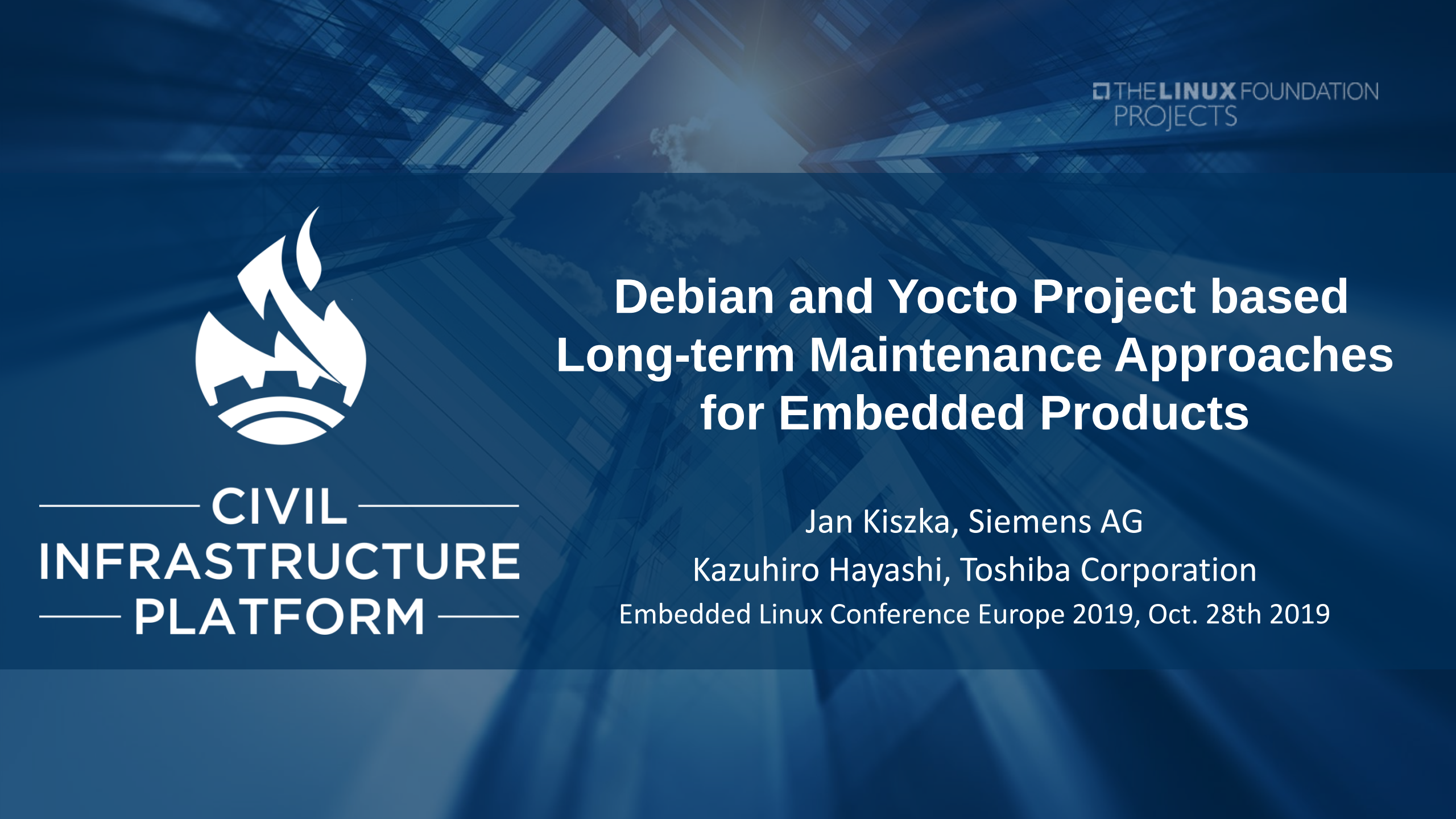


Debian and Yocto Project based Long-term Maintenance Approaches for Embedded Products

