test
Regression test
…



Reliability

Reliability is enhanced by features that help to avoid, detect and repair hardware faults [72]

Security

Quick response in resolving CVE/ vulnerabilities and attacks in platform

24/ 7 Long-term Platform Test

Daily test for CVE [63]
…

Security

Quick response in resolving CVE/ vulnerabilities and attacks in platform

For Stable Kernel Maintenance

KernelCI

- **Automated Linux Kernel Testing** [73][74]
  - **Detect, bisect, report and fix regressions on upstream Kernel trees before release**
  - **Short tests on many configurations**

Reproducible Builds [75]

- **Create an independently-verifiable path from source to binary**
  - **Ensure builds have identical results**
  - **Act as part of a chain of trust**
  - **Prove the source code has not been tampered/modified**

Open Source Testing Tools

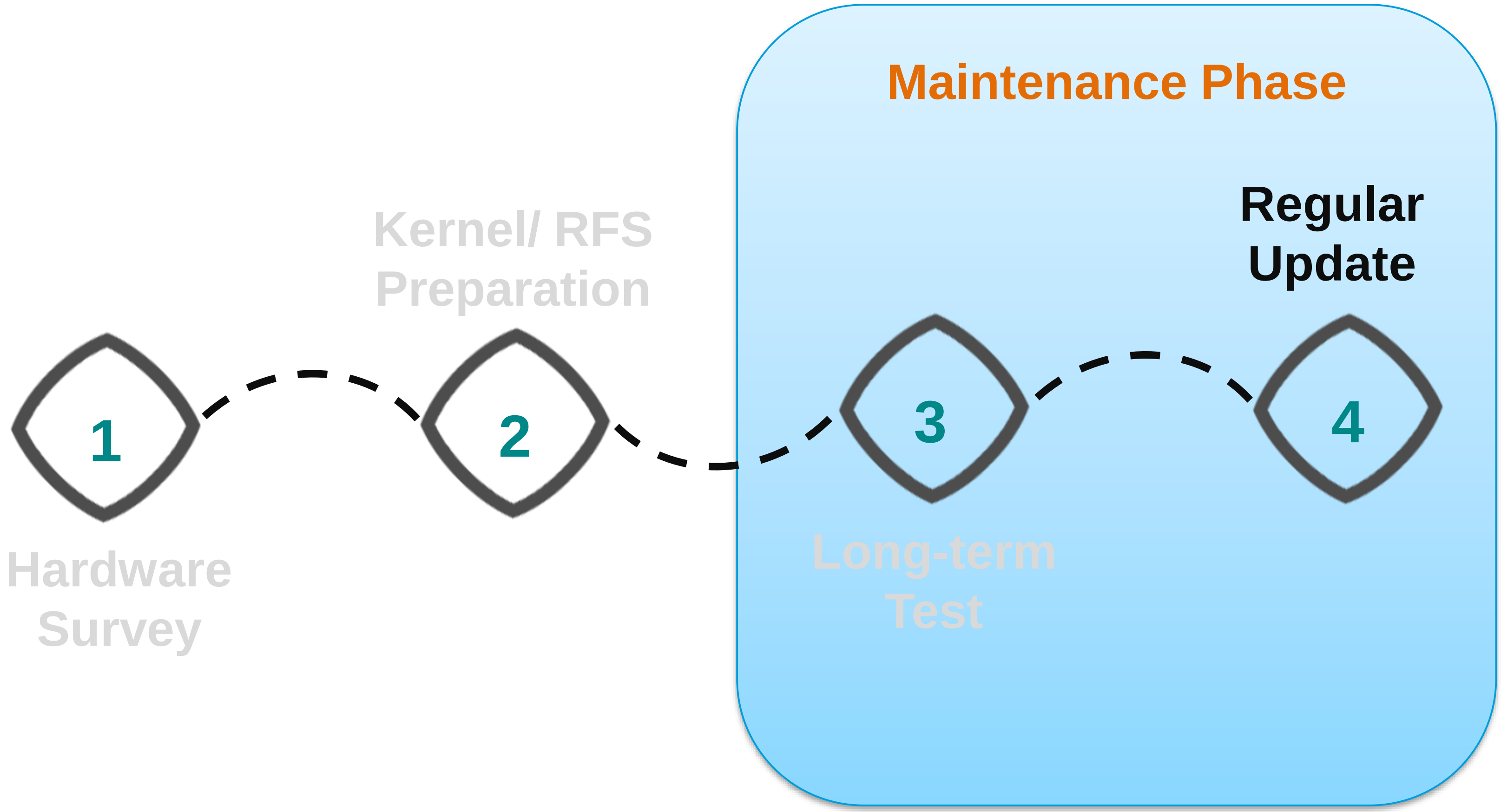
Continuous Integration	• Jenkins [78] • Jenkins X [79]
Continuous Delivery/ Deployment	• LAVA 2 [57]
Distributed compiler service	• icecc [55] • GOMA [80][81] • distcc [82]
Test Case Management	• Jenkins • LAVA 2 • Fuego [68][69]
Version Control	• Git with gitlab [83]
Static Program Analysis	• Coding style • OWASP [52] • Infer [53] • Sonarqube [54]
Dynamic Program Analysis	• Gcov [59] • Valgrind [60] • Profiling tools [61]
Security Testing	• OpenVAS [63] • Vuls [84]
Fuzzing Testing	• Syzkaller [64] • Trinity [65] • OSS-fuzz [66]

