



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

# Civil Infrastructure Platform : Industrial Grade SLTS Kernel and Base-Layer Development

**Yoshitake Kobayashi, CIP**

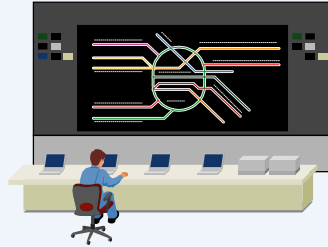
CELP Japan Technical Jamboree, June 23, 2017



# Our Civilization is Run by Linux



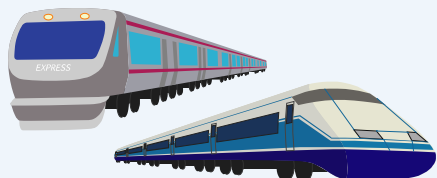
## Transport



**Rail automation**

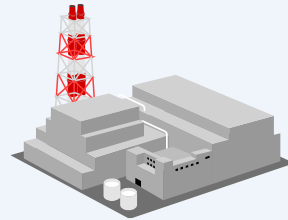


**Automatic ticket gates**

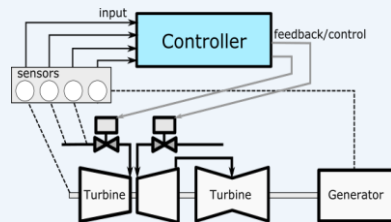


**Vehicle control**

## Energy

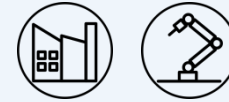


**Power Generation**



**Turbine Control**

## Industry



**Industry automation**



**Industrial communication**



**CNC control**

## Others



**Healthcare**



**Building automation**



**Broadcasting**



**But there are issues to be  
solved...**



The background image shows a large, complex railway signaling cabinet. It features multiple rows of electronic modules and a dense network of multi-colored cables (yellow, green, blue, red) connected to various ports. The top of the cabinet has a series of colored indicator lights or status panels in red, orange, green, and blue. The overall structure is made of light-colored metal or plastic panels.

# 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**



# What we have done on Linux for civil infrastructure systems



- Improve real-time performance and test
- Improve reliability and test
- Improve security and test
- Improve stability and test
- Create a lot of documents and review
  - Open source software licenses compliance
  - Export control classification
- Then, support for long-time such as 20-60 years
- ...

# **We have a problem...**

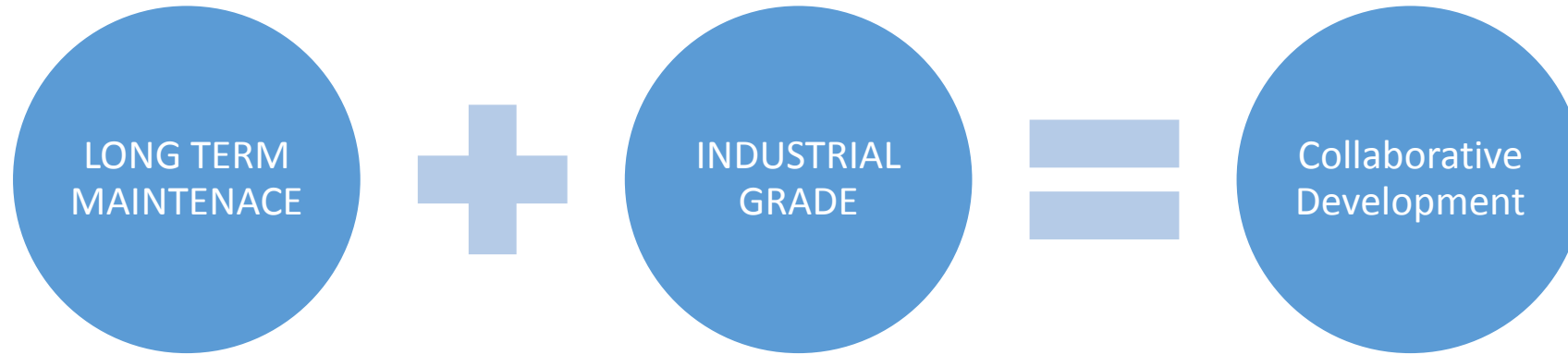


# The Problems we face ...



- 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...