Jan Kiszka, Siemens AG
Kazuhiro Hayashi, Toshiba Corporation

Embedded Linux Conference Europe 2019, Oct. 28th 2019

About us

- Kazuhiro Hayashi <kazuhiro3.hayashi@toshiba.co.jp>
  - Working for Toshiba Corporation
  - (In-house) Embedded Linux development for products, consultant
  - Member of CIP, developing tools of CIP Core
  - Contribute to Linux and other OSS projects
- Jan Kiszka <jan.kiszka@siemens.com>
  - Working for Siemens Corporate Technology
  - (In-house) Embedded Linux consultant & developer
  - Member of CIP, isar-cip-core development, some CIP kernel backports
  - Maintainer of and contributor to various OSS projects

Agenda

- Requirements in Industrial Product Development
- CIP as the solution
- What is “CIP Core”?
- CIP Core implementation
  - Isar
  - Deby
- Building products on top of CIP Core
- Future plans

Requirements in Industrial Product Development

- Porting
- OS/middleware development
- Application development
- Installation
- Testing
- Development release

- Architecture & board support
- Customizability, Scalability
- Standalone SDK
- Ready-to-use image generation
- Stability

- Product Certification
- Publication

- License clearing

- Bug fix, security fix
- System update

- Long-term maintenance
- Verified & automated update

- Quick release

- Modularity for multiple projects

CIVIL INFRASTRUCTURE PLATFORM

What to Do First…

- Select appropriate base system
  - Linux distribution
- Provide tools for integrating the base system into products
  - Build system, etc.

- Architecture & board support
- Customizability, Scalability
- Standalone SDK
- Ready-to-use image generation
- Stability

- License clearing

- Long-term support
- Verified & automated update

- Quick release

- Modularity for multiple projects

CIP is the Solution

Industrial grade

- Reliability
- Functional Safety
- Real-time capabilities

Sustainability

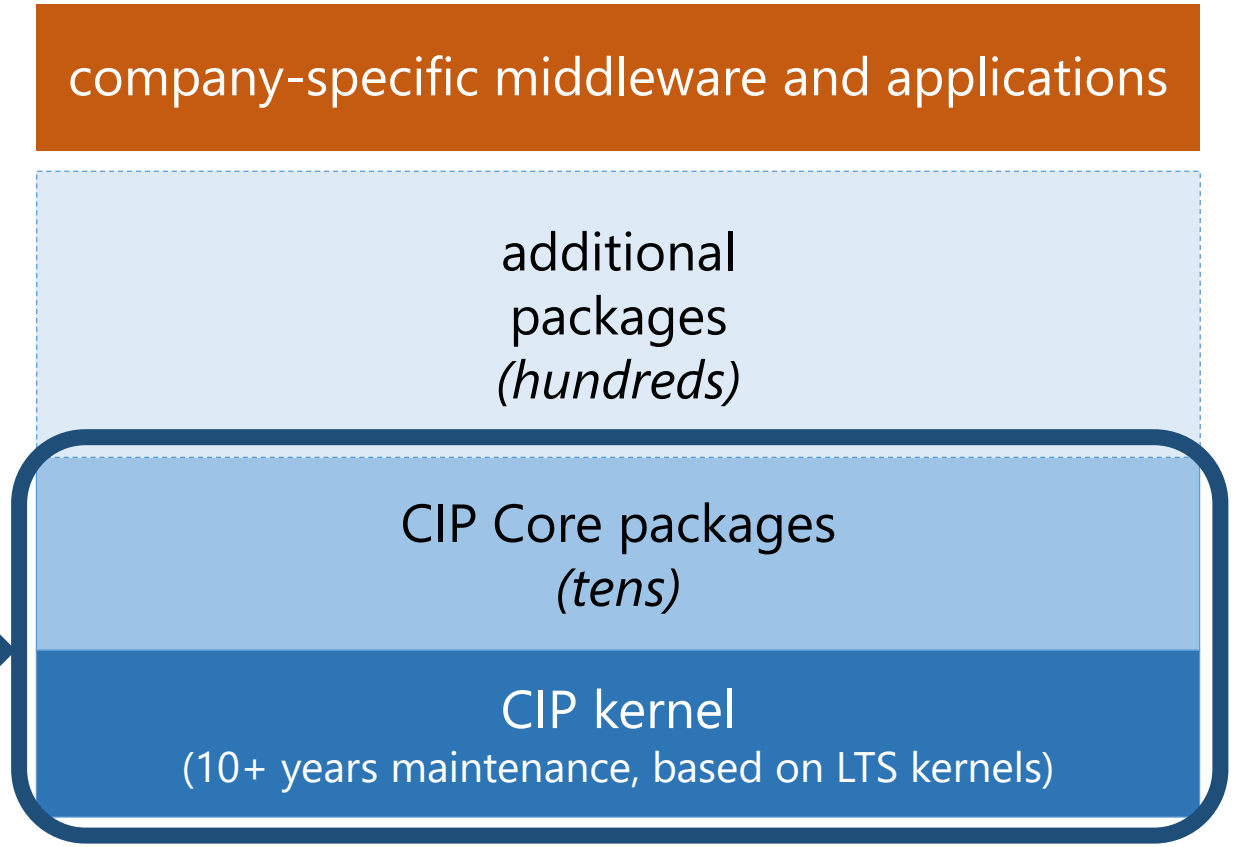
- Product life-cycles of decades
- Backwards compatibility
- Standards