CI/ CD/ LT are

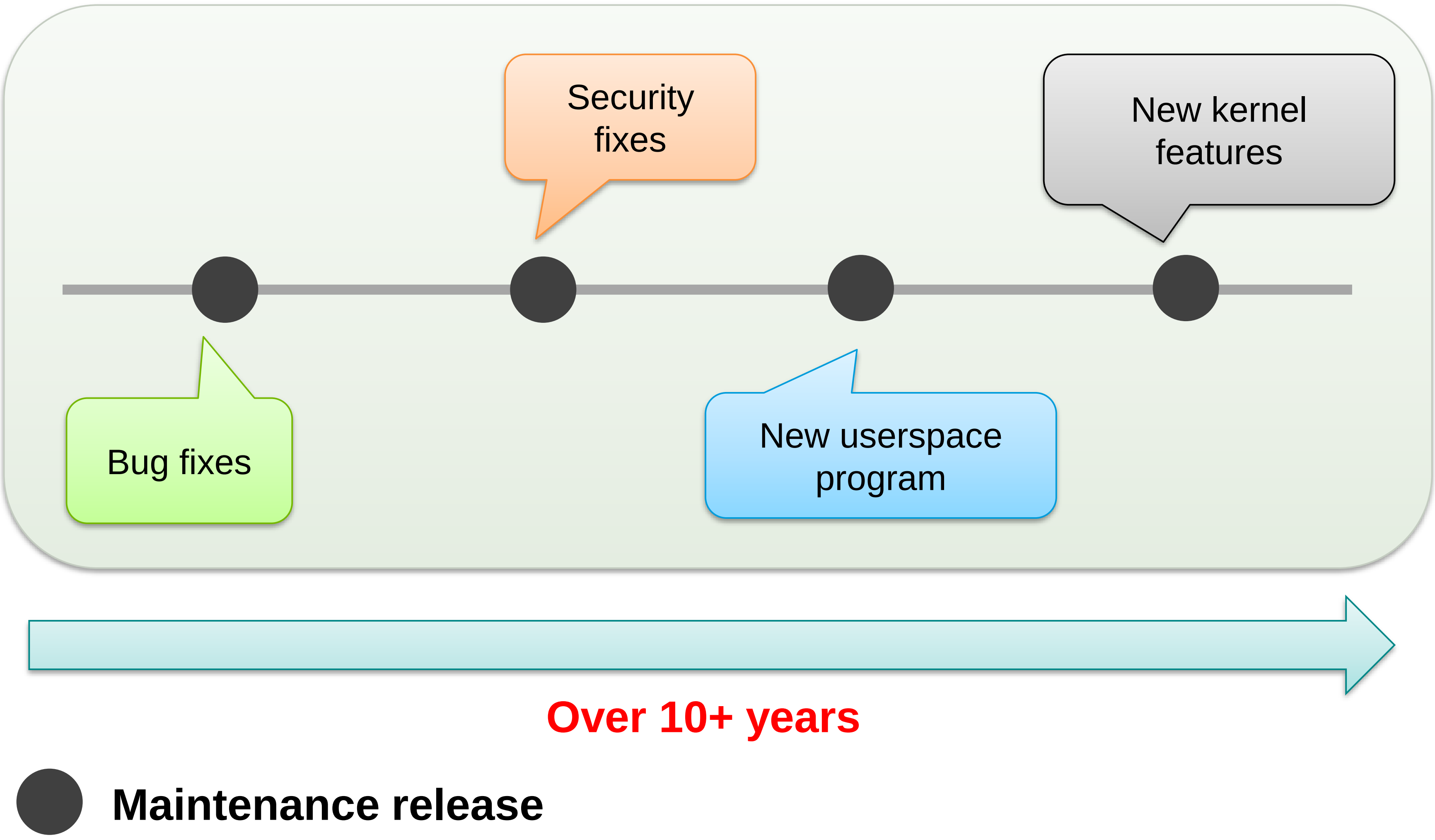
concepts of software engineering

instead of

tools or procedures


Maintenance Phase
Kernel/ RFS Preparation
Regular Update
1
2
3
4
Hardware Survey
Long-term Test

Why We Need Software Update?



The Components Might Be Updated

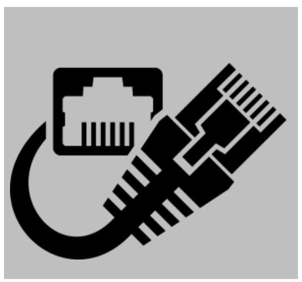
Components	Size	Update frequency	Risk
Peripheral devices firmware	< 10 MB	Rarely	Mid
Bootloader (including SPL)	< 1 MB	Rarely	High
Device tree	<100 kB	Rarely	High
Linux kernel	< 10 MB	Regularly	High
Root file system	Variant	Regularly	High
System configuration	< 1 MB	Rarely	Low
Application	Variant	Often	Low

MOXA

Characteristics of Industrial Embedded Linux Platform

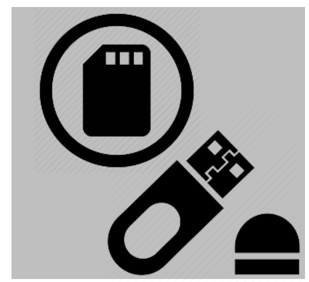
1. **Harsh environment**
   Unreliable network and power supply
2. **Middle of nowhere**
   Human-less warehouse or site
3. **Bandwidth limited**
   Wireless focus
4. **Multiple version supported**
   Rollback version
5. **Multiple devices**
   Remote management
6. **Longevity**
   Long-term support at least **10 years life cycle**