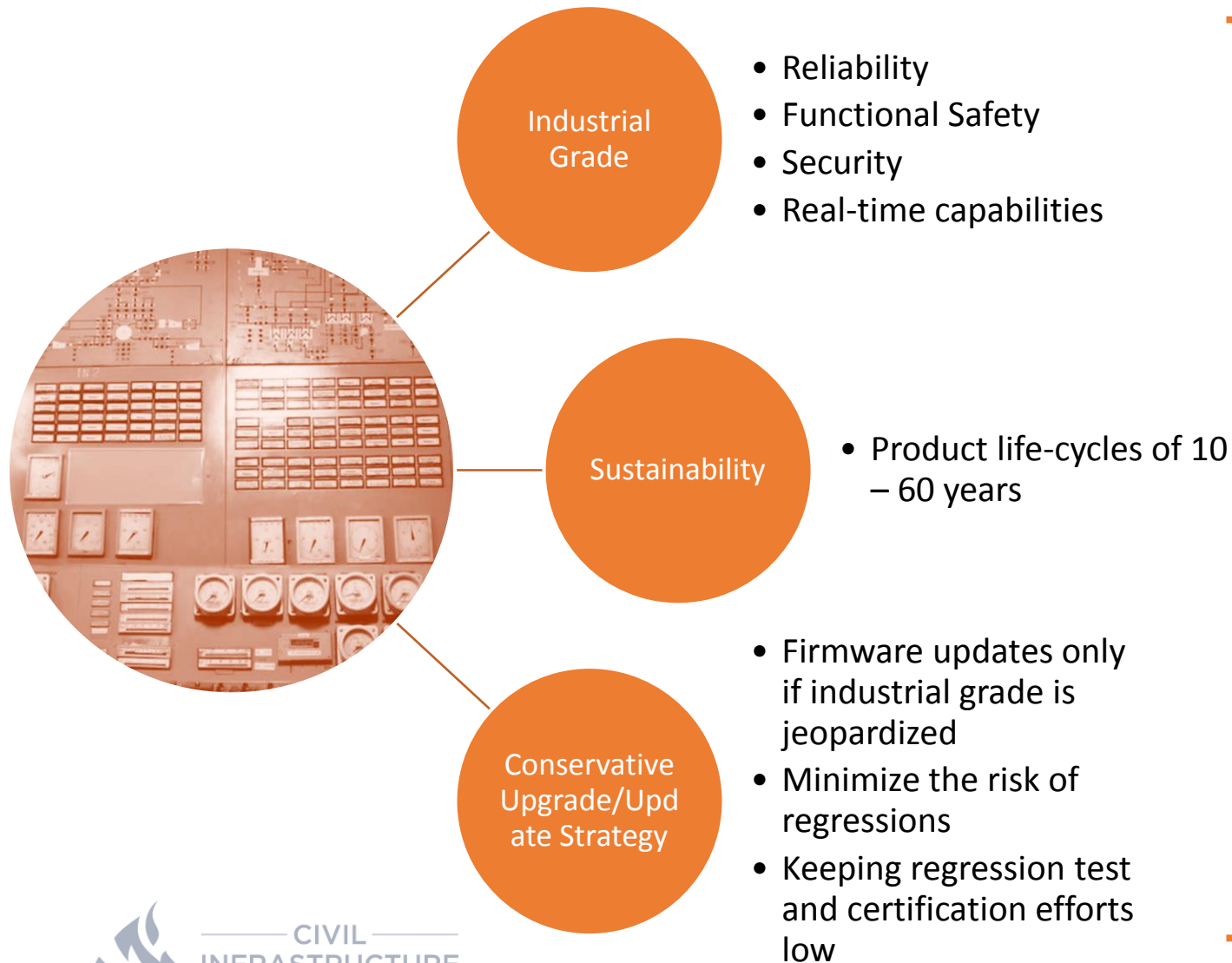
**Establishing an Open Source Base Layer of industrial-grade software to enable the use and implementation of software building blocks for Civil Infrastructure Systems**