Security

- Security & vunerability managment
- Firmware updates
- Minimize risk of regressions

CIP is the Solution

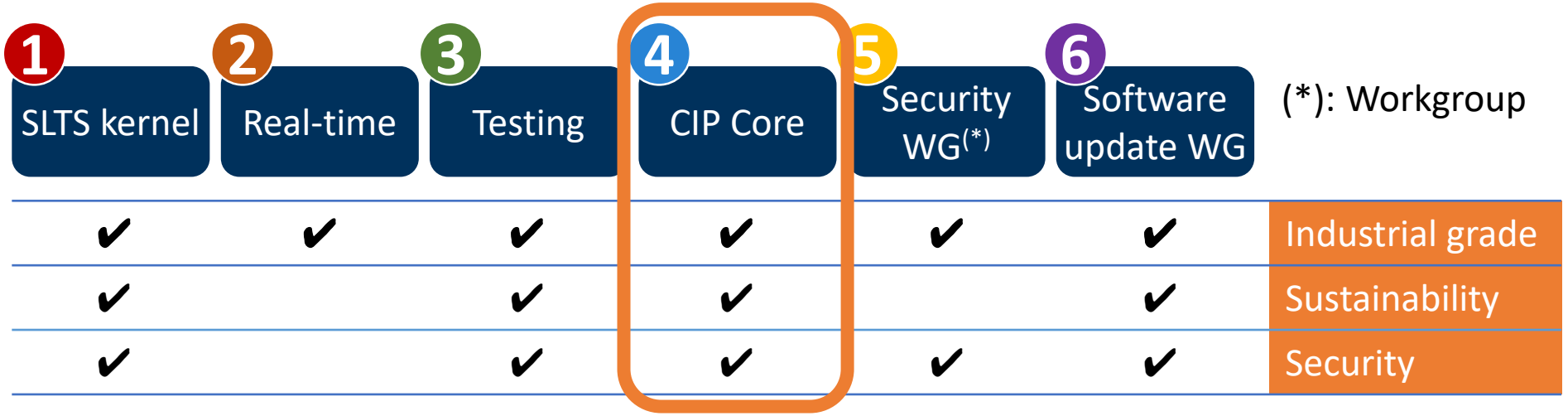
CIVIL INFRASTRUCTURE PLATFORM



Establishes an "Open Source Base Layer (OSBL)"

