



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

Industrial-grade Open Source Base Layer

Yoshitake Kobayashi, Toshiba Corporation
Embedded Linux Conference North America,
March 12-14, 2018

What is CIP?

What is CIP?



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

What is CIP?



- One of the most conservative open source projects 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 runs on Linux

Transport



Rail automation



Vehicle control



Automatic ticket gates

Energy

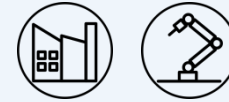


Power Generation



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 hills are visible in the distance.

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

Power Plant Control Example

3 – 5 years development time

0.5 – 4 years customer specific extensions

6 – 8 years supply time

15+ years hardware maintenance after latest shipment

20 – 60 years product lifetime

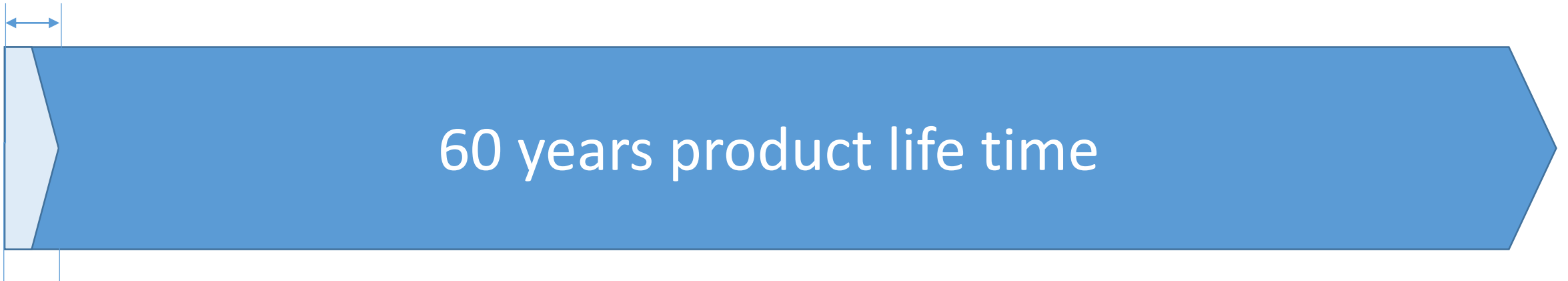
2 years

**CIP started from
April 2016**

Possible product life time and CIP current age.



2 years



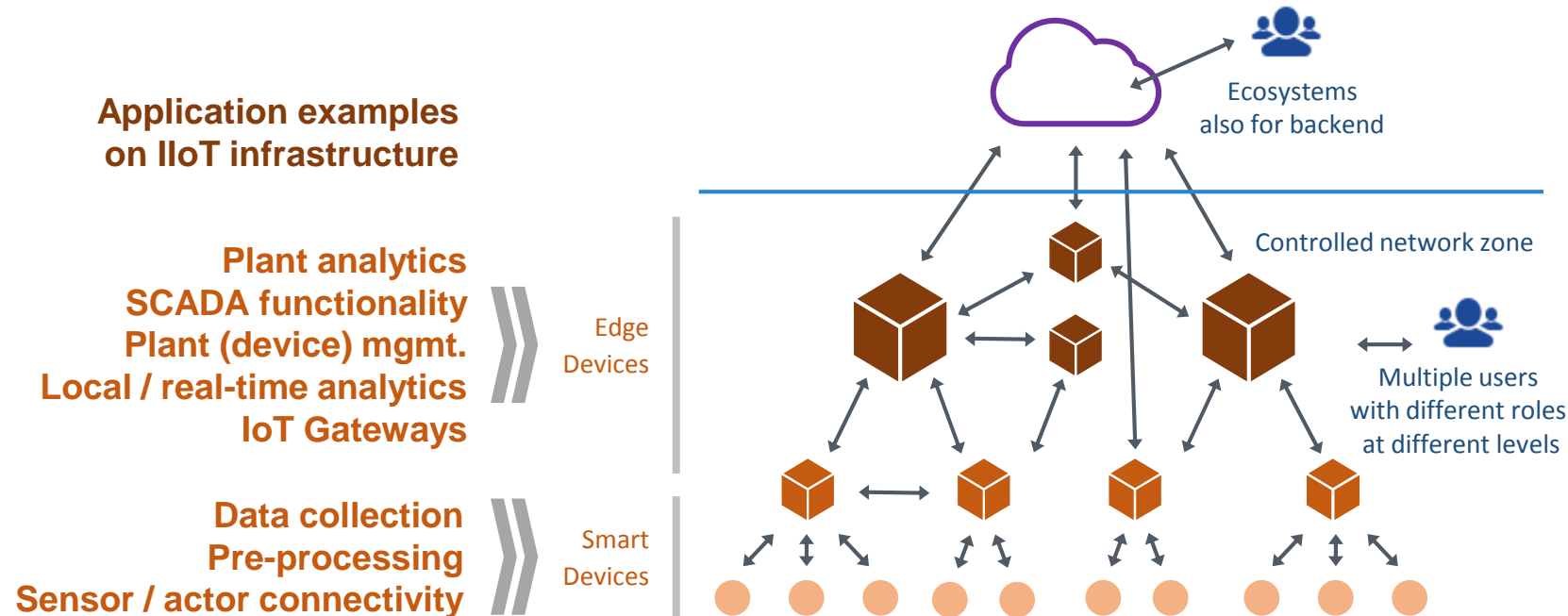
Things change a lot during this product life time

