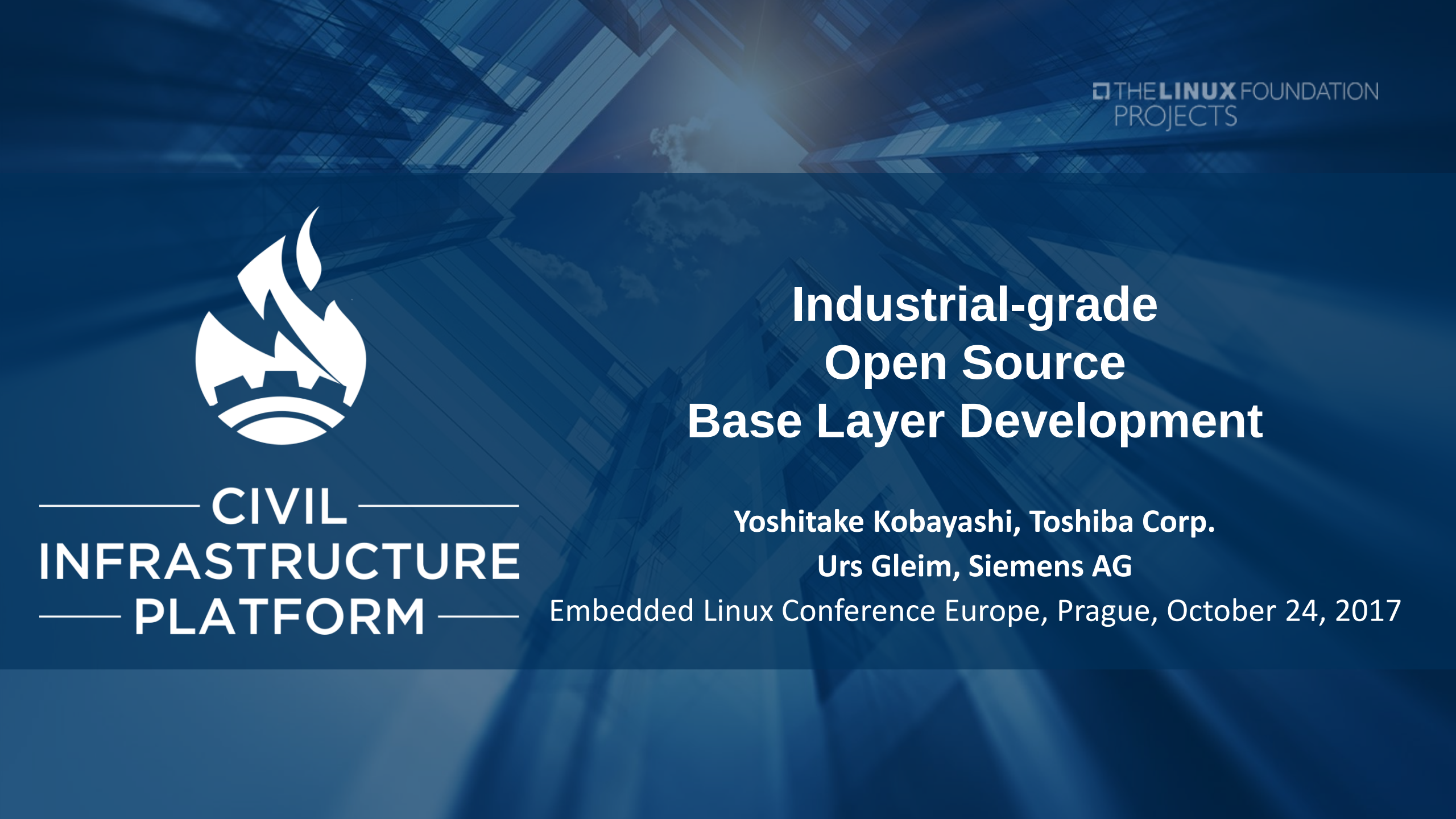


Industrial-grade Open Source Base Layer Development

Yoshitake Kobayashi, Toshiba Corp.
Urs Gleim, Siemens AG

Embedded Linux Conference Europe, Prague, October 24, 2017

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

CIVIL INFRASTRUCTURE PLATFORM

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

CIVIL INFRASTRUCTURE PLATFORM

Our Civilization is run by Linux

https://www.airpano.com/360Degree-VirtualTour.php?3D=San-Francisco-USA

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

https://www.airpano.com/360Degree-VirtualTour.php?3D=San-Francisco-USA

There are issues to be solved…

https://www.airpano.com/360Degree-VirtualTour.php?3D=San-Francisco-USA

A Railway System:

25-50 years products life-cycle

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

Image: http://www.deutschebahn.com/contentblob/10862328/20160301+Stw+M%C3%BClheim+Innenansicht+1+(1)/data.jpg

Railway Example

3 – 5 years development time

2 – 4 years customer specific extensions

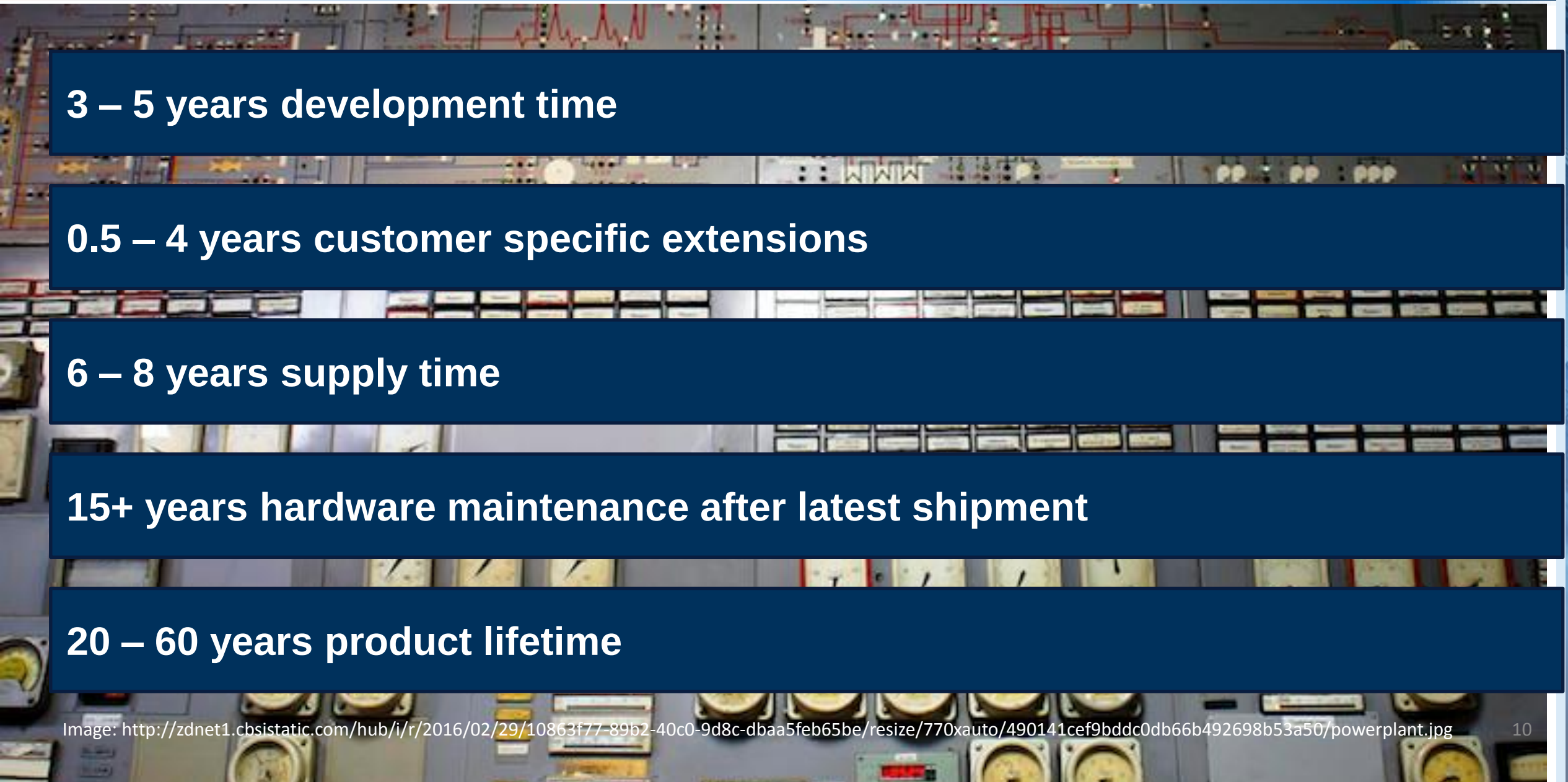
1 year initial safety certifications / authorization

