



———— CIVIL ————  
**INFRASTRUCTURE**  
—— PLATFORM ——

# Two Years Experience of Industrial-grade Open Source Base Layer Development and its Future

**Yoshitake Kobayashi**, Toshiba Corp., CIP TSC Chair

**Urs Gleim**, Siemens AG, CIP Board Chair

Open Source Summit Europe, Edinburgh, October 22, 2018

# What is CIP?



# Our Civilization runs on Linux



This morning around the corner...



## Transport



**Rail automation**



**Vehicle control**



**Automatic ticket gates**

## Energy

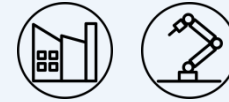


**Power Generation**



**Turbine Control**

## Industry



**Industry automation**



**CNC control**



**Industrial communication**

## Others



**Healthcare**



**Building automation**



**Broadcasting**

There are issues to be solved...





# A Power Plant System:

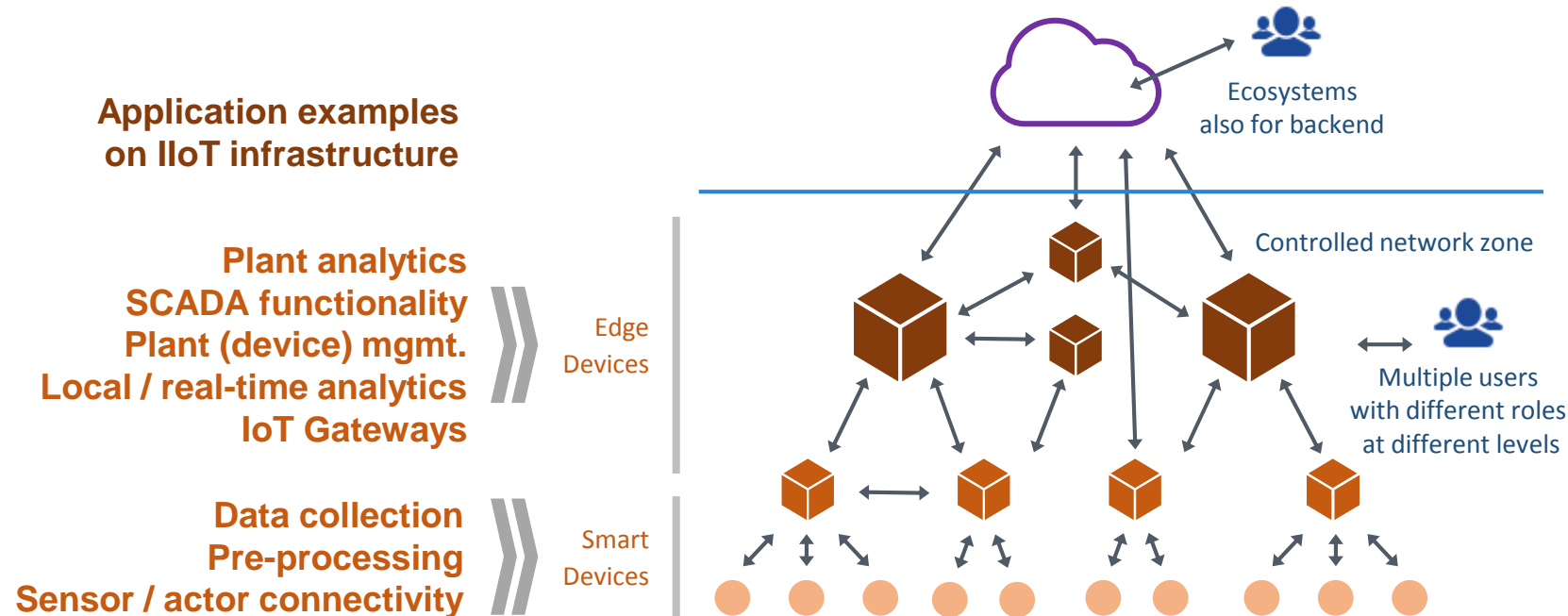
**25-60 years products life-cycle**

Very reluctant to perform product updates and upgrades of hardware and base software platform

# Industrial IoT: Edge and Fog Computing

Functionality is moving from the cloud to the “Edge”

- Increasing number of networked industrial-grade devices
- Security management requires harmonized software landscape



**IIoT:** Industrial IoT    **SCADA:** Supervisory Control And Data Acquisition

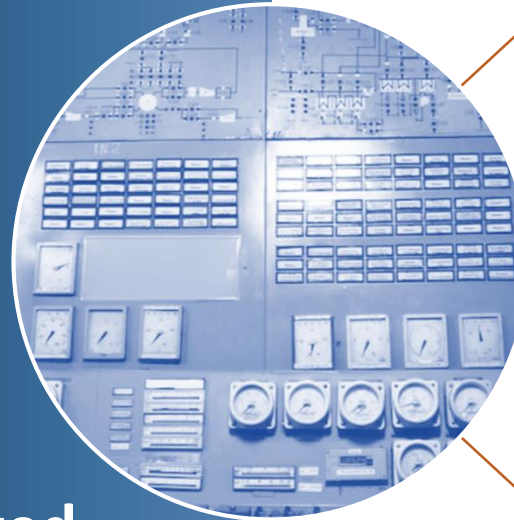
# The key challenges



**Apply IoT concepts to industrial systems.**

**Ensure quality and longevity of products.**

**Keep millions of connected systems secure.**



## Industrial gradeness

- Reliability
- Functional Safety
- Real-time capabilities

## Sustainability

- Product life-cycles of decades
- Backwards compatibility
- Standards

## Security

- Security & vulnerability management
- Firmware updates
- Minimize risk of regressions