The Media for Software Update



Wire cable

OTA



Portable storage

On-site

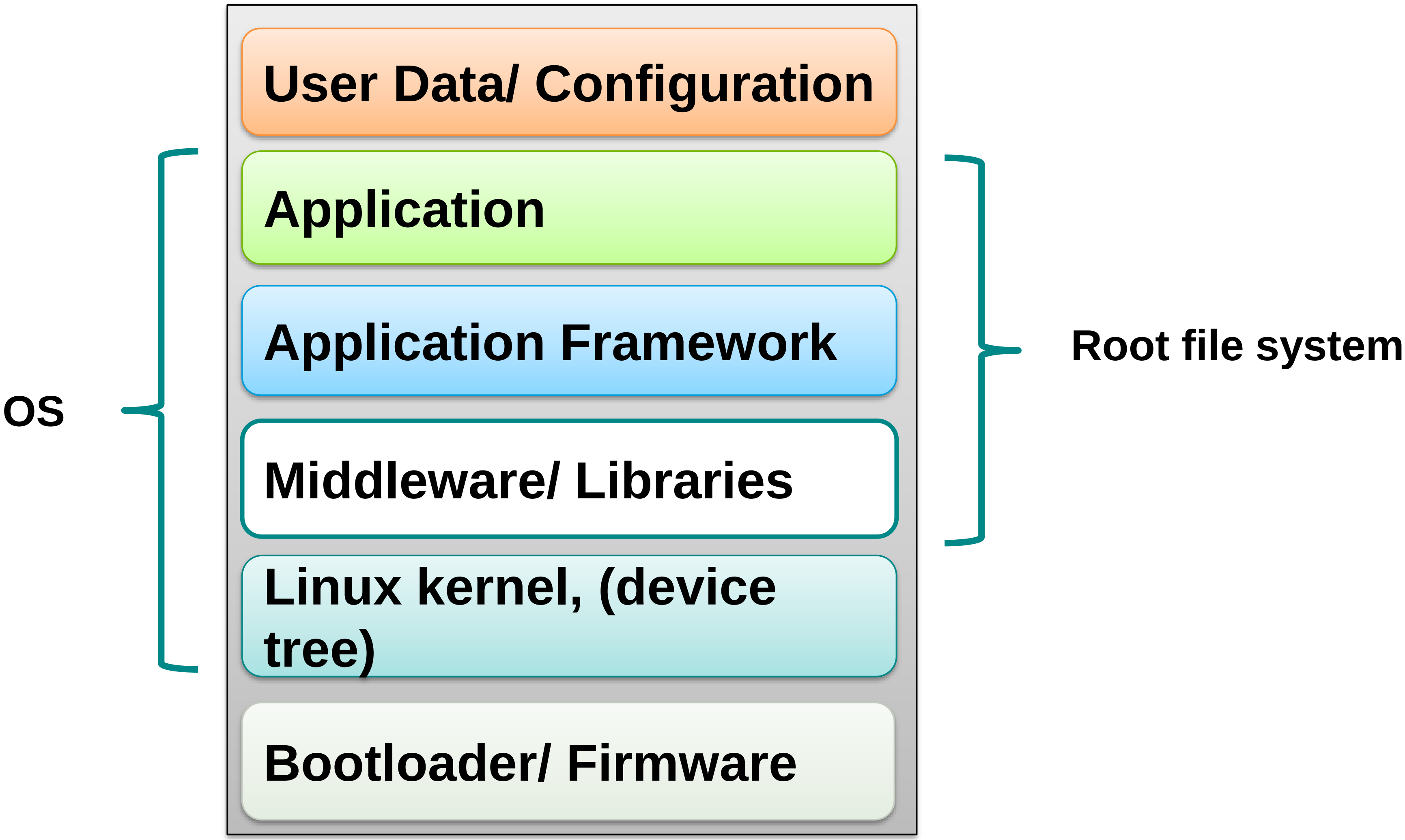
Software Update Requirements

Basic Features	
Fail-safe	
Roll-back	
Size reduction	
Signatures	
Multiple storage type support (e.g., NOR/NAND flash, eMMC)	
Build system integration	
Remote access (e.g., OTA)	
Additional Features	
Online and offline updates	
Encryption	
Delta-updates	
Successful update detection	
Proactive updating	

Update Approaches

Components	Size	Complexity	Time Cost
Image/ block based	Large	Low	Very High
File based	Variant	Low	Variant
Package based (e.g., deb, rpm)	Variant	Low	Variant
Delta based	Low	Very High	Low

Partition Architecture



Asymmetric/ Symmetric Firmware Updates [85]

User Data/ Configuration	User Data/ Configuration
Recovery OS	Main OS – A (Active)
Main OS	Main OS – B (Inactive)
Bootloader/ Firmware	Bootloader/ Firmware