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

25 – 50 years lifetime

Image: http://www.deutschebahn.com/contentblob/10862328/20160301+Stw+M%C3%BClheim+Innenansicht+1+(1)/data.jpg

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

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

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**

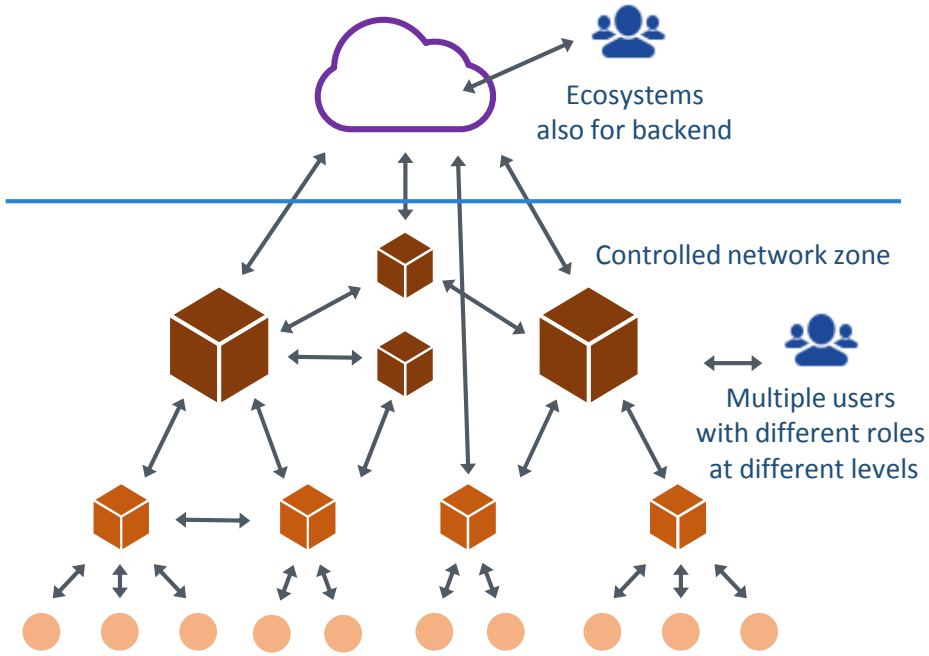
Application examples on IIoT infrastructure

Plant analytics
SCADA functionality
Plant (device) mgmt.
Local / real-time analytics
IoT Gateways

Edge Devices

Data collection
Pre-processing
Sensor / actor connectivity

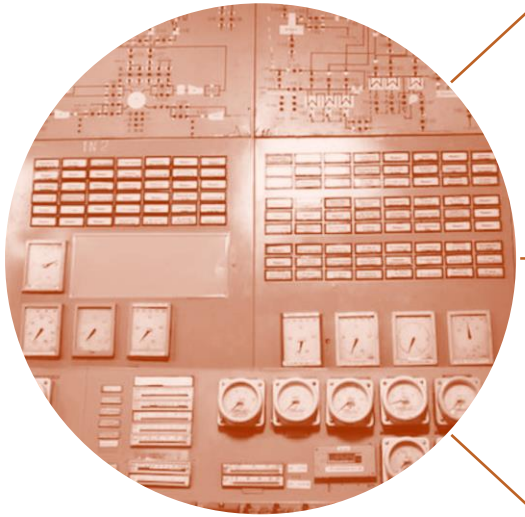
Smart Devices



CIVIL INFRASTRUCTURE PLATFORM

IoT: Internet of Things **IIoT:** Industrial IoT **SCADA:** Supervisory Control And Data Acquisition

Requirements for the Civil infrastructure systems



Industrial Grade

- Reliability
- Functional Safety
- Security
- Real-time capabilities

Sustainability

- Product life-cycles of 10 – 60 years

Conservative Upgrade/Update Strategy

- Firmware updates only if industrial-gradeness is jeopardized
- Minimize the risk of regressions
- Keeping regression test and certification efforts low

This has to be achieve with …

Maintenance costs

- Low maintenance costs for commonly uses software components
- Low commissioning and update costs

Development costs

- Don‘t re-invent the wheel

Development time

- Shorter development times for more complex systems

CIVIL INFRASTRUCTURE PLATFORM

The Problems we face …

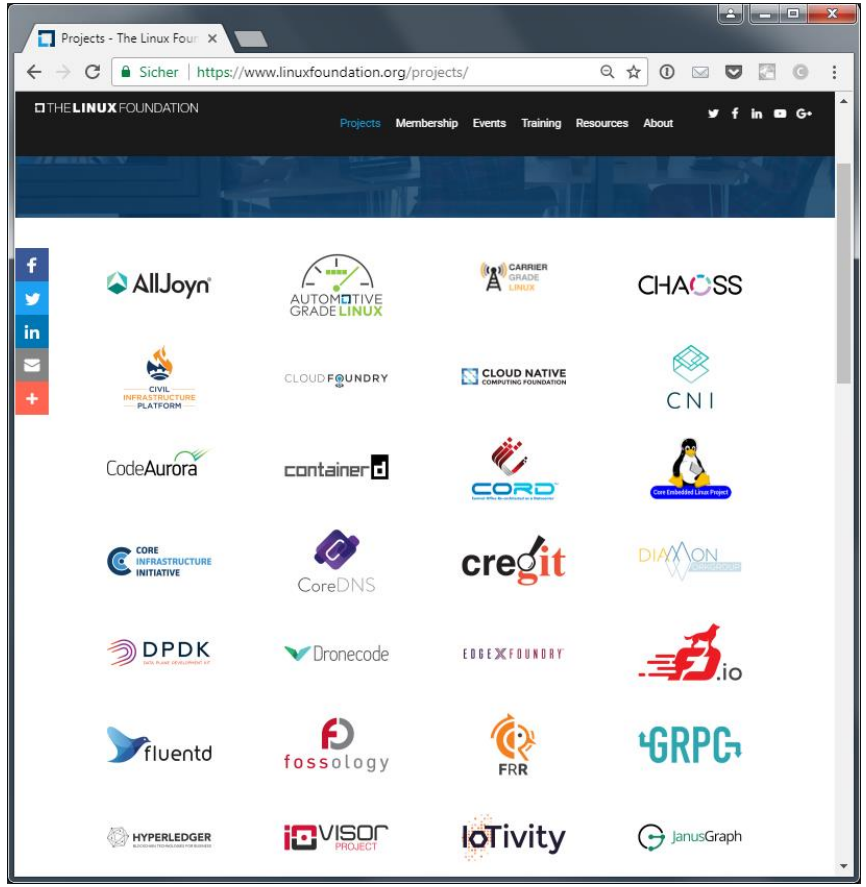
- The systems that support our modern civilization need to **survive for a VERY LONG TIME.** Until now the corresponding industrial grade super long term maintenance has been **done individually by each company**.