# Strategic Partnerships – Motivation & Context



**We maintain different flavors and version of Linux**

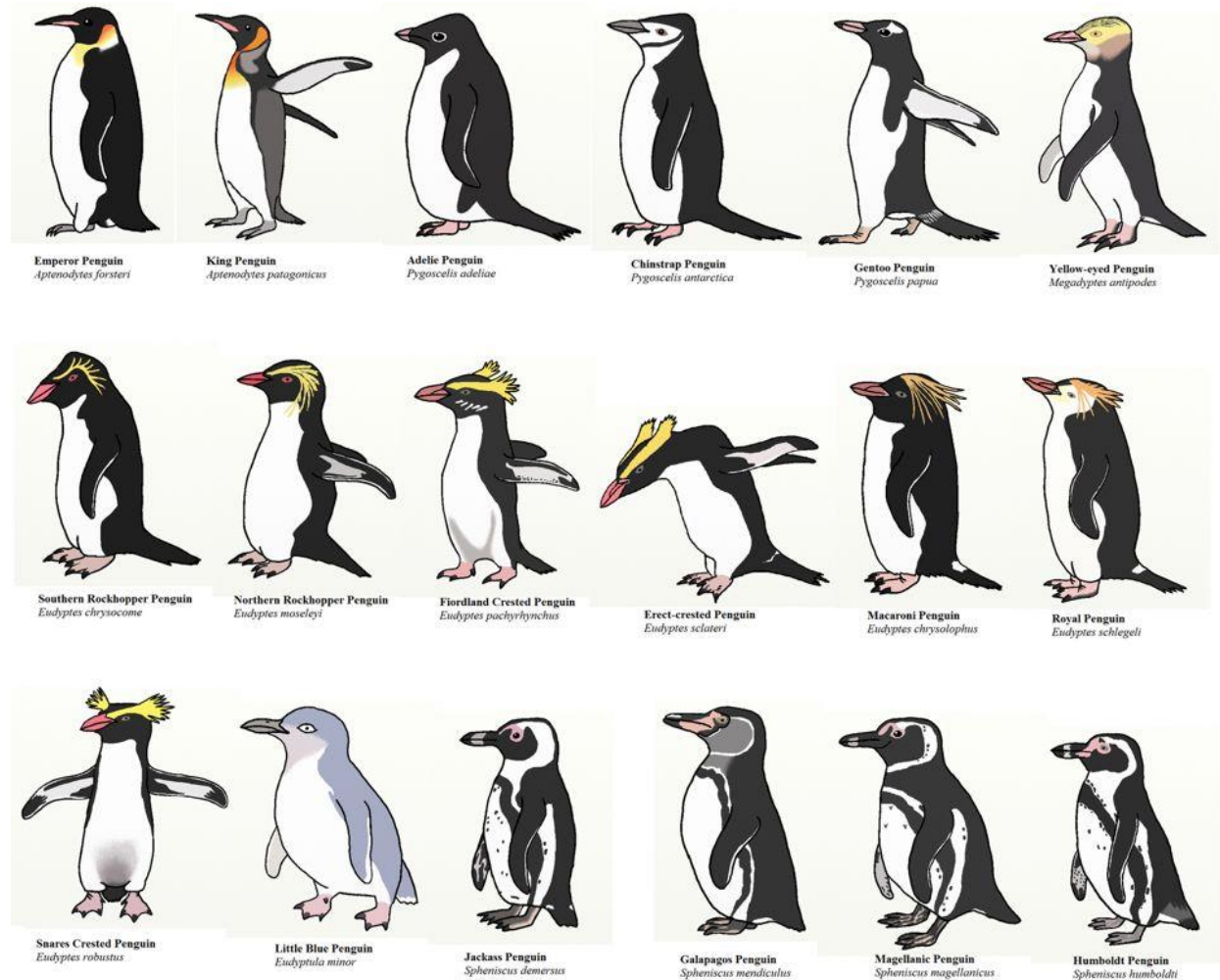
**...in each division...**

**...for several products...**

**...for many years.**

**...without having business advantages from doing this.**

**And other companies do the same.**



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

# 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

<https://www.cip-project.org/>



— CIVIL —  
INFRASTRUCTURE  
— PLATFORM —

since April 2016

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



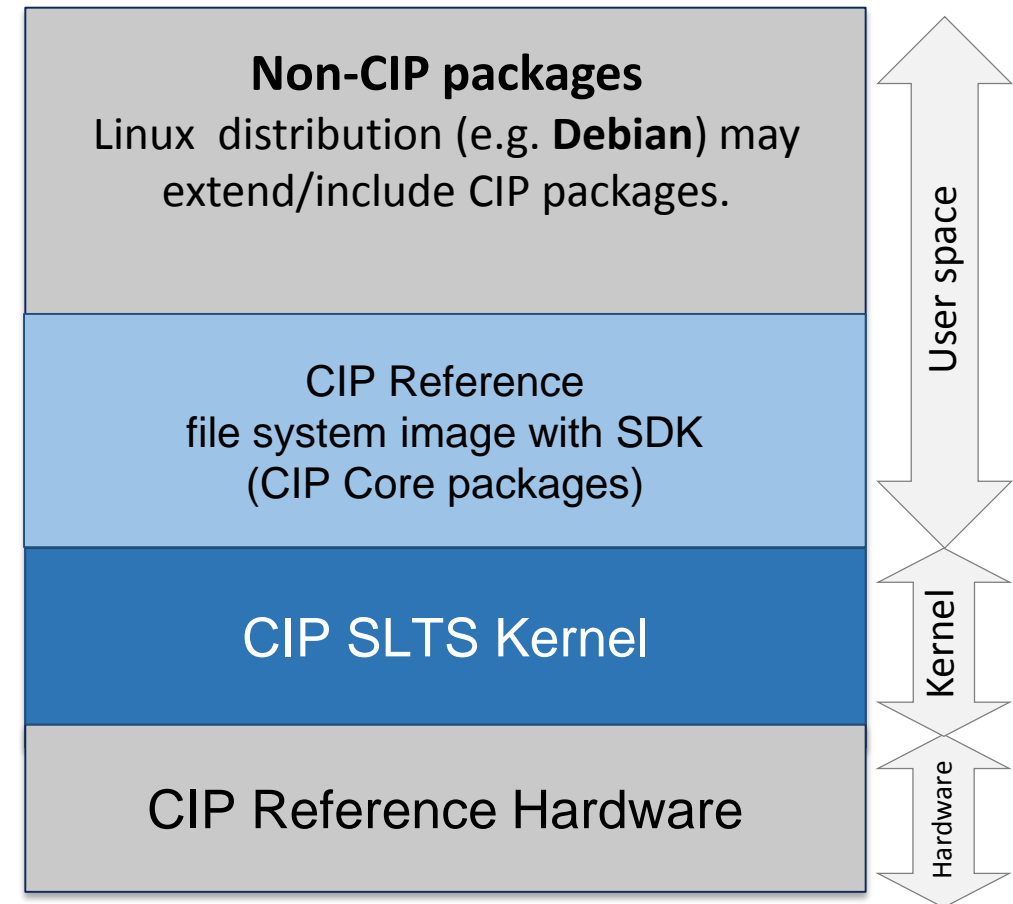
- OSBL is a set of industrial grade core open source software components, tools and methods
  - Reference implementation
  - Start for a minimal set for controllers in industrial-grade systems

3 months  
upstream  
kernel

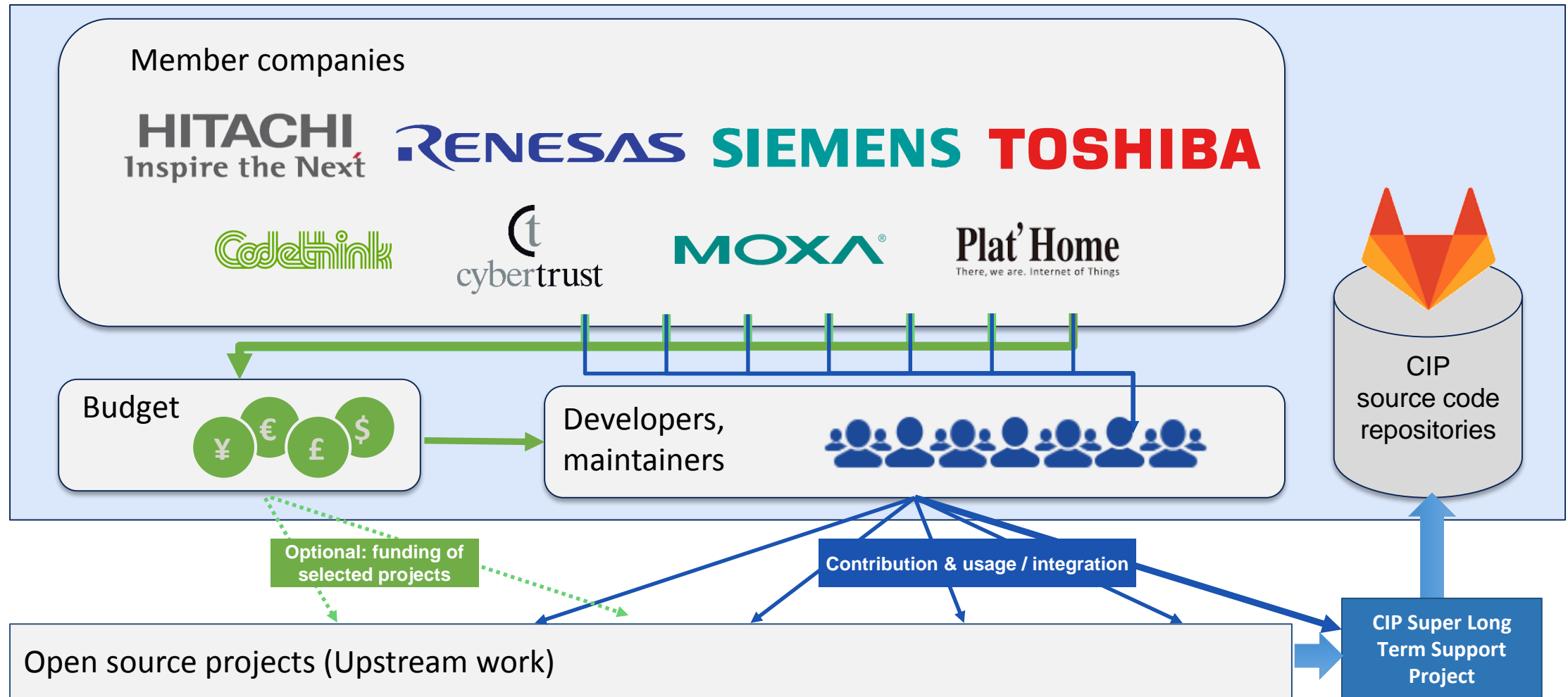
2-6 years Long Term Support (LTS) for  
desktop/server

2-6 years LTSI, support of embedded hardware

10-15 years super long support incl. core packages by <https://cip-project.org> (Siemens, Hitachi, Toshiba, Renesas et al.)

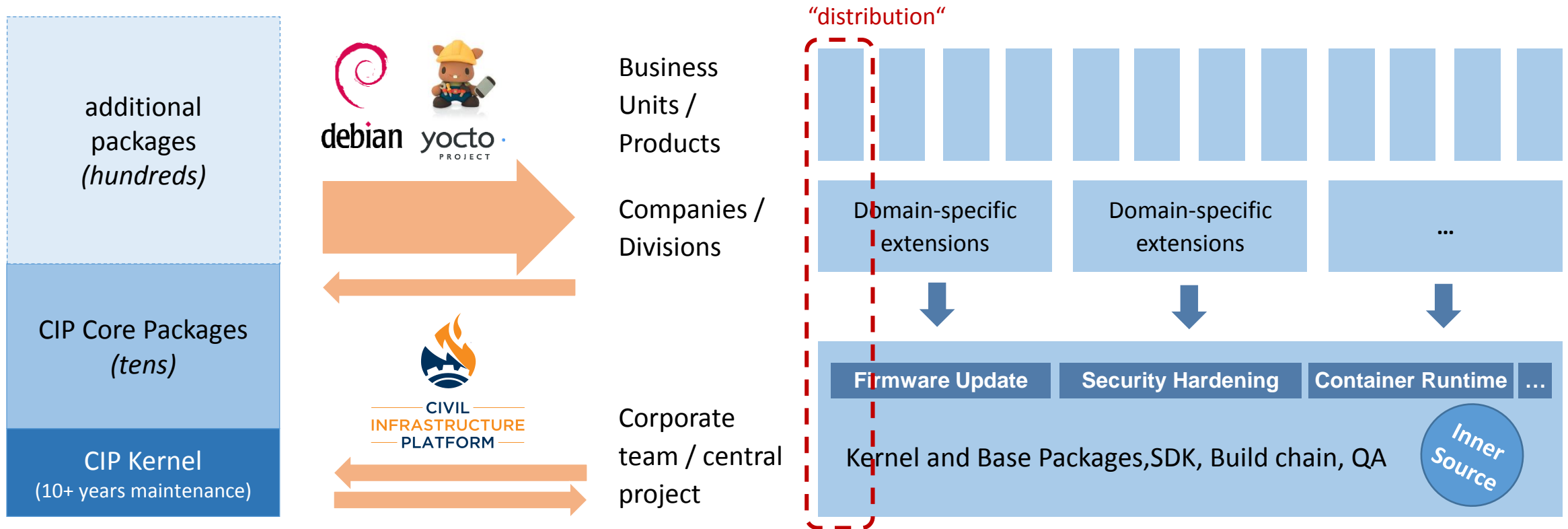


# The backbone of CIP are the member companies



# Mapping CIP into the company

Layered Linux distribution for industrial products, utilizing and influencing the relevant Open Source projects:

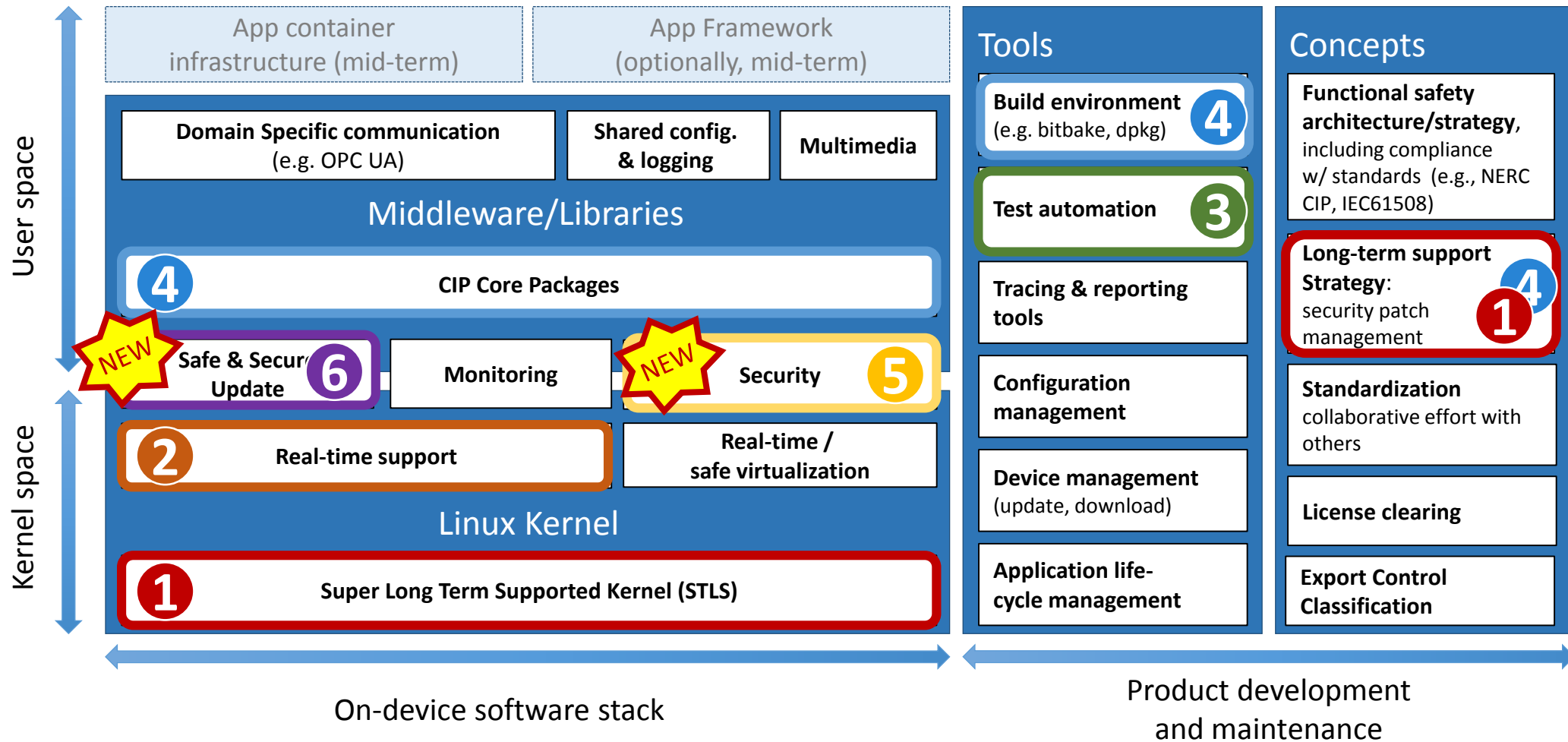


Up to 70% effort reduction achievable for OSS license clearing and vulnerability monitoring, kernel and package maintenance, application adaptation and testing for an individual product.



# CIP activities and status

# Scope of activities



# CIP focuses on upstream development

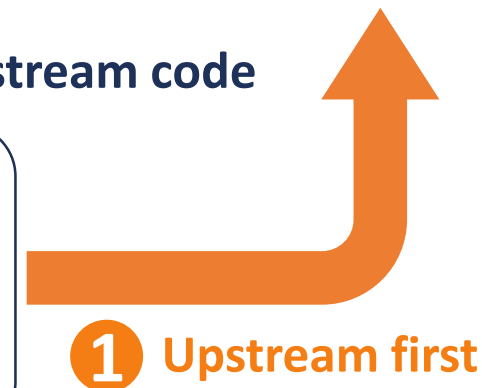


Upstream  
Projects

Contribute, Collaborate and use by CIP



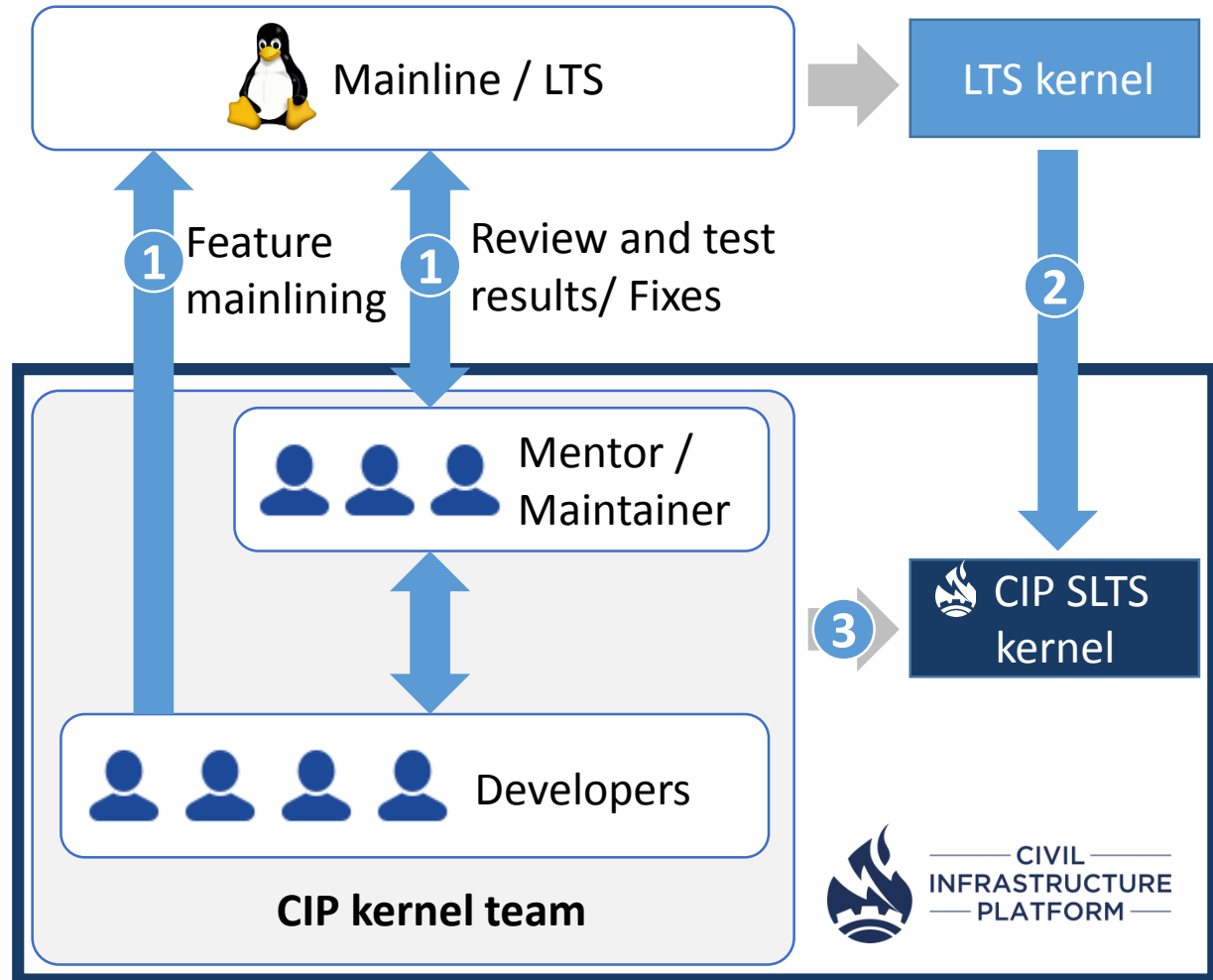
Contributing by CIP members as  
future candidates



CIP Open Source Base Layer (OSBL)

# 1 CIP SLTS kernel development (Upstream first development model)

- Upstream projects for CIP
  - Linux mainline and LTS
- How CIP collaborate with upstream?
  - Mainline
    - All backport patches should be up-streamed before merging
    - Many patches especially Renesas board related features has already up-streamed
  - Linux stable
    - CIP SLTS is based on LTS
    - CIP kernel team participate into LTS review process



1 up-streaming 2 use 3 integrate

# 1 CIP SLTS Kernel development (Join Linux review process)



4.4-stable review patch. If anyone has any objections, please let me know.