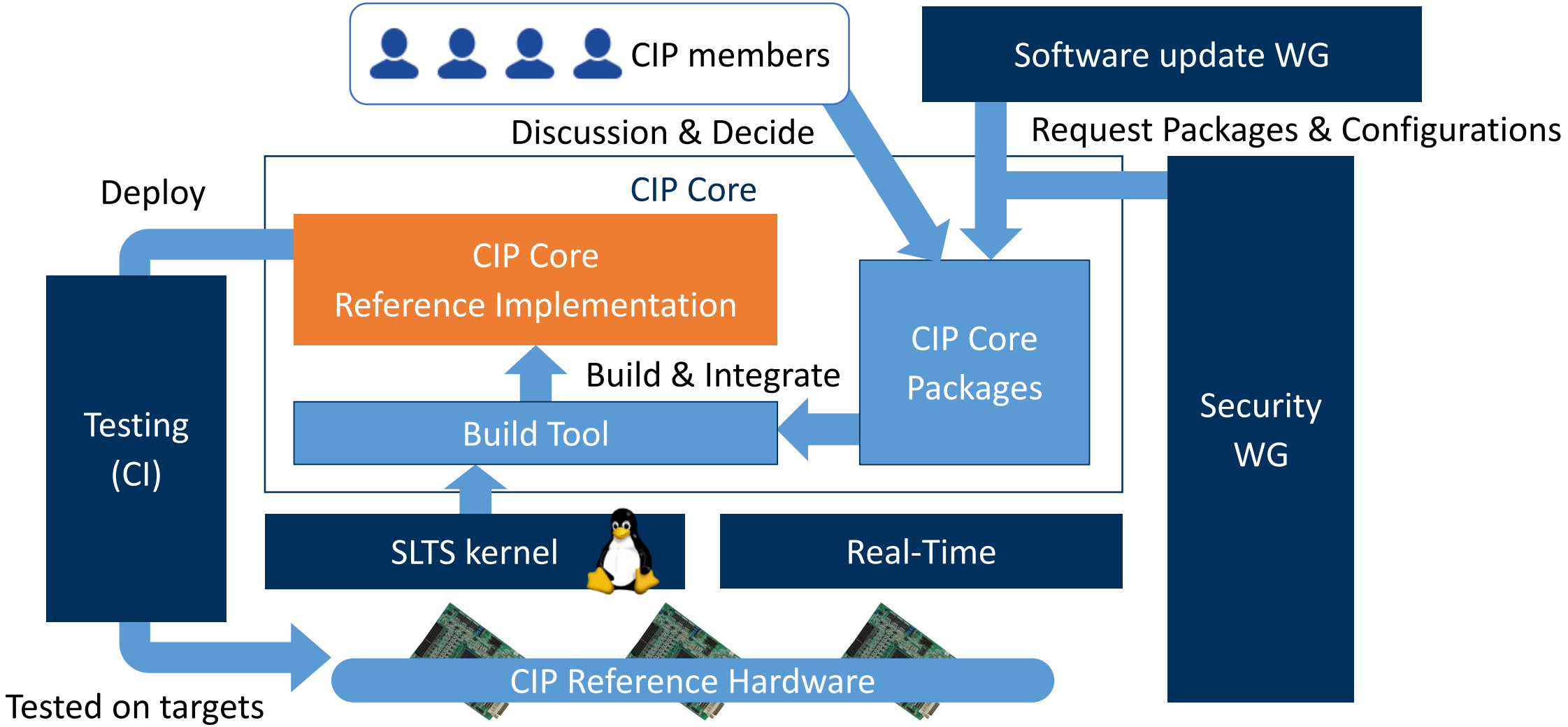
CIP Core

- One of the CIP projects focusing on user land software and tools
- Goals
  - Define a list of “CIP Core packages” maintained for long-term
  - Provide a reference implementation including “CIP Core packages”
  - Test the implementation on the “CIP reference hardware”