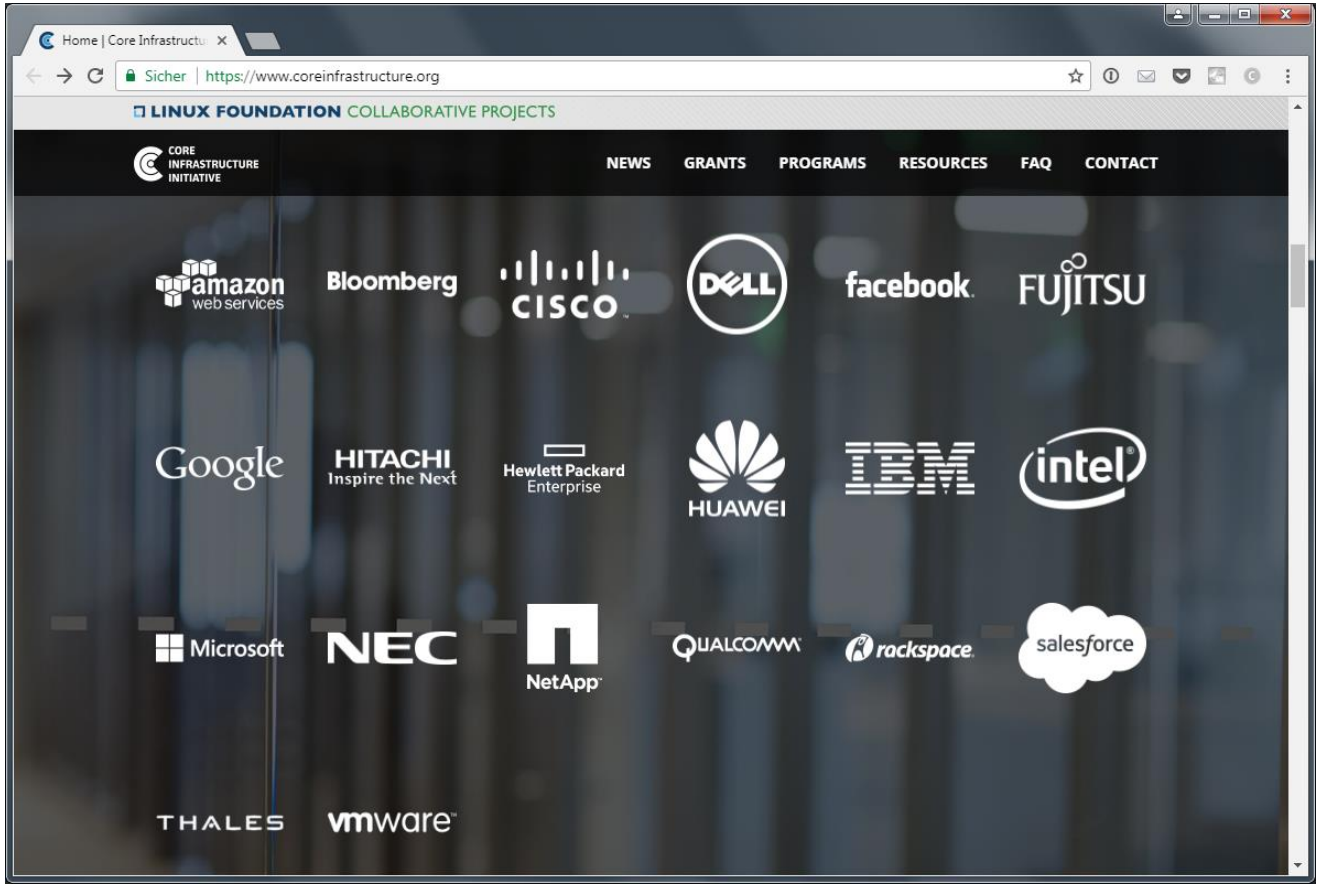
- These systems not only have to survive for a long time, they must be "**INDUSTRIAL GRADE**" (robust, secure and reliable). And at the same time the industry will also need to **catch up with the latest technology trends**

The genesis of a collaborative project

Linux Foundation Projects

Driving joint efforts and backing them with people and budget.

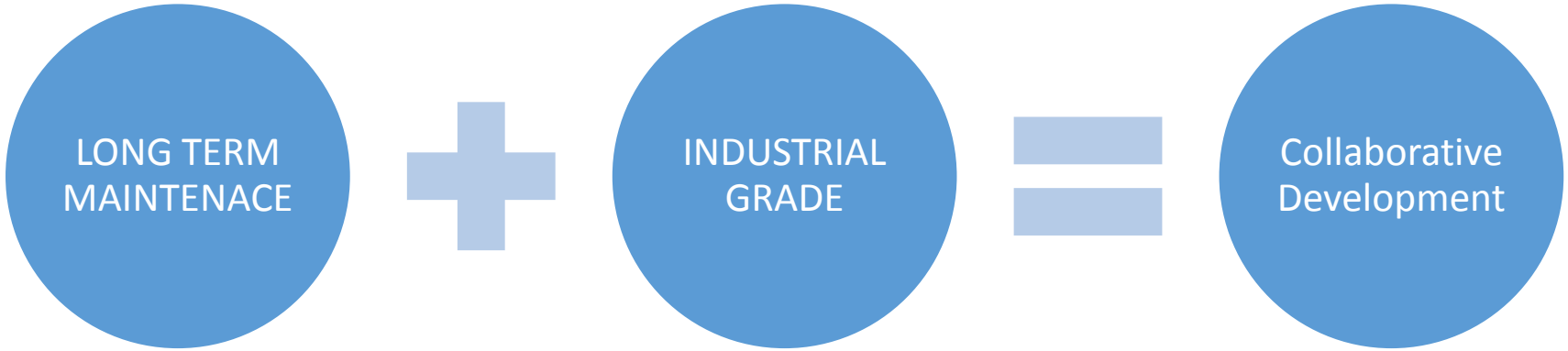




CIVIL INFRASTRUCTURE PLATFORM

The majority focusses in IT, enterprise, cloud technologies.

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…

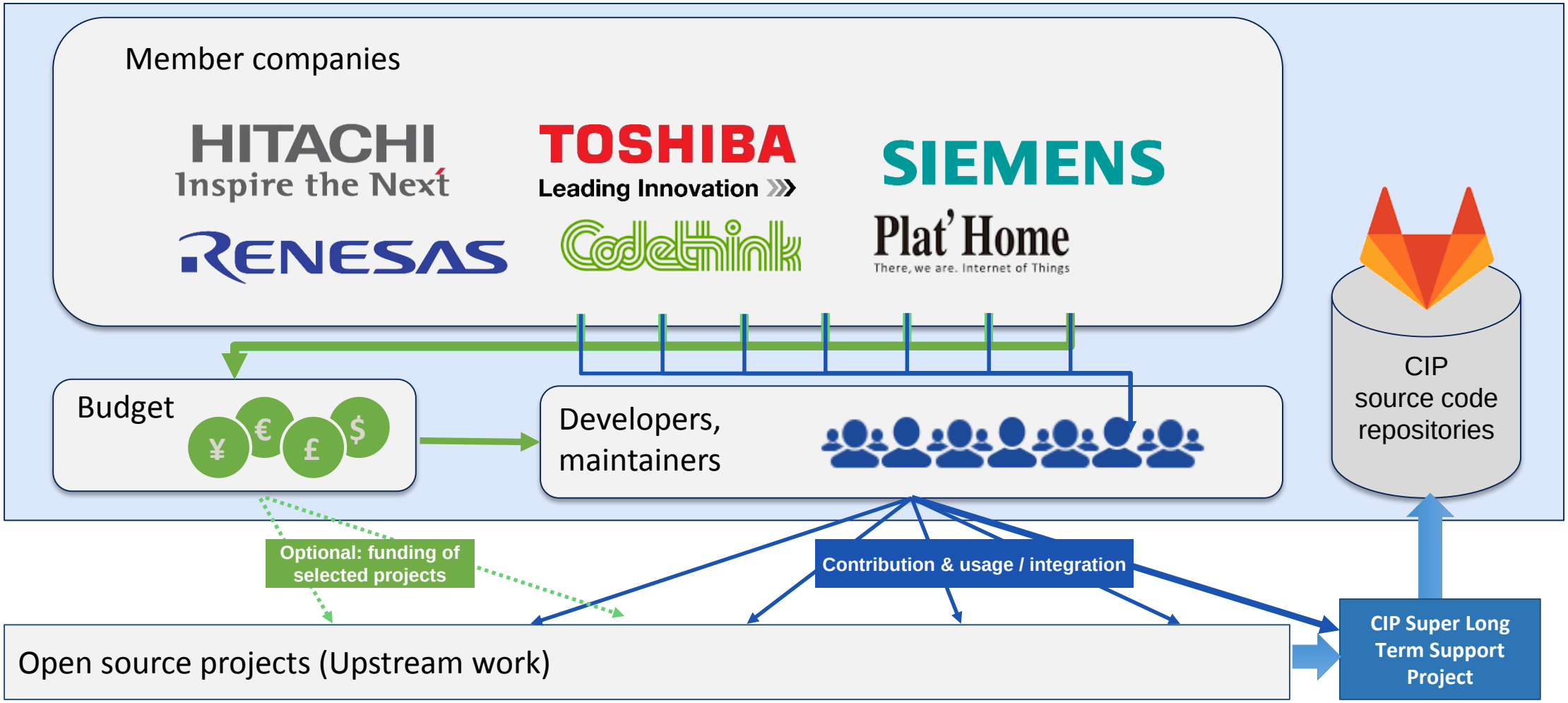
Establishing an Open Source Base Layer of industrial-grade software to enable the use and implementation of software building blocks for Civil Infrastructure Systems

