



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

Industrial-grade Open Source Base Layer Development

Yoshitake Kobayashi, Toshiba Corp.

Urs Gleim, Siemens AG

Embedded Linux Conference Europe, Prague, October 24, 2017

**More updates will be
announced at CIP session
on October 24th @ 11:45am.**

What is CIP?

What is CIP?



- One of the most conservative open source project in the Linux Foundation
- One of the most important projects for our civilization

What is CIP?



- One of the most conservative open source project in the Linux Foundation
- CIP aims to
 - Provide an **open source base layer** for CIP related embedded systems
 - Work closely with the upstream community
- CIP **does not** aim to
 - Create a new Linux distribution

Our Civilization is run by Linux

Transport



Rail automation



Vehicle control



Automatic ticket gates

Energy

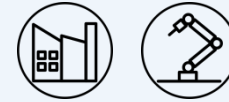


Power Generation



Turbine Control

Industry



Industry automation



CNC control



Industrial communication

Others



Healthcare



Building automation



Broadcasting

An aerial photograph of San Francisco, showing the dense city skyline with numerous skyscrapers, including the Transamerica Pyramid. The Golden Gate Bridge is visible in the background, spanning the Golden Gate Strait. The city is surrounded by water, and the hills are visible in the distance.

There are issues to be solved...

The background image shows a large, multi-bay electronic cabinet, likely a railway signaling system. The top section features a row of colorful terminal blocks in red, orange, green, and blue. Below this, the cabinet is filled with numerous modules, each with a large number of pins or connectors. Bundles of multi-colored wires are organized and connected to these modules. The overall appearance is that of a complex, industrial-grade electronic system.

A Railway System:

25-50 years products life-cycle

with very reluctant nature for product update and upgrade of hardware and base software platform

Railway Example



3 – 5 years development time

2 – 4 years customer specific extensions

1 year initial safety certifications / authorization

**3 – 6 months safety certifications / authorization for follow-up releases
(depending on amount of changes)**

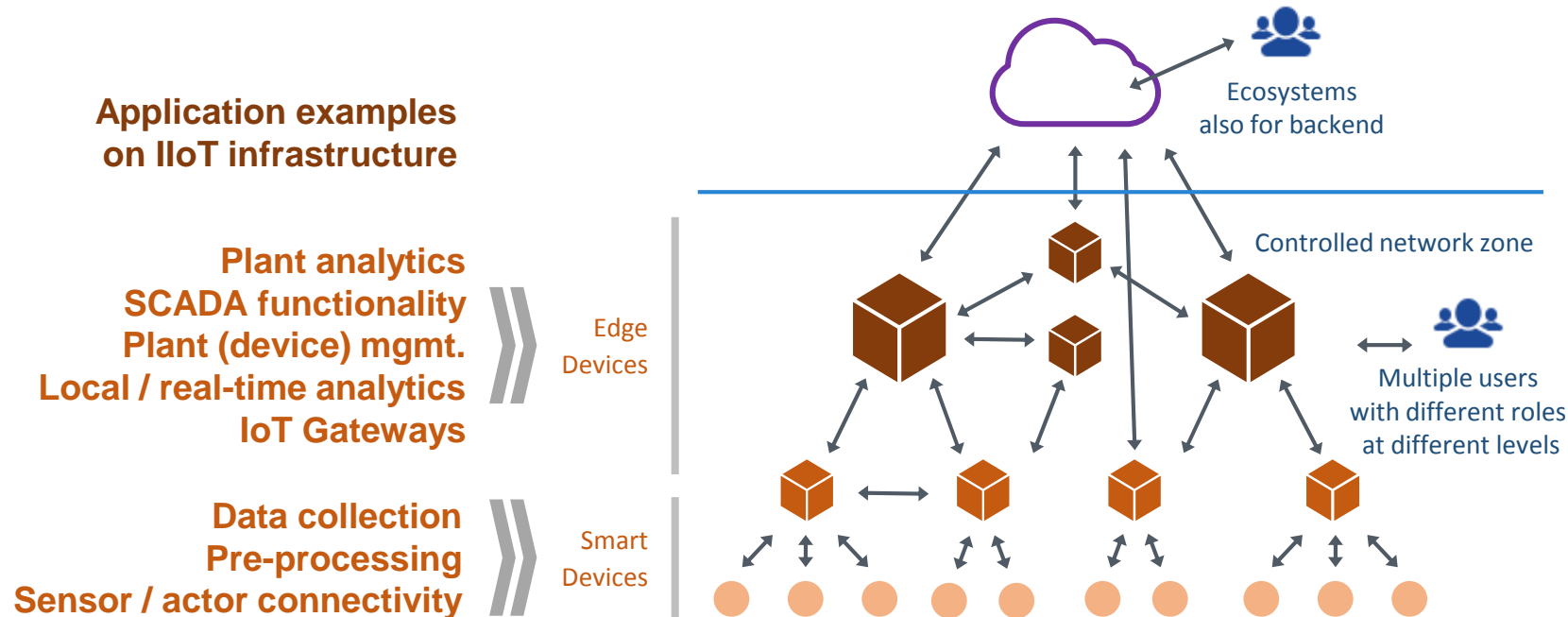
25 – 50 years lifetime

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



The Problems we face ...