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



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

# Requirements for the Civil infrastructure systems



**This has to be achieved with ...**

## **Maintenance costs**

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

## **Development costs**

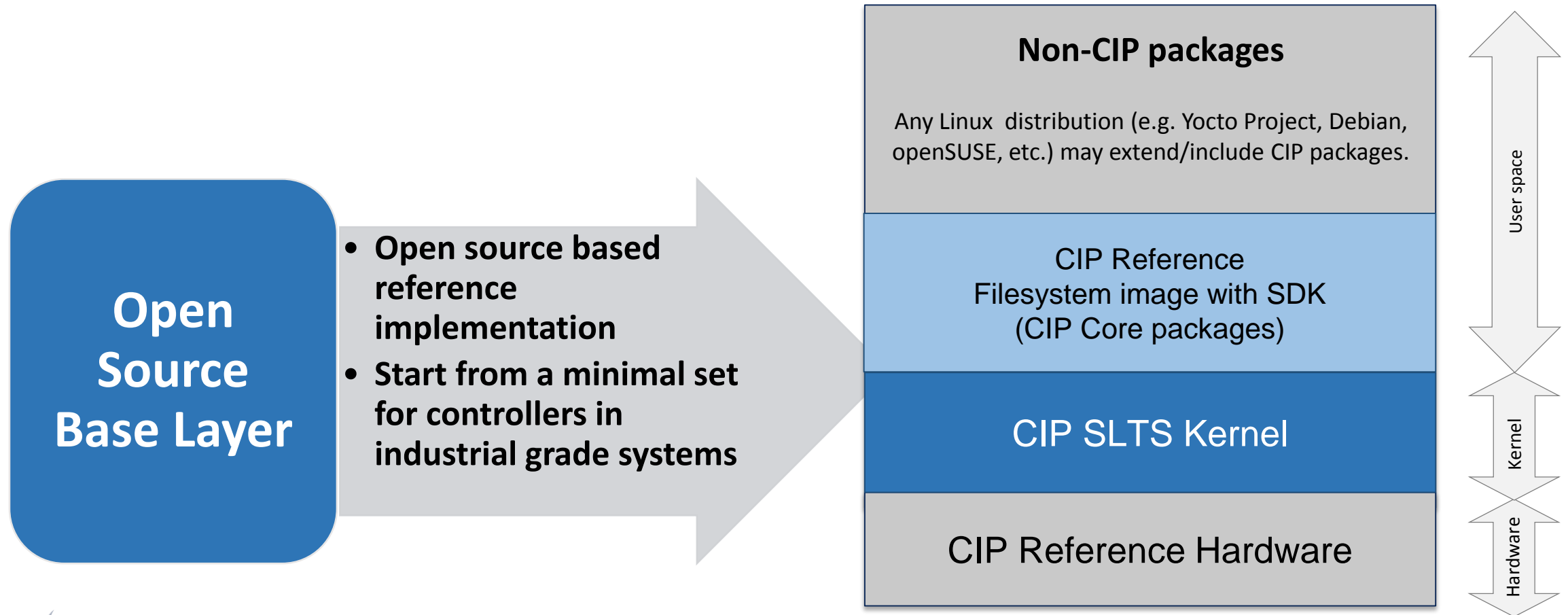
- Don't re-invent the wheel

## **Development time**

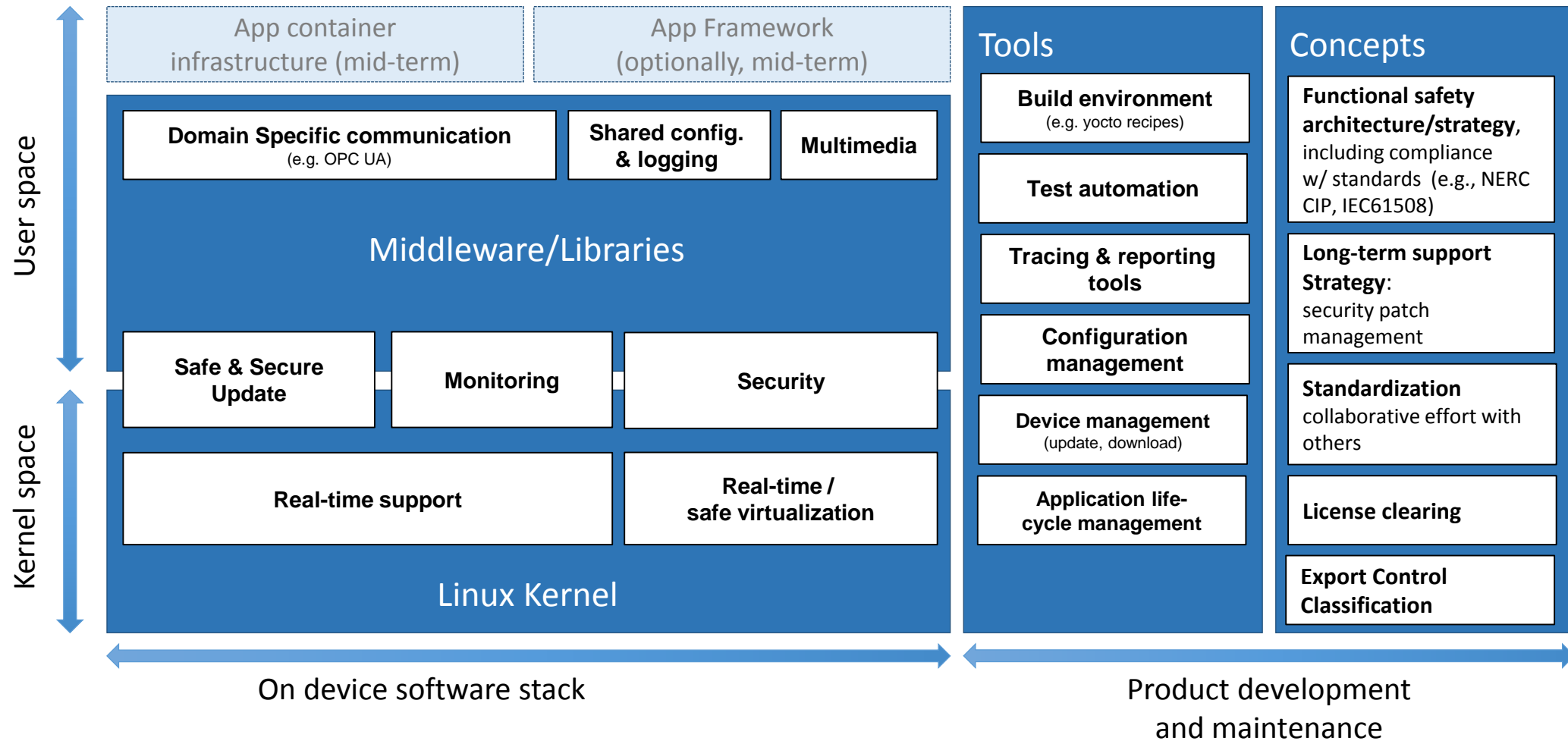
- Shorter development times for more complex systems