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

CIVIL INFRASTRUCTURE PLATFORM

since April 2016

The backbone of CIP are the member companies



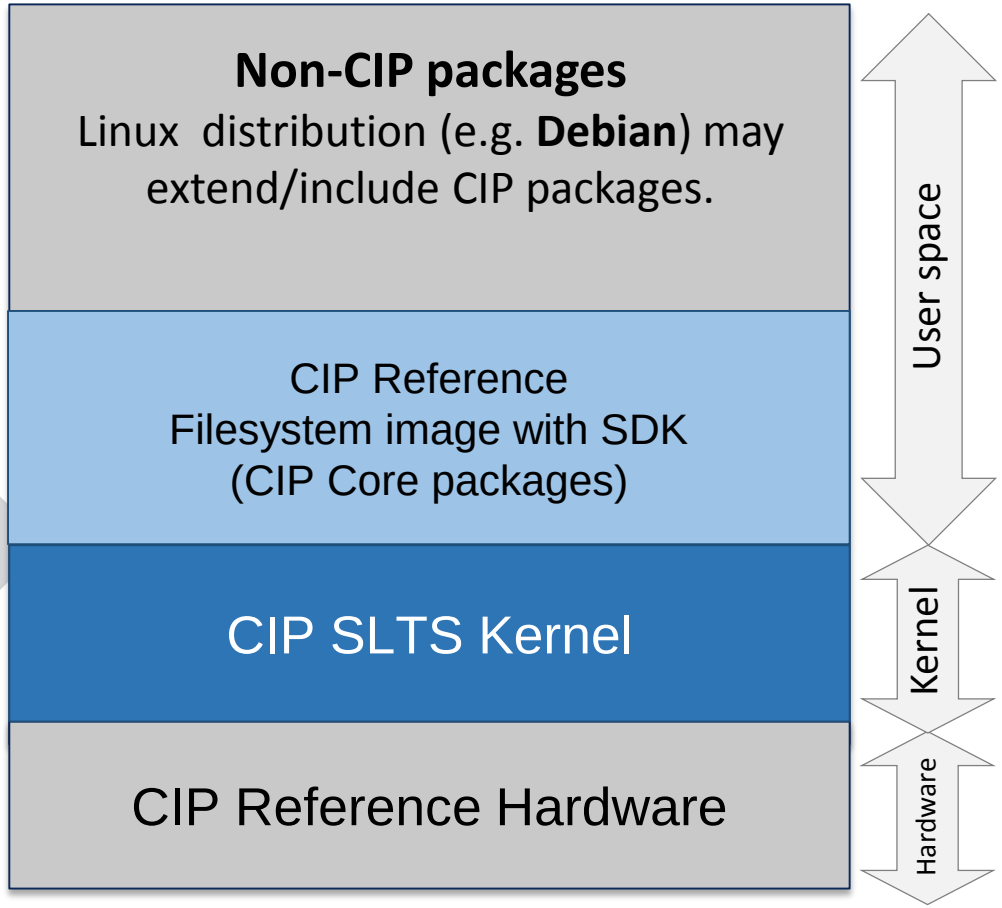
What is CIP, again?

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

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

Open Source Base Layer

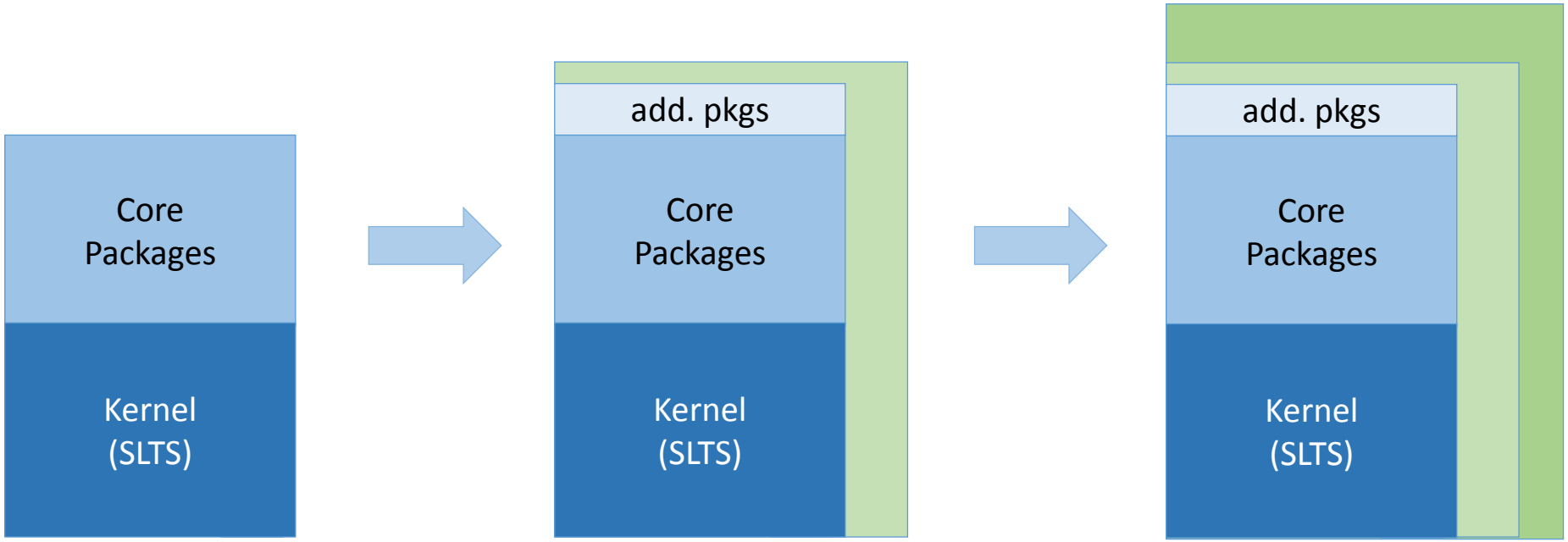
- **Open source based reference implementation**
- **Start from a minimal set for controllers in industrial grade systems**



CIVIL INFRASTRUCTURE PLATFORM

Development plan

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



Phase 1:

- Define supported kernel subsystems, arch.
- Initial SLTS component selection
- Select SLTS versions
- Set-up maintenance infrastructure (build, test)

Phase 2:

- Patch collection, stabilization, back port of patches for CIP kernel packages
- Support more subsystems
- Additional core packages