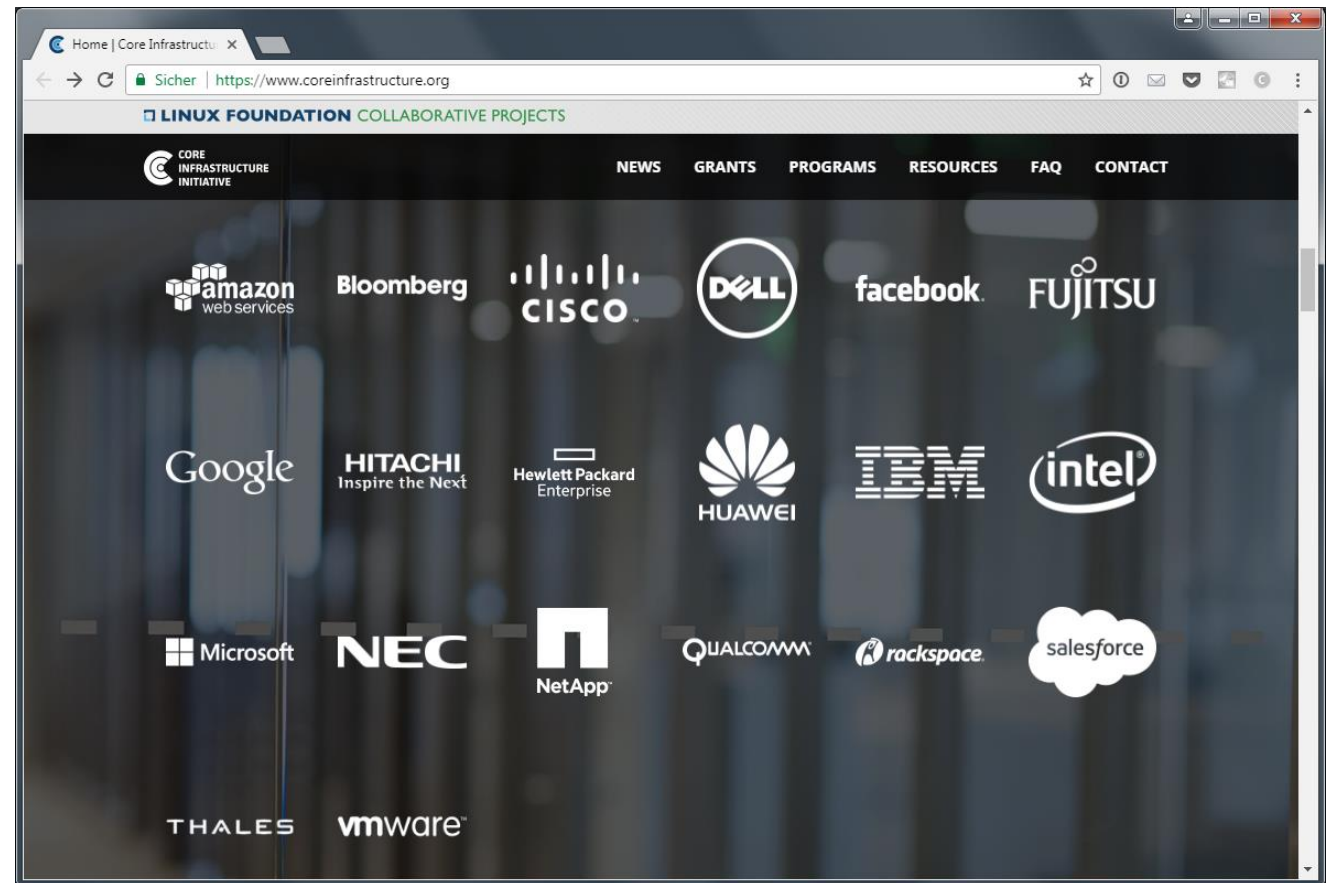
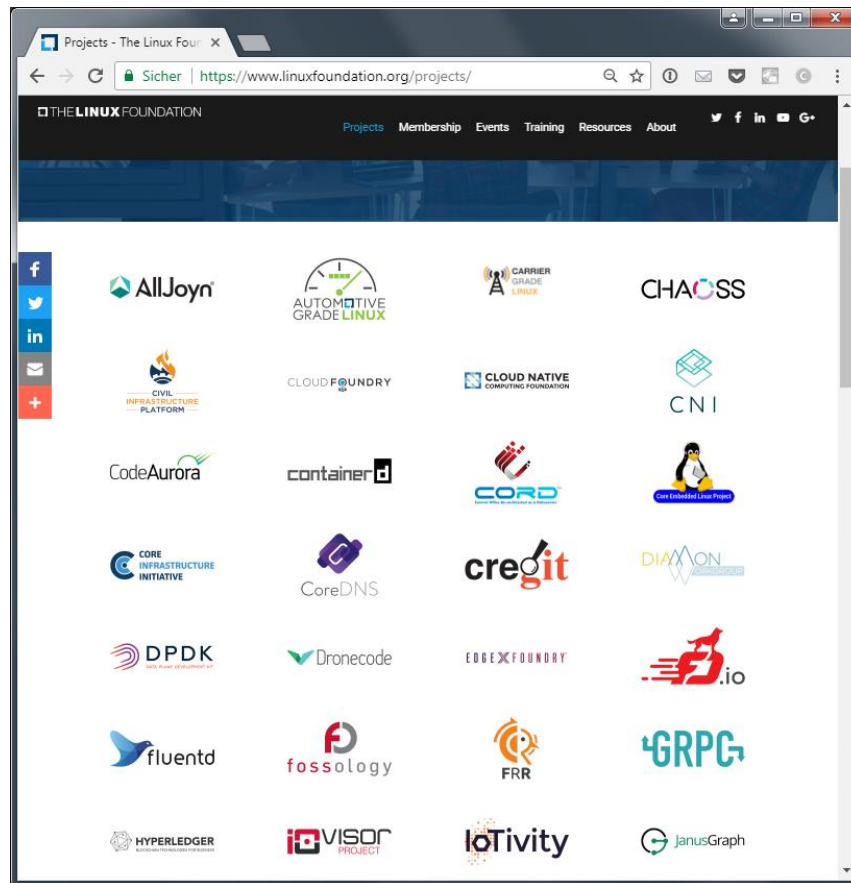


- The systems that support our modern civilization need to **survive for a VERY LONG TIME**. Until now the corresponding industrial grade super long term maintenance has been **done individually by each company**.
- These systems not only have to survive for a long time, they must be **“INDUSTRIAL GRADE”** (robust, secure and reliable). And at the same time the industry will also need to **catch up with the latest technology trends**

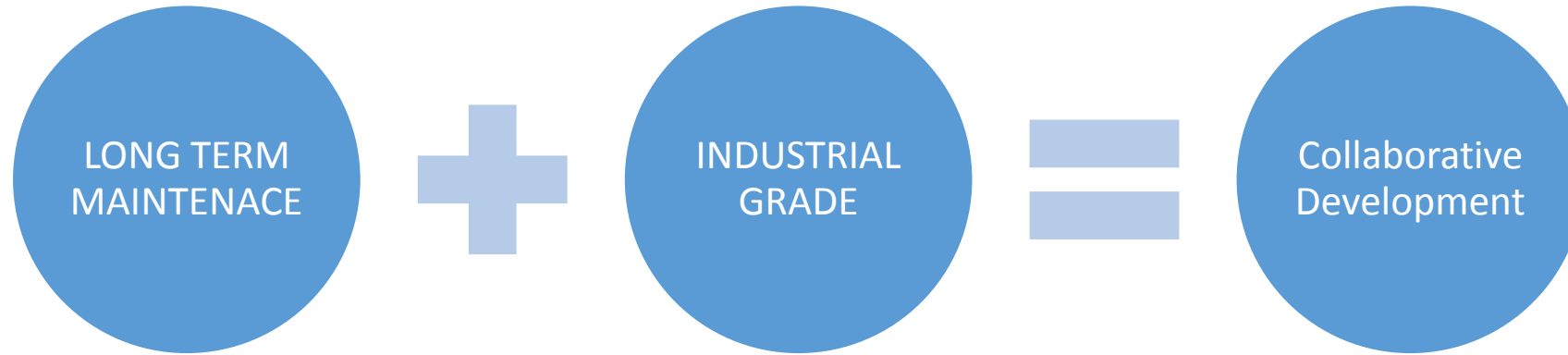
The genesis of a collaborative project

Linux Foundation Projects

Driving joint efforts and backing them with people and budget.



The Solutions we need ...



- **We need a Collaborative framework** to maintain the same open source based system for many, many, many years to keep it secure, robust and reliable.
- AND most importantly, we need to do this collaboratively in the **upstream communities**, not locally.

