LINUX FOUNDATION
COLLABORATIVE PROJECTS

CIVIL INFRASTRUCTURE PLATFORM

CIP Kernel Team Activities to Accomplish Super Long Term Support

SZ Lin (林上智) @ Moxa Inc.
Masashi Kudo @ Cybertrust Japan

June 29th , 2020

About US

- **SZ Lin (林上智)** <sz.lin@moxa.com>
  - Working for Moxa Inc.
  - CIP Technical Steering Committee Member
  - OpenChain Project Governing Board Member
  - Debian Developer

- **Masashi Kudo** <masashi.kudo@miraclelinux.com>
  - Working for Cybertrust Japan Co., Ltd.
  - Acted as OpenDaylight (LF Networking) Ambassador
  - CIP Kernel Team Chair

CIVIL INFRASTRUCTURE PLATFORM

Table of Contents

What is CIP?

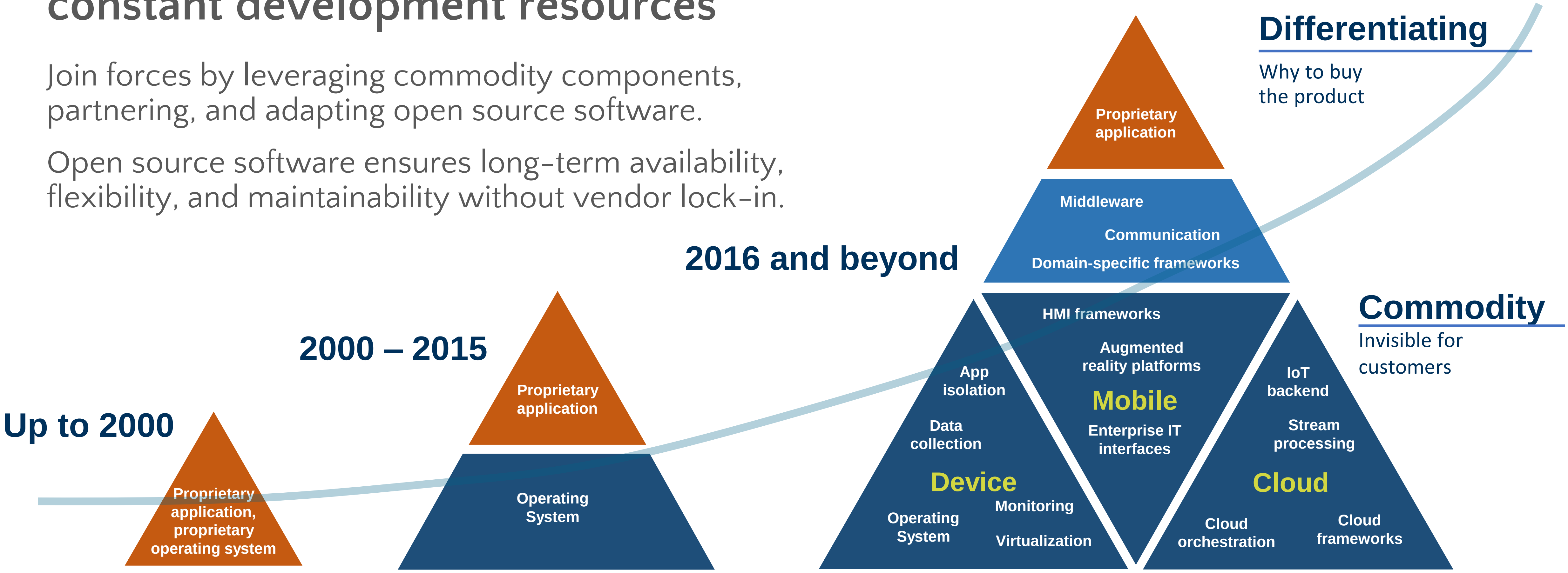
Image: http://zdnet1.cbsistatic.com/hub/i/r/2016/02/29/10863f77-89b2-40c0-9d8c-dbaa5feb65be/resize/770xauto/490141cef9bddc0db66b492698b53a50/pc

Speed and efficiency: focus on differentiating parts

Handling increasing complexity with constant development resources

Join forces by leveraging commodity components, partnering, and adapting open source software.

Open source software ensures long-term availability, flexibility, and maintainability without vendor lock-in.

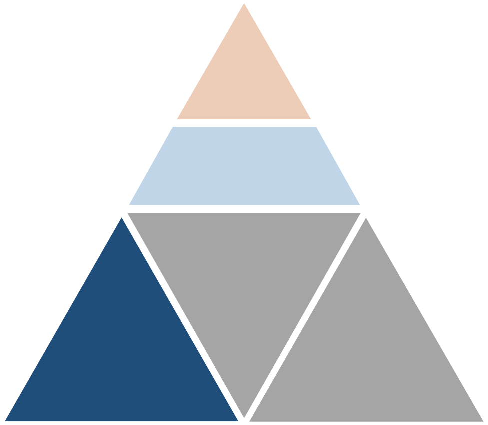


CIVIL INFRASTRUCTURE PLATFORM

Facts and Issues: Silo Development

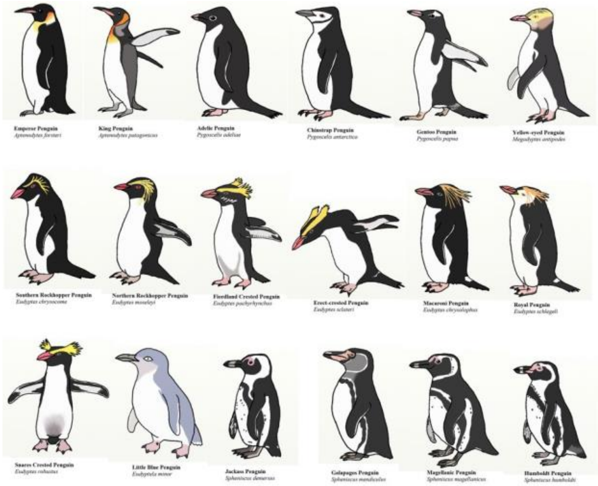
Facts

- Millions or trillions Industrial devices, including smart devices
- Similar software components (e.g. Linux)
- Industrial IoT requirements
  - Security
  - Sustainability
  - Industrial-gradeness



Issues

- A lot of products have to meet industrial requirements
- Same development and maintenance efforts spent by many companies or even business units
- **No common solution** for base building blocks



picture taken from Pinterest
https://www.pinterest.de/pin/554646510344033382/

CIVIL INFRASTRUCTURE PLATFORM

CIP is the Solution

Establishing an

Open Source Base Layer

of industrial-grade software to enable the use and implementation of software building blocks for Civil Infrastructure Systems

CIVIL INFRASTRUCTURE PLATFORM

What is “Open Source Base Layer (OSBL)”?

system-specific middleware and applications

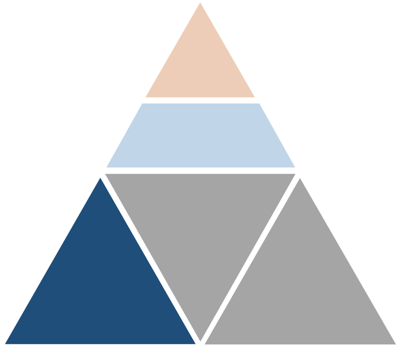
additional packages
(hundreds)

CIP Core packages
(tens)

CIP SLTS kernel
(10+ years maintenance, based on LTS kernels)

OSBL

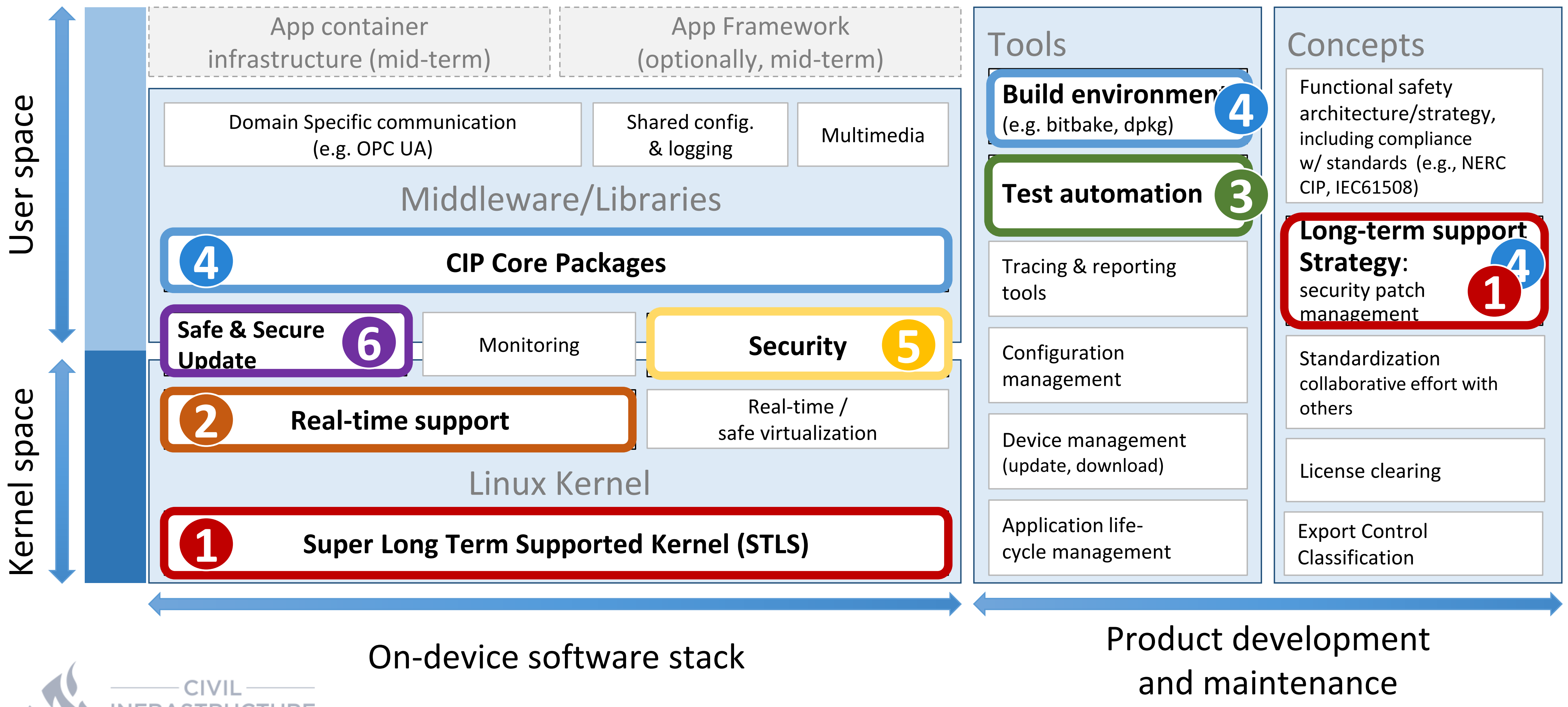
scope of
a typical Linux
distribution



CIVIL INFRASTRUCTURE PLATFORM

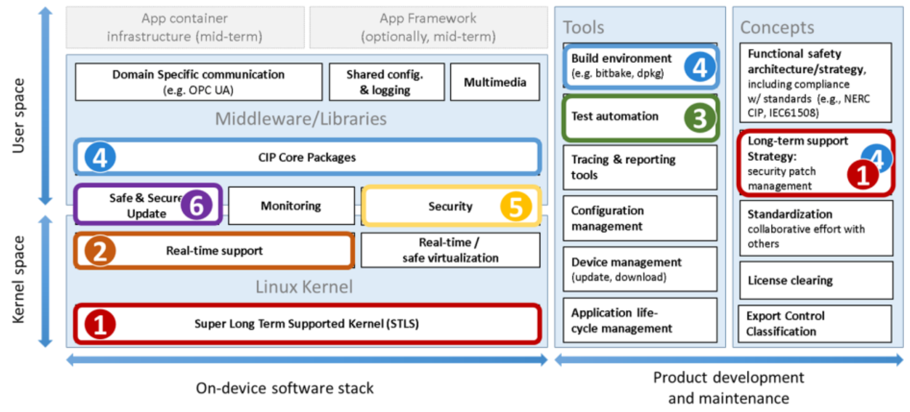
CIP Civil Infrastructure Platform Project (https://www.cip-project.org/) **SLTS** Super Long Term Support

Scope of activities



CIVIL INFRASTRUCTURE PLATFORM

CIP governance structure and projects



Governing Board (GB)

Technical Steering Committee (TSC)

(*): Workgroup

	1 SLTS kernel	2 Real-time	3 Testing	4 CIP Core	5 Security WG(*)	6 Software update WG(*)
Security	✔		✔	✔	✔	✔
Sustainability	✔		✔	✔		✔
Industrial grade	✔	✔	✔	✔	✔	✔