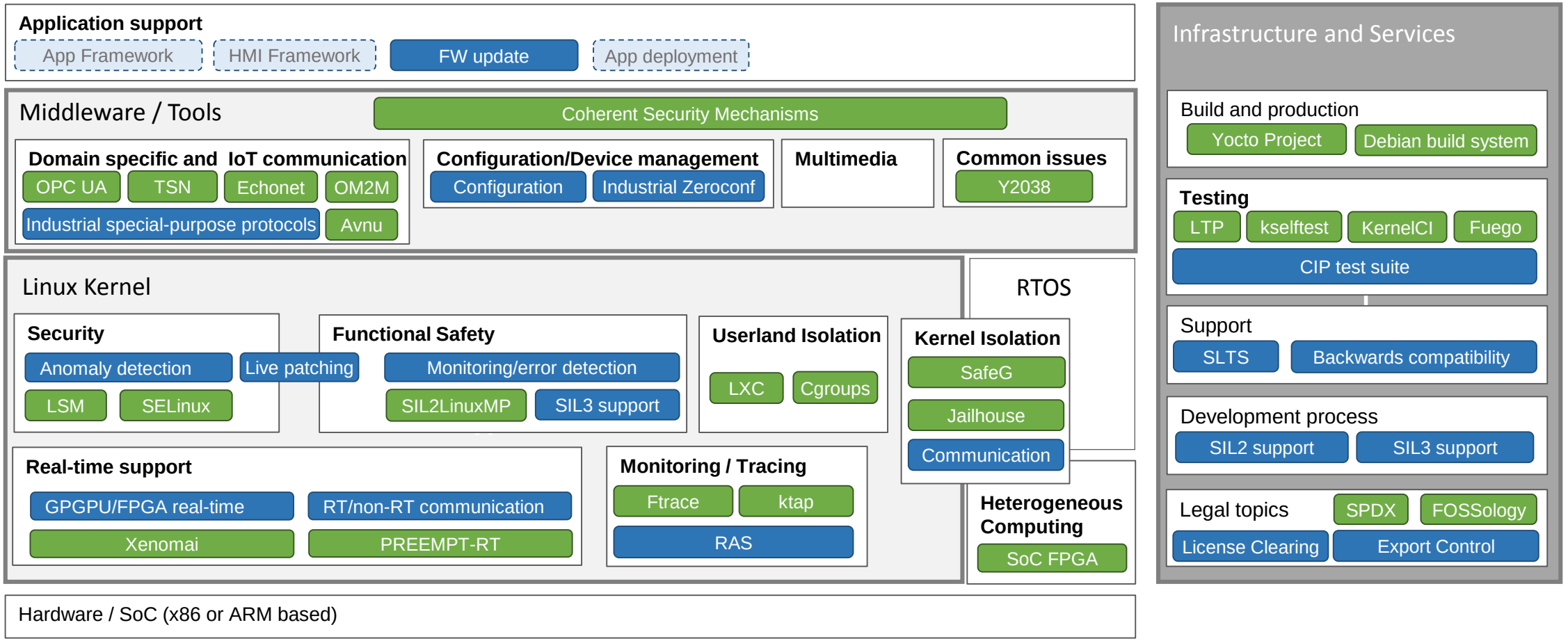
Phase 3:

- Domain specific enhancements, e.g. communication protocols, industrial IoT middleware
- Optionally: more subystems
- Optionally: more core packages

CIVIL INFRASTRUCTURE PLATFORM

Vision: Technical topics and related projects

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



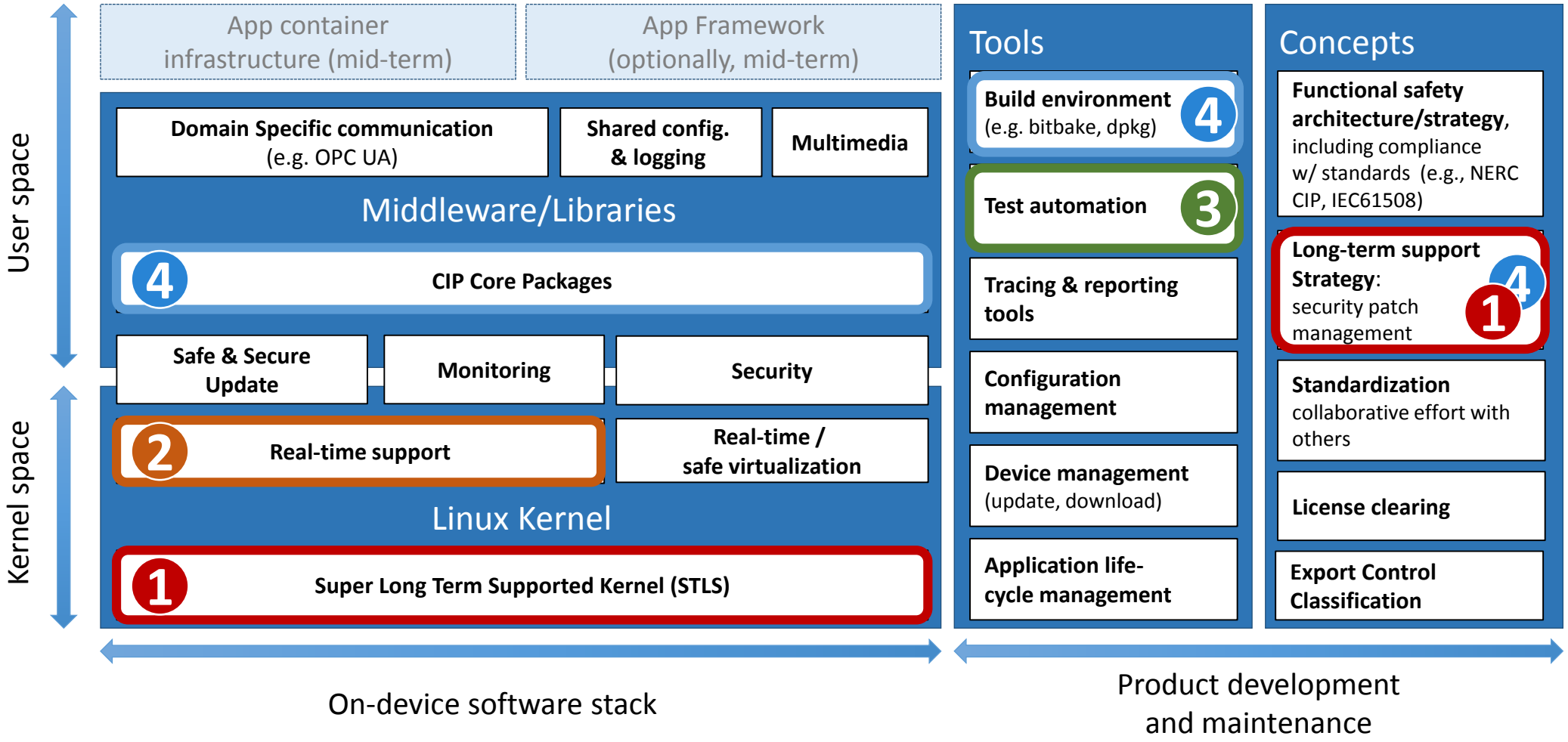
Legend: To be specified / implemented by CIP; Integration / cooperation

CIP activities and status

Announcements

- CIP testing project released B@D v1.0
- CIP Core project launched
- CIP decided to take Debian as a primary reference distribution

Scope of activities



CIP 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.

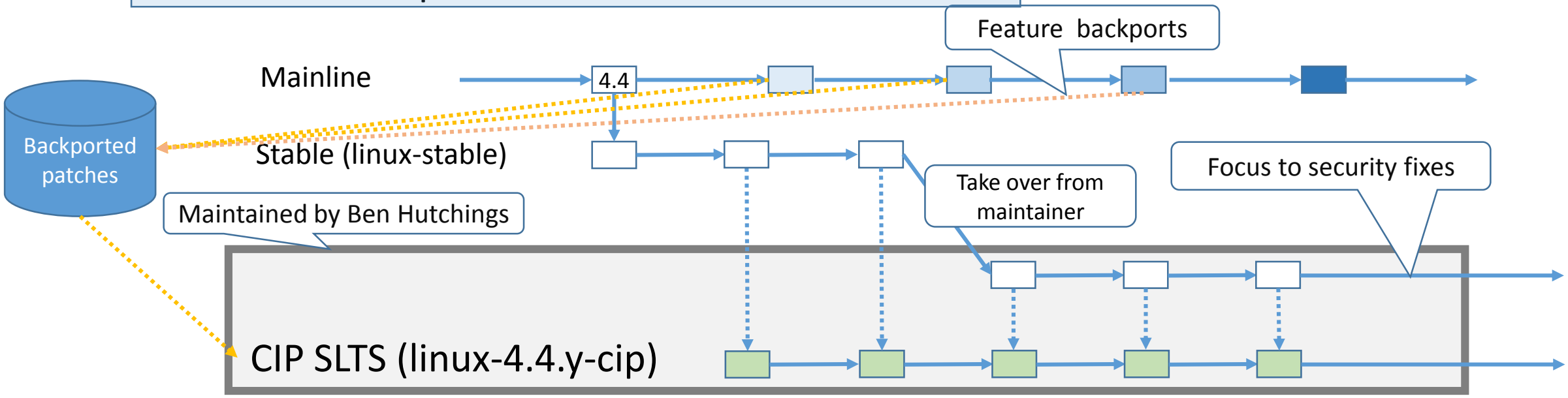
4 CIP Core

- 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.