CIP is our 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

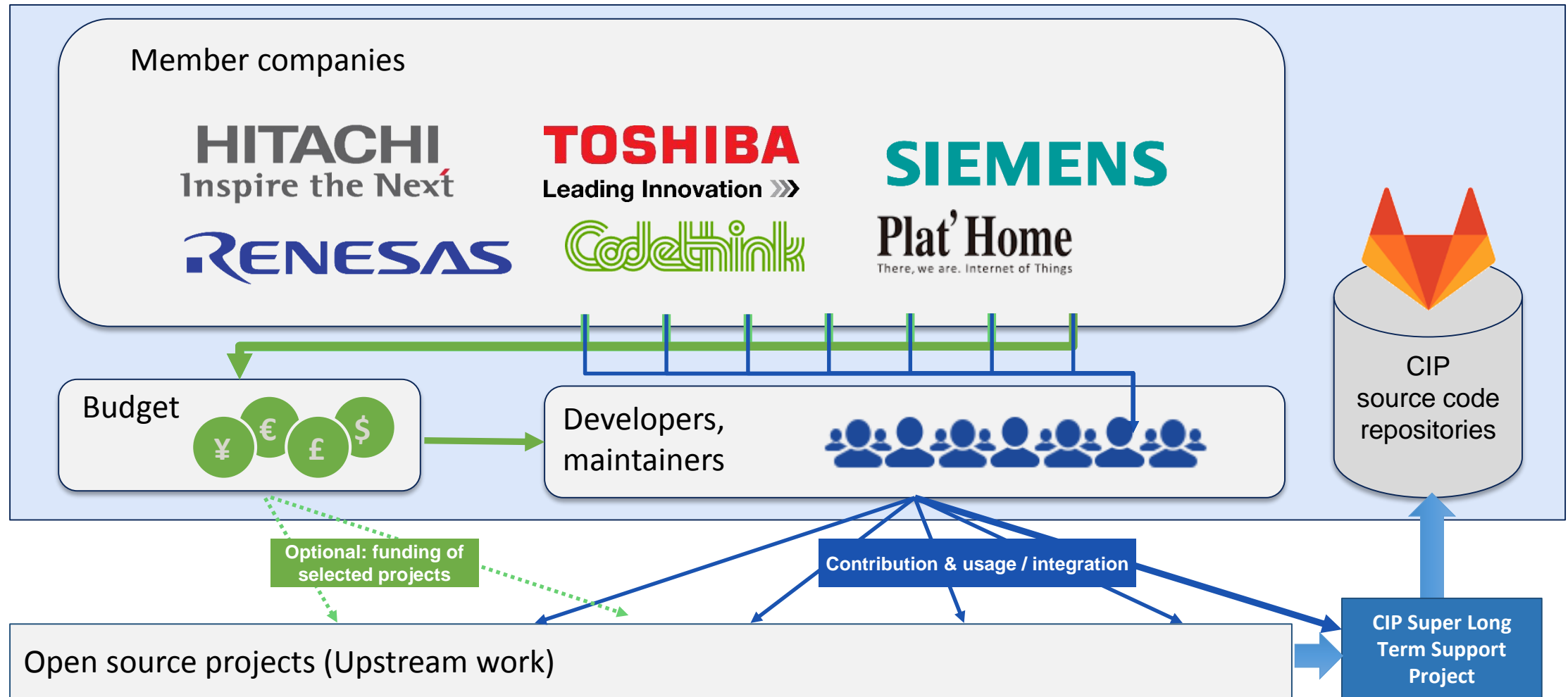
<https://www.cip-project.org/>



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

since April 2016

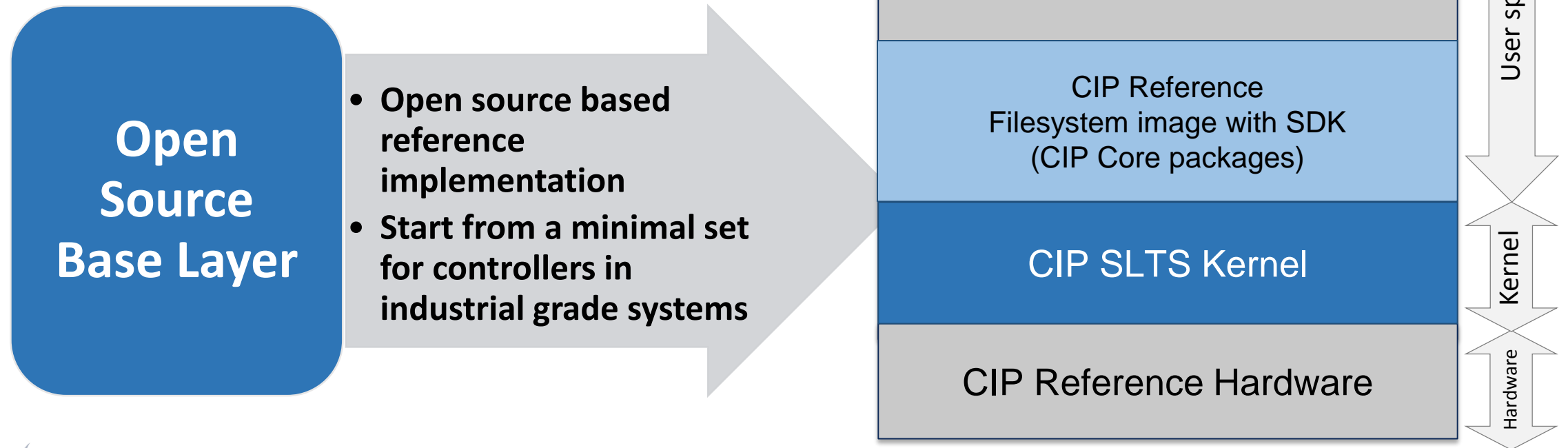
The backbone of CIP are the member companies



What is CIP, again?

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

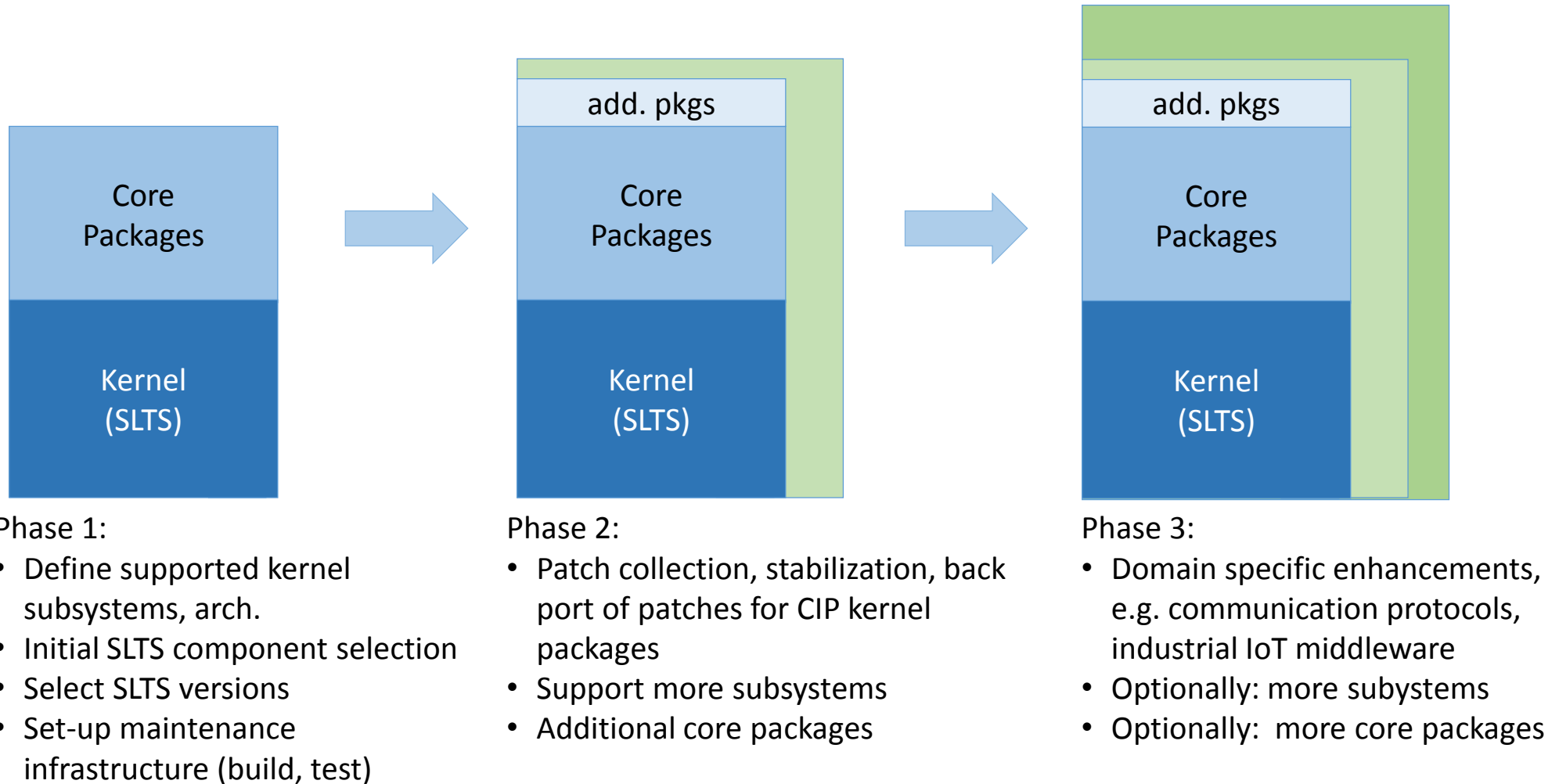
- OSBL is a set of industrial grade core open source software components, tools and methods