Asymmetric Firmware Updates

- Fail-safe
- Downtime

Symmetric Firmware Updates

- Seamless update
- Roll-back
- Fail-safe
- Double copy of OS

Comparison - Features

Category	Fail-Safe	Roll-Back	Delta-Updates	Signatures	Multiple Storage Type Support	Build System Integration
SWUpdate	Y	Y	librsync	Y	•NOR NAND flashes •UBI volumes •SD / eMMC	Yocto/ Buildroot
RAUC	Y	Y	casync	Y	•NOR NAND flashes •UBI volumes •SD / eMMC	Yocto/ Buildroot
OSTree	N	Y	archive-z2	Y	?	Yocto

Comparison - Others

Method	Asymmetric/ Symmetric Image Updates	Type	Language	License
SWUpdate	Both	Image-based File-based	C99	GPLv2 With openssl exception
RAUC	Both	Image-based File-based	C	LGPLv2.1
OSTree	Asymmetric	File-based	C/C++	MPL 2.0 /LGPLv2+

Conclusion

Preparedness Planning

Longevity, stability and security

Community Collaboration

Different approach for multiple target applications

Thank You

Q & A

References

[1] https://www.openchainproject.org
[2] https://www.openinventionnetwork.com/
[3] https://spdx.org/
[4] https://www.fossology.org/
[5] https://en.wikipedia.org/wiki/Das_U-Boot
[6] https://en.wikipedia.org/wiki/Coreboot
[7] https://en.wikipedia.org/wiki/Booting#Modern_boot_loaders
[8] http://www.rodsbooks.com/refind/
[9] https://en.wikipedia.org/wiki/REFInd
[10] https://www.kernel.org
[11] https://wiki.linuxfoundation.org/realtime/start
[12] https://tiny.wiki.kernel.org/start
[13] https://bootlin.com/pub/conferences/2017/jdll/opdenacker-embedded-linux-in-less-than-4mb-of-ram/opdenacker-embedded-linux-in-less-than-4mb-of-ram.pdf
[14] https://xenomai.org/
[15] https://www.rtai.org/

References

[16] https://www.kernel.org/category/releases.html
[17] https://ltsi.linuxfoundation.org/
[18] https://events.linuxfoundation.org/wp-content/uploads/2017/11/Using-Linux-for-Long-Term-Community-Status-and-the-Way-We-Go-OSS-Tsugikazu-Shibata.pdf
[19] https://www.cip-project.org/
[20] https://elisa.tech/
[21] http://www.osadl.org/SIL2LinuxMP.sil2-linux-project.0.html
[22] https://wiki.linuxfoundation.org/gsoc/2019-gsoc-safety-critical-Linux
[23] https://lists.elisa.tech/login?r=%2Ftopics
[24] https://events.static.linuxfound.org/sites/events/files/slides/libc-talk.pdf
[25] https://www.gnu.org/software/libc/
[26] https://uclibc-ng.org/
[27] http://events.linuxfoundation.org/sites/events/files/slides/uclibc-still-makes-sense-brodkin-elce2017.pdf
[28] https://www.musl-libc.org/
[29] https://en.wikipedia.org/wiki/Year_2038_problem

References

[30] https://en.wikipedia.org/wiki/Linux_startup_process

[31] http://upstart.ubuntu.com/faq.html

[32] https://en.wikipedia.org/wiki/Systemd

[33] https://sysdfree.wordpress.com/2019/03/09/135/

[34] https://wiki.gentoo.org/wiki/Comparison_of_init_systems

[35] https://elinux.org/images/6/69/Demystifying_Systemd.pdf

[36] http://proteanos.com/

[37] https://www.piboxproject.com/

[38] https://lists.debian.org/debian-devel/2016/02/msg00122.html

[39] https://busybox.net/FAQ.html#libc

[40] https://wiki.yoctoproject.org/wiki/Stable_branch_maintenance

[41] https://www.debian.org/social_contract#guidelines

[42] https://bootlin.com/pub/conferences/2018/elc/petazzoni-buildroot-whats-new/petazzoni-buildroot-whats-new.pdf

References

[43] https://events.static.linuxfound.org/sites/events/files/slides/libc-talk.pdf

[44] http://events17.linuxfoundation.org/sites/events/files/slides/ELC%202016%20-%20Designing%20a%20distro%20from%20scratch%20using%20OpenEmbedded.pdf