CIP Projects and its scopes

CIVIL INFRASTRUCTURE PLATFORM

The backbone of CIP are the member companies

CIVIL INFRASTRUCTURE PLATFORM

RENESAS, SIEMENS, TOSHIBA, Codethink, cybertrust, HITACHI Inspire the Next, MOXA, Plat'Home There, we are. Internet of Things

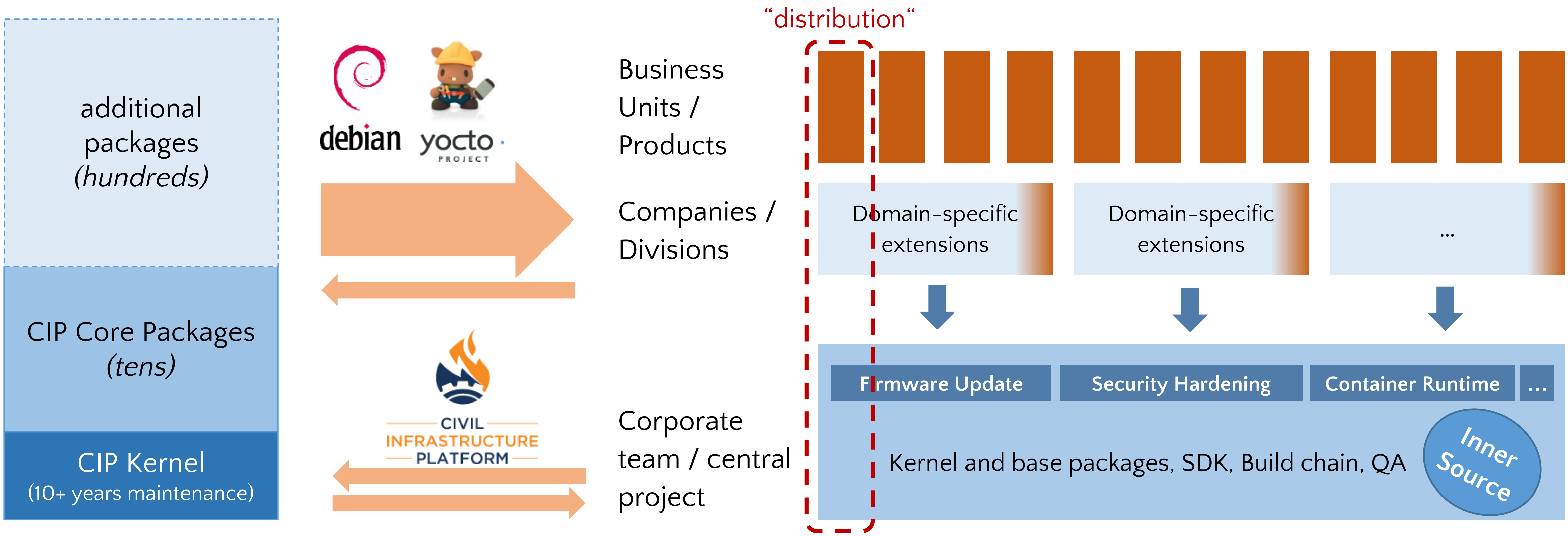
Budget ¥ € £ $ → Developers, maintainers

Optional: funding of selected projects

Contribution & usage / integration

Open Source Projects (Upstream work): Real-Time LINUX, debian, Reproducible Builds, EDGEX FOUNDRY

Mapping CIP into the company



Up to 70% effort reduction achievable for OSS license clearing and vulnerability monitoring, kernel and package maintenance, application adaptation and testing for an individual product.

OSS Open Source Software **QA** quality ssurance **SDK** software development kit

Upstream First

CIP Kernel Team

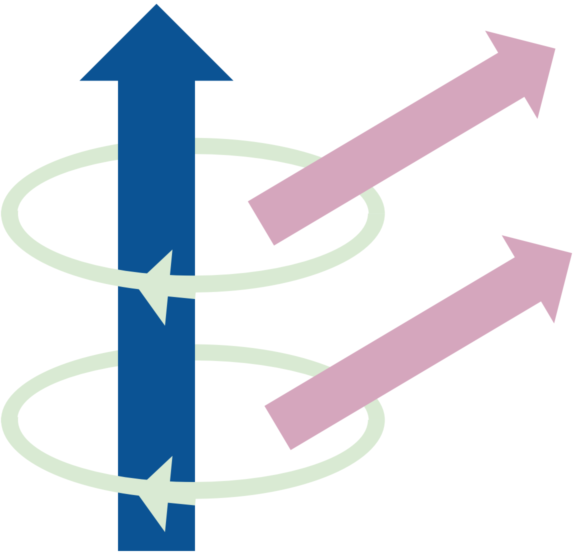
- **Primary Goal**
  - Provide CIP SLTS kernels with ten+ years maintenance period **by fixing versions** to fulfill the required level of reliability, sustainability, and security

- **Team Members**
  - Masashi Kudo – **Chairperson**
  - Nobuhiro Iwamatsu – **Kernel Maintainer**
  - Pavel Machek – **Kernel Maintainer**
  - Ben Hutchings – **Kernel Mentor**
  - SZ Lin – **Kernel Developer**
  - Chen-Yu Tsai – **Kernel Developer**

Development Models

“Own Community” Model

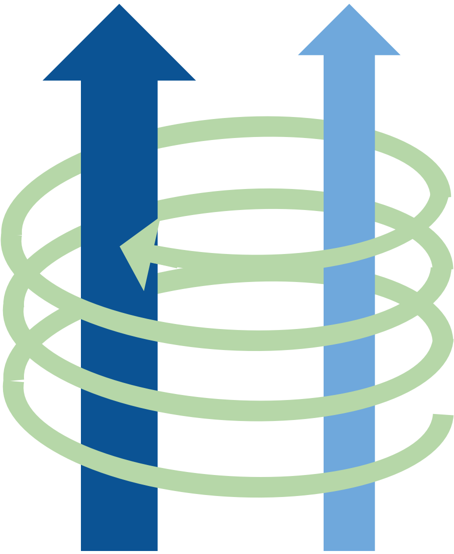
The project branches its base from upstream and evolves by its own.



Upstream **Project**

“Upstream First” Model

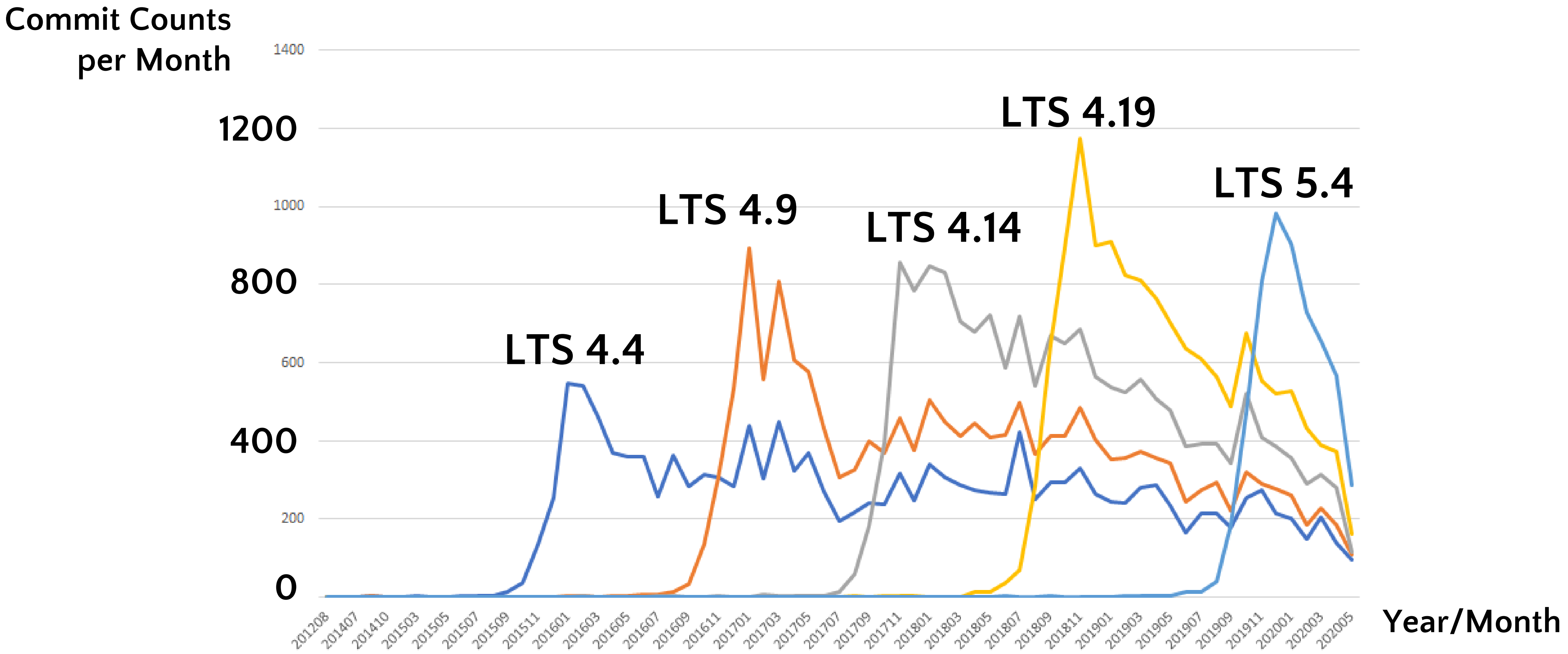
The project only allows patch commits if those patches are already in the upstream.