# Things to be done: Creation of an “Open Source Base Layer”



# Scope of activities

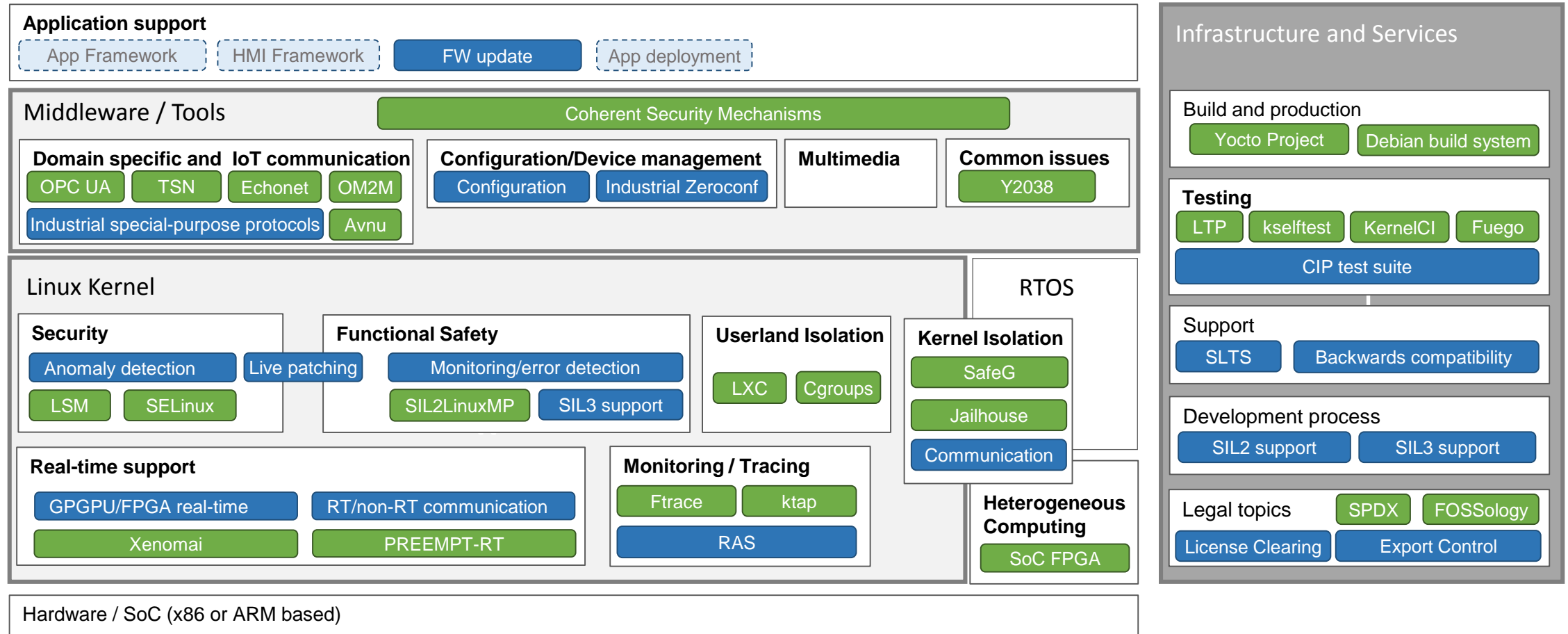




# Technical topics and related projects (Feb. 2017 version)



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

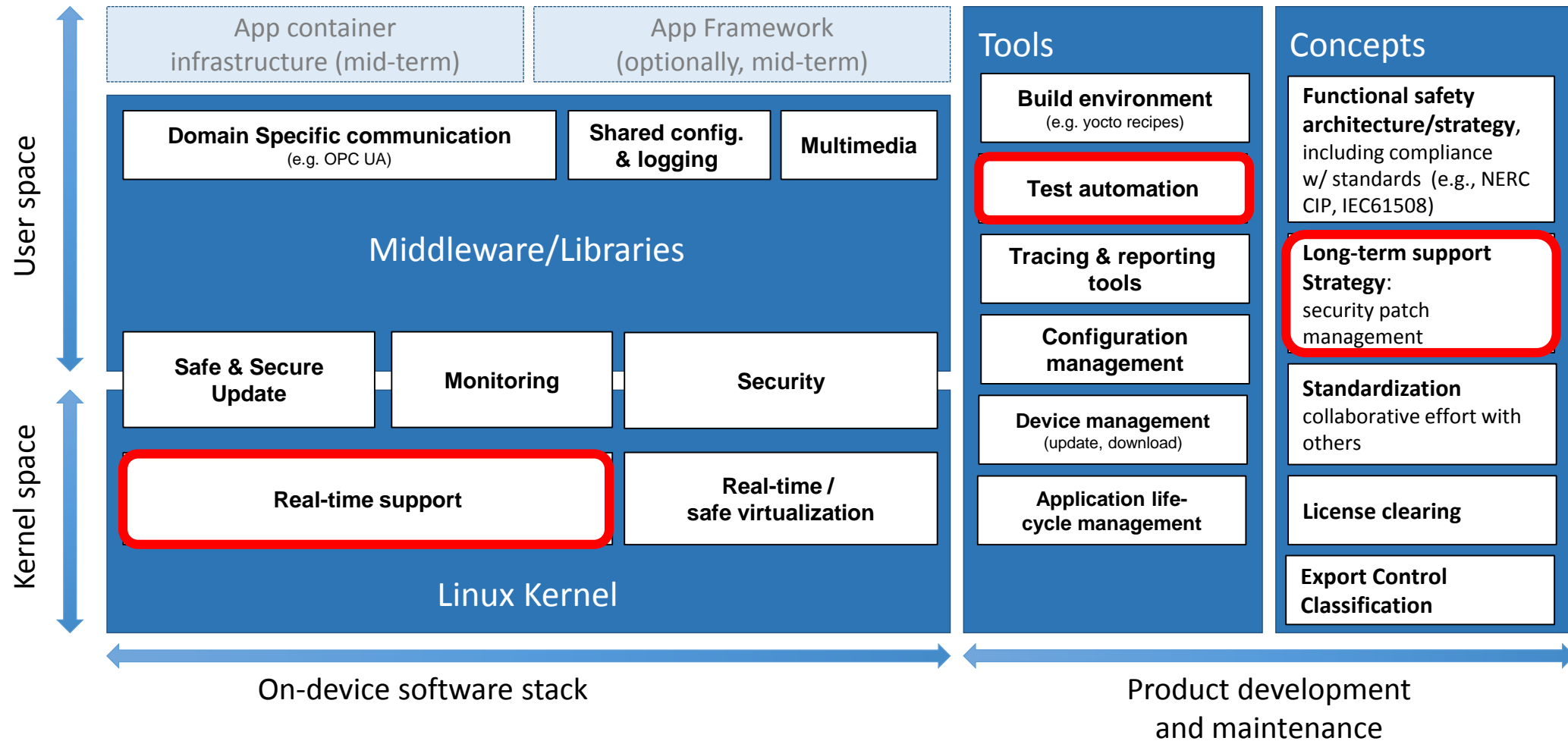


Legend

To be specified / implemented by CIP

Integration / cooperation

# Scope of activities





# Current status of CIP base layer development



- CIP SLTS kernel development
  - Decide the CIP kernel version
    - 4.4 is the first CIP kernel. Maintenance expected for 10 years and more (SLTS).
  - Select a maintainer
    - Ben Hutchings is the initial CIP-kernel maintainer
  - Define a kernel maintenance policies
    - <https://wiki.linuxfoundation.org/civilinfrastructureplatform/cipkernelmaintenance>
  - Start maintenance
    - Linux 4.4.69-cip4 released on 25<sup>th</sup> May 2017
  - Create CIP kernel test framework
- CIP core package development
  - Define an initial component set
  - Define component version
  - Contribute to upstream project
  - Start maintenance for SLTS