[45] https://github.com/meta-debian/meta-debian

[46] https://events.linuxfoundation.org/wp-content/uploads/2017/12/ELCE2018_Debian-Yocto-State-of-the-Art_r6_Kazuhiro-Hayashi.pdf

[47] https://events.linuxfoundation.org/wp-content/uploads/2017/12/Buildroot-vs-Yocto-Differences-for-Your-Daily-Job-Luca-Ceresoli-AIM-Sportline.pdf

[48] https://events.static.linuxfound.org/sites/events/files/slides/belloni-petazzoni-buildroot-oe_0.pdf

[49] https://www.debian.org/intro/why_debian.en.html

[50] https://www.debian.org/security/index.en.html

[51] http://events.linuxfoundation.org/sites/events/files/slides/Building%2C%20Deploying%20and%20Testing%20an%20Industrial%20Linux%20Platform.pdf

References

[52] https://wiki.jenkins-ci.org/display/JENKINS/Plugins
[53] http://fbinfer.com/
[54] https://www.sonarqube.org/
[55] https://github.com/icecc
[56] https://www.slideshare.net/szlin/distributed-compiler-icecc
[57] https://validation.linaro.org/static/docs/v2/#
[58] http://elinux.org/images/3/35/LAVA_Project_Update.pdf
[59] https://gcc.gnu.org/onlinedocs/gcc/Gcov.html
[60] http://valgrind.org/
[61] https://perf.wiki.kernel.org/index.php/Main_Page
[62] http://linux-test-project.github.io/
[63] http://www.openvas.org/
[64] https://github.com/google/syzkaller
[65] http://codemonkey.org.uk/projects/trinity/
[66] https://github.com/google/oss-fuzz
[67] https://kselftest.wiki.kernel.org

References

[68] https://elinux.org/Fuego
[69] http://fuegotest.org/
[70] https://elinux.org/Automated_Testing_Summit_2019
[71] https://en.wikipedia.org/wiki/Robustness_(computer_science)
[72] https://en.wikipedia.org/wiki/Reliability,_availability_and_serviceability
[73] https://kernelci.org/
[74] https://fosdem.org/2019/schedule/event/kernelci_a_new_dawn/attachments/slides/3300/export/events/attachments/kernelci_a_new_dawn/slides/3300/gtucker_kernelci_fosdem_2019_v2_3_1024x768.pdf
[75] https://reproducible-builds.org/
[76] http://layer-acht.org/slides/2019-06-08-MiniDebConf-Hamburg--aiming-for-bullseye/#/7
[77] https://wiki.debian.org/ReproducibleBuilds
[78] https://jenkins.io
[79] https://jenkins.io/projects/jenkins-x/
[80] https://chromium.googlesource.com/infra/goma/server/
[81] https://chromium.googlesource.com/infra/goma/client

References

[82] https://github.com/distcc/distcc

[83] https://about.gitlab.com/

[84] https://vuls.io/

[85] https://mkrak.org/wp-content/uploads/2018/04/FOSS-NORTH_2018_Software_Updates.pdf

[86] https://events.linuxfoundation.org/wp-content/uploads/2017/12/Strategies-for-Developing-and-Deploying-your-Embedded-Applications-and-Images-Mirza-Krak-Mender.io_.pdf

[87] http://events17.linuxfoundation.org/sites/events/files/slides/ELC2017_SWUpdate.pdf

[88] https://events.linuxfoundation.org/wp-content/uploads/2017/12/ELCE-2018-Update-Tools-BoF_Jan-Lubbe.pdf

[89] https://events.linuxfoundation.org/wp-content/uploads/2017/12/ELCE-2018-Update-Tools-BoF_Jan-Lubbe.pdf

[90] https://elinux.org/images/f/f5/Embedded_Systems_Software_Update_for_IoT.pdf

[91] https://rauc.readthedocs.io/en/latest/

[92] http://elinux.org/images/6/6e/End_of_Time_--_Embedded_Linux_Conference_2015.pdf

[93] https://android.googlesource.com/platform/bionic