Development plan



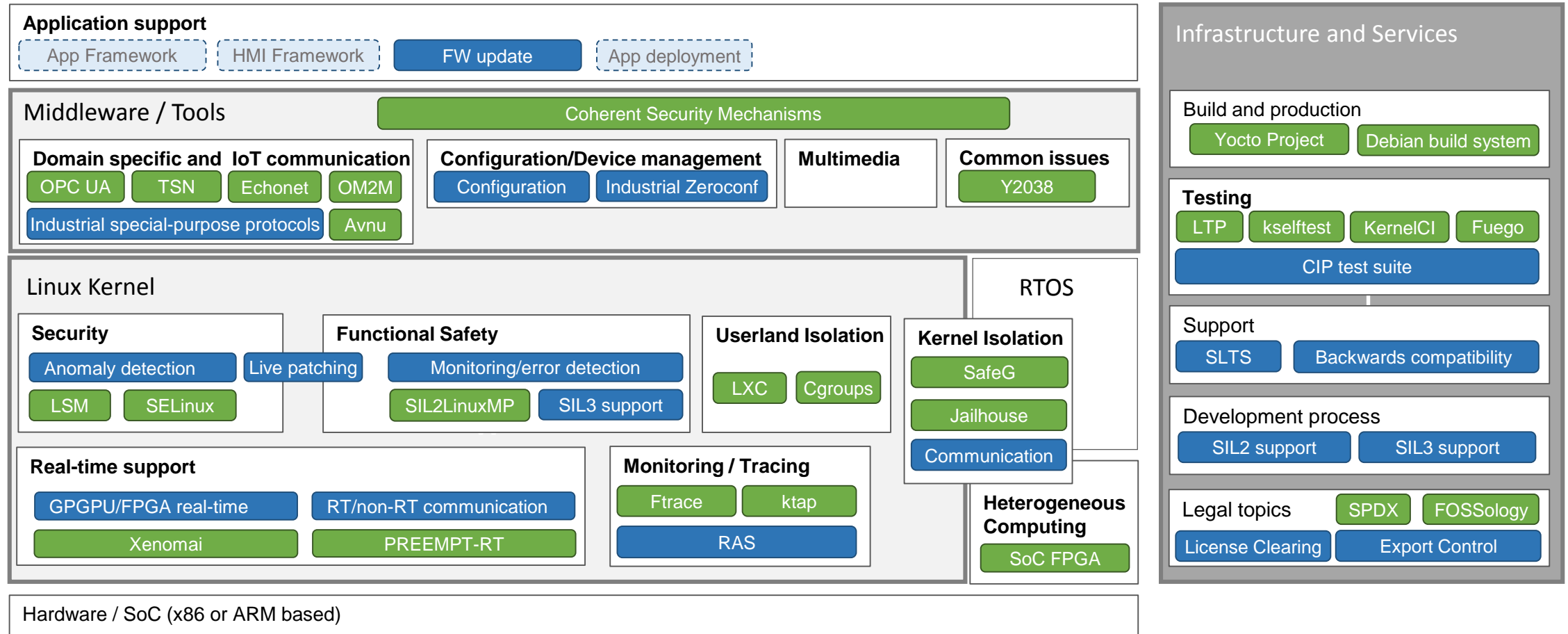
CIP will increase the development effort to create a industrial grade common base-layer