Upstream **Project**

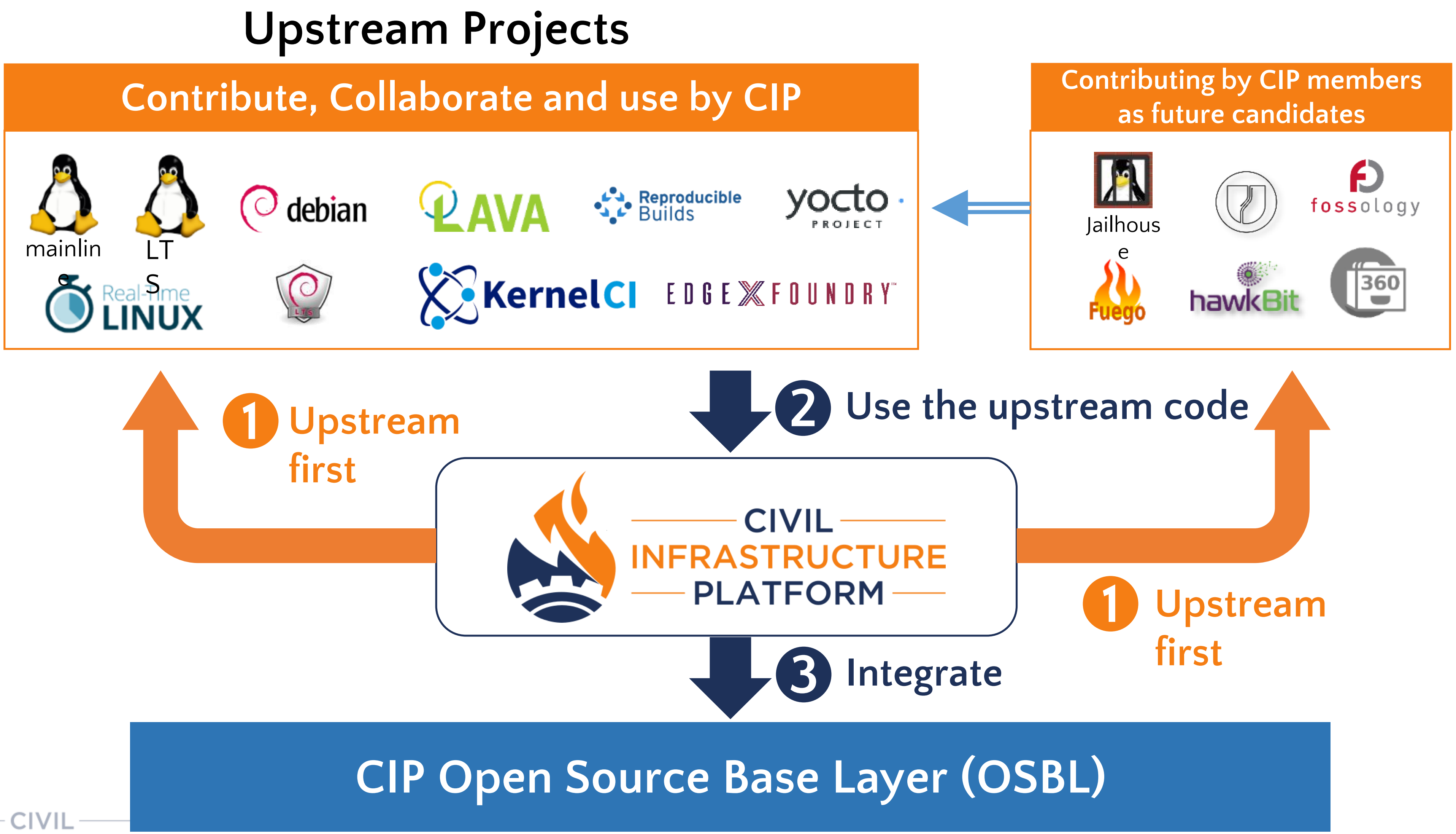
CIVIL INFRASTRUCTURE PLATFORM

Commit Counts per LTS

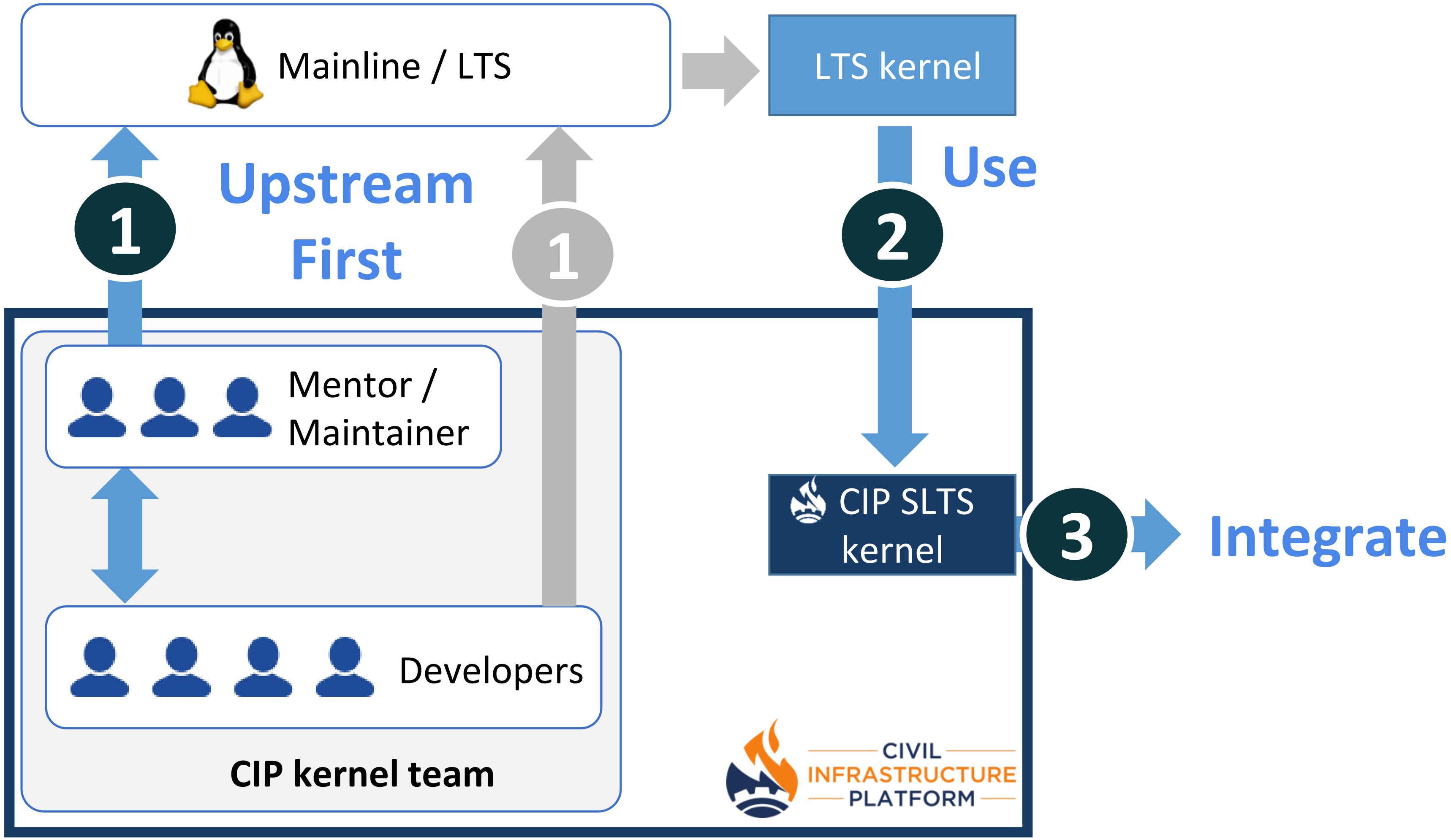


Note: If a patch has an original patch, the date of the patch is that of the original one.

Collaborative development with other OSS projects

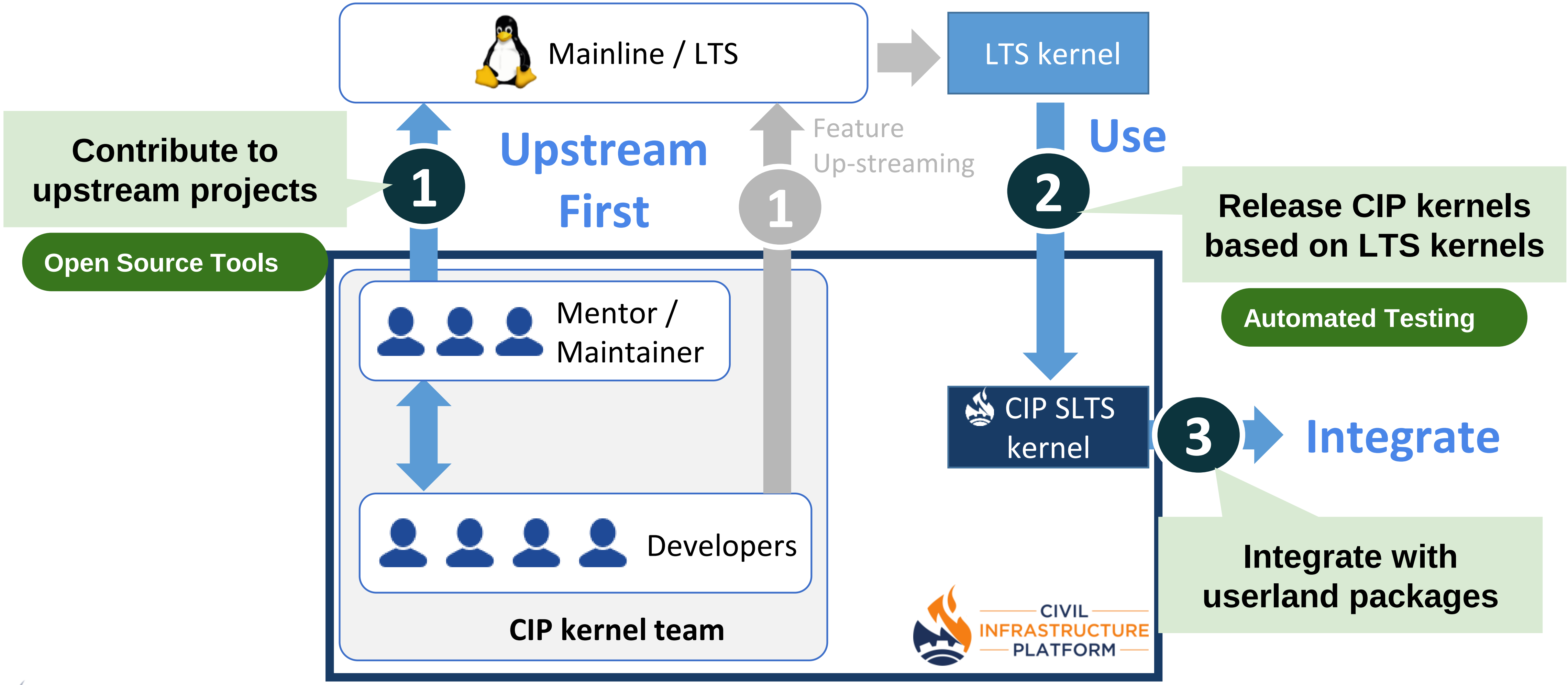


CIP SLTS kernel development

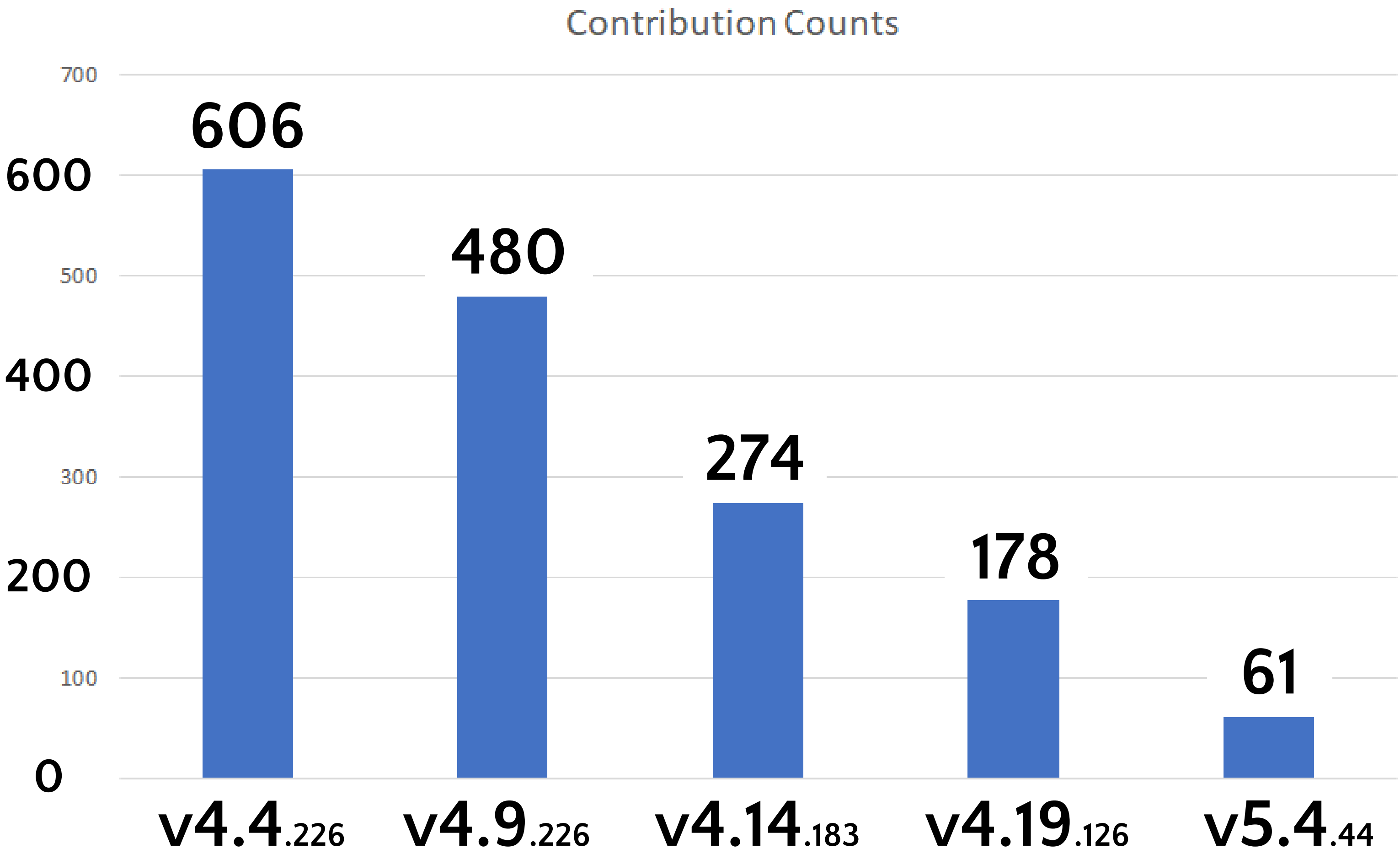


CIVIL INFRASTRUCTURE PLATFORM

CIP SLTS kernel development



1 Contributions – Statistics



as of June 7, 2020

1 Contributions – Details

as of June 7, 2020

	v4.4.226	v4.9.226	v4.14.183	v4.19.126	v5.4.44	TOTAL
Suggested-by:	1	1	1	2	1	**6**
Reported-by:	44	35	29	16	6	**130**
Signed-off-by:	429	319	137	74	24	**983**
Debugged-by:	1	1				**2**
Author:	78	81	53	33	14	**259**
Acked-by:	26	29	33	43	11	**142**
Reviewed-by:	2	4	9	6	3	**24**
Tested-by:	4	4	6	3		**17**
Cc:	103	96	70	49	23	**341**
TOTAL / LTS	**606**	**480**	**274**	**178**	**61**	**1599**

CIVIL INFRASTRUCTURE PLATFORM

Note: There could be multiple contributions by a same personnel in one commit. such duplicates are eliminated in total numbers. Therefore, the summation of each item may not equal to “Total”.