# CIP SLTS Kernel Development

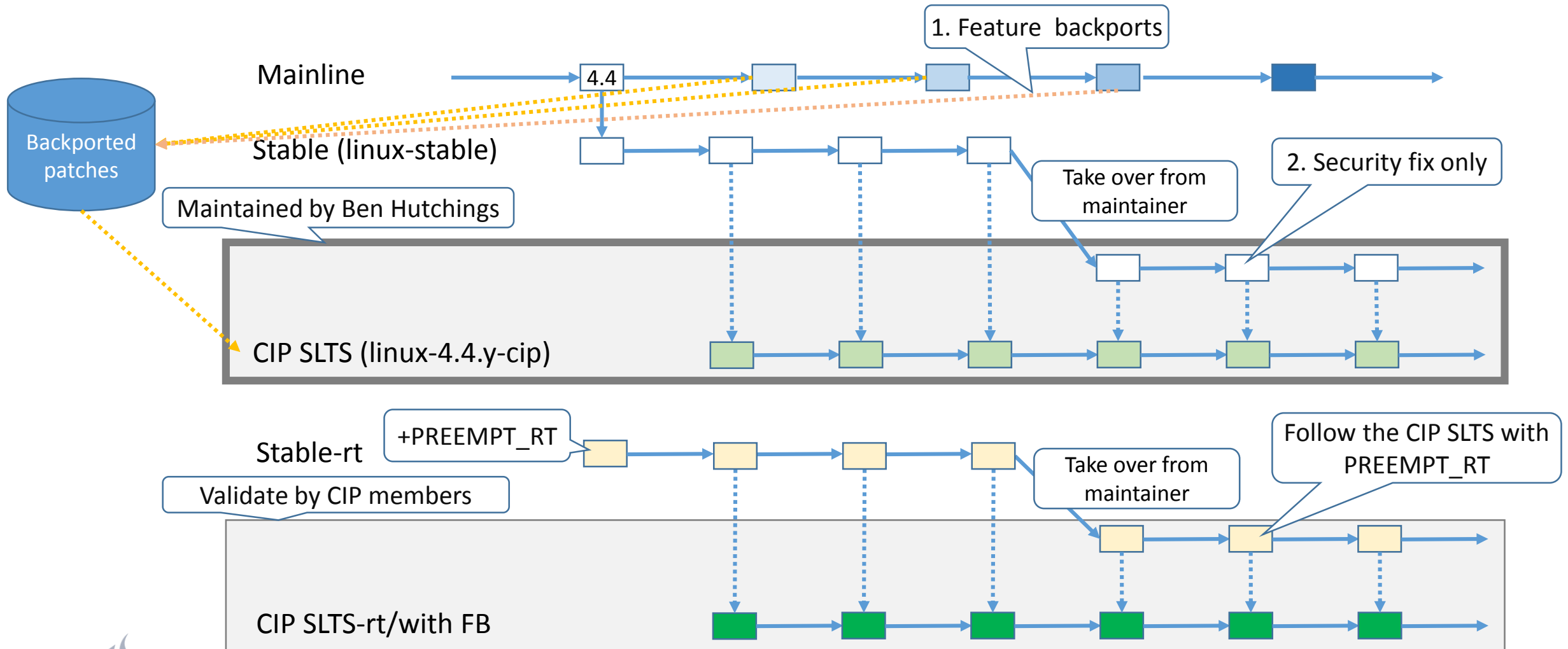
# Overview of CIP SLTS kernel



- Kernel trees
  - CIP SLTS (linux-4.4.y-cip)
    - Official CIP SLTS kernel tree
      - <https://git.kernel.org/cgit/linux/kernel/git/bwh/linux-cip.git/>
      - Based on linux-stable.git
    - Maintainer: Ben Hutchings
    - Validation will be done by CIP
  - 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
- Maintenance period
  - 10 years and more (10-20 years)



# CIP SLTS Kernel development trees



# CIP SLTS Kernel development



- 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

# CIP's participation in the Real-time Linux Project



- CIP has become a Gold Member of the Real Time Linux Project
- What's next
  - Work together with the RTL Project
  - CIP member Daniel Wagner (Siemens) is trying 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>





# 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.

# Major version release cycle (Next CIP SLTS kernel version)



- CIP will take a LTS kernel every **2-4 years**
- Planning to synchronize with LTSI for next CIP SLTS kernel
  - LTSI: <http://ltsi.linuxfoundation.org/>

# CIP Kernel testing



# Purpose of CIP testing



- Detecting bugs
- Detecting regressions
- Provide test results in a timely manner

# 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

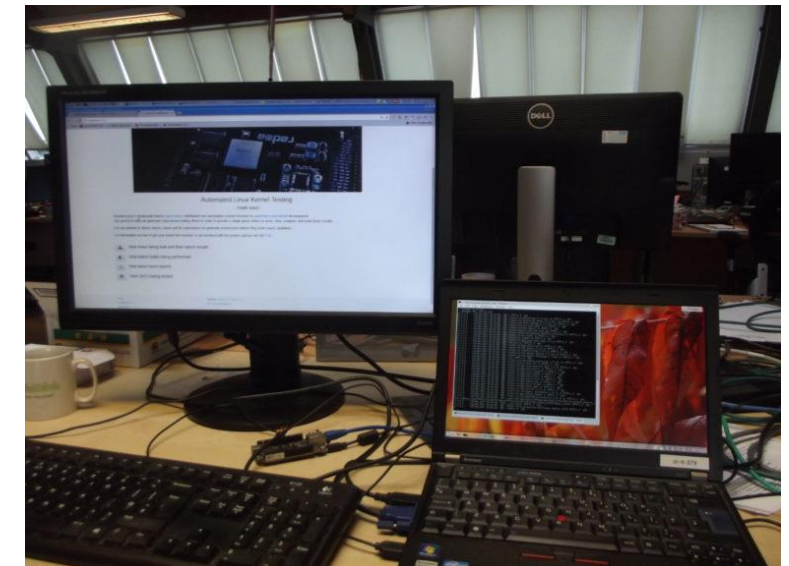
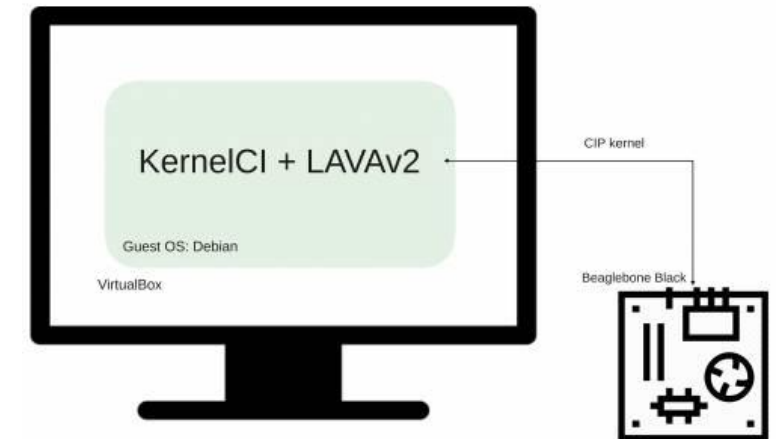
- Once the testing environment has been ready and works for the kernel, explore how to extend it to the entire CIP platform.