Vision: Technical topics and related projects



** Topics will be added or removed to reflect CIP technical interests*



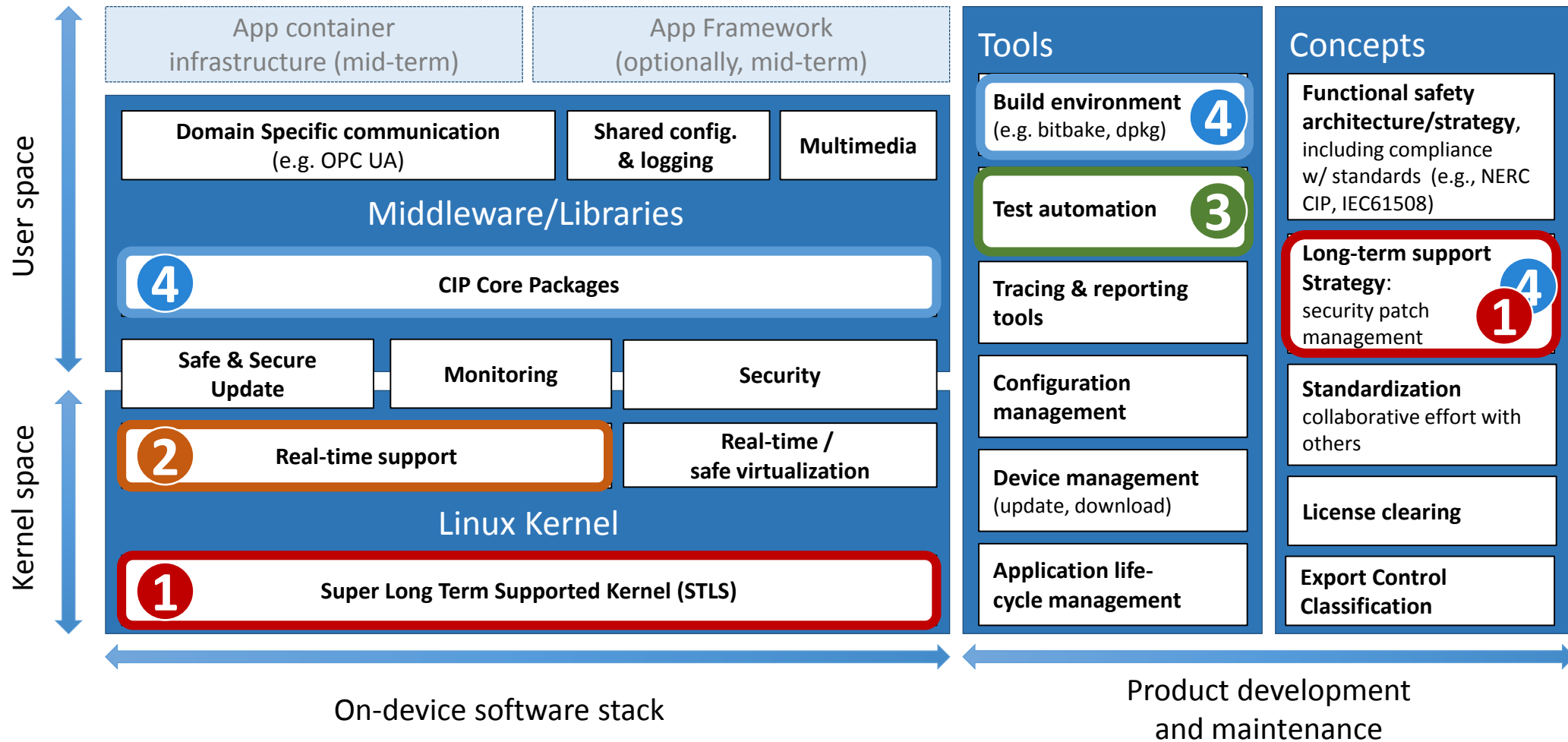
Legend

To be specified / implemented by CIP

Integration / cooperation

CIP activities and status

Scope of activities





1 Kernel maintenance

- The first action taken by the CIP project is to select and maintain Linux kernels for very long time (+15 years). To achieve goal a group of experts has been assigned.

2 PREEMPT_RT patches are added to the CIP kernel

3 Testing

- Civil infrastructure industry has high stability, reliability and security standards in order to ensure safety critical systems. The CIP Testing project has been formed to address this reality. So far the efforts are focused on testing the CIP kernel. In the future they will be extended to the complete CIP platform.

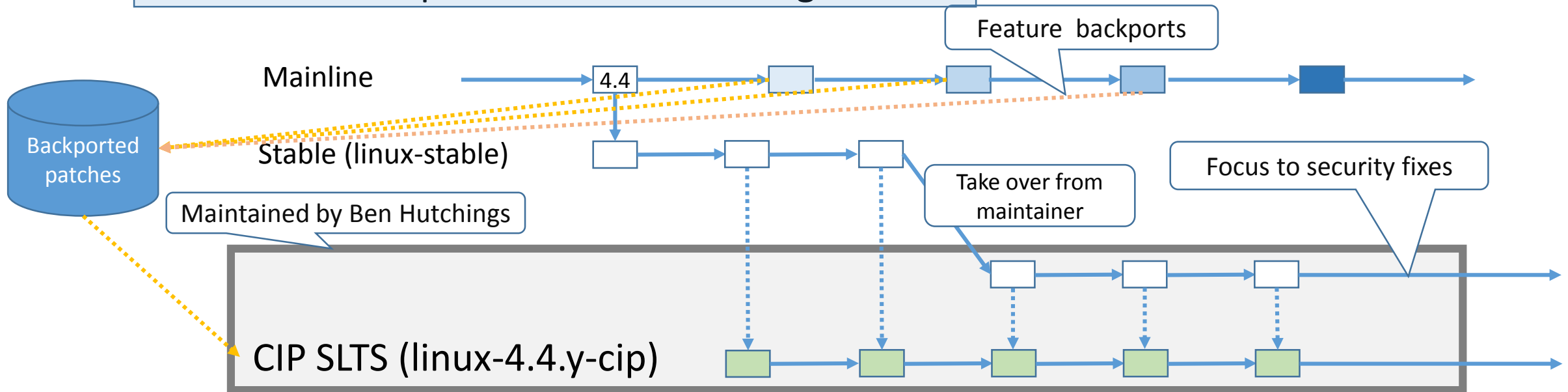
4 CIP Core

- This project focus to create reference minimal file system images that allow testing the CIP Core packages: a set of industrial-grade components that require super long-term maintenance.

1 CIP SLTS Kernel development (1/5)

CIP SLTS (linux-4.4.y-cip), Maintenance period 10 years and more (10-20 years)

- Official CIP SLTS kernel tree based on linux-stable.git
 - <https://git.kernel.org/cgit/linux/kernel/git/bwh/linux-cip.git/>
- Maintainer: Ben Hutchings
- Linux 4.4.83-cip8 released on 21st August 2017



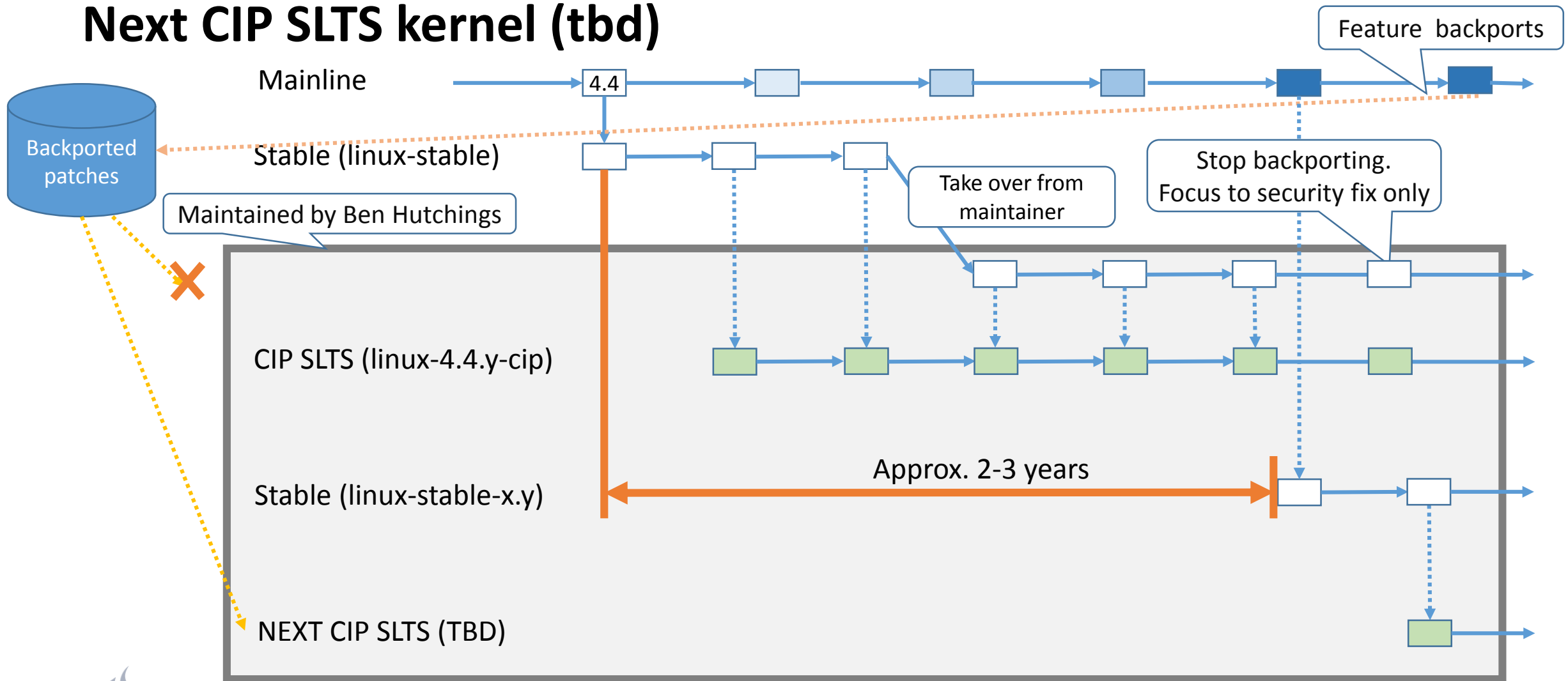
1 CIP SLTS Kernel development (2/5)