❶ 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 4.4.92-cip11 released on 18th October 2017



❶ 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 IoT2000 series

❶ CIP SLTS Kernel development (3/5)

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.

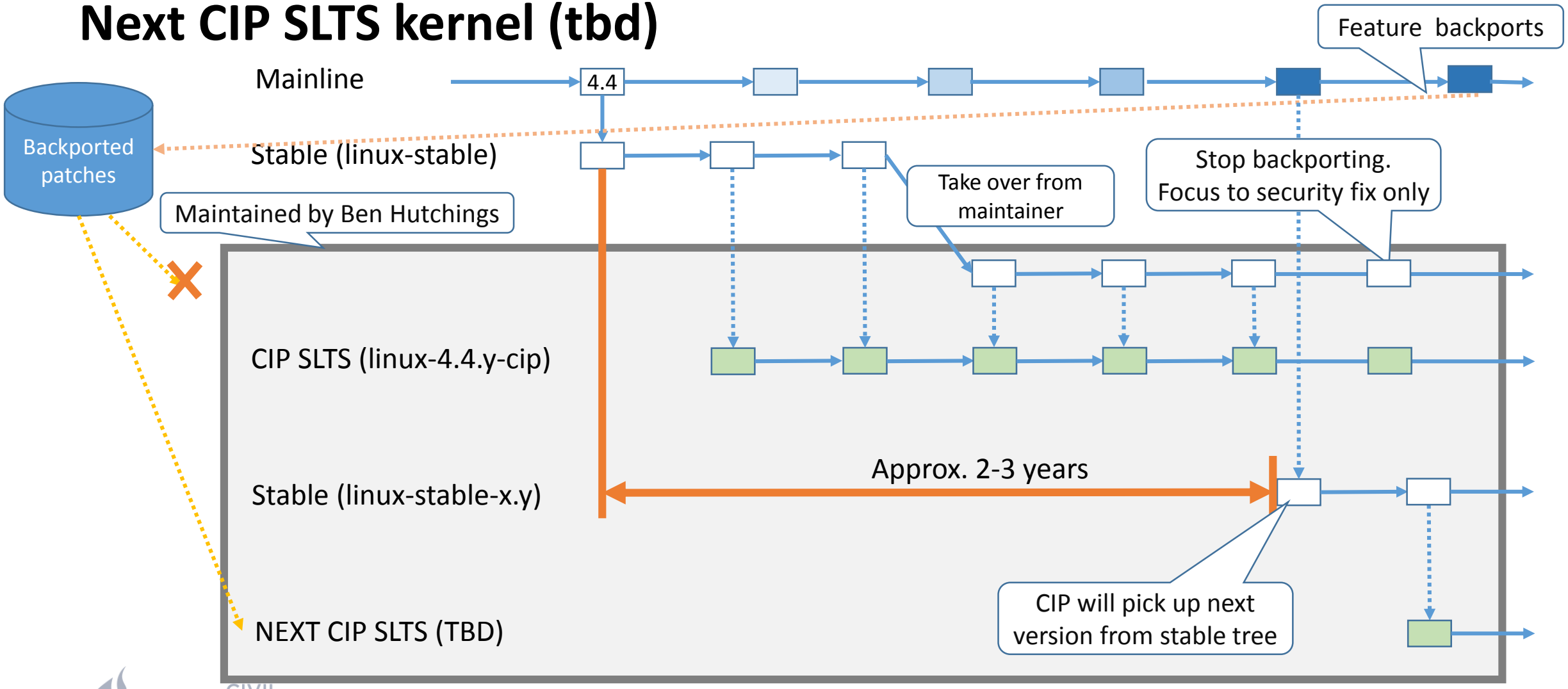
< -- snip -- >

--
Ben Hutchings
Software Developer, Codethink Ltd.

CIVIL INFRASTRUCTURE PLATFORM

❶ CIP SLTS Kernel development (4/5)

Next CIP SLTS kernel (tbd)



❶ CIP SLTS Kernel development (5/5)

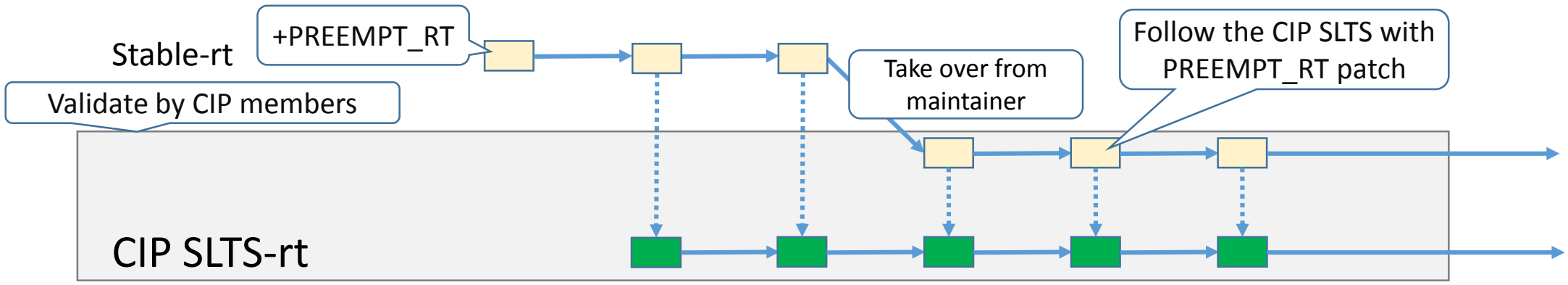
Out-of-tree drivers

- 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.

❷ 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
- CIP RT is currently under development at the following URL
  - https://github.com/igaw/linux-cip-rt



❷ 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

❸ CIP testing (1/3)

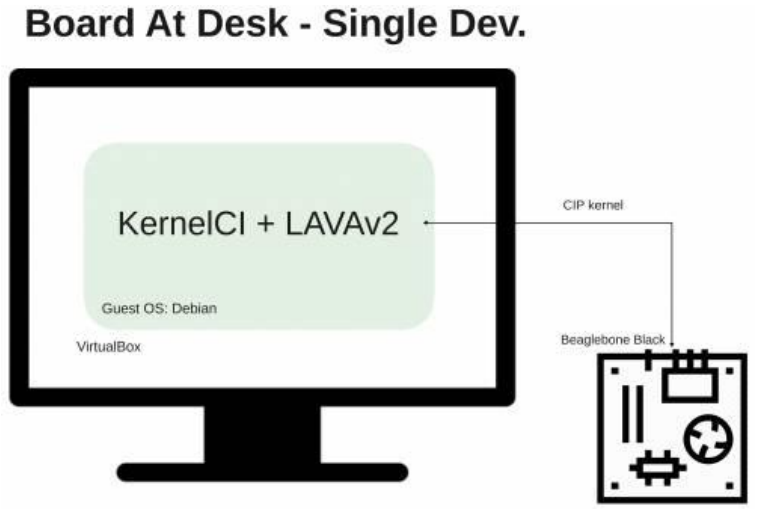
Milestones of CIP testing and current status

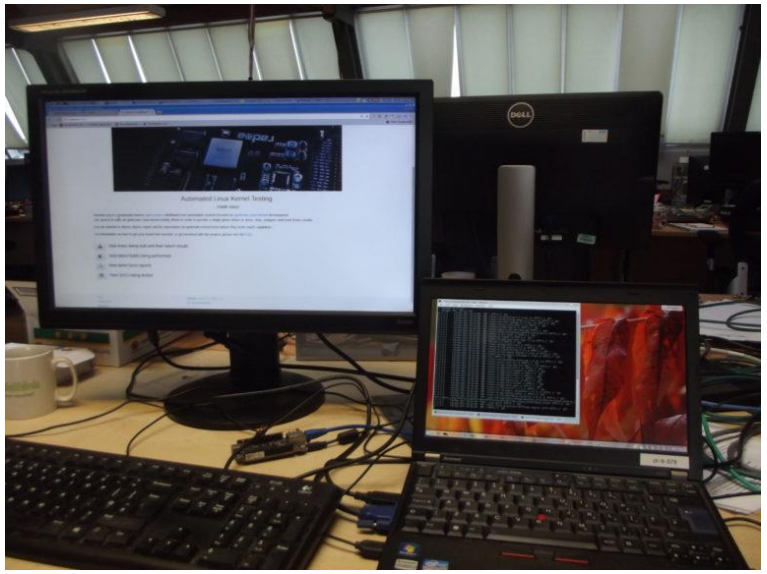
1. Board at desk - single dev
   - A setup that allows a developer to test the CIP kernel on the CIP selected hardware platform connected locally to her development machine using kernelCI tools.
2. CIP kernel testing
   - Test the CIP kernel on a regular basis and share the results with other CIP community 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