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



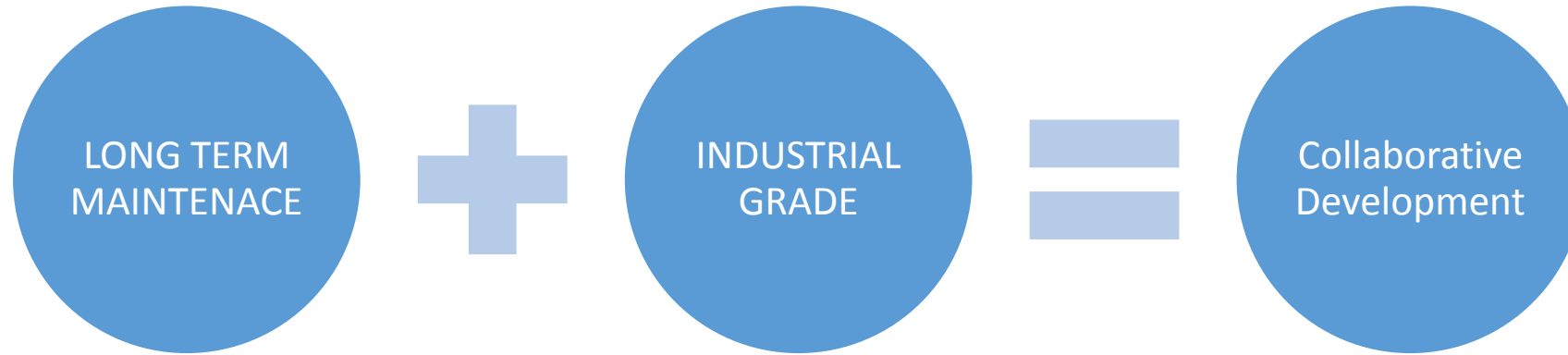
IIoT: Industrial IoT **SCADA:** Supervisory Control And Data Acquisition

The 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 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

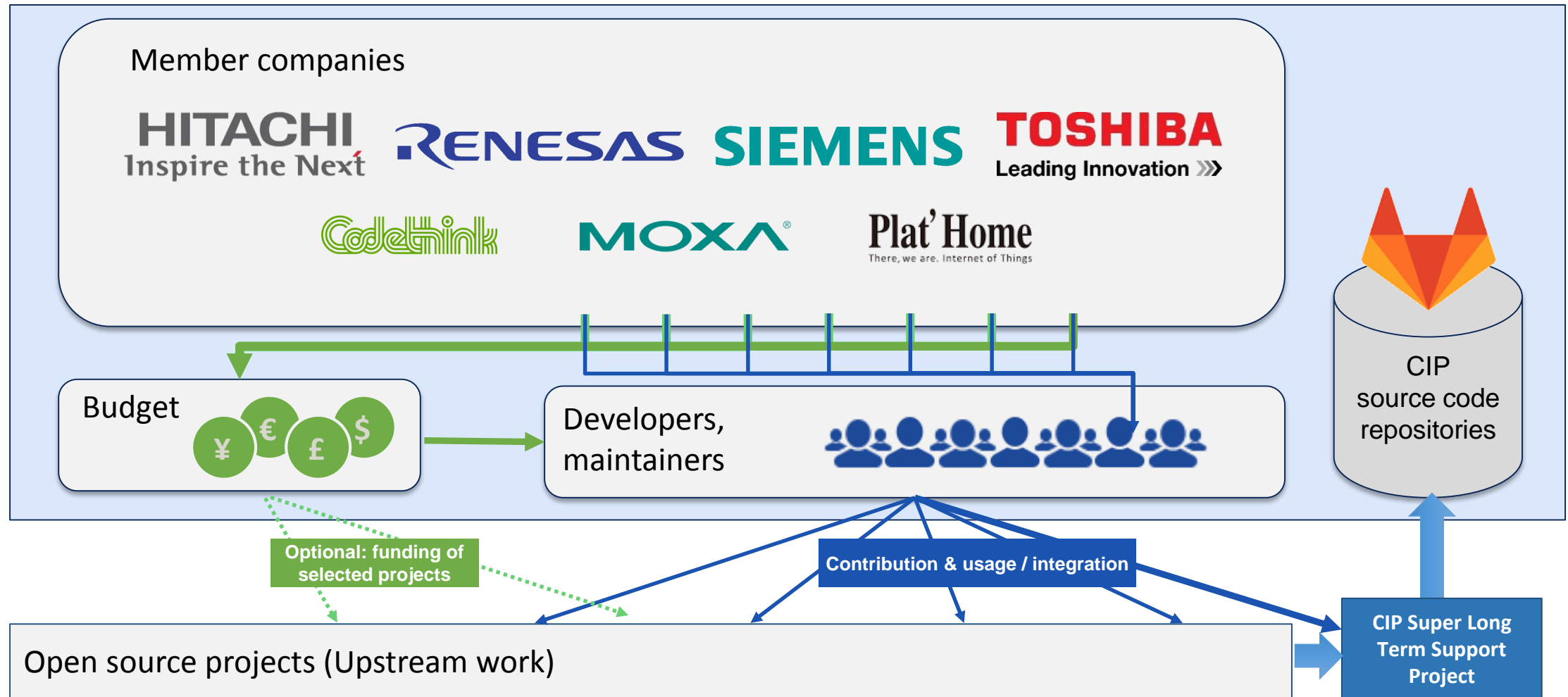
<https://www.cip-project.org/>



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

since April 2016

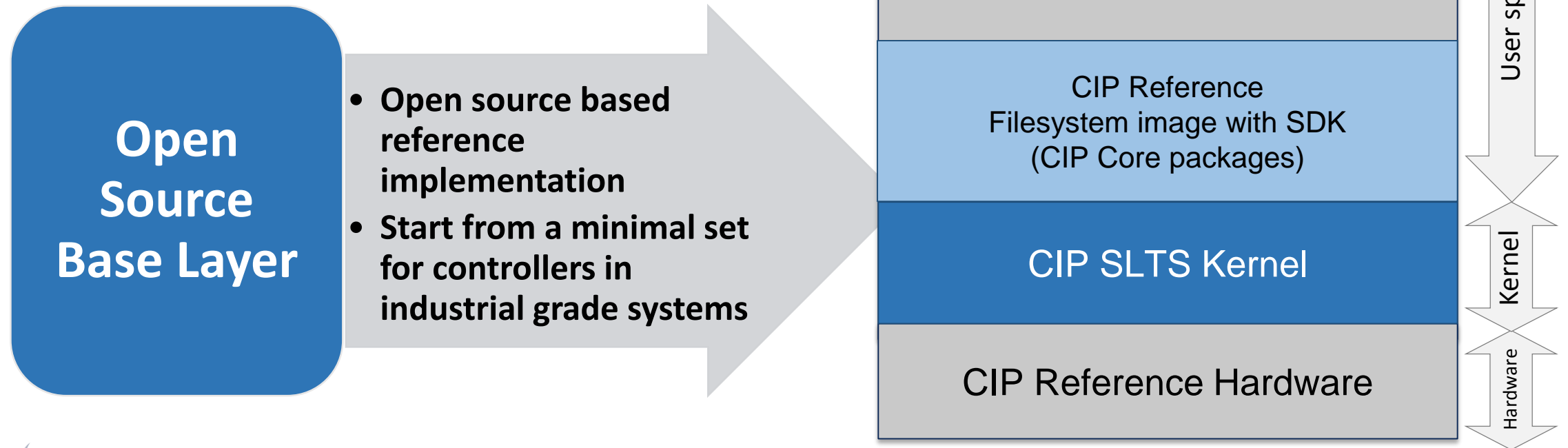
The backbone of CIP are the member companies