- Kernel maintenance policy
 - <https://wiki.linuxfoundation.org/civilinfrastructureplatform/cipkernelmaintenance>
 - Follow the stable kernel development rule as the basis
 - Feature backports are acceptable
 - All features has to be in upstream kernel before backport to CIP kernel
 - **CIP has “Upstream first” policy**
 - Validation will be done by CIP test infrastructure and/or members
- Current backported features on 4.4.y-CIP
 - Kernel Self Protection Project related features
 - Address Space Layout Randomization for user space process (ASLR)
 - GCC’s undefined behaviour Sanitizer (UBSAN)
 - Faster page poisoning
 - Board support patches for Renesas RZ/G and Siemens IoT2020

1 CIP SLTS Kernel development (3/5)

Next CIP SLTS kernel (tbd)



1 CIP SLTS Kernel development (4/5)



Out-of-tree drivers

- In general, all out-of-tree drivers are unsupported by CIP
- Users can use CIP kernel with out-of-tree drivers
 - If a bug is found in such a modified kernel, users will first demonstrate that it exists in the CIP kernel source release in order for the CIP maintainers to act on it.

1 CIP SLTS Kernel development (5/5)

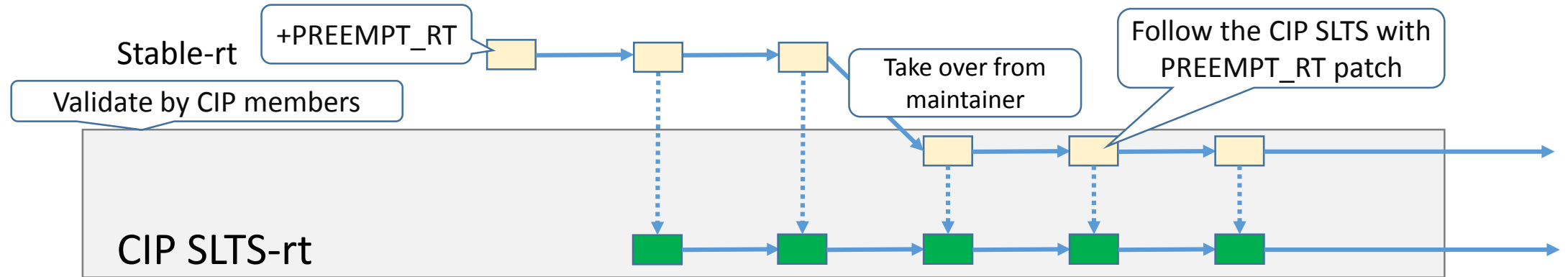


- CIP Kernel maintenance (<https://gitlab.com/cip-project/linux-cip>)
- Ben Hutchings is the CIP Kernel maintainer.
 - CIP kernel v4.4.92-cip11 released on Oct 18th:
 - Regressions and security fixes that land in LTS.
 - Specific industrial grade hardware support.
 - Talk about CIP kernel at ELCE
 - Tue Oct 24th at 16:55
 - Congress Hall II

2 CIP SLTS real-time support (1/2)

CIP SLTS+PREEMPT_RT (will be separately maintained by CIP members)

- CIP kernel tree based on linux-stable-rt and patches from CIP SLTS
- Validation will be done by CIP



2 CIP SLTS real-time support (2/2)



- CIP has become a Gold Member of the Real Time Linux Project
- What's next
 - Work together with the RTL Project
 - A CIP member is working to become the maintainer of 4.4.y-stable-rt, the base version of the CIP Kernel.
- More information
 - <https://wiki.linuxfoundation.org/realtime/rtl/start>



③ CIP testing (1/3)



Milestones of CIP testing and current status

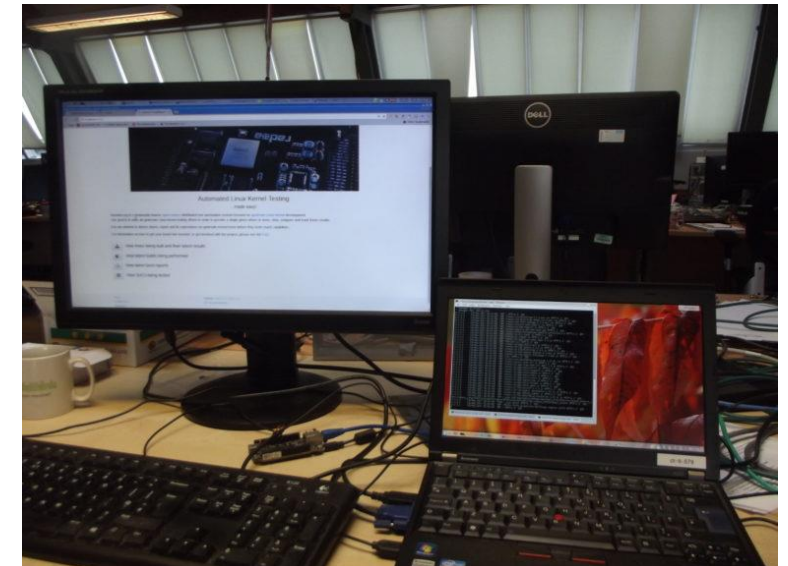
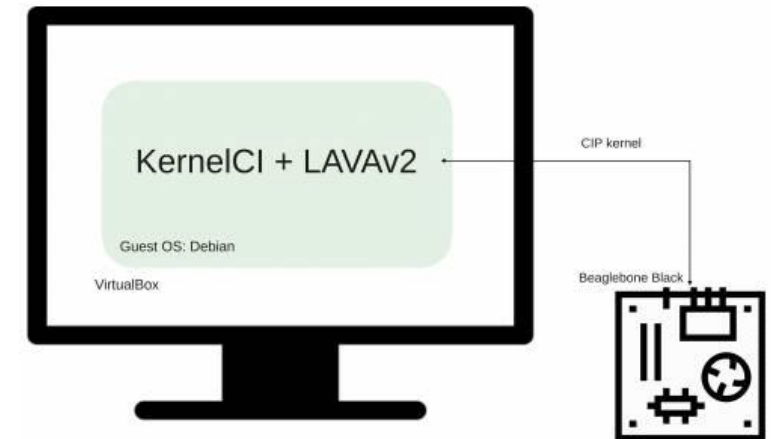
1. Board at desk - single dev
 - A setup that allows a developer to test the CIP kernel on the CIP selected hardware platform connected locally to her development machine using kernelCI tools.
2. CIP kernel testing
 - Test the CIP kernel on a regular basis and share the results with other CIP community members.
3. Define kernel testing as a service within CIP
 - Define the testing environment within CIP assuming that, in some cases, some members may share the tests, test results or laboratories while others may not.
4. From kernel testing to system testing
 - Once the testing environment has been ready and works for the kernel, explore how to extend it to the entire CIP platform.