2 Use – Current SLTS Versions

Version	Maintainer	First Release	Projected EOL
4.19	Nobuhiro Iwamatsu & Pavel Machek	2019-01-11 • v4.19.13-cip1	2029-01
4.19-rt	Pavel Machek	2019-01-11 • v4.19.13-cip1-rt1	2029-01
4.4	Nobuhiro Iwamatsu & Pavel Machek	2017-01-17 • v4.4.42-cip1	2027-01
4.4-rt	Pavel Machek	2017-11-16 • v4.4.75-cip6-rt1	2027-01

2 Use – CIP Kernel Release Process

1. Review stable patches – status tracked in Gitlab [1]
    - Mark the review and the name of the worker under the commit.
    - **Start to review stable kernel patches in rc stage**
2. Review patch from CIP members via cip-dev [2]
    - Update the status of the commit in patchwork
3. Start testing
4. Tag release candidate
5. Ack by other maintainers
6. Release and send the news to cip-dev

```
# Stable Kernel Patches Review Status
Please list your name and review result below the patch item

  * UR:  Under Review
  * ACK: Acknowledge (if the patch is accepted)
  * TBB: To be backported (if other patches should be also backported)
  * NAK: Negative acknowledge (if the patch is rejected, please list the reason)
  * IGN: Patch was not reviewed as it is out of scope for CIP project


## v4.4.184
  - 72d1ee93e931 Linux 4.4.184
  - 46c7b5d6f2a5 tcp: refine memory limit test in tcp_fragment()
```

[1] https://gitlab.com/cip-project/cip-kernel/lts-commit-list

[2] https://patchwork.kernel.org/project/cip-dev/list/

CIVIL INFRASTRUCTURE PLATFORM

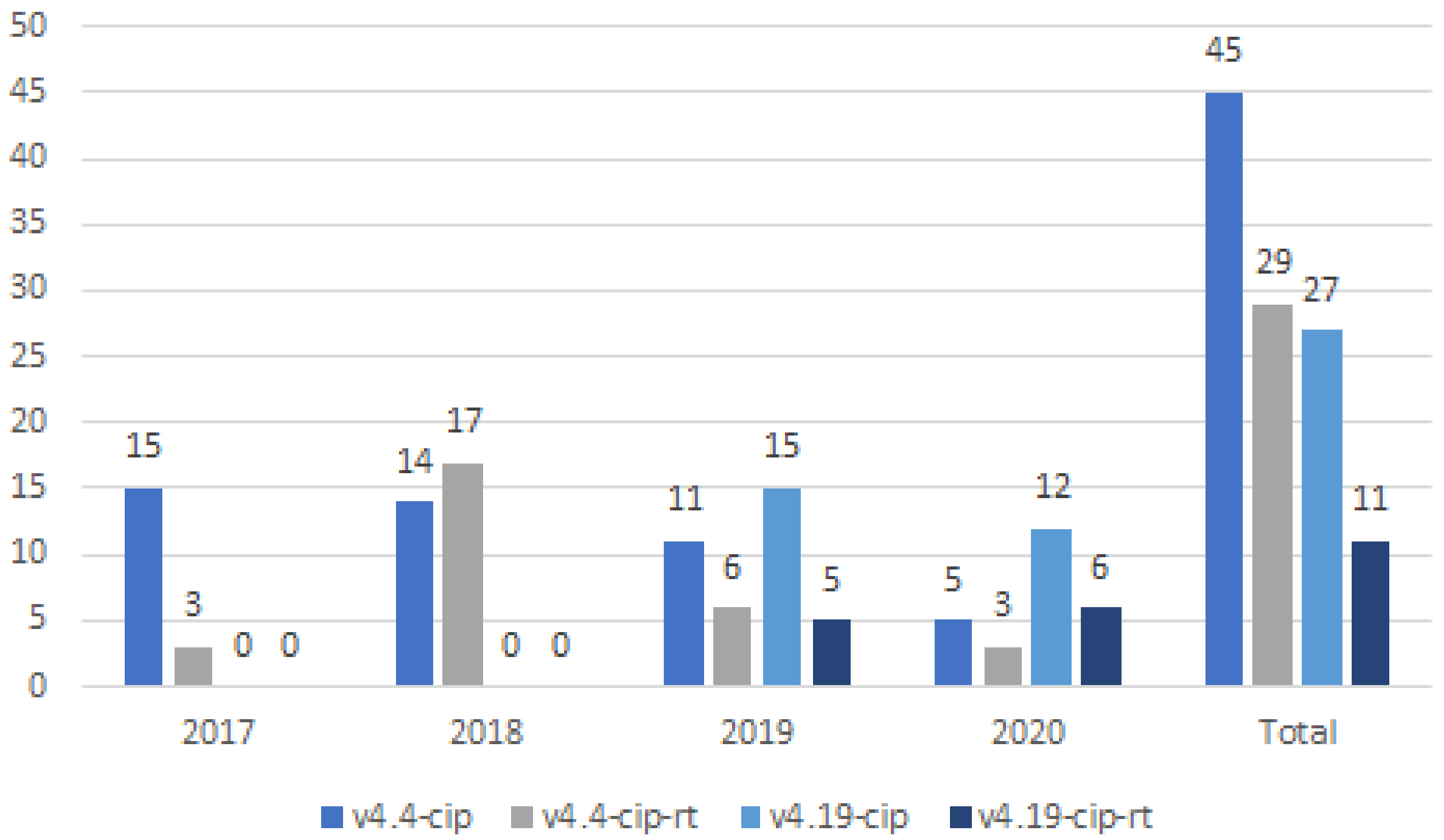
2 Use – CIP SLTS Kernel Release Policy

Release regularly	Release on demand
SLTS 4.19: twice a month **SLTS 4.4**: once a month	Depends on criticality of bug / security fixes
SLTS 4.19-rt: once a month **SLTS 4.4-rt**: once every two months	Ditto

Note: Difficult to estimate actual release date because of number of patches depends on each stable release