What is CIP, again?

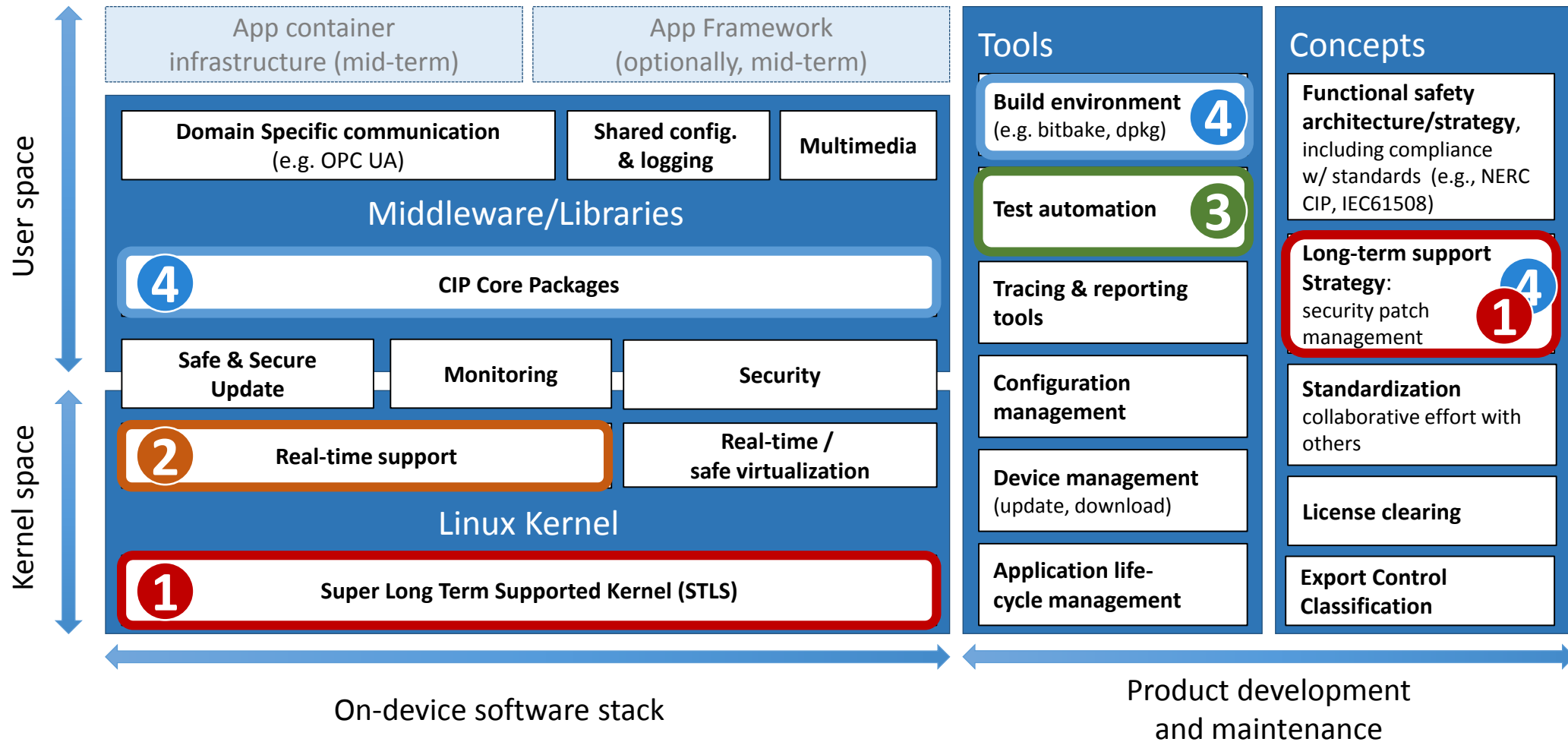
What is “Open Source Base Layer (OSBL)”?

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



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. 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 continuity of 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.