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

③ CIP testing (2/3)

- CIP Testing project (<https://wiki.linuxfoundation.org/civilinfrastructureplatform/ciptesting>)
- B@D designed to:
 - Test Linux kernels and base systems.
 - Locally: no need of a centrally managed service.
 - On hardware connected to your dev machine.
- Latest status
 - **CIP testing environment (B@D v1.0) just released** (<https://goo.gl/4RFrJ1>)
 - Based on kernelci.org
 - Linux and Windows 10 as Host OS supported.
 - Shipped as a VM and Vagrant based environment.
 - 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.



③ CIP testing (3/3)



Next Steps

- Discussing with AGL members for ALG and CIP testing collaboration
- During the coming months the team will focus on:
 - Defining how tests should look like.
 - Defining how results should be shared.
 - Increasing the test coverage of the CIP Kernel
- More updates at Embedded Linux Conference 2018

4 CIP Core Packages (1/5)



An example of minimal package set for CIP base layer

Candidates for initial component set

- | CIP Kernel |
|--|
| <ul style="list-style-type: none">• Kernel<ul style="list-style-type: none">• Linux kernel + backported patches• PREEMPT_RT patch• Bootloader<ul style="list-style-type: none">• U-boot• Shells / Utilities<ul style="list-style-type: none">• Busybox• Base libraries<ul style="list-style-type: none">• Glibc• Tool Chain<ul style="list-style-type: none">• Binutils• GCC• Security<ul style="list-style-type: none">• OpenSSL |

Keep these packages for Reproducible build

- | Dev packages |
|--|
| <ul style="list-style-type: none">• Flex• Bison• autoconf• automake• bc• bison• Bzip2• Curl• Db• Dbus• Expat• Flex• gawk• Gdb• Git• Glib• Gmp• Gzip• gettext• Kbd• Libibverbs• Libtool• Libxml2• Mpclib• Mpfr4• Ncurses• Make• M4• pax-utils• Pciutils• Perl• pkg-config• Popt• Procps• Quilt• Readline• sysfsutils• Tar• Unifdef• Zlib |

NOTE: The maintenance effort varies considerably for different packages.

④ CIP Core Packages (2/5)



Current status of the Base layer development

1. Define an initial component set



1.5 Talk to open source communities

2. Define component version

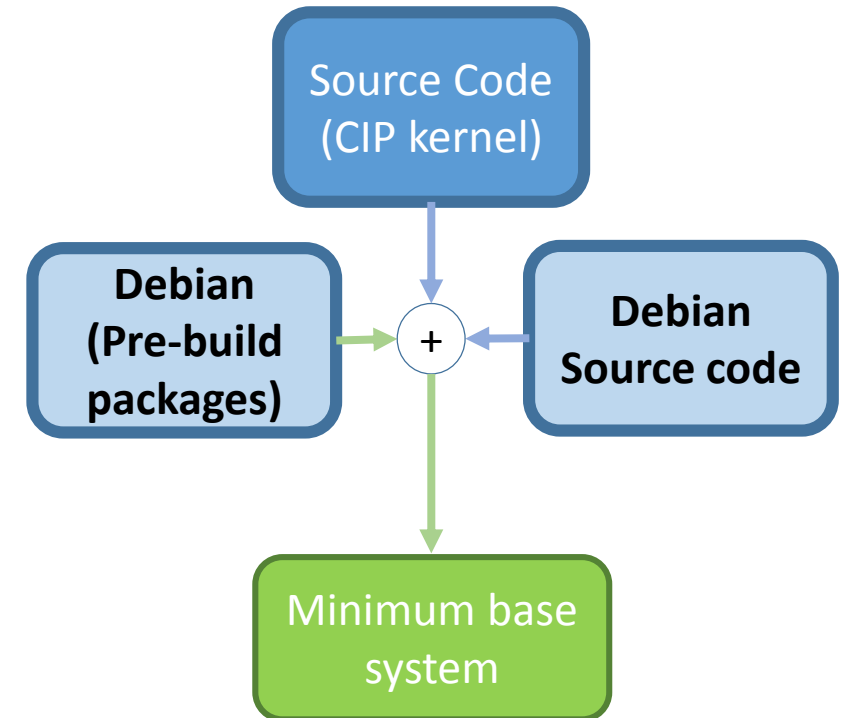
3. Contribute to upstream project

4. Start maintenance for SLTS

4 CIP Core Packages (3/5)

CIP Core

- **CIP Core is now become CIP official project**
 - CIP Core aims to provide a way to create and test installable images
- **Goal**
 - **Input:** Debian sources/binaries and cip kernel
 - **Build mechanism:** Bitbake and/or Debian build system
 - **Output:** Minimum deployable base system image for testing
- **Current status**
 - Minimal rootfs can be build for the following hardware
 - Renesas RZ/G1M (iwg20m)
 - BeagleBone Black
 - Cyclone-V
 - QEMUx86