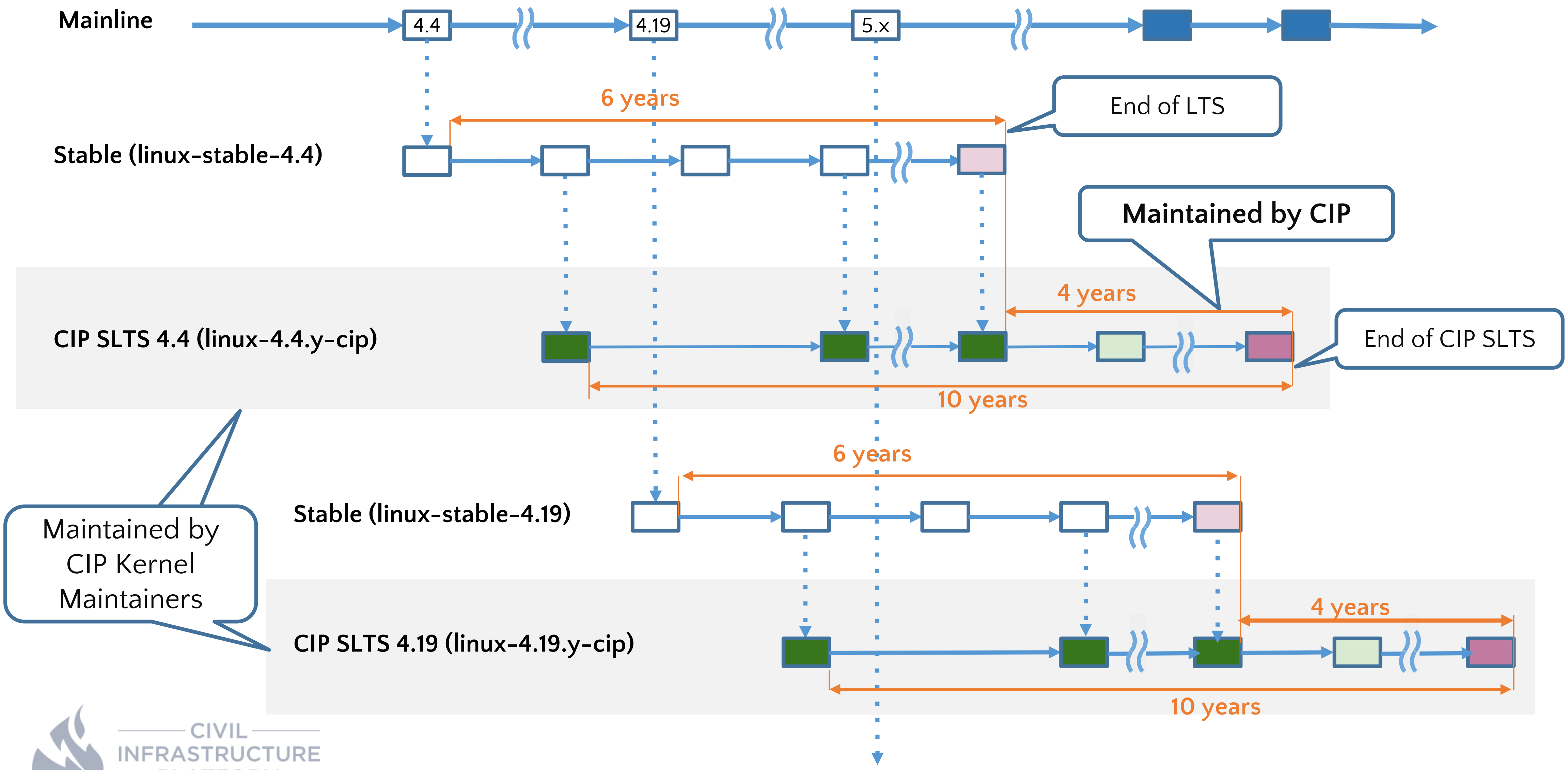
CIVIL INFRASTRUCTURE PLATFORM

Use – CIP SLTS Kernel Release Statistics



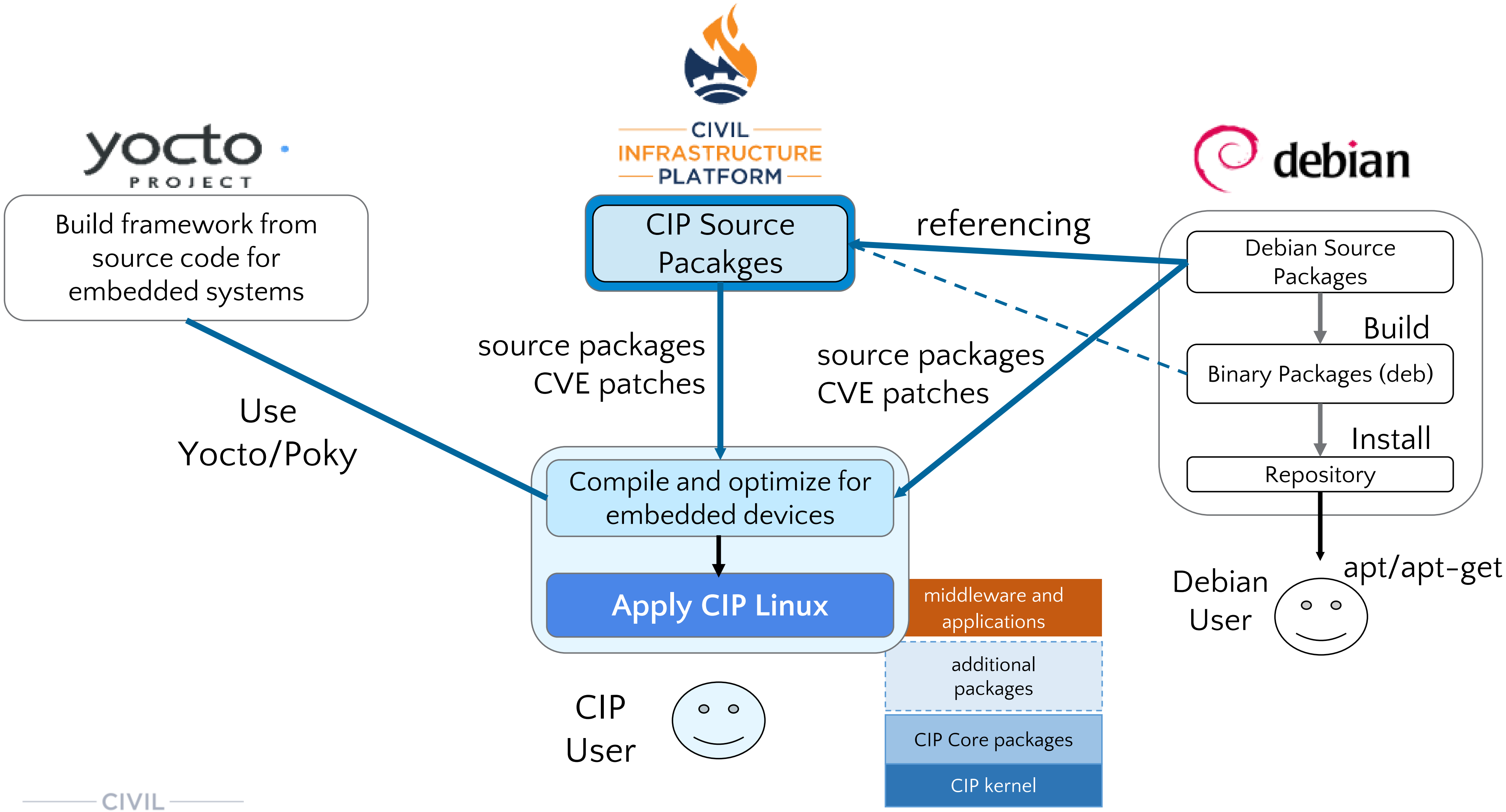
as of June 7, 2020

2 Use – CIP SLTS Kernel Releases



CIVIL INFRASTRUCTURE PLATFORM

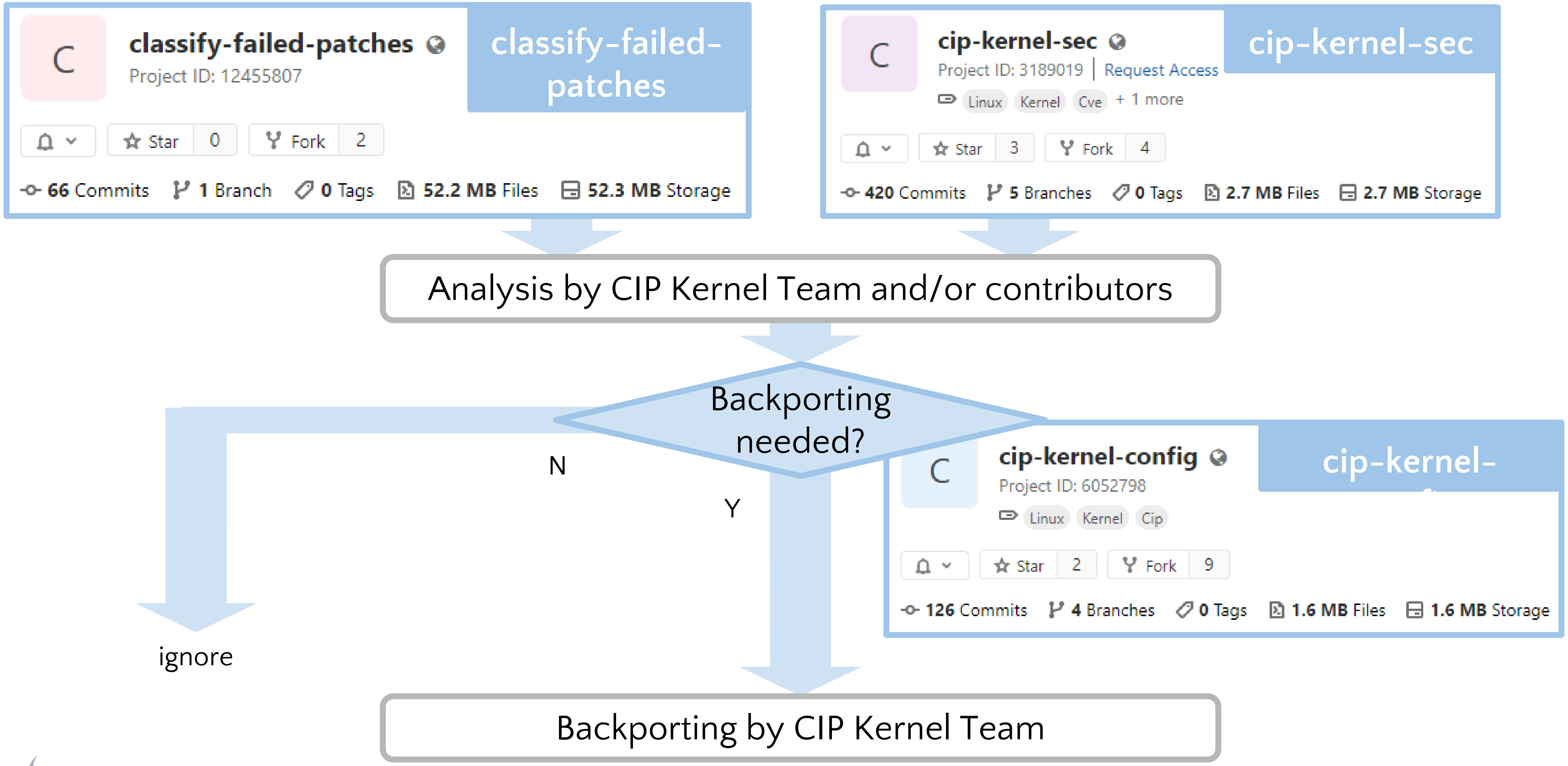
3 Integrate



CIVIL INFRASTRUCTURE PLATFORM

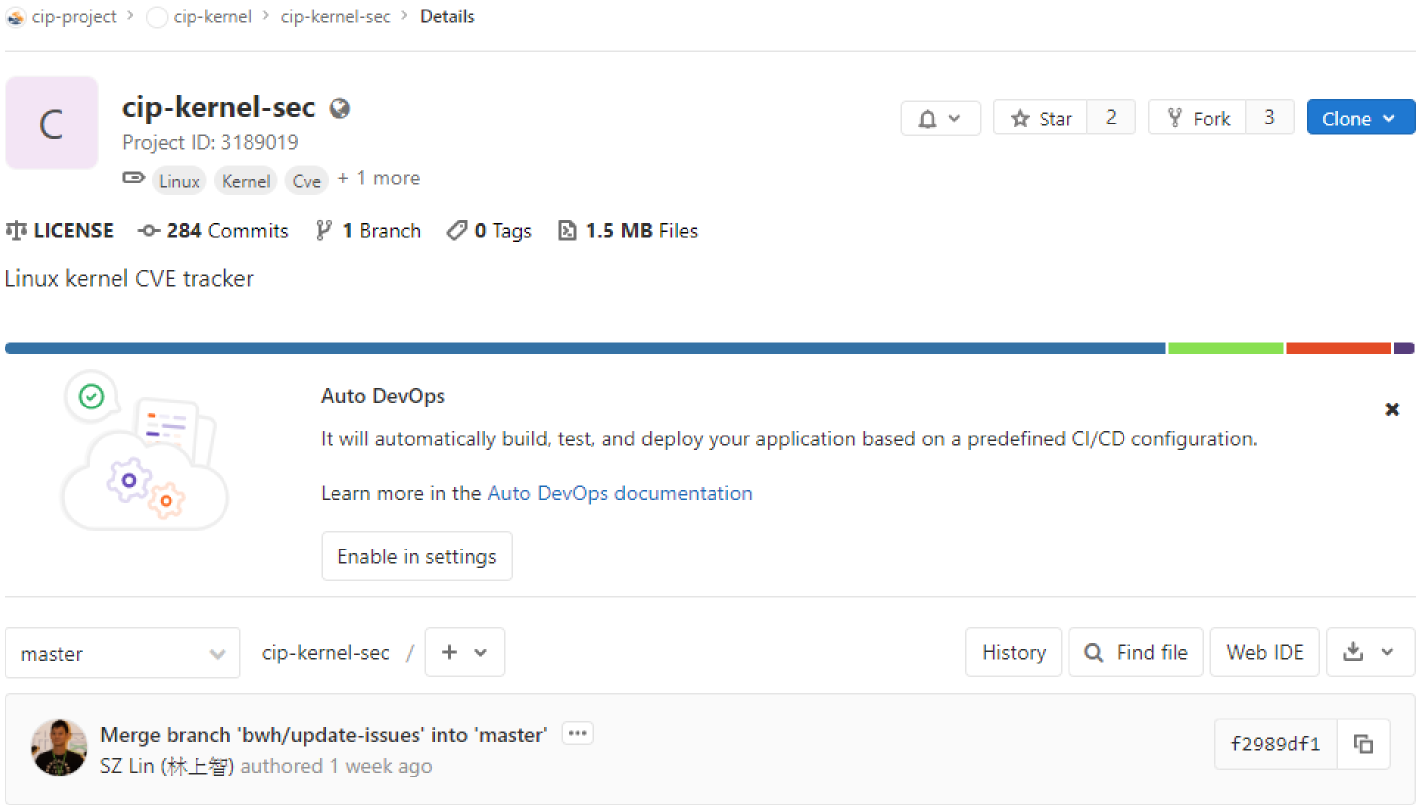
CIP
Open Source
Tools

Open Source Tools for Backporting Process



cip-kernel-sec

- Tracks the status of security issues, identified by CVE ID, in mainline, stable, and other configured branches.