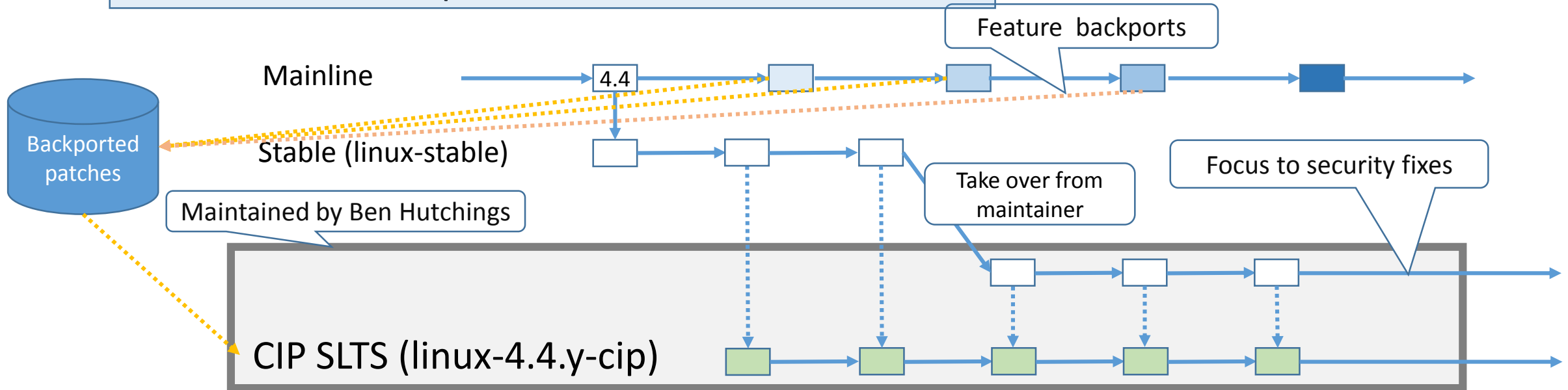
4 CIP Core

- 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

- 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, from Codethink
- Linux 4.4.120-cip20 released on 9th March 2018



1 CIP SLTS Kernel development (2/5)



- What's new in CIP kernel
 - Ensuring Meltdown and Spectre fixes land in the CIP kernel
 - Platform specific backported patches, especially for Renesas RZ/G platforms.
- Other 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 Siemens IoT2000 series

1 CIP SLTS Kernel development (3/5)



- Kernel maintenance policy
 - <https://wiki.linuxfoundation.org/civilinfrastructureplatform/cipkernelmaintenance>
 - Follow the stable kernel development rule as the basis
 - Feature backports are acceptable
 - **CIP has “Upstream first” policy**
 - All features has to be in upstream kernel before backport to CIP kernel
 - Validation will be done by CIP test infrastructure and/or members
- CIP kernel release period
 - Ben Hutchings, from Codethink, releases a kernel version approx. every 4 to 6 weeks. Depends on the amount and relevance of upstream patches
 - Collaborates in the 4.4 LTS process directly

1 CIP SLTS Kernel development (4/5)



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 (5/5)

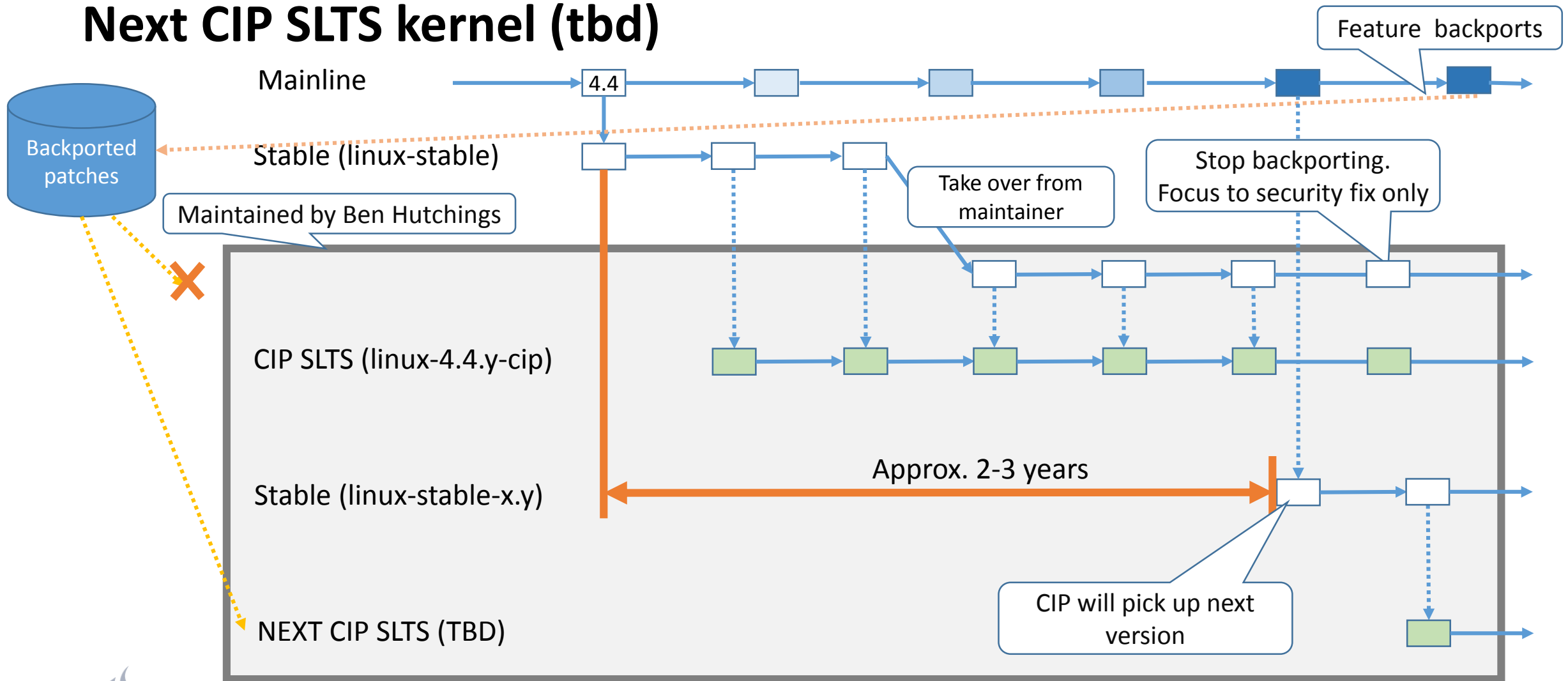


Out-of-tree drivers

- 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 next SLTS Kernel version

Next CIP SLTS kernel (tbd)



1 CIP next SLTS Kernel version



- It's time to consider the next SLTS kernel version
- Kernel version alignment with other project is a key
 - LTS / LTSI / AGL / Debian and more
- Planning to have a F2F meeting during Open Source Summit Japan with those project and stable maintainers