-----  
From: Christoph Hellwig <hch@lst.de>

commit f507b54dccfd8000c517d740bc45f20c74532d18 upstream.

The job structure is allocated as part of the request, so we should not free it in the error path of bsg\_prepare\_job.

Signed-off-by: Christoph Hellwig <hch@lst.de>

Reviewed-by: Ming Lei <ming.lei@redhat.com>

Signed-off-by: Jens Axboe <axboe@kernel.dk>

Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

---  
block/bsg-lib.c | 1 -  
1 file changed, 1 deletion(-)

```
--- a/block/bsg-lib.c
+++ b/block/bsg-lib.c
@@ -147,7 +147,6 @@ static int bsg_create_job(struct device
failjob_rls_rqst_payload:
    kfree(job->request_payload.sg_list);
failjob_rls_job:
-    kfree(job);
    return -ENOMEM;
}
```

## Reviewed by Ben Hutchings for 4.4-stable

On Tue, 2017-10-03 at 14:21 +0200, Greg Kroah-Hartman wrote:

```
> 4.4-stable review patch. If anyone has any objections, please let me know.
>
> -----
>
> From: Christoph Hellwig <hch@lst.de>
>
> commit f507b54dccfd8000c517d740bc45f20c74532d18 upstream.
>
> The job structure is allocated as part of the request, so we should not
> free it in the error path of bsg_prepare_job.
```

That function doesn't exist here (it was introduced in 4.13). Instead, **this backport has modified bsg\_create\_job(), creating a leak**. Please revert this on the 3.18, 4.4 and 4.9 stable branches.

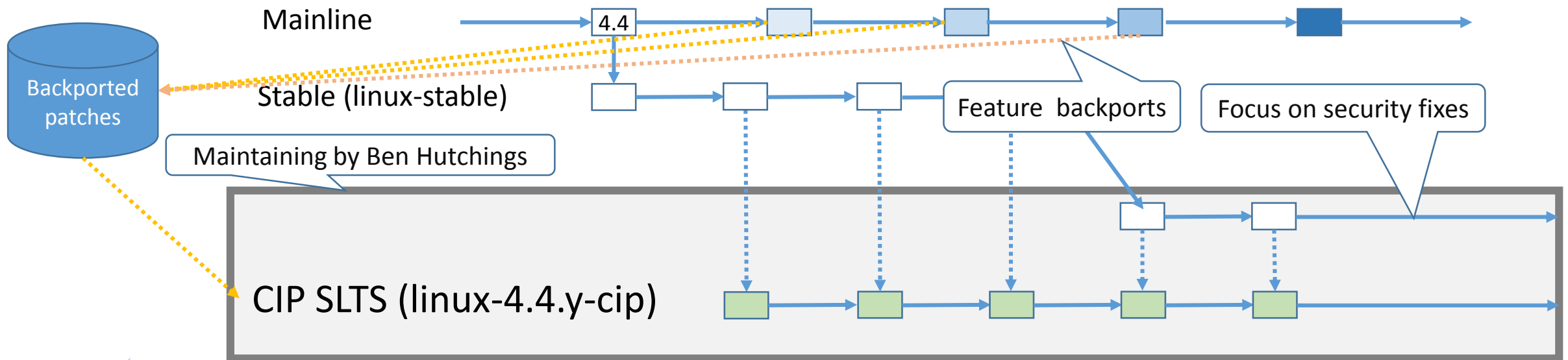
< -- snip -- >

--  
Ben Hutchings  
Software Developer, Codethink Ltd.

# 1 CIP SLTS Kernel development (Maintenance plan)

## CIP SLTS (linux-4.4.y-cip), Maintenance period 10 years

- CIP SLTS kernel tree is now available on kernel.org under the CIP group
  - <https://git.kernel.org/pub/scm/linux/kernel/git/cip/linux-cip.git/>
  - CIP RT kernel will be available soon
- Mentor: Ben Hutchings (Codethink)
- **Maintainer: Nobuhiro Iwamatsu (Cybertrust), N.N. (will be announced soon)**



# 1 CIP SLTS Kernel development

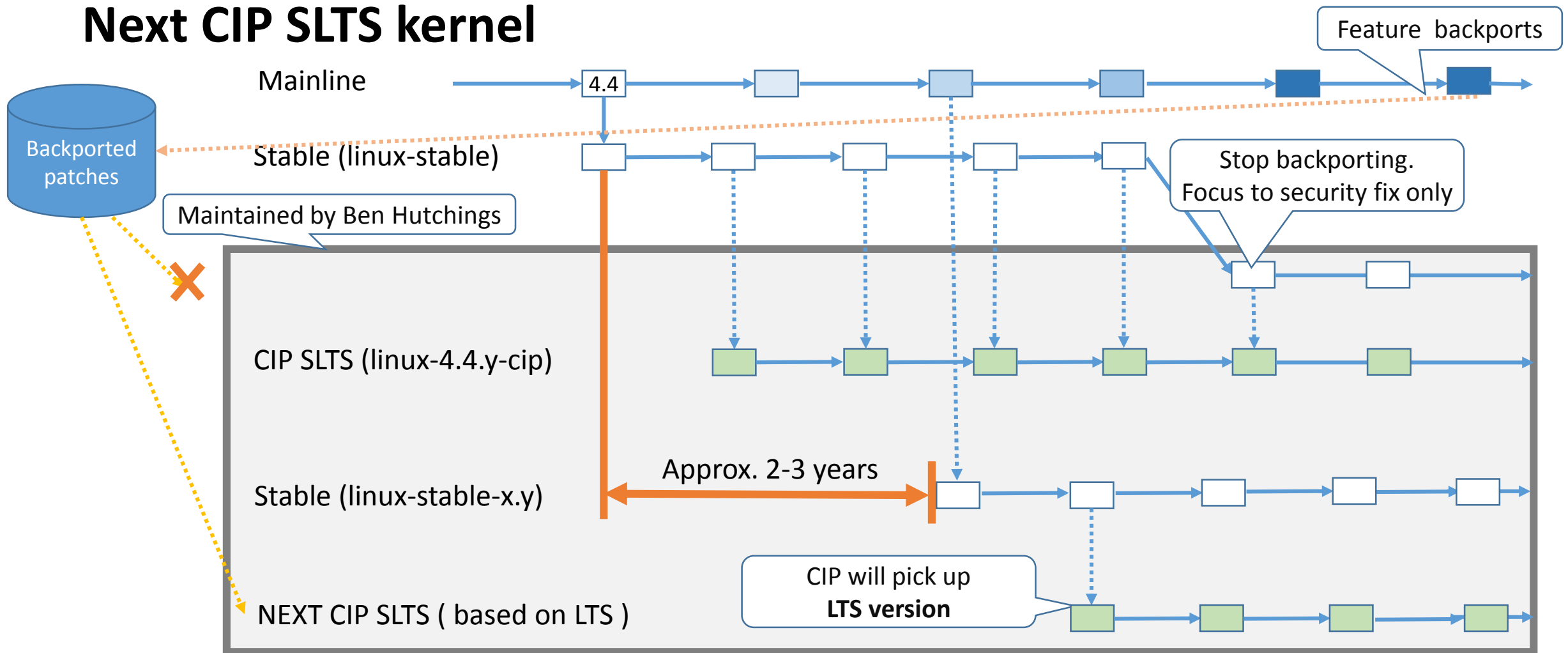


- Development status
  - The latest CIP kernel 4.4.154-cip28
- CIP Reference boards
  - QEMU x86\_64
  - AM335x Beaglebone Black (Armv7)
  - RZ/G1M iWave Qseven Development Kit (Armv7)
  - RZ/G2M-96CE(tentative name) (**Armv8**)
- CIP Reference board candidates (under consideration)
  - Physical x86\_64 board
  - Cyclone V Development Kit