cip-kernel-sec

Mainline/LTS

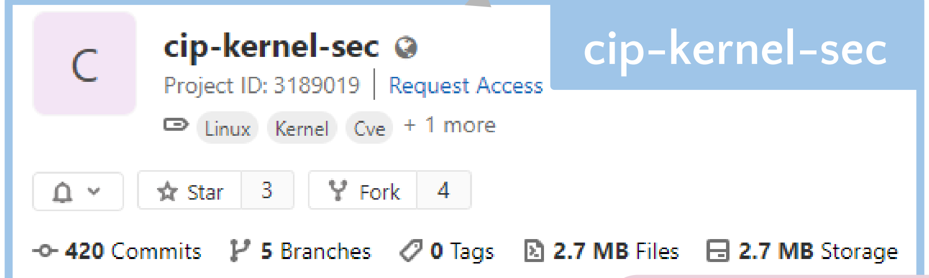
ubuntu

debian

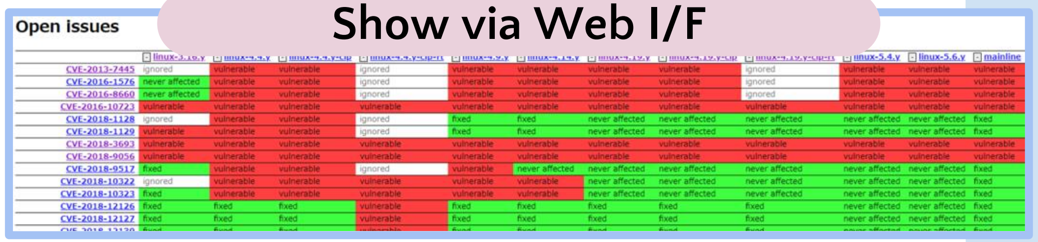
Gather CVE Information for Kernel

CIVIL INFRASTRUCTURE PLATFORM

cip-kernel-sec

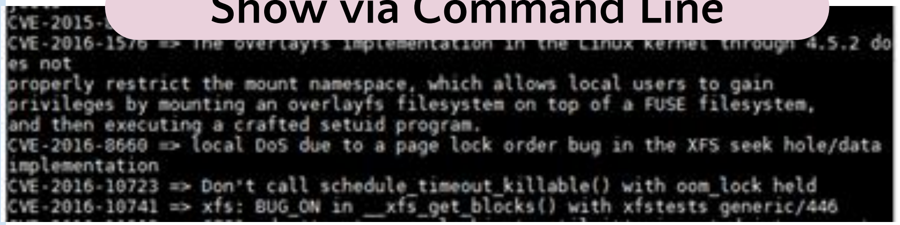


Show via Web I/F



Webview

Show via Command Line

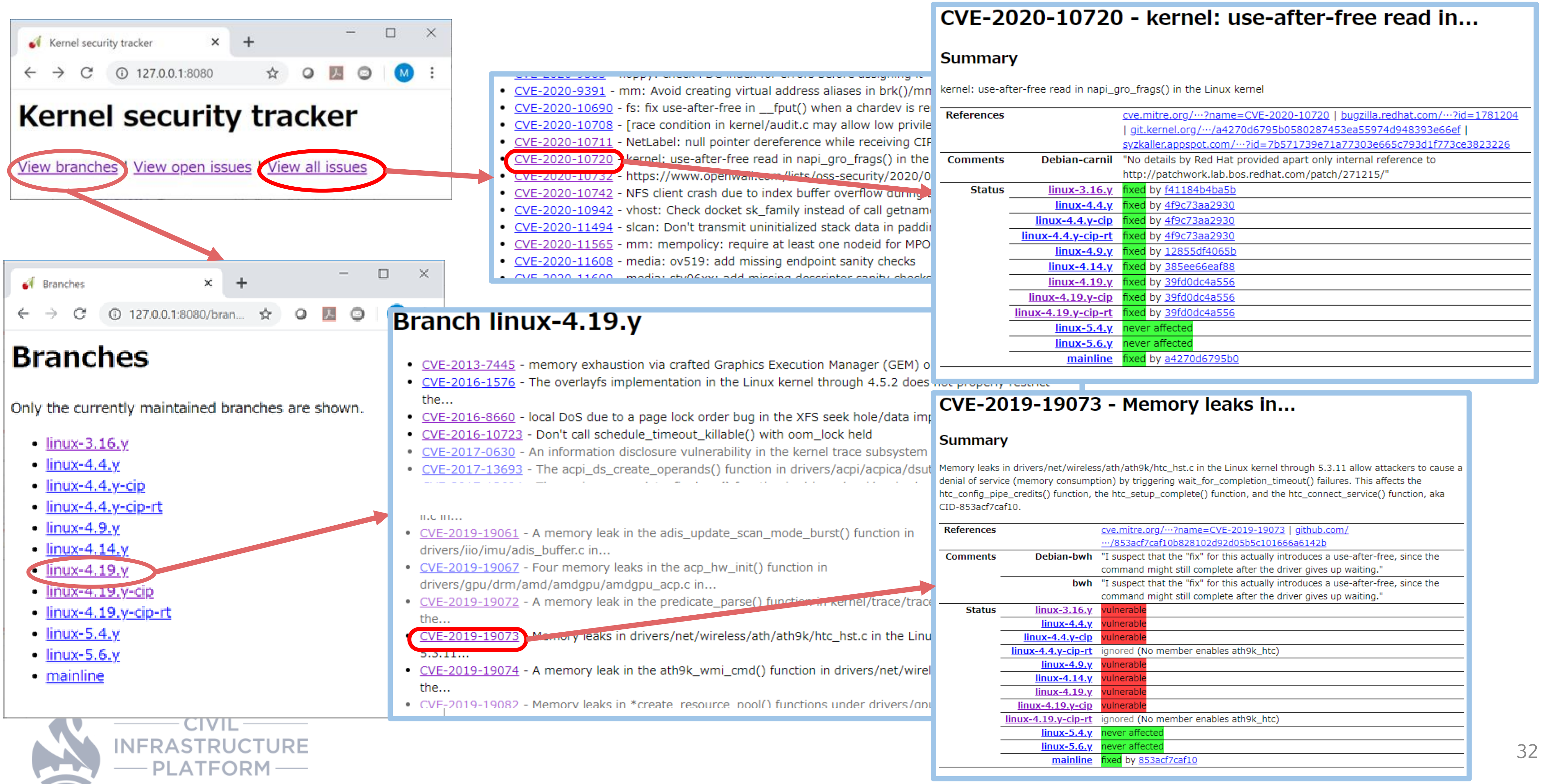


Command line view

Backporting by CIP Kernel Team

CIVIL INFRASTRUCTURE PLATFORM

cip-kernel-sec Webview



Kernel security tracker

127.0.0.1:8080

Kernel security tracker

View branches | View open issues | View all issues

Branches

127.0.0.1:8080/bran...

Branches

Only the currently maintained branches are shown.

- linux-3.16.y
- linux-4.4.y
- linux-4.4.y-cip
- linux-4.4.y-cip-rt
- linux-4.9.y
- linux-4.14.y
- linux-4.19.y
- linux-4.19.y-cip
- linux-4.19.y-cip-rt
- linux-5.4.y
- linux-5.6.y
- mainline

- CVE-2020-9391 - mm: Avoid creating virtual address aliases in brk()/mm
- CVE-2020-10690 - fs: fix use-after-free in __fput() when a chardev is re
- CVE-2020-10708 - [race condition in kernel/audit.c may allow low privile
- CVE-2020-10711 - NetLabel: null pointer dereference while receiving CIP
- CVE-2020-10720 - kernel: use-after-free read in napi_gro_frags() in the
- CVE-2020-10732 - https://www.openwall.com/lists/oss-security/2020/0
- CVE-2020-10742 - NFS client crash due to index buffer overflow during
- CVE-2020-10942 - vhost: Check docket sk_family instead of call getnam
- CVE-2020-11494 - slcan: Don't transmit uninitialized stack data in paddi
- CVE-2020-11565 - mm: mempolicy: require at least one nodeid for MPO
- CVE-2020-11608 - media: ov519: add missing endpoint sanity checks

Branch linux-4.19.y

- CVE-2013-7445 - memory exhaustion via crafted Graphics Execution Manager (GEM) o
- CVE-2016-1576 - The overlayfs implementation in the Linux kernel through 4.5.2 does not properly restrict the...
- CVE-2016-8660 - local DoS due to a page lock order bug in the XFS seek hole/data imp
- CVE-2016-10723 - Don't call schedule_timeout_killable() with oom_lock held
- CVE-2017-0630 - An information disclosure vulnerability in the kernel trace subsystem
- CVE-2017-13693 - The acpi_ds_create_operands() function in drivers/acpi/acpica/dsut
- CVE-2019-19061 - A memory leak in the adis_update_scan_mode_burst() function in drivers/iio/imu/adis_buffer.c in...
- CVE-2019-19067 - Four memory leaks in the acp_hw_init() function in drivers/gpu/drm/amd/amdgpu/amdgpu_acp.c in...
- CVE-2019-19072 - A memory leak in the predicate_parse() function in kernel/trace/trace the...
- CVE-2019-19073 - Memory leaks in drivers/net/wireless/ath/ath9k/htc_hst.c in the Linu 5.3.11...
- CVE-2019-19074 - A memory leak in the ath9k_wmi_cmd() function in drivers/net/wirel the...

CVE-2020-10720 - kernel: use-after-free read in...

Summary

kernel: use-after-free read in napi_gro_frags() in the Linux kernel

References		cve.mitre.org/…?name=CVE-2020-10720 | bugzilla.redhat.com/…?id=1781204 | git.kernel.org/…/a4270d6795b0580287453ea55974d948393e66ef | syzkaller.appspot.com/…?id=7b571739e71a77303e665c793d1f773ce3823226
Comments	Debian-carnil	"No details by Red Hat provided apart only internal reference to http://patchwork.lab.bos.redhat.com/patch/271215/"
Status	linux-3.16.y	fixed by f41184b4ba5b
	linux-4.4.y	fixed by 4f9c73aa2930
	linux-4.4.y-cip	fixed by 4f9c73aa2930
	linux-4.4.y-cip-rt	fixed by 4f9c73aa2930
	linux-4.9.y	fixed by 12855df4065b
	linux-4.14.y	fixed by 385ee66eaf88
	linux-4.19.y	fixed by 39fd0dc4a556
	linux-4.19.y-cip	fixed by 39fd0dc4a556
	linux-4.19.y-cip-rt	fixed by 39fd0dc4a556
	linux-5.4.y	never affected
	linux-5.6.y	never affected
	mainline	fixed by a4270d6795b0

CVE-2019-19073 - Memory leaks in...

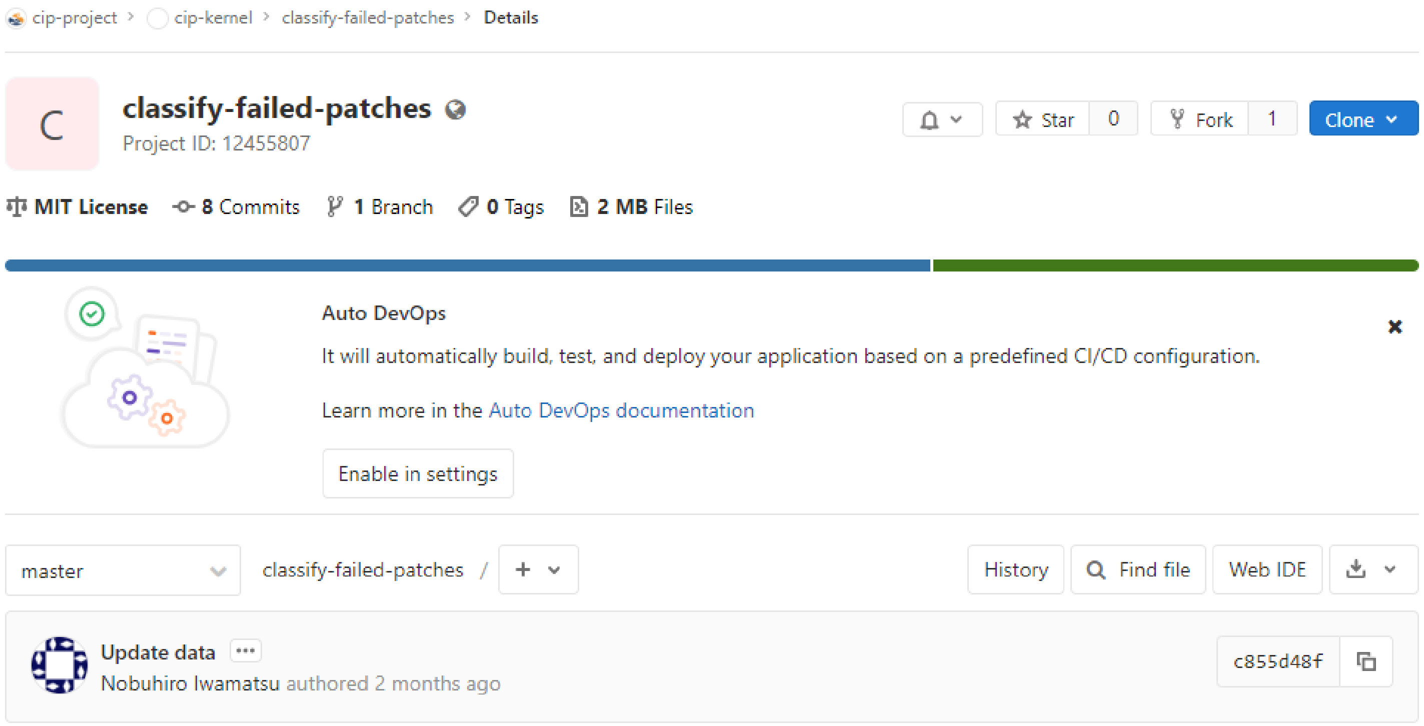
Summary

Memory leaks in drivers/net/wireless/ath/ath9k/htc_hst.c in the Linux kernel through 5.3.11 allow attackers to cause a denial of service (memory consumption) by triggering wait_for_completion_timeout() failures. This affects the htc_config_pipe_credits() function, the htc_setup_complete() function, and the htc_connect_service() function, aka CID-853acf7caf10.

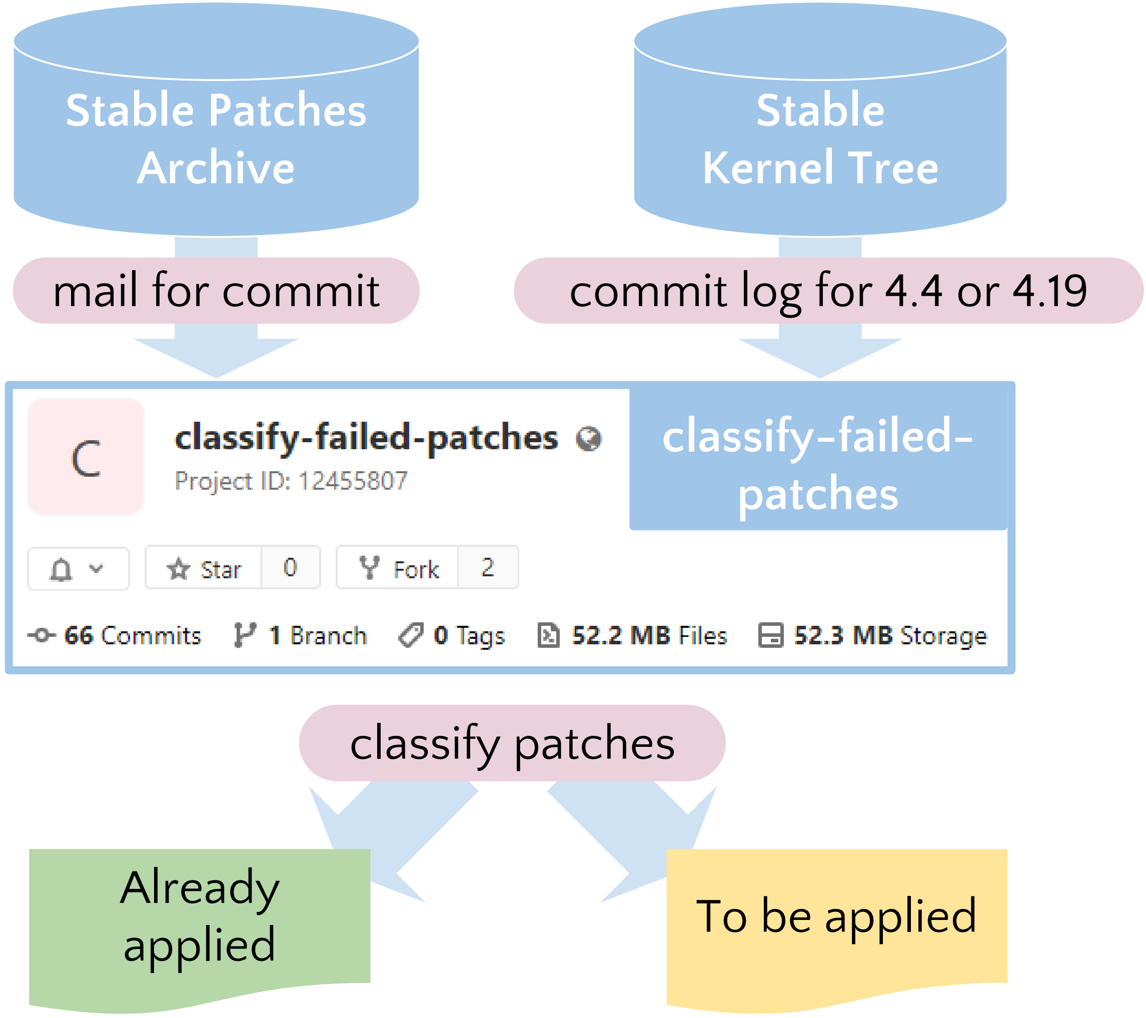
References		cve.mitre.org/…?name=CVE-2019-19073 | github.com/…/853acf7caf10b828102d92d05b5c101666a6142b
Comments	Debian-bwh	"I suspect that the "fix" for this actually introduces a use-after-free, since the command might still complete after the driver gives up waiting."
	bwh	"I suspect that the "fix" for this actually introduces a use-after-free, since the command might still complete after the driver gives up waiting."
Status	linux-3.16.y	vulnerable
	linux-4.4.y	vulnerable
	linux-4.4.y-cip	vulnerable
	linux-4.4.y-cip-rt	ignored (No member enables ath9k_htc)
	linux-4.9.y	vulnerable
	linux-4.14.y	vulnerable
	linux-4.19.y	vulnerable
	linux-4.19.y-cip	vulnerable
	linux-4.19.y-cip-rt	ignored (No member enables ath9k_htc)
	linux-5.4.y	never affected
	linux-5.6.y	never affected
	mainline	fixed by 853acf7caf10

classify-failed-patches

- This project tracks the status of failed patches, and classifies patches into “applied” and “ToApply” types.