CIP Projects and its scopes

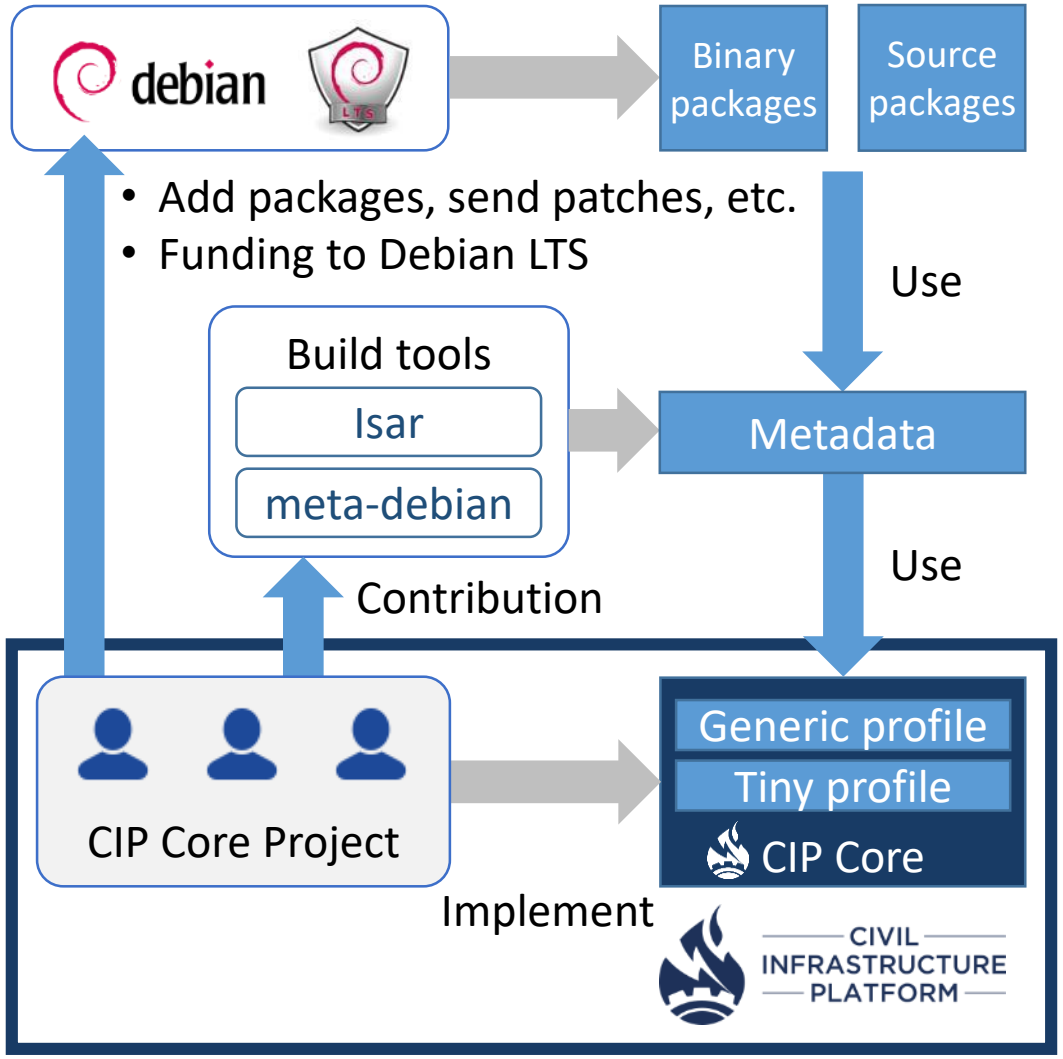
CIP Core: Position in CIP Projects



CIP Core: Implementation

- Debian-based implementation
  - Mature, high-quality, mainstream distro.
  - Many new & old architecture supports
  - Suitable for small and big installations
  - Security updates
- Profiles