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

# CIP testing

- Goal
  - Create and publish a VM image that contains KernelCI & LAVA
  - Single developer can test the CIP kernel (or any other kernels)
- News
  - **B@D v0.9.1 has been release at OSSJ 2017**
    - <https://www.cip-project.org/news/2017/05/30/bd-v0-9-1>
  - Download the VM or deploy the environment through Vagrant
    - <https://wiki.linuxfoundation.org/civilinfrastructureplatform/cipdownload>
  - Check the tools and software packages included in this release.
    - <https://wiki.linuxfoundation.org/civilinfrastructureplatform/ciptestingboardatdesksingledevfeaturepage>
    - The CIP testing team has invested a significant effort in writing step by step instructions to deploy, configure and run tests.
- Check the source code involved
  - <https://gitlab.com/cip-project/cip-testing/board-at-desk-single-dev/tree/master>

## Board At Desk - Single Dev.





# CIP testing: next steps



- 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 Europe 2017 this October

# CIP Core Package Development

# Current status of the Base layer development



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

# Current status of the Base layer development



1. Define an initial component set



1.5 Talk to upstream maintainer

2. Define component version

3. Contribute to upstream project

4. Start maintenance for SLTS



# Initial component set for CIP base layer



CIP will start with a minimal set of packages. “CIP kernel” and “CIP core” packages run on hardware.

## Candidates for initial component set

## Keep these packages for Reproducible build

CIP  
Kernel

CIP Core  
Packages

- Kernel
  - Linux kernel 4.4 + backported patches
  - PREEMPT\_RT patch
- Bootloader
  - U-boot
- Shells / Utilities
  - Busybox
- Base libraries
  - Glibc
- Tool Chain
  - Binutils
  - GCC
- Security
  - OpenSSL

Dev  
packages

• Flex	• Git	• pax-utils
• Bison	• Glib	• Pciutils
• autoconf	• Gmp	• Perl
• automake	• Gzip	• pkg-config
• bc	• gettext	• Popt
• bison	• Kbd	• Procps
• Bzip2	• Libibverbs	• Quilt
• Curl	• Libtool	• Readline
• Db	• Libxml2	• sysfsutils
• Dbus	• Mpclib	• Tar
• Expat	• Mpfr4	• Unifdef
• Flex	• Ncurses	• Zlib
• gawk	• Make	
• Gdb	• M4	