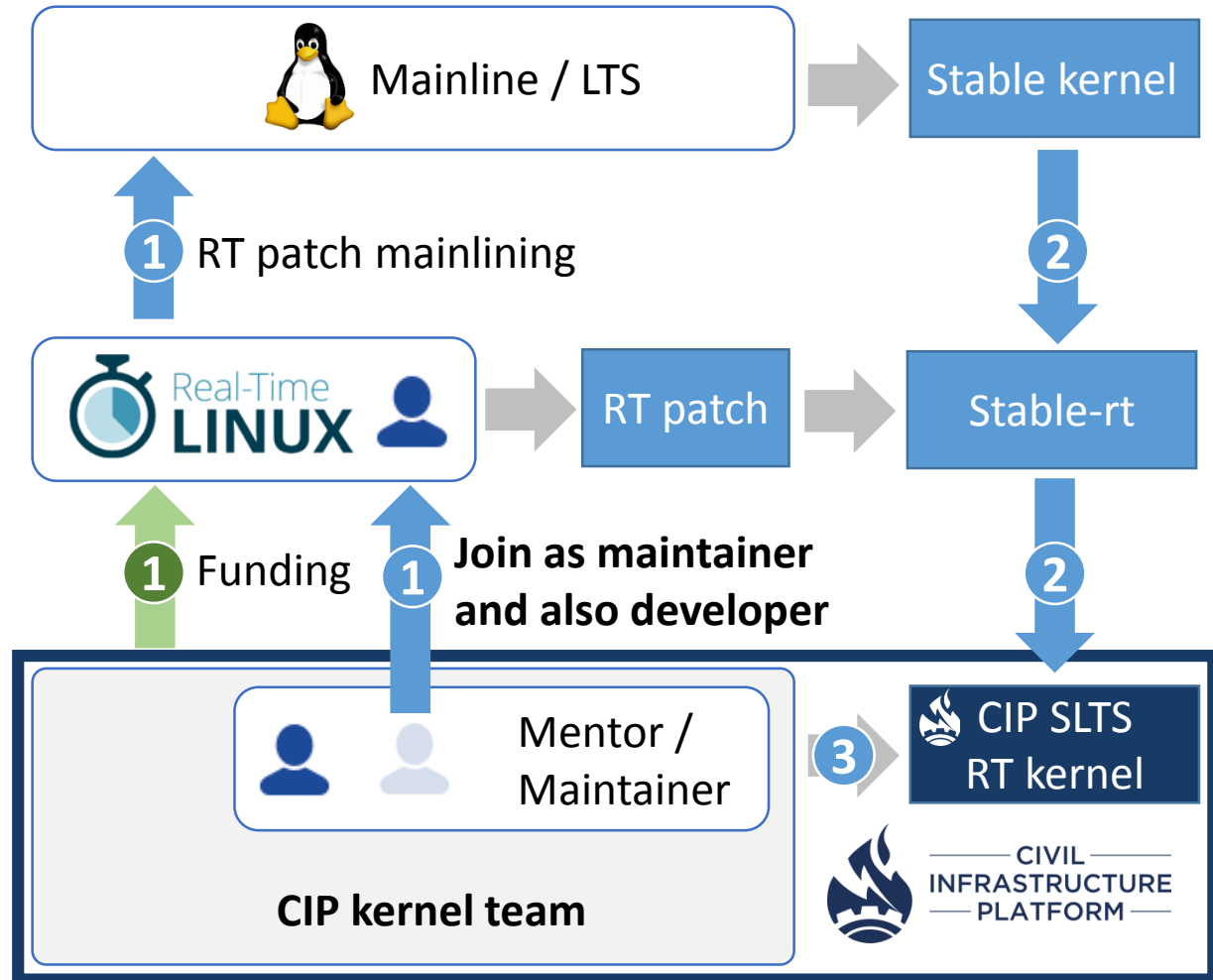
# 1 Next SLTS kernel version

## Next CIP SLTS kernel



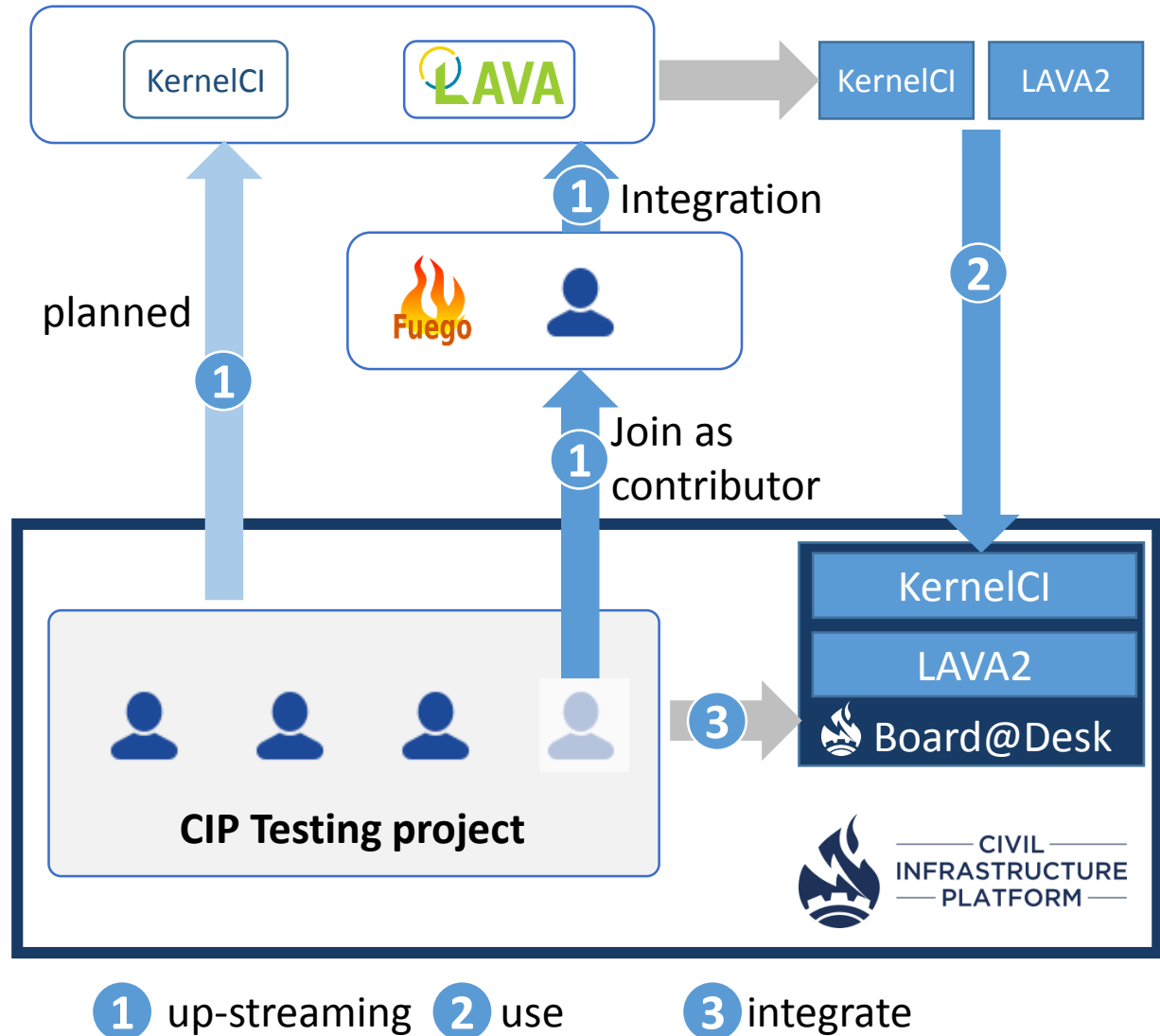
## 2 Real-time Linux development (PREEMPT\_RT)

- Upstream projects for CIP
  - Real-time Linux Project
- How CIP collaborate with upstream
  - Contributing PREEMPT\_RT mainlining
    - Join Real-time Linux Project as Gold member
  - Contributing RT stable maintenance
    - CIP member Daniel Wagner from Siemens is maintaining 4.4.y-stable-rt
    - CIP 4.4-rt-cip kernel based on 4.4.y-stable-rt
  - Test results are available on CI-RT
    - <https://ci-rt.linutronix.de/RT-Test/>



### 3 CIP Testing

- Upstream projects
  - CIP is using LAVA2 and KernelCI for testing environment
  - CIP member contributing
    - Fuego
- How CIP collaborate with upstream?
  - CIP testing B@D created with KernelCI and LAVA2
  - Sharing CIP testing results to public
    - <https://lists.cip-project.org/pipermail/cip-testing-results/>
  - Posting patches to creating features
    - Send test results from Fuego to KernelCI
    - LAVA support on Fuego
- What's next?
  - Closer collaboration with KernelCI



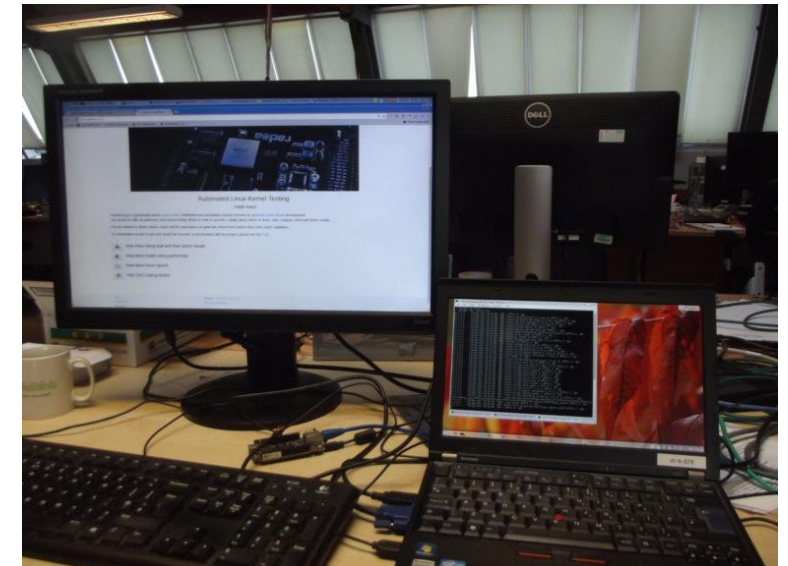
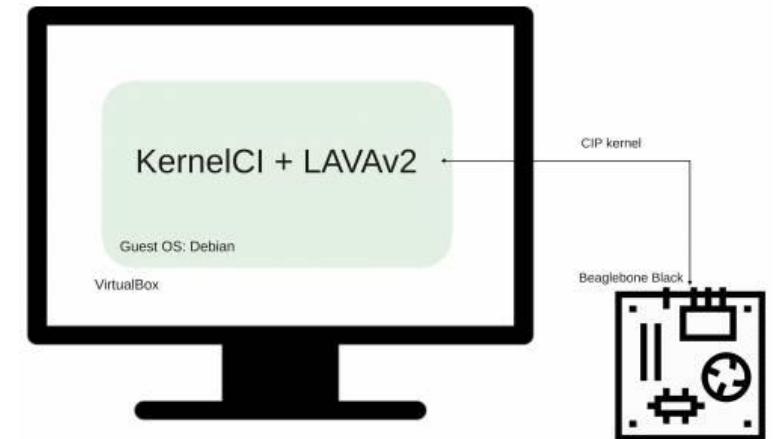
### ③ CIP testing