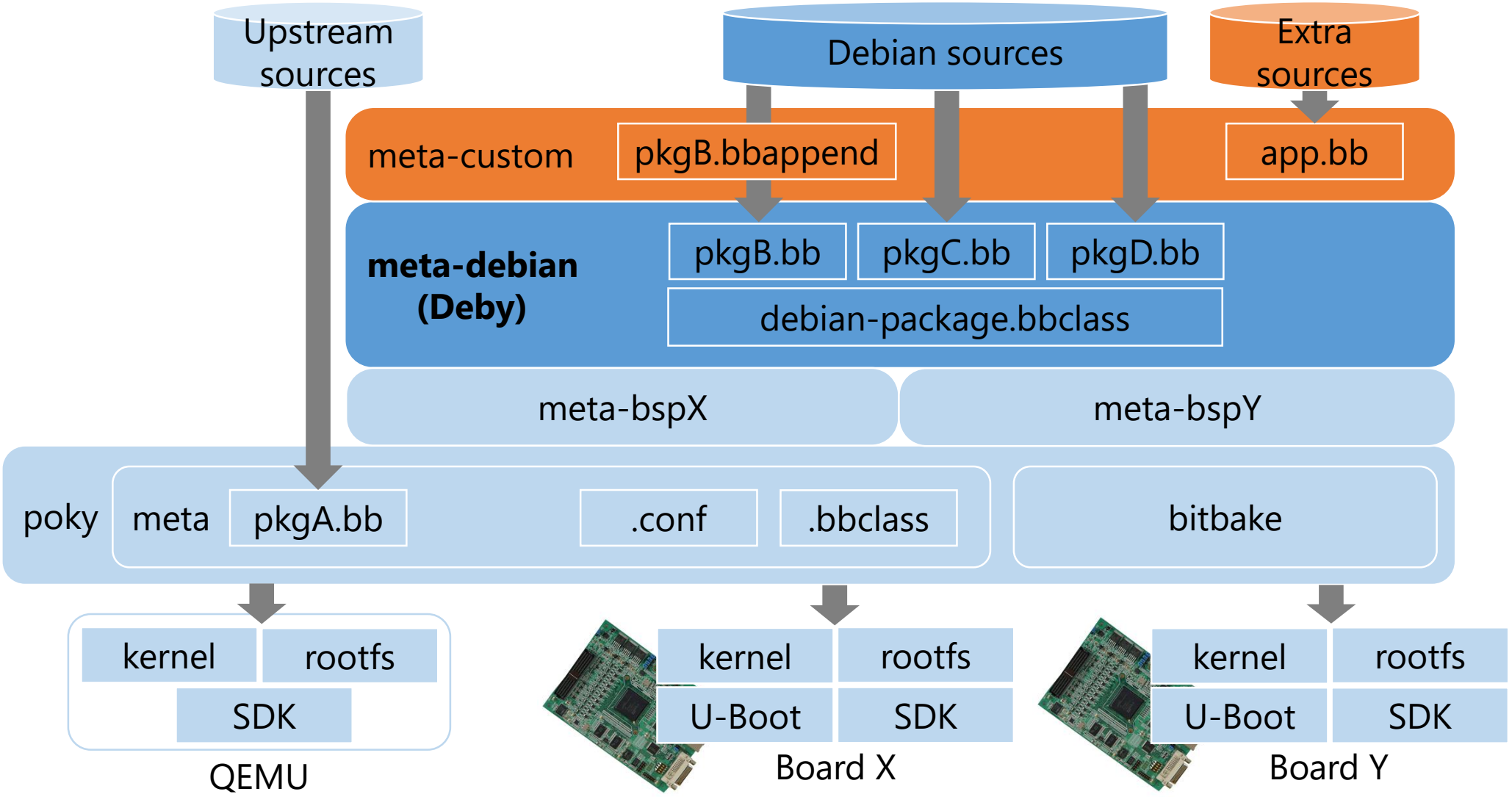
	Generic profile	Tiny profile
Approach	Binary packages	Source packages
Tool	Isar	Deby



Deby (meta-debian)