classify-failed-patches



classify-failed-patches

Applied patches

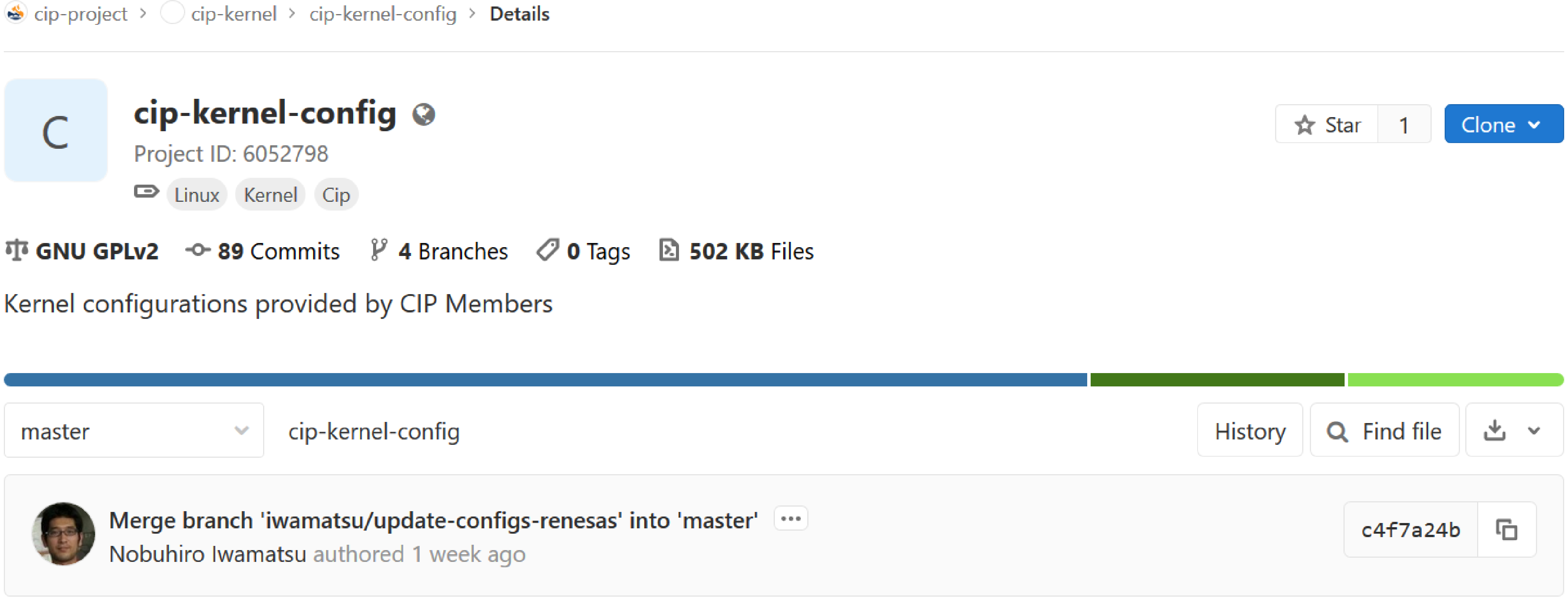
```
[APPLIED] arm64: Disable unhandled signal log messages by default
        56b57bd20f5bcdd353eacf7b7c41ee18ffe0c963 arm64: Disable unhandled signal log messages by default
[APPLIED] ARC: hide unused function unw_hdr_alloc
        4d28512bfca84456b6f65e6800c003cf0810480b ARC: hide unused function unw_hdr_alloc
[APPLIED] btrfs: Ensure replaced device doesn't have pending chunk allocation
        986543fcf50c8a3681be44cac42dc498fe25ab34 btrfs: Ensure replaced device doesn't have pending chunk allocation
[APPLIED] btrfs: Ensure replaced device doesn't have pending chunk allocation
        986543fcf50c8a3681be44cac42dc498fe25ab34 btrfs: Ensure replaced device doesn't have pending chunk allocation
[APPLIED] btrfs: Ensure replaced device doesn't have pending chunk allocation
        986543fcf50c8a3681be44cac42dc498fe25ab34 btrfs: Ensure replaced device doesn't have pending chunk allocation
[APPLIED] btrfs: Ensure replaced device doesn't have pending chunk allocation
        986543fcf50c8a3681be44cac42dc498fe25ab34 btrfs: Ensure replaced device doesn't have pending chunk allocation
```

To be Applied Patches

```
[TOAPPLY] inet: update the IP ID generation algorithm to patch 355b98553789b646ed97ad801a619ff898471b92 standards.
[TOAPPLY] scsi: ufs: Fix RX_TERMINATION_FORCE_ENABLE define value
[TOAPPLY] inet: update the IP ID generation algorithm to patch 355b98553789b646ed97ad801a619ff898471b92 standards.
[TOAPPLY] IB/hfi1: Failed to drain send queue when QP is put into error state
[TOAPPLY] arm64: mm: Ensure tail of unaligned initrd is reserved
[TOAPPLY] fs/proc/task_mmu.c: fix uninitalized variable warning
[TOAPPLY] tpm: Fix the type of the return value in calc_tpm2_event_size()
[TOAPPLY] block: bio_map_user_iov should not be limited to BIO_MAX_PAGES
[TOAPPLY] clk: ingenic/jz4725b: Fix parent of pixel clock
[TOAPPLY] i2c-piix4: Add Hygon Dhyana SMBus support
[TOAPPLY] TTY: serial_core, add ->install
```

cip-kernel-config

- Necessity of backporting is determined to be fixed base on kernel configurations provided by CIP members



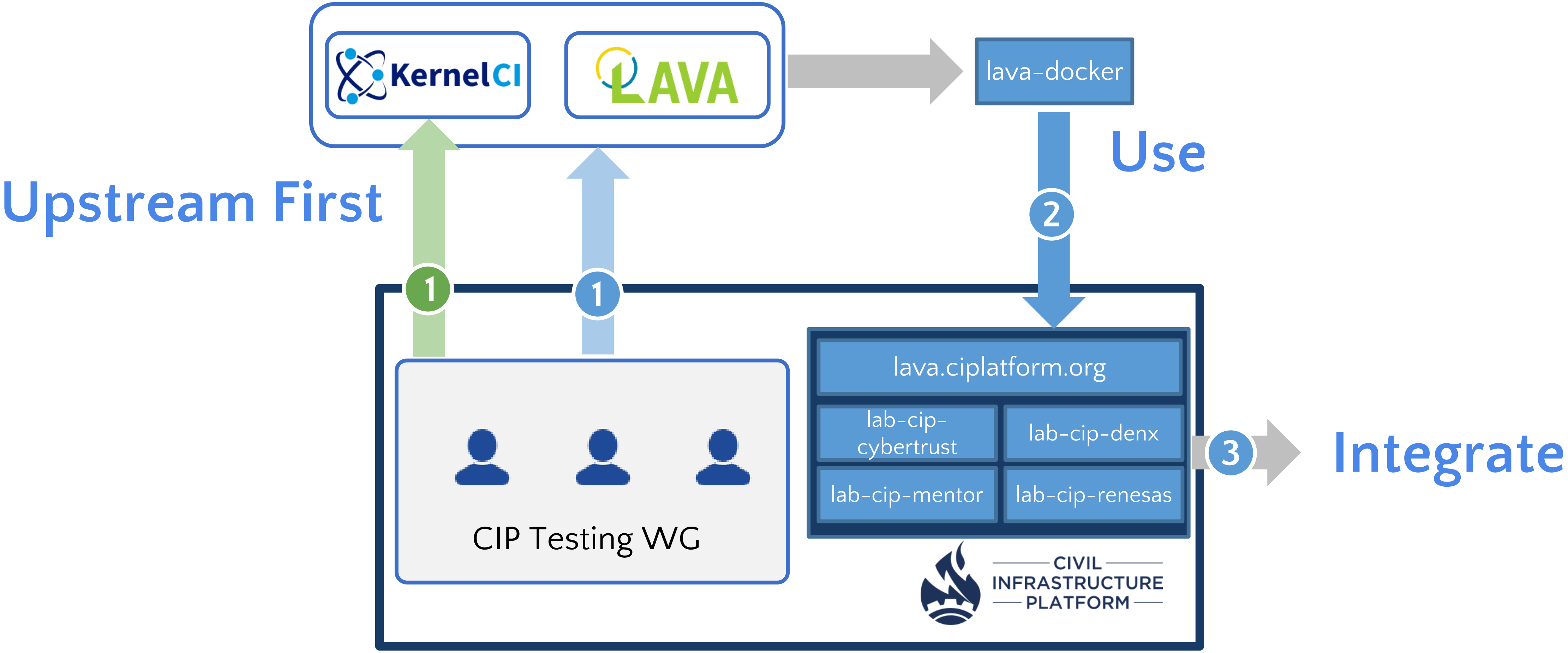
CIVIL INFRASTRUCTURE PLATFORM

CIP Automated Testing

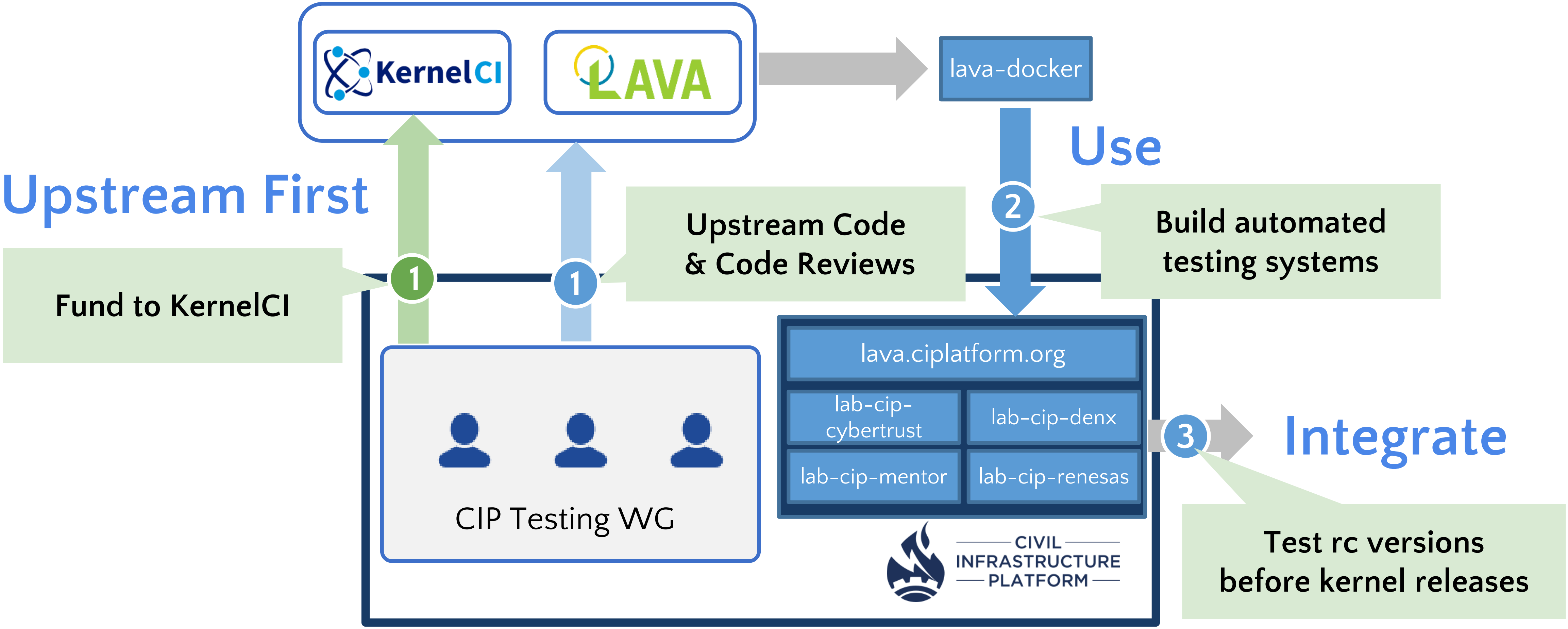
CIP Testing Goals

- **Centralized control / distributed testing**
  - CIP developers who are distributed over the world should be able to test CIP reference platforms which are hosted at 4 labs located in Europe, India and Japan.
- **Automated testing with Continuous Integration (CI)**
  - Sustain periodical and long-term kernel releases cost-effectively
- **Support all CIP reference platforms**
  - There are currently 7 different reference platforms