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

# CIP Project X

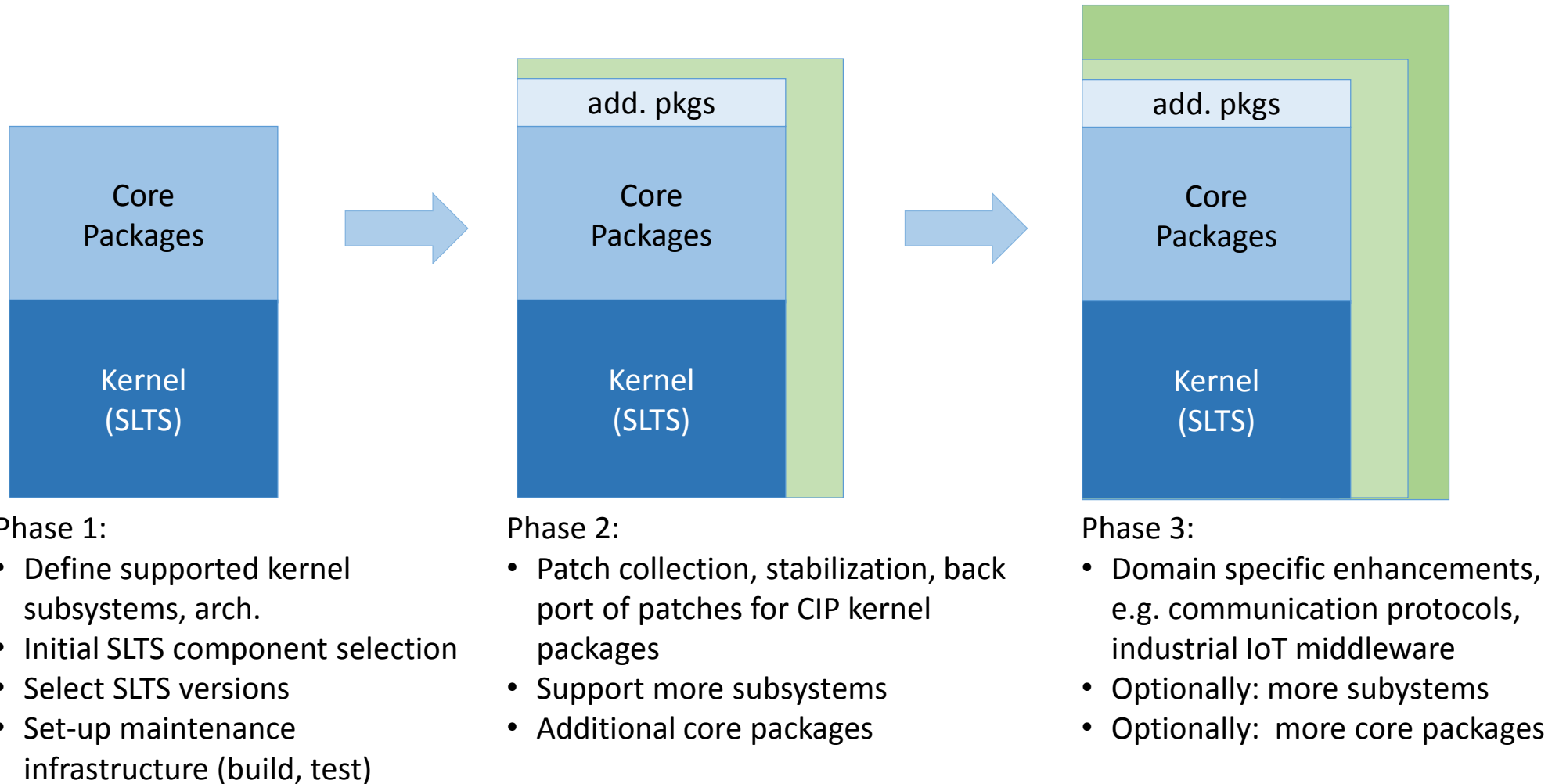


- **Started an incubation project for the minimum base system**
  - This project will provide a way to test the installable image
- **Goal**
  - **Input:** Debian sources/binaries and cip kernel
  - **Build mechanism:** bitbake and/or Debian build system
  - **Output:** Minimum deployable base system
- **Current status**
  - Minimal rootfs available for the following hardware
    - QEMUx86
    - BeagleBone Black
    - Cyclone-V
- **Source code**
  - <https://gitlab.com/cip-playground/project-x>

# Development plan



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



# CIP whitepaper release

- 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](https://wiki.linuxfoundation.org/_media/civilinfrastructureplatform/whitepaper_short.pdf)





# Summary



- Selected the first CIP kernel and initial maintainer
  - 4.4 as first CIP kernel. Maintenance expected for 10+ years (SLTS).
  - Ben Hutchings as initial CIP kernel maintainer.
  - Defined CIP Kernel maintenance policies.
  - Defining CIP kernel + RT maintenance.
- Defined initial board platforms and provide support for them.
  - Renesas RZ/G and Beaglebone Black
- Released Board @ Desk for CIP kernel testing
- Started CIP Project X
- Published a whitepaper

# Next Steps

# Next steps by CIP



- Board @desk - Single dev
  - Start Action-2.  
<https://wiki.linuxfoundation.org/civilinfrastructureplatform/ciptesting>
  - Increase test coverage.
- Kernel maintenance
  - Define Kernel features
  - Create a branch for 4.4-cip-rt
- Analysis
  - Select additional software as part of CIP base layer.
  - Review requirements from CIP members (e.g. Functional Safety)
- Collaboration: kernelCI, LAVA, Fuego, y2038, KSPP, Real-time Linux Project

# CIP booth at OSSJ 2017



- CIP use cases
  - Industrial controller
  - Power plant simulator with real controller
  - IoT (OpenBlocks IoT)
  - CIP testing on reference board (Renesas RZ/G)
- Whitepaper



# **Please Join us!**



# Why joining CIP?



- **Steer**  
participate in project decisions and technical direction.
- **Participate**  
bring your use cases and ideas to the right forum.
- **Learn**  
by working on daily basis in the open with others with common interest.
- **Collaborate**  
share effort and knowledge. Stand on the shoulders of giants.

# Contact Information and Resources



To get the latest information, please contact:

- Noriaki Fukuyasu: [fukuyasu@linuxfoundation.org](mailto:fukuyasu@linuxfoundation.org)

## Other resources

- CIP Web site: <https://www.cip-project.org>
- CIP Mailing list: [cip-dev@lists.cip-project.org](mailto:cip-dev@lists.cip-project.org)
- CIP Wiki: <https://wiki.linuxfoundation.org/civilinfrastructureplatform/>
- Collaboration at CIP: <http://www.gitlab.com/cip-project>
- CIP kernel: [git://git.kernel.org/pub/scm/linux/kernel/git/bwh/linux-cip.git](https://git.kernel.org/pub/scm/linux/kernel/git/bwh/linux-cip.git)

# Call for new participants!



Provide a super long-term maintained industrial-grade embedded Linux platform.

## Platinum Members

**HITACHI**  
Inspire the Next

**SIEMENS**

**RENESAS**

**TOSHIBA**

## Silver Members

**CodeThink**

**Plat'Home**  
There, we are. Internet of Things





# Questions?



# Thank you!