#### CIP Testing project

(<https://wiki.linuxfoundation.org/civilinfrastructureplatform/ciptesting>)

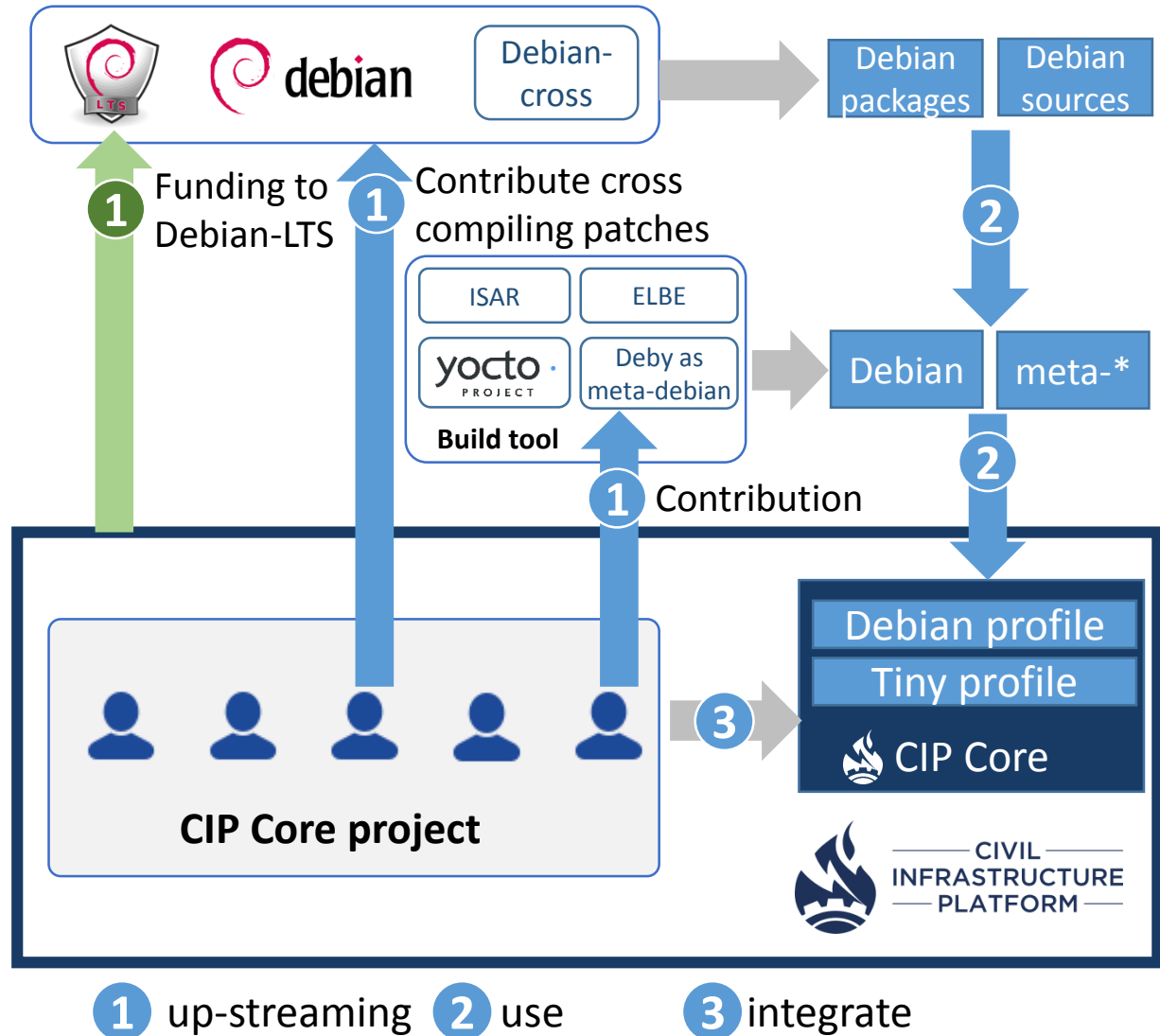
- B@D designed to:
  - Test Linux kernels and base systems locally.
  - On hardware connected to your dev machine.
- B@D features
  - Based on kernelci.org
  - Linux and Windows 10 as Host OS supported.
  - Shipped as VM and Vagrant based environments.
  - Results and logs sharing capabilities.
- Check the source code involved
  - <https://gitlab.com/cip-project/cip-testing/board-at-desk-single-dev/tree/master>

Board At Desk - Single Dev.



## 4 CIP Core

- Upstream projects for CIP
  - Debian
    - Debian-LTS
    - Debian-Cross
  - Deby / ISAR / ELBE / Yocto Project
- CIP Core will move to have 2 profiles
  - Tiny (Bitbake + Debian source code)
  - Debian (Debian binary (dpkg) based)
- How CIP collaborate with upstream?
  - Support Debian-LTS project
  - Contributing patches to Debian-cross
  - CIP Core uses Yocto Project and Deby
    - Deby is a layer for Poky to use Debian source code



## 4 CIP Core

- Upstream projects for CIP

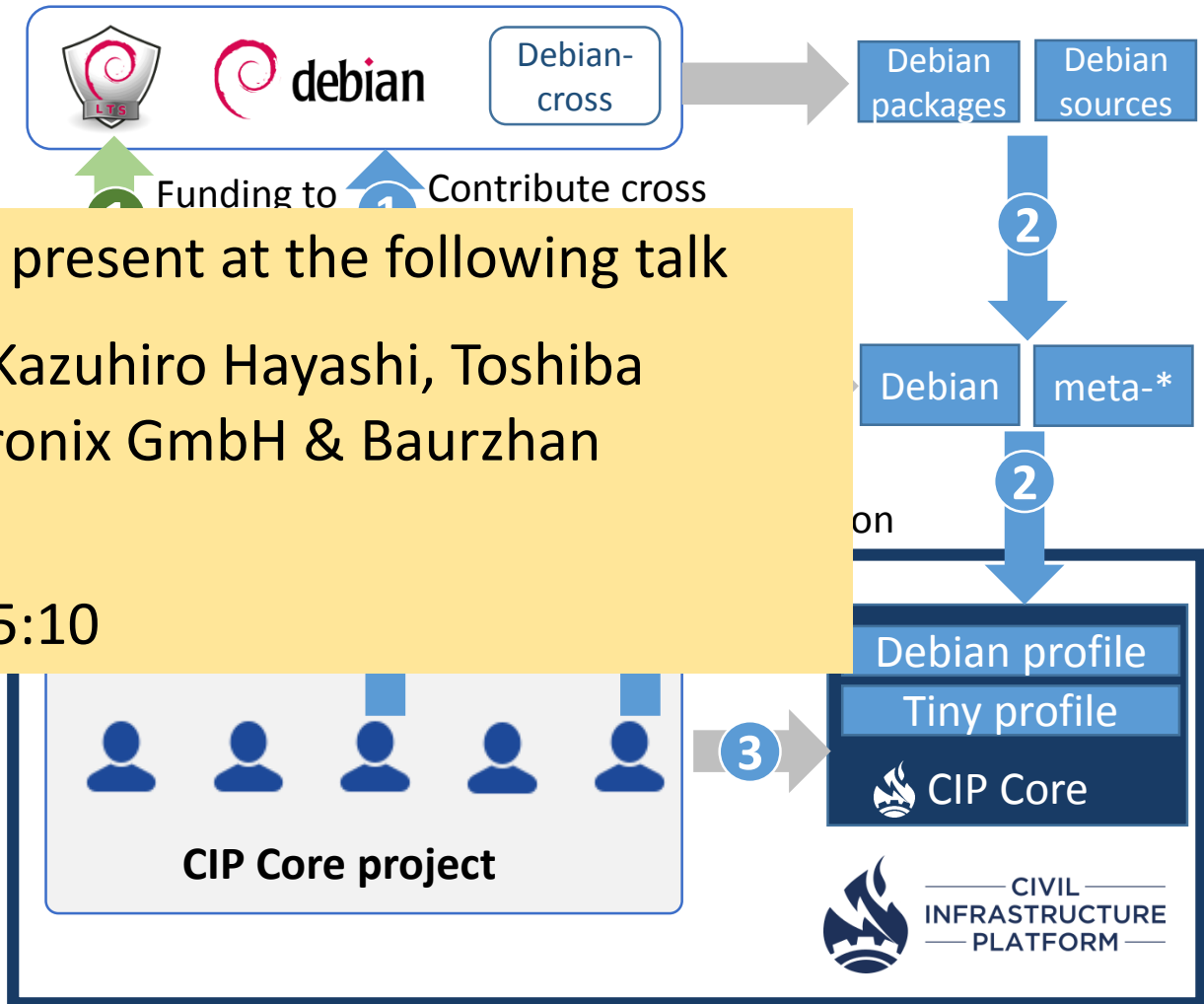
- Debian
  - Debian-LTS

- The latest status of this activity will present at the following talk

- CIP Core
    - Tiny
    - Debian
- Debian & Yocto: State of the Art** - Kazuhiro Hayashi, Toshiba Corporation & Manuel Traut, Linutronix GmbH & Baurzhan Ismagulov