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

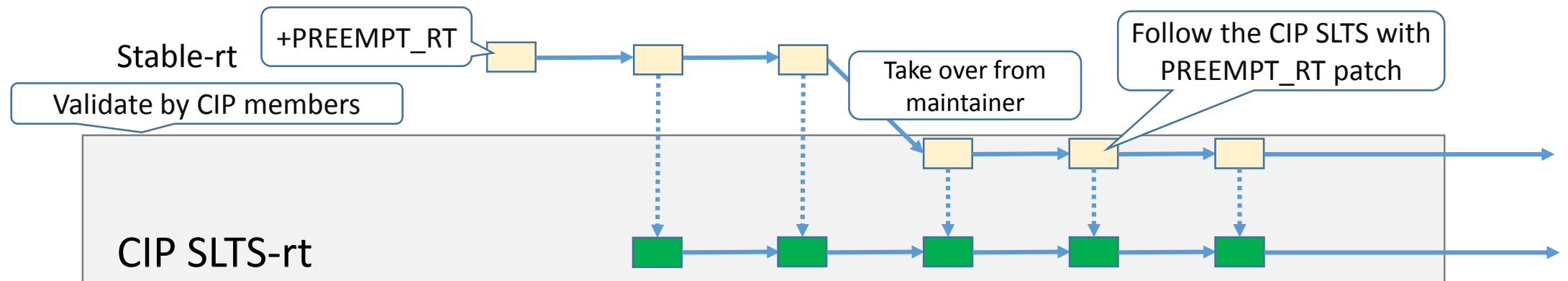


- CIP is a Gold Member of the Real Time Linux Project
 - Work together with the RTL Project
 - Daniel Wagner from Siemens is working to become the maintainer of 4.4.y-stable-rt, the base version of the CIP Kernel.
- More information
 - <https://wiki.linuxfoundation.org/realtime/rtl/start>

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

CIP SLTS Real time kernel is available

- CIP kernel tree based on linux-stable-rt and patches from CIP SLTS
 - <https://git.kernel.org/pub/scm/linux/kernel/git/wagi/linux-cip-rt.git>
 - Maintainer: Daniel Wagner, from Siemens AG
- Validated by CIP



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 selected CIP hardware platforms 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 is ready and works for the kernel, explore how to extend it to the entire CIP platform.

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

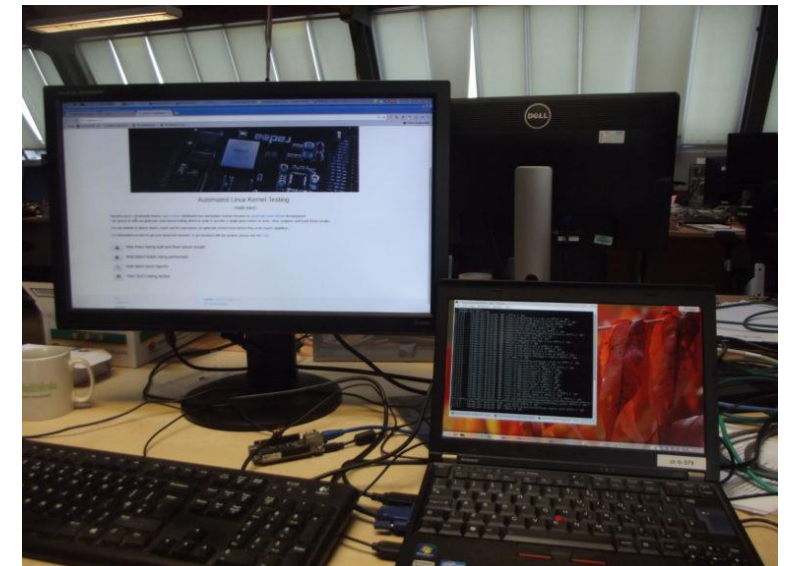
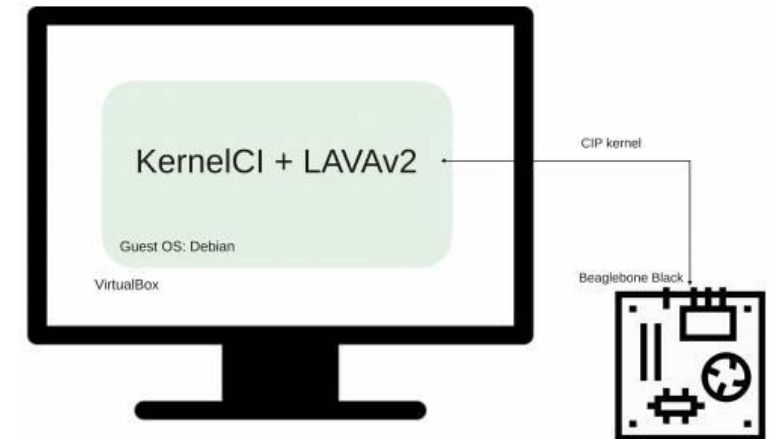
③ CIP testing (2/3)

CIP Testing project

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

- B@D designed to:
 - Test Linux kernels and base systems locally.
 - 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.



③ The latest status for CIP testing (3/3)



1. Updated to include the latest kernelCI
 2. Supporting Renesas iw20gm
 3. Shared testing results
 - <https://lists.cip-project.org/pipermail/cip-testing-results/>
- **Next steps**
 - Deployment through containers.
 - Collaboration with other testing effort
 - CIP had a meeting with AGL members for testing collaboration
 - Increasing the test coverage of the CIP Kernel

④ CIP Core Packages (1/6)



Current status of the Base layer development

1. Define an initial component set
2. Define component version
3. Contribute to upstream projects
4. Start maintenance for SLTS

④ CIP Core Packages (2/6)



Debian as CIP primary reference distribution

- What does the primary distribution means?
 - CIP will select CIP Core package from Debian packages
 - CIP would like to work with Debian community
- Status
 - Supporting to solving security related issues
 - Adding support for staging repositories to security-master
 - <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=817286>
 - Status: Paperwork is ongoing
 - CIP Governing Board approved to support **Debian-LTS as Platinum level**
 - Status: Just started