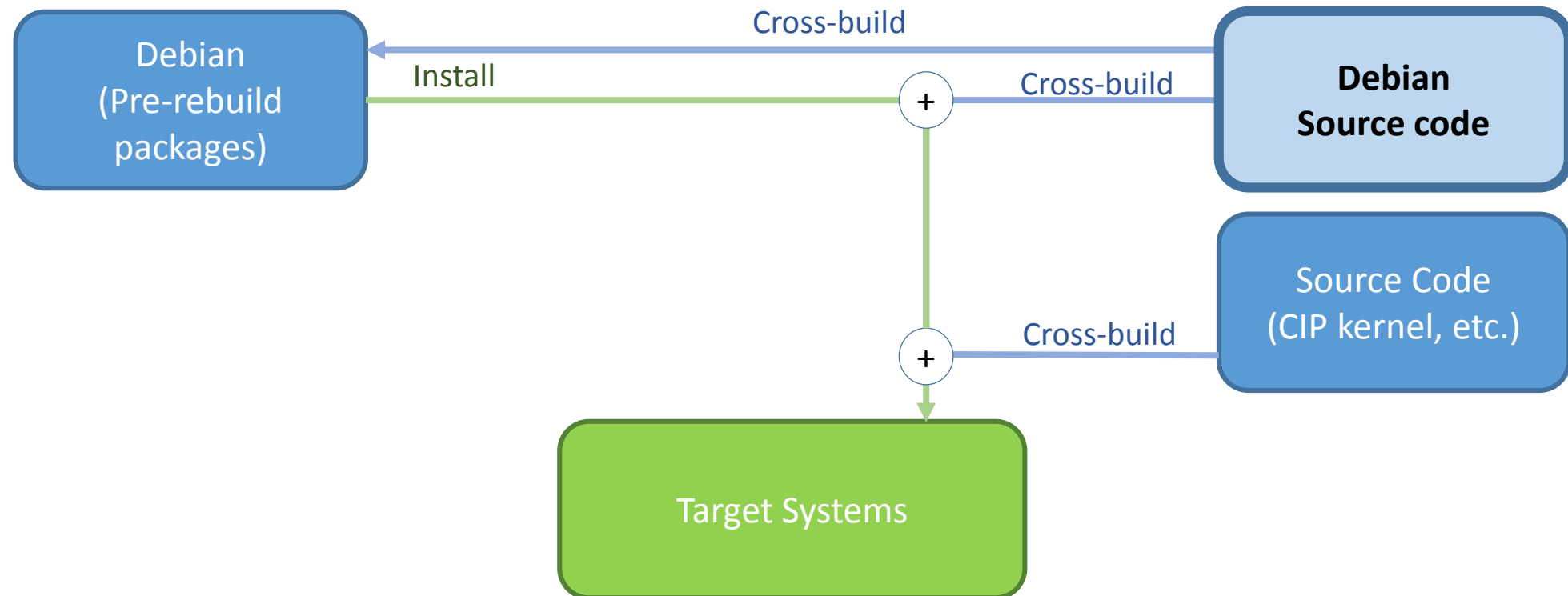


Source code: <https://gitlab.com/cip-project/cip-core>

4 CIP Core Packages (4/5)

Creating Debian-based image (Current status)

Deby: <https://github.com/meta-debian/meta-debian>



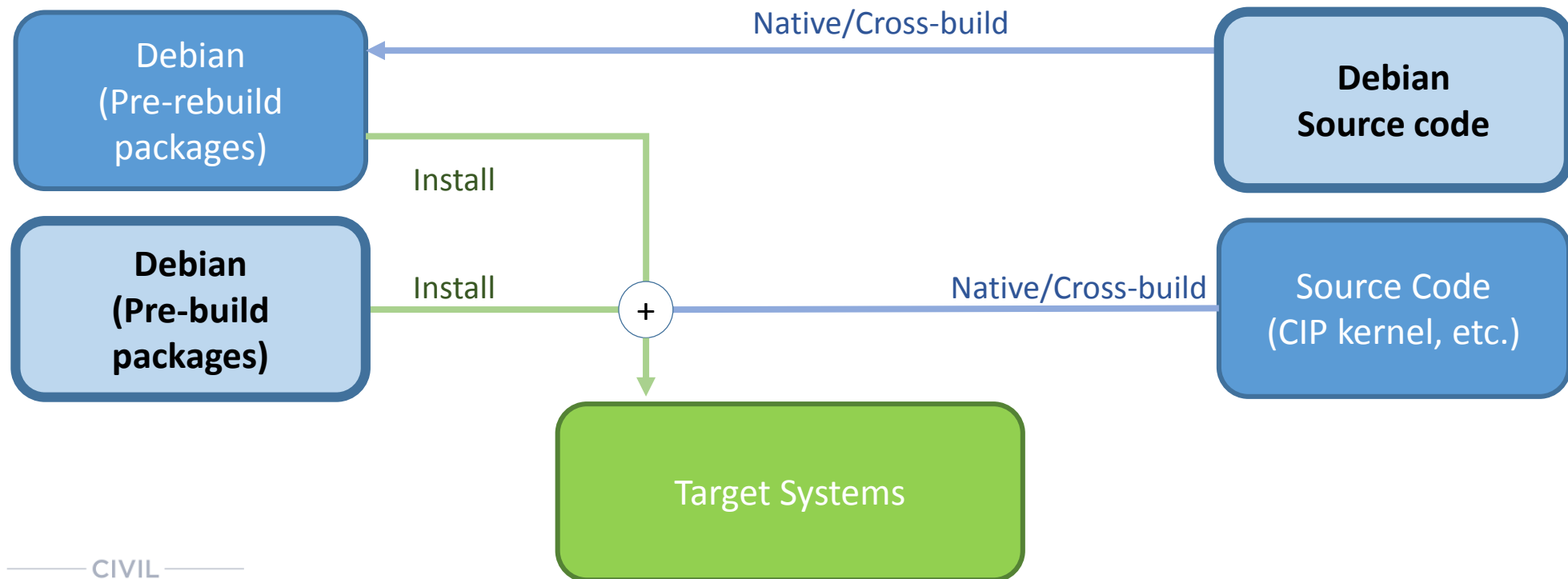
4 CIP Core Packages (5/5)



Creating Debian-based image (Other options)

ISAR: <https://github.com/ilbers/isar>

ELBE: <https://elbe-rfs.org/>



④ Potential build tools for CIP Core (Comparison Elbe, Isar and Deby)

	Elbe	Isar	Deby
Base system	Debian binary packages (no rebuilding)		Binary packages cross-built from Debian source packages
Build system	Custom	Bitbake	
Host tools	Debian: debootstrap, qemu, elbe-pbuilder	Debian: multistrap, dpkg-buildpackage, qemu	Poky
Metadata	✓ ELBE-XML for project description	<ul style="list-style-type: none"> ✓ Recipes for building product packages ✓ Recipes for image generation 	<ul style="list-style-type: none"> ✓ Common function to unpack Debian source packages ✓ Full recipes for cross-building every Debian source package
Compilation	Native		Cross
Benefits	<ul style="list-style-type: none"> ✓ Re-use Debian binaries and QA ✓ Fast (re-use, parallel builds) ✓ Lower development costs 		<ul style="list-style-type: none"> ✓ Affinity with Poky recipes ✓ Fully customizability ✓ No need to keep binary pkgs
Common features	<ul style="list-style-type: none"> ✓ Based on Debian packages (stability, long-term maintenance) ✓ Generate images by installing binary packages ✓ Manage multiple products as a custom setting (layers or configuration files) 		

http://events.linuxfoundation.jp/sites/events/files/slides/ISAR-DEBY-OSSJ2017_r10.pdf

Summary and conclusion



- **The CIP Open Source Base Layer of industrial-grade software materializes**
- CIP today focusses on
 - **Kernel maintenance:** maintaining Linux kernels for very long time (+15 years) 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

Conclusion



- **Our Civilization needs an Open Source Base Layer of industrial-grade software**
- CIP provides this, based on Linux
- Sustainability is ensured by
 - The backing of big industrial and semiconductor companies
 - Close cooperation with and build on mature Open Source projects (Debian, PREEMPT_RT, kernelci, ...)
 - Providing elaborated tool chains
 - Ensuring in-depth tests
- CIP gets traction in the member companies

Contact Information and Resources



To get the latest information, please contact:

- CIP Mailing list: 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/bwh/linux-cip.git>

CIP whitepaper

- Year One Update + Whitepaper Release
 - <https://www.cip-project.org/blog/2017/05/31/cip-year-one-update-whitepaper-release>
- Everyone can download the whitepaper
 - https://wiki.linuxfoundation.org/_media/civilinfrastructureplatform/whitepaper_short.pdf



CIP developers meeting/gathering at ELCE



- CIP developers meeting/gathering at ELCE which it will be immediately after the talk. Link:
<https://wiki.linuxfoundation.org/civilinfrastructureplatform/cipconferences/cipwselce2017#cip-developers-meeting-gathering>
- Date & Time: October 24th, 13:00-14:00
- Room: Liben Room, Mezzanine Level



Thank you!



Questions?