- How CIP Core → Tuesday, October 23 • 14:30 - 15:10

- Support Debian-LTS project
- Contributing patches to Debian-cross
- CIP Core uses Yocto Project and Deby
  - Deby is a layer for Poky to use Debian source code

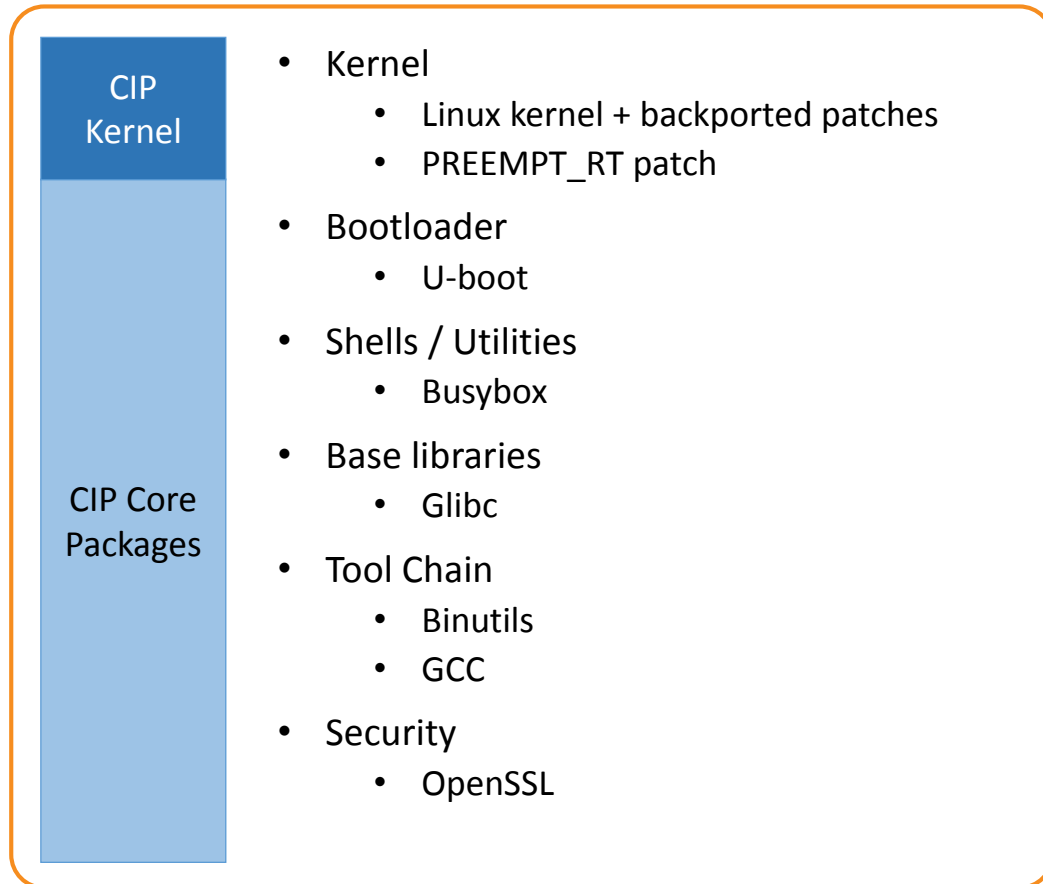


## 4 Example for CIP Core Tiny profile



### An example of minimal package set for CIP base layer

#### Candidates for initial component set



#### Keep these packages for Reproducible build



*NOTE: The maintenance effort varies considerably for different packages.*

## ④ Gaps and Common Goals between Debian and CIP

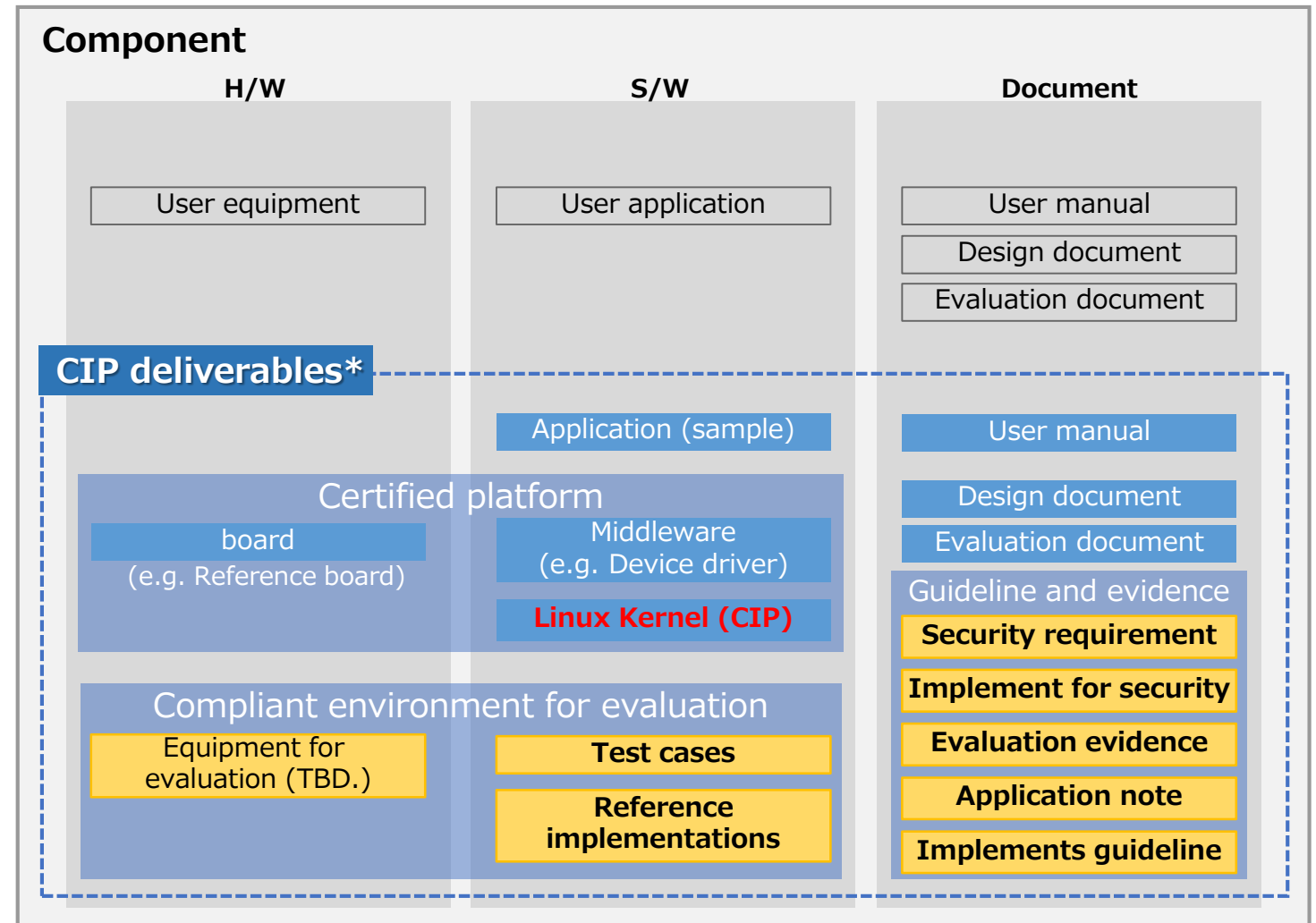


Debian	CIP requires	Chance to collaborate with Debian
<p><b>Support</b></p> <ul style="list-style-type: none"><li>▪ Term: 3+2 years by <b>Debian-LTS</b></li><li>▪ Num of source pkgs: over 25000 (67776 binary pkgs)</li></ul> <p><b>Build</b></p> <ul style="list-style-type: none"><li>▪ Should support native build</li><li>▪ Working on cross build packaging (Debian-cross)</li><li>▪ Reproducible build</li></ul> <p><b>OSS license compliance</b></p> <ul style="list-style-type: none"><li>▪ DEP-5 adoption is ongoing</li></ul> <p><b>Testing</b></p> <ul style="list-style-type: none"><li>▪ Packages has to be tested</li><li>▪ autopkgtest</li></ul>	<p><b>Support</b></p> <ul style="list-style-type: none"><li>▪ Term: 10+ years</li><li>▪ Num of pkgs: 10+ (minimum)</li></ul> <p><b>Build</b></p> <ul style="list-style-type: none"><li>▪ Need to have both native and cross build</li><li>▪ Binary / Source code should be managed and reproducible</li></ul> <p><b>OSS license compliance</b></p> <ul style="list-style-type: none"><li>▪ Generate reports automatically</li><li>▪ Easy to redistribute</li></ul> <p><b>Testing</b></p> <ul style="list-style-type: none"><li>▪ All packages should be tested in timely manner</li></ul>	<ul style="list-style-type: none"><li>▪ Longer term maintenance for limited number of packages <b>(CIP joined Debian-LTS)</b></li><li>▪ Contributing to <b>Debian-cross</b> <b>(RFC posted to Debian-cross)</b></li><li>▪ Exchange and share the <b>license review results</b></li><li>▪ Contributing <b>test cases</b> to upstream</li></ul>

## 5 Security working group



- CIP launched a new working group to focus on cybersecurity
- Goal
  - Provide guidelines and reference implementations to help developers to meet cybersecurity standard requirements (IEC 62443)
- Status
  - Just started



\*: Noted that this image is under planning and for only illustrative purposes.