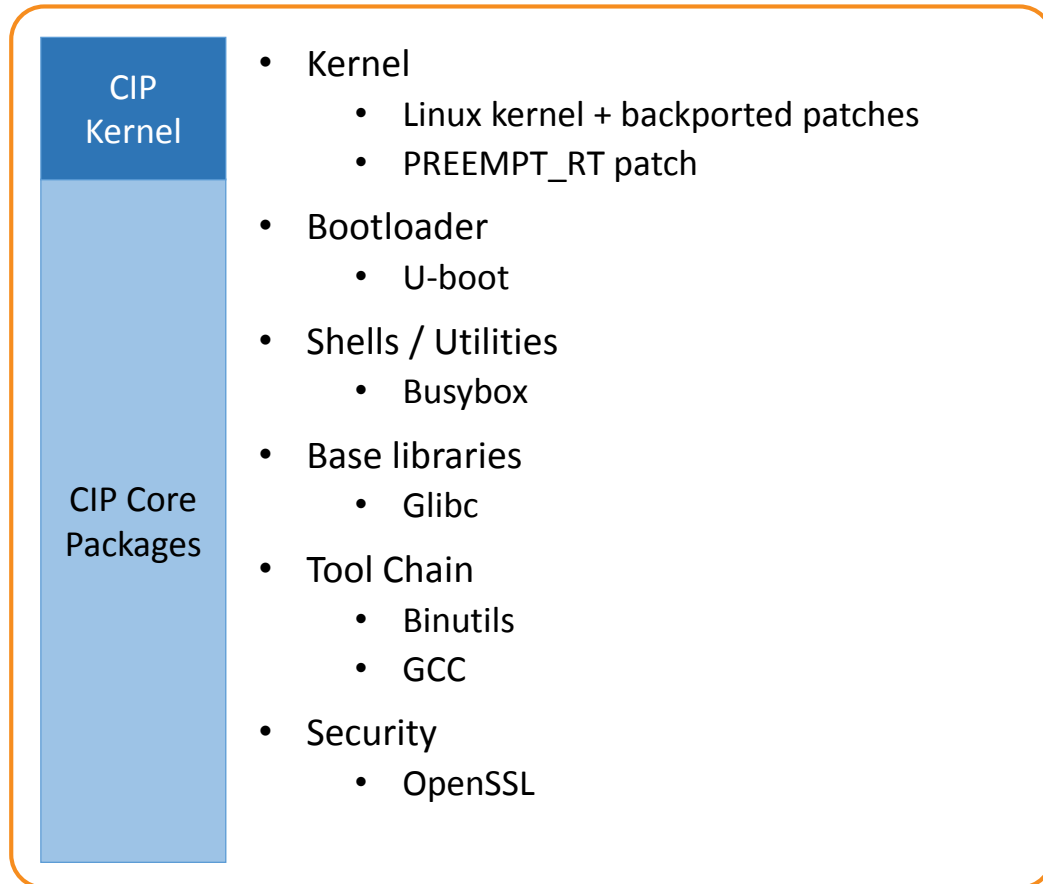


4 Initial focus for CIP Core Packages (3/6)

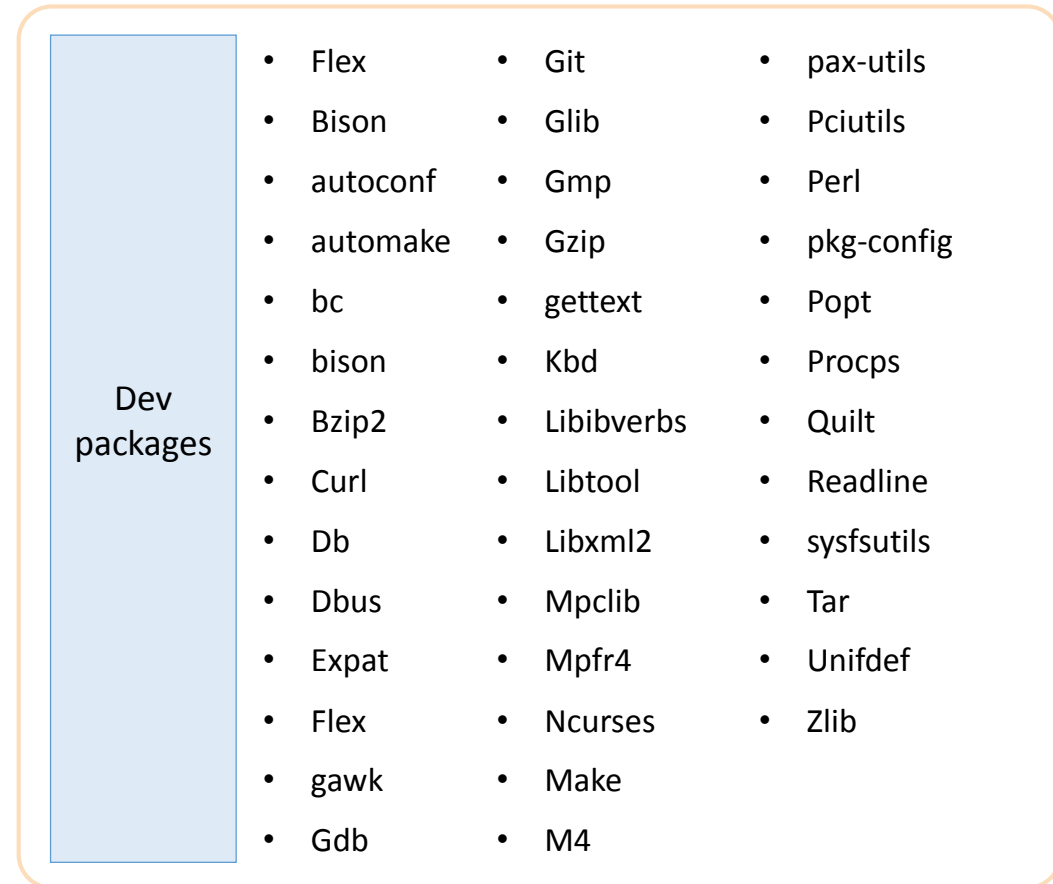


An example of minimal package set for CIP base layer

Candidates for initial component set



Keep these packages for Reproducible build

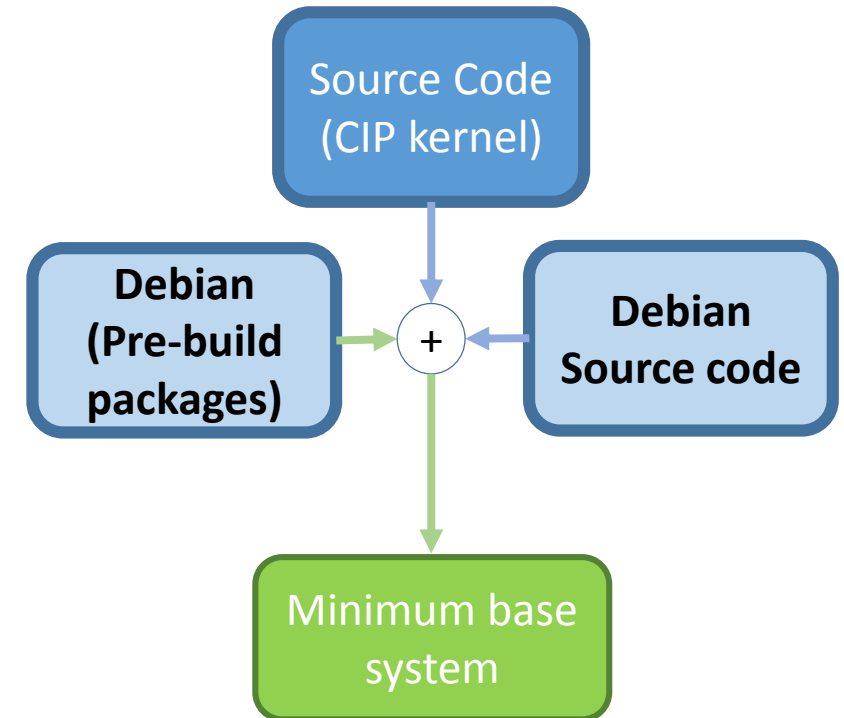


NOTE: The maintenance effort varies considerably for different packages.