❸ 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





CIVIL INFRASTRUCTURE PLATFORM

❸ CIP testing (3/3)

Next Steps

- Collaboration with other testing effort
  - CIP had a meeting with AGL members for 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

❹ 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

- CIP members also interested in Yocto Project as a build tool
  - CIP might create 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)

❹ CIP Core Packages (1/5)

An example of minimal package set for CIP base layer

Candidates for initial component set

CIP Kernel

- Kernel
  - Linux kernel + backported patches
  - PREEMPT_RT patch

CIP Core Packages

- Bootloader
  - U-boot
- Shells / Utilities
  - Busybox
- Base libraries
  - Glibc
- Tool Chain
  - Binutils
  - GCC
- Security
  - OpenSSL

Keep these packages for Reproducible build

Dev packages

- 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.

❹ CIP Core Packages (2/5)

Current status of the Base layer development

1. Define an initial component set

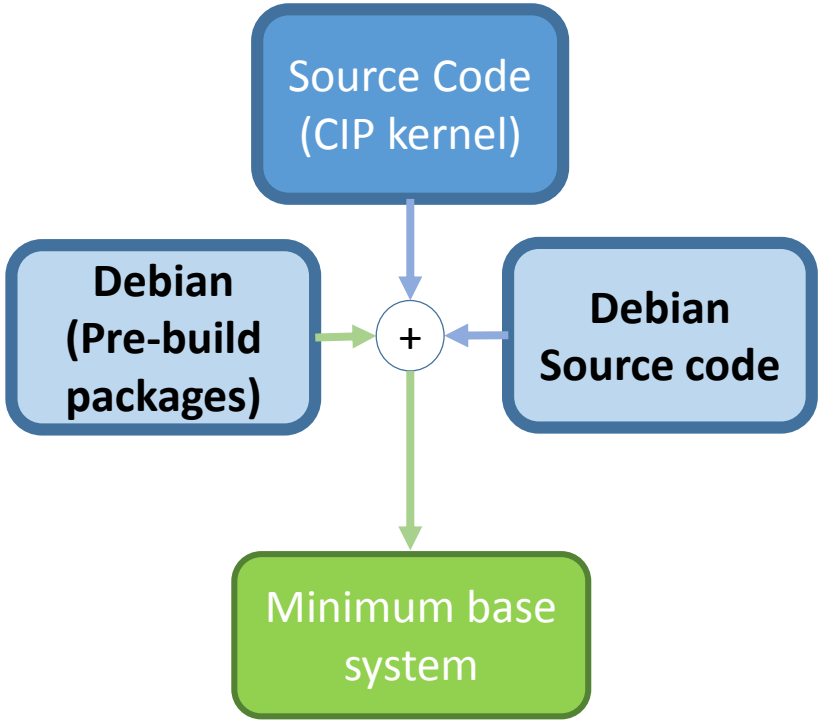
1.5 Talk to open source communities

2. Define component version
3. Contribute to upstream project
4. Start maintenance for SLTS

❹ 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

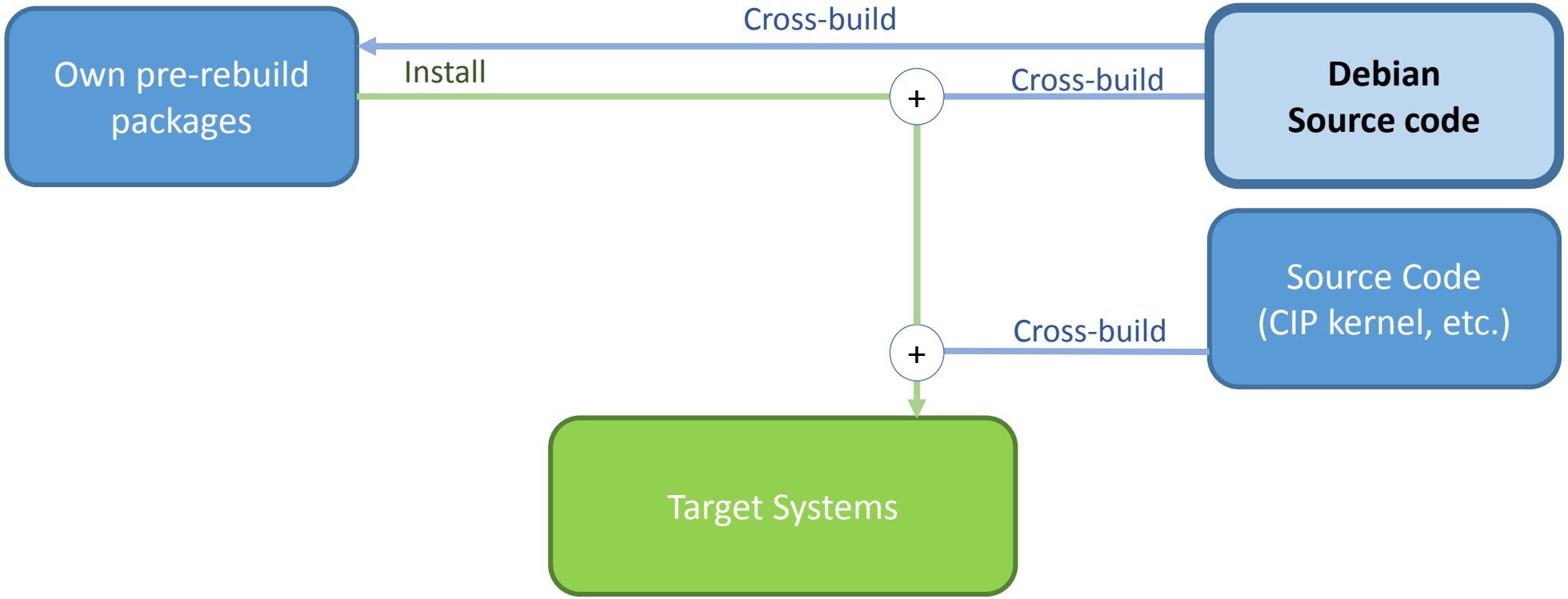


Source code: https://gitlab.com/cip-project/cip-core

❹ CIP Core Packages (4/5)

Creating Debian-based image (Currently supported)