## 6 Software update working group

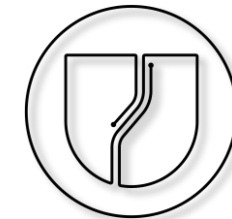


- CIP launched a new working group to focus software update
- **Goal**
  - Incorporate a common solution for software updates into CIP core
- **Status**
  - Just started



Eclipse IoT hawkBit

<https://www.eclipse.org/hawkbite/>



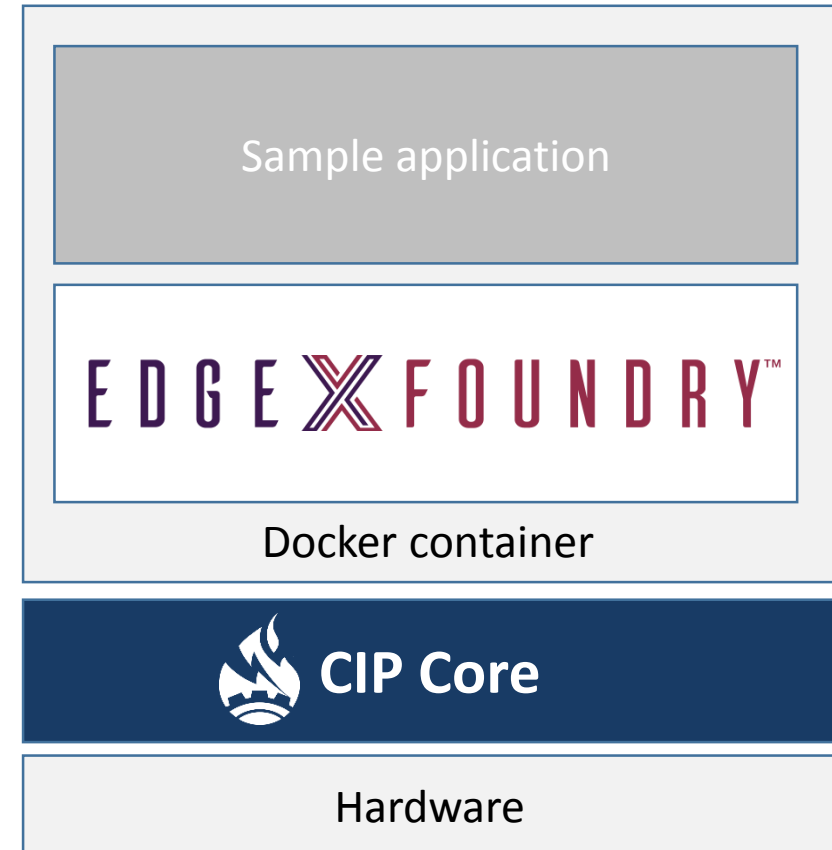
SWUpdate

<https://github.com/sbabic/swupdate>

# Collaboration: EdgeX Foundry on CIP Core



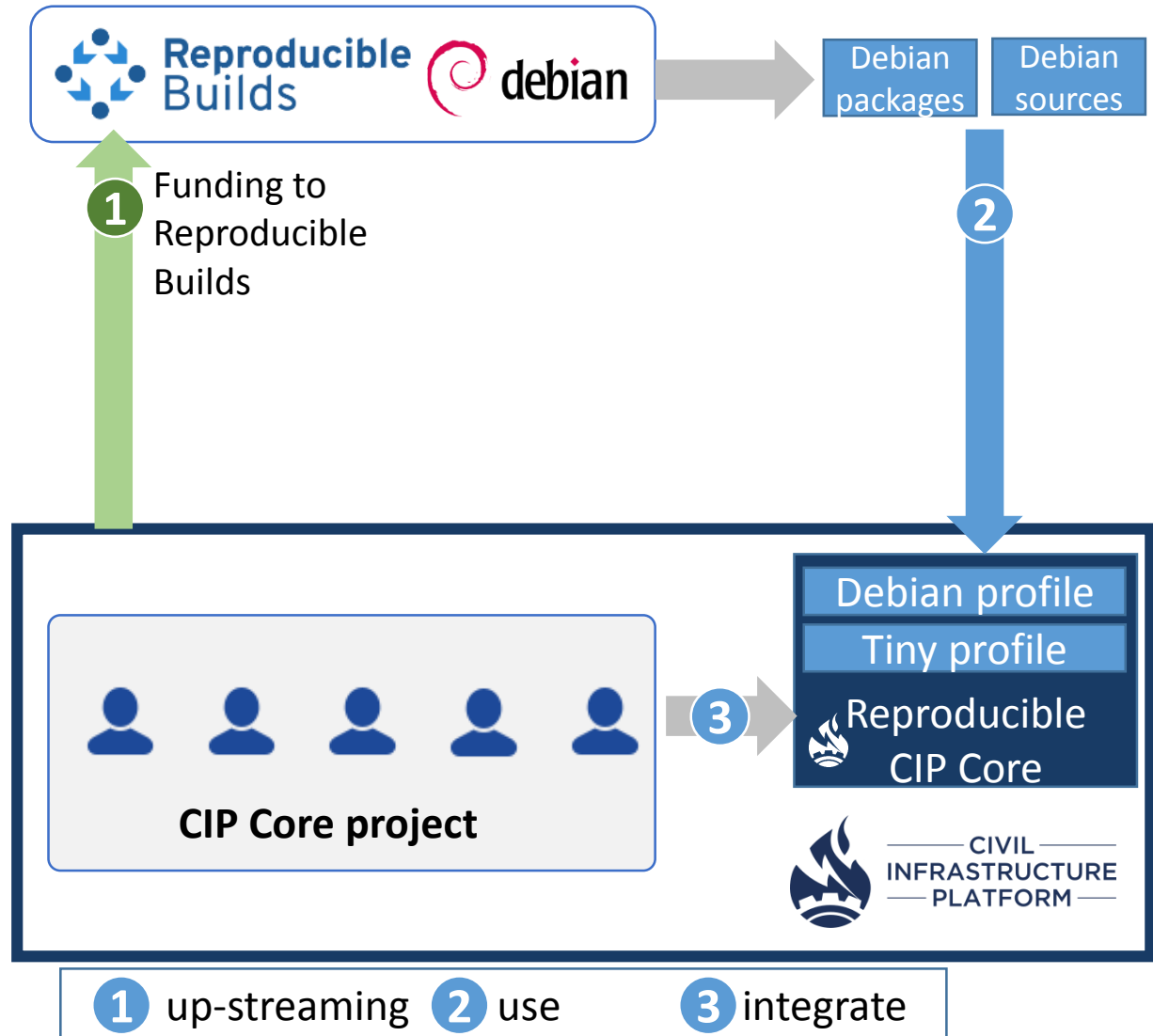
- CIP has joined EdgeX Foundry as Associate Member
- **EdgeX Foundry on CIP Core**
  - Purpose
    - To demonstrate CIP Core provides maintained base system for IoT systems
  - Goal
    - Create a sample implementation to run EdgeX Foundry on CIP Core
    - Proof of concept project
  - Status
    - Source code is available on CIP GitLab  
<https://gitlab.com/cip-playground/edgex-cip>



# Collaboration: Reproducible builds



- CIP became a sponsor of Reproducible builds
- Collaboration plan
  - Ensure reproducible build for CIP Core over lifetime



# Summary



- CIP today focuses on
  - **Kernel maintenance:** maintaining Linux kernels for very long time including real-time support
  - **Testing:** providing a test infrastructure and evolve tests
  - **CIP Core packages:** a set of industrial-grade components that require super long-term maintenance including the required build tool chains
  - **Security:** Improving to have security features and to follow Cyber Security Standard
  - **Collaboration:** Linux stable, Debian/Debian-LTS, Real Time Linux, Reproducible Builds, EdgeX Foundry
- New activities started: Security (IEC 62443-4-2), SW update

# Conclusion



- Our Civilization needs an Open Source Base Layer of industrial-grade software
  - CIP provides this, using Linux
- Sustainability is ensured by
  - The backing of big industrial and semiconductor companies
  - Close cooperation with and building with mature Open Source projects (Debian, PREEMPT\_RT, KernelCI, ...)
  - Providing suitable tool chains
  - Ensuring in-depth tests
- **Contribution and collaboration with upstream projects are the key CIP activities**



# Questions?

# Contact Information and Resources



To get the latest information, please contact:

- CIP Mailing list: [cip-dev@lists.cip-project.org](mailto:cip-dev@lists.cip-project.org)

## Other resources

- CIP Web site: <https://www.cip-project.org>
- CIP Wiki: <https://wiki.linuxfoundation.org/civilinfrastructureplatform/>
- CIP source code
  - CIP GitLab: <http://www.gitlab.com/cip-project>
  - CIP kernel: <git://git.kernel.org/pub/scm/linux/kernel/git/cip/linux-cip.git>



# Thanks for your attention!