4 CIP Core Packages (4/6)

CIP Core

- **CIP Core is a 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 built for the following hardware
 - Renesas RZ/G1M (iwg20m)
 - BeagleBone Black
 - Cyclone-V
 - QEMUx86

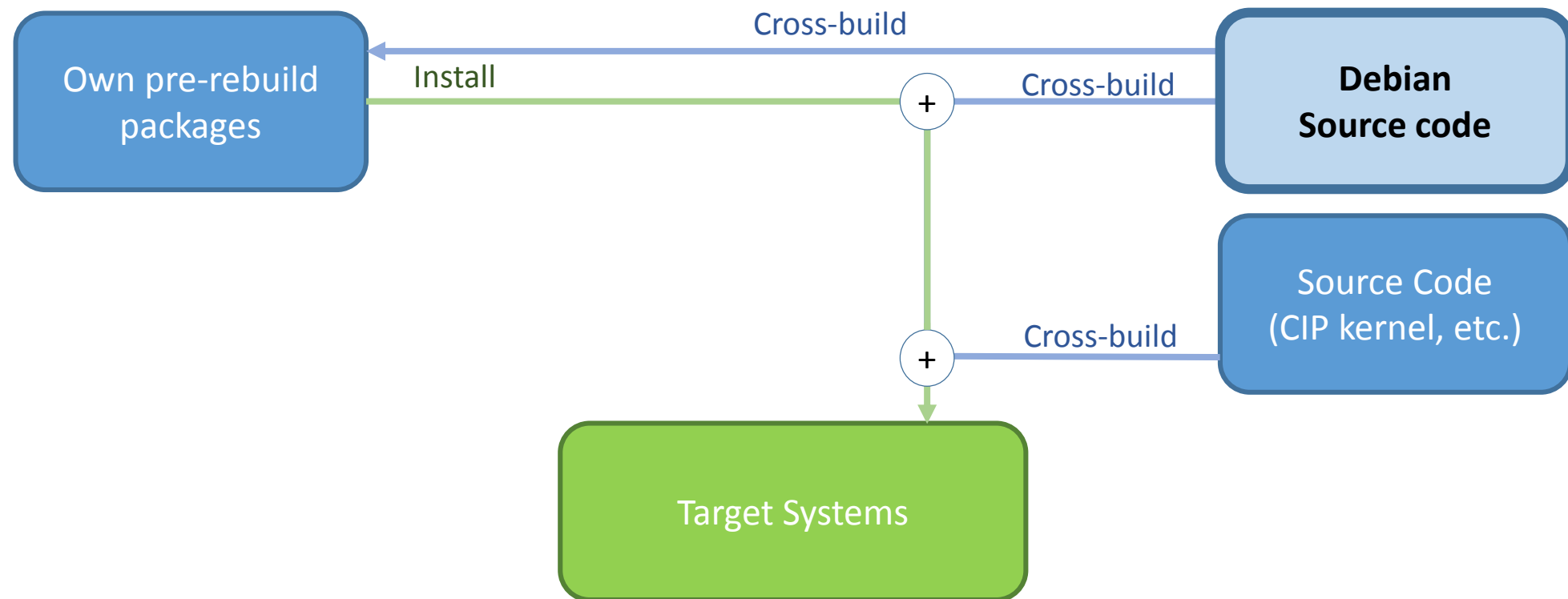


Source code: <https://gitlab.com/cip-project/cip-core>

4 CIP Core Packages (5/6)

Creating Debian-based image (Currently supported)