CIP Testing Team

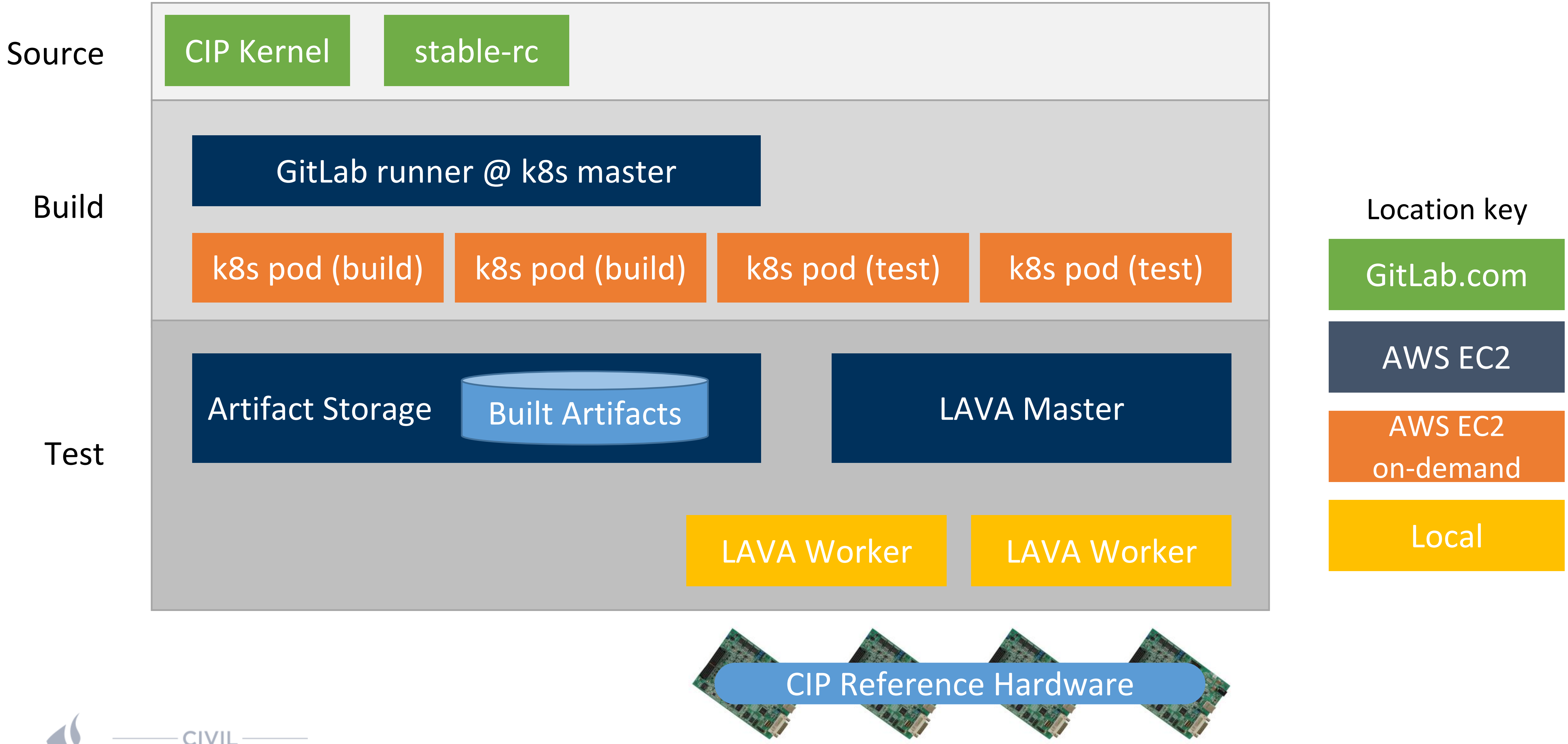


CIP Testing Team



CIVIL INFRASTRUCTURE PLATFORM

Testing Architecture Overview



CIP Reference Boards

CIP Reference Boards		Supported Kernels			
Platform	**Architecture**	**SLTS v4.4**	**SLTS v4.4-rt**	**SLTS v4.19**	**SLTS v4.19-rt**
AM335x Beaglebone Black	Armv7	Y	Y[1]	Y	Y[1]
Cyclone V DE0-Nano-SoC Development Kit	Armv7	N	N	Y	Y[1]
QEMU	x86_64	Y	Y[1]	Y	Y[1]
RZ/G1M iWave Qseven Development Kit	Armv7	Y	Y[1,2]	Y	Y[1,2]
RZ/G2M HopeRun HiHope	Armv8	N	N	Y	Y[1,2]
SIMATIC IPC227E	x86-64	N	N	Y	Y[1]
OpenBlocks IoT VX2	x86-64	N	N	Y	Y[1]

CIP Reference Board Candidate		Supported Kernels			
Platform	**Architecture**	**SLTS v4.4**	**SLTS v4.4-rt**	**SLTS v4.19**	**SLTS v4.19-rt**
Zynq UltraScale+ MPSoC ZCU102 Evaluation Kit	Armv8	N	N	Y	Y[1]

[1] Tested with standard Kernel configuration (non-RT)
[2] Tested with Real-Time enabled Kernel configuration

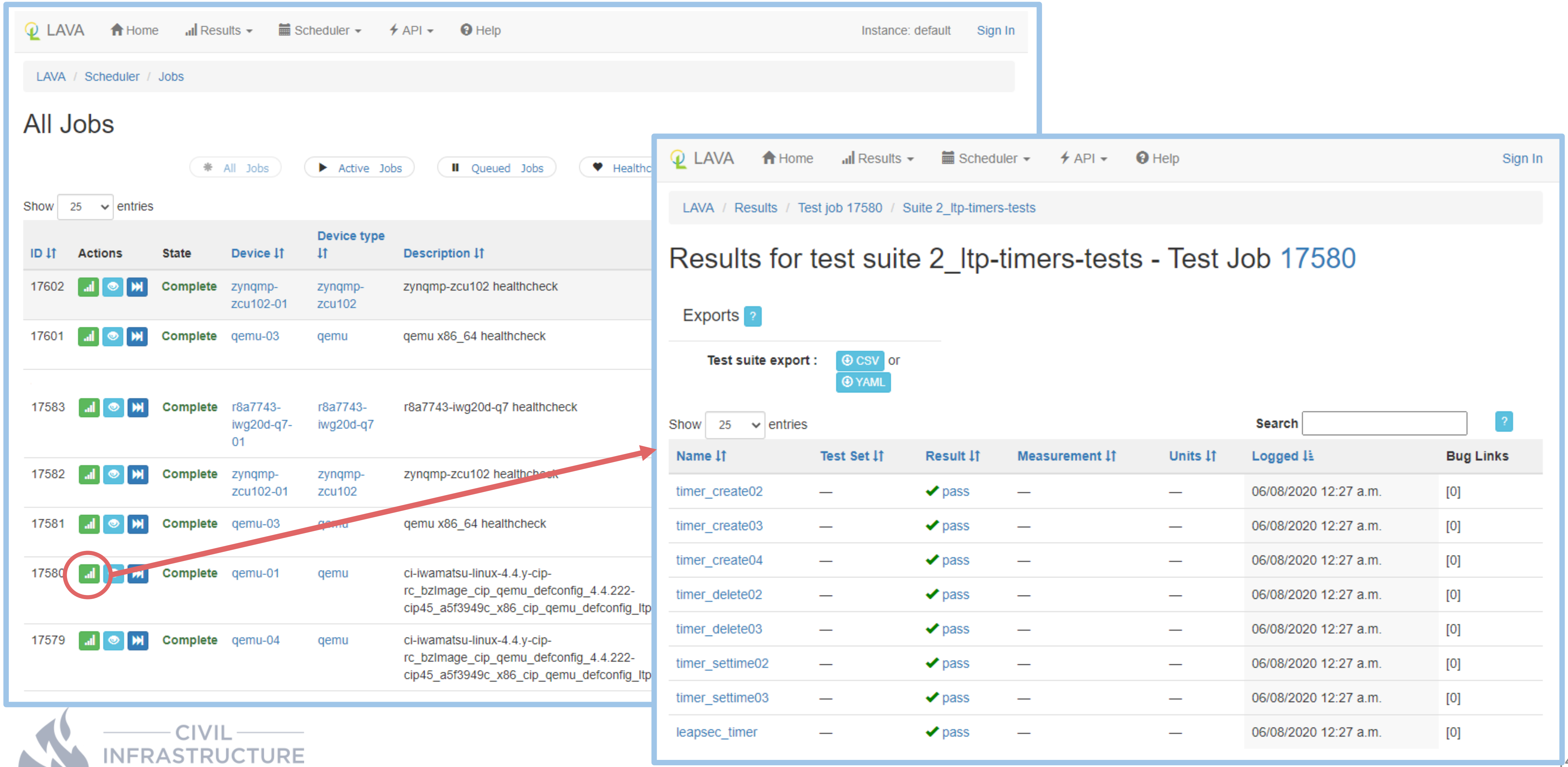
CIVIL INFRASTRUCTURE PLATFORM

Automated Testing

- Currently CIP is running the following tests:
  - Boot test
    - uname -a
  - Spectre/Meltdown checker
  - LTP
    - ltp-cve-tests, ltp-dio-tests, ltp-fs-tests, ltp-ipc-tests, ltp-math-tests, ltp-open-posix-tests, ltp-sched-tests, ltp-syscalls-tests and ltp-timers-tests
  - Cyclictest+Hackbench
    - This test measures event latency in the Linux Kernel, with hackbench running in the background to stress the system.

- In Development:
  - Kselftest

LAVA Results



Collaboration with KernelCI

- kernelci.org is now a Linux Foundation project, sponsored by Baylibre, **CIP**, Collabra, Foundries.io, Google, Microsoft and Redhat: https://foundation.kernelci.org/
- CIP are collaborating with KernelCI to improve the range of tests supported by KernelCI, starting with LTP
- Further collaboration is being discussed between CIP and KernelCI

CIVIL INFRASTRUCTURE PLATFORM

KernelCI

Summary

Summary

- CIP Kernel Team follows "Upstream First" principle, and contributes to upstream.
- CIP open source tools are developed to facilitate the contribution activities.
- By taking advantage of kernel LTS, the team steadily releases CIP SLTS kernels, and aims to maintain them for 10 years or more.
- To reduce CIP SLTS kernel release cost, the team is closely working with CIP testing team to build automated testing systems.

Please join us to sustain Civil Infrastructure together !

Weekly Regular Online Meeting

- CIP IRC weekly meeting – Every Thursday UTC (GMT) 09:00

US-West	US-East	UK	DE	TW	JP
02:00	05:00	10:00	11:00	17:00	18:00

- Channel:
  * irc:chat.freenode.net:6667/cip

- The meeting is used to share status among CIP developers (Kernel Team, Test Team, SW Update WG, Security WG)

CIVIL INFRASTRUCTURE PLATFORM

CIP Kernel Workgroup Repository

- **CIP Linux kernel & real-time kernel**
  - https://git.kernel.org/pub/scm/linux/kernel/git/cip/linux-cip.git
- **CIP Linux kernel CVE tracker**
  - https://gitlab.com/cip-project/cip-kernel/cip-kernel-sec
- **CIP Linux kernel failed patches tracker**
  - https://gitlab.com/cip-project/cip-kernel/classify-failed-patches

Contact Information and Resources

To get the latest information, please contact:

- CIP Mailing List: cip-dev@lists.cip-project.org

Other resources

- Twitter: @cip_project
- CIP Web Site: https://www.cip-project.org
- CIP News: https://www.cip-project.org/news/in-the-news
- CIP Wiki: https://wiki.linuxfoundation.org/civilinfrastructureplatform/
- CIP Source Code
  - CIP repositories hosted at kernel.org: https://git.kernel.org/pub/scm/linux/kernel/git/cip/
  - CIP GitLab: https://gitlab.com/cip-project

Thank You