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

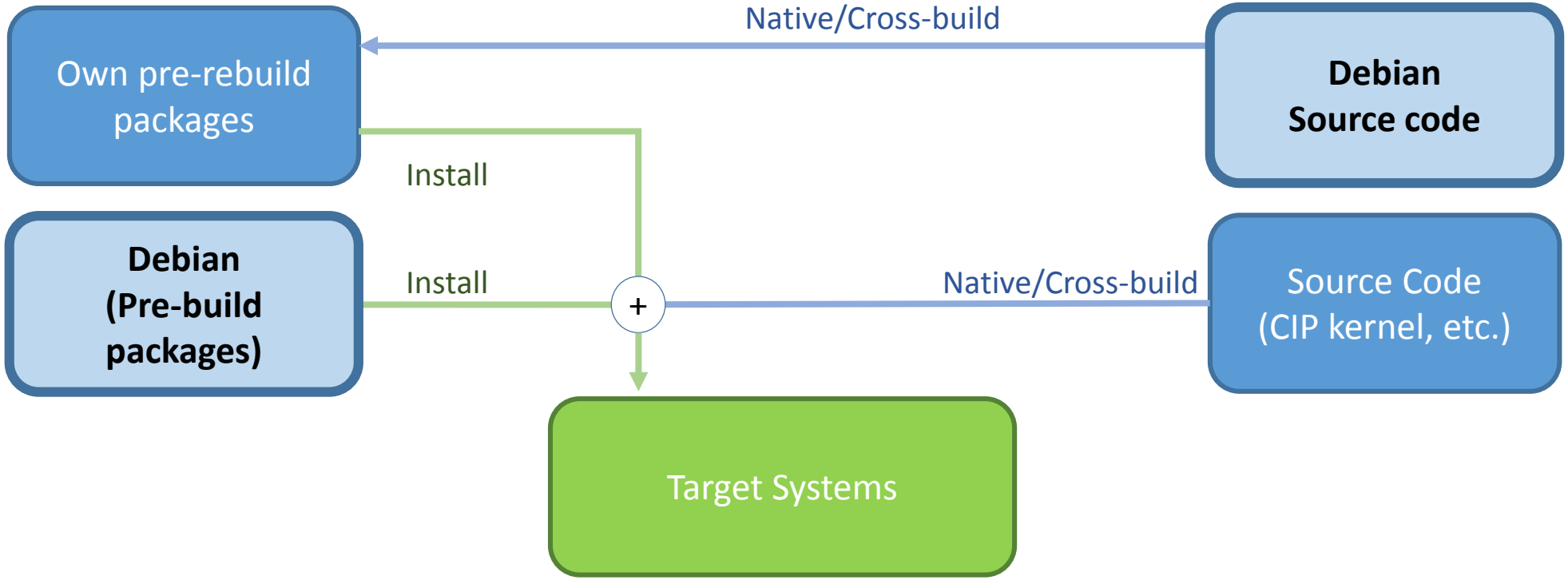


❹ CIP Core Packages (5/5)

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	✓ Recipes for building product packages ✓ Recipes for image generation	✓ Common function to unpack Debian source packages ✓ **Full recipes** for cross-building every Debian source package
Compilation	**Native**		Cross
Benefits	✓ **Re-use Debian binaries and QA** ✓ Fast (re-use, parallel builds) ✓ **Lower development costs**		✓ Affinity with Poky recipes ✓ **Fully customizability** ✓ No need to keep binary pkgs
Common features	✓ **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

❹ Gaps and Common Goals between Debian and CIP

Debian	CIP requires	Chance to collaborate with Debian
Support ▪ Term: 3+2 years by Debian-LTS ▪ Num of pkgs: 67776	**Support** ▪ Term: 10+ years ▪ Num of pkgs: 10+ (minimum)	▪ **Longer** term maintenance for limited number of packages
Build ▪ Should support native build ▪ Working on cross build packaging (Debian-cross) ▪ Reproducible build	**Build** ▪ Need to have both native and cross build ▪ Binary / Source code should be managed and reproducible	▪ Contributing to **Debian-cross**
OSS license compliance ▪ DEP-5 adoption is ongoing	**OSS license compliance** ▪ Generate reports automatically ▪ Easy to redistribute	▪ Exchange and share the **license review results**
Testing ▪ Packages has to be tested ▪ autopkgtest	**Testing** ▪ All packages should be tested in timely manner	▪ Contributing **test cases** to upstream

What’s currently under discussion in CIP

- Functional safety
- Security standards for industry
  - E.g. IEC62443-4
- Y2038

Summary and conclusion

Summary

- **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

CIVIL INFRASTRUCTURE PLATFORM

CIP meeting and presentations @ ELCE

- Tuesday, October 24th, 13:00-14:00 (**Immediately after this talk**)

  CIP developers meeting/gathering at ELCE (Liben Room, Mezzanine Level)

- Tuesday, October 24, 16:55 - 17:35 (Congress Hall II)

  Maintaining a Linux Kernel for 13 Years? You Must be Kidding Me. We Need at Least 30?,
  Agustin Benito Bethencourt & Ben Hutchings (Codethink Ltd)

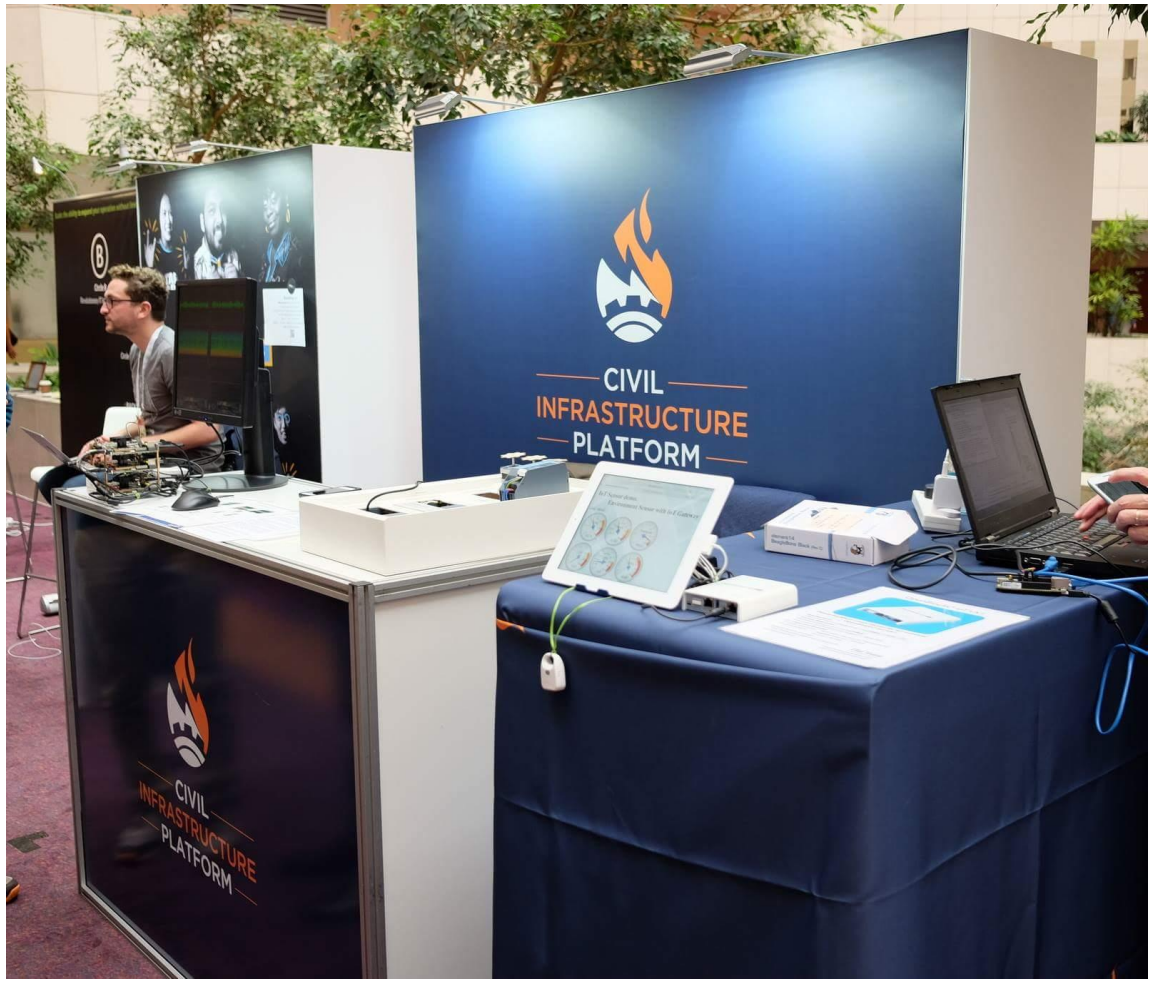
- Wednesday, October 25, 09:50 - 10:10 (Congress Hall)

  Keynote: Challenges in Industrializing OSS and How Siemens Tackles Them,
  Jan Kiszka (Siemens AG)

Need more information?

- Please come to CIP booth!
  - Location: Mezzanine Level

- Demo contents
  - **CIP RT kernel** (Renesas RZ/G)
  - **Industrial IoT** (Siemens IoT2020)
  - **IoT sensors** (Plat'Home OpenBlocks)
  - **B@D v1.0 (at 3:00pm-4:30pm)**



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

Thank you!

CIVIL INFRASTRUCTURE PLATFORM

Questions?