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



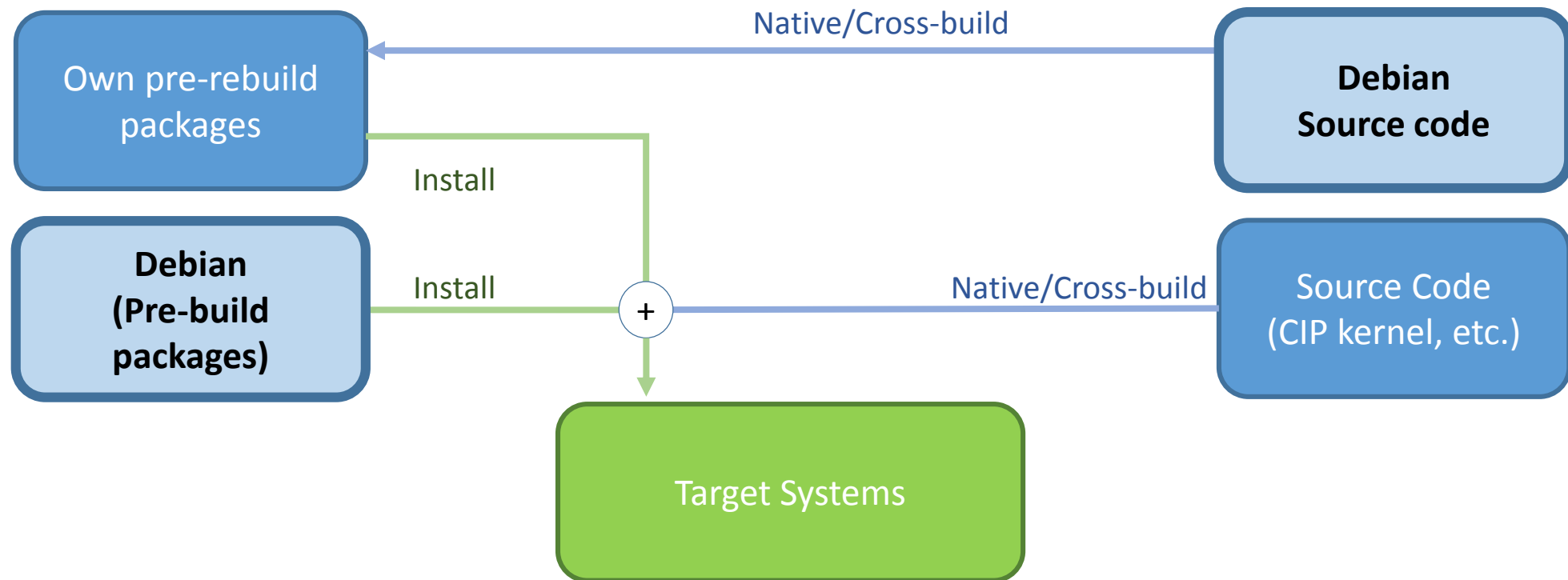
4 CIP Core Packages (6/6)



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 customizable ✓ 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 using custom configuration 		

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

4 Gaps and Common Goals between Debian and CIP



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

4 Other concern with CIP Core



- CIP members are also interested in Yocto Project as a build tool
 - Using Poky / Bitbake as buildtool to customize rootfs
 - CIP might create a meta-cip layer
 - Users can get SLTS benefit from CIP Core packages
 - Other OE-layers could be extend CIP Core (Will not SLTS by CIP)

Under discussion in CIP

Cybersecurity Standard for Industry (IEC62443)



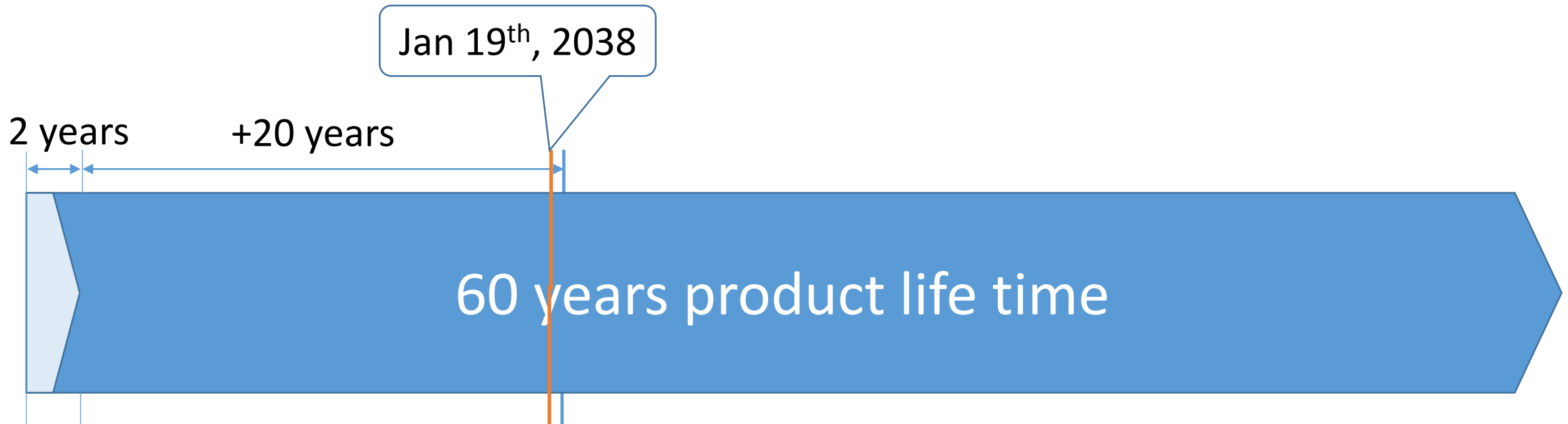
- **ISA/IEC-62443**

- Specifications to develop secure Industrial Automation and Control Systems
 - <http://www.isasecure.org/en-US/Certification>
- Certification requirements include
 - Development procedure
 - Tests
 - Certification scheme

- **How CIP is involved**

- Development procedures and certification schemes are out of scope
- Developing the following items which may help to get a certification with Linux based system (also CIP based system)
 - Documents for recommended settings and configuration for open source packages
 - Test cases and tools

Q: 20 years from now?



- Y2038 is a blocking issue for 20 year commitments
- CIP members had a meeting during ELC-EU with key developers

Other items currently under discussion in CIP



- Functional safety
- Software updates for industrial systems
- CIP joined EdgeX Foundry as Associate member



Summary and conclusion

Summary



- **The CIP Open Source Base Layer of industrial-grade software has materialized**
- 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:** Debian/Debian-LTS, Real Time Linux, EdgeX Foundry

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
- Collaboration is one of the key CIP activities



Thank you!



Questions?

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>