- Yocto Project extension for using Debian source packages
  - Source code: Debian
  - Build system: Yocto Project
- Goals
  - Achieve stability and long-term support with the Yocto advantages
- Features: Yocto based flexibility & extensibility
  - High customizability by own recipes
  - Small footprint (Around 2MB)
  - Various target CPUs and tunings
  - Adaptation to BSP layers provided by board vendors
- Repositories
  - Upstream: https://github.com/meta-debian/meta-debian
  - CIP Core: https://gitlab.com/cip-project/cip-core/deby

Deby: How it works

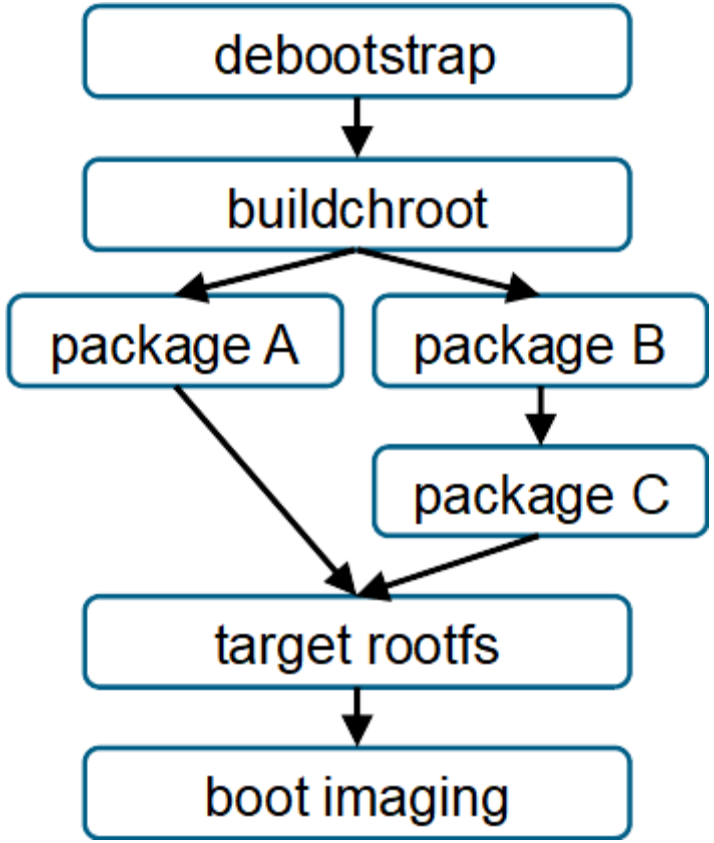


Isar

- **I**ntegration **S**ystem for **A**utomated **R**oot filesystem generation
  https://github.com/ilbers/isar
- Goals
  - Build systems in a Debian way
  - Developer-centric workflow: One-command building
  - Make customizations easy and repeatable
  - Efficient building
- The best of both worlds
  - Debian: Tested binary packages, tools, security updates
  - OpenEmbedded / Yocto: bitbake, recipes, layers
- Reuse Yocto knowledge of developers

Image Generation Sequence of Isar

1. debootstrap Debian for target,
   also for host if cross-building
2. Create buildchroots (target and host)
3. Build custom Debian packages
   - pre-debianized packages
   - ad-hoc debianized packages (customizations, u-boot, kernel, …)
4. Assemble rootfs
   - debootstrap output
   - external packages
   - self-built packages
5. Run images (typically wic)
   - Filesystem image generation
   - Partitioning
   - Bootloader installation and configuration



Example: Building Images for BeagleBone Black

- Generic profile (Isar)

```
$ git clone https://gitlab.com/cip-project/cip-core/isar-cip-core && cd isar-cip-core
$ wget https://raw.githubusercontent.com/siemens/kas/master/kas-docker
$ chmod a+x kas-docker
$ ./kas-docker --isar build kas.yml:board-bbb.yml
$ dd if=/path/to/cip-core-image-cip-core-buster-bbb.wic.img of=/dev/mmcblk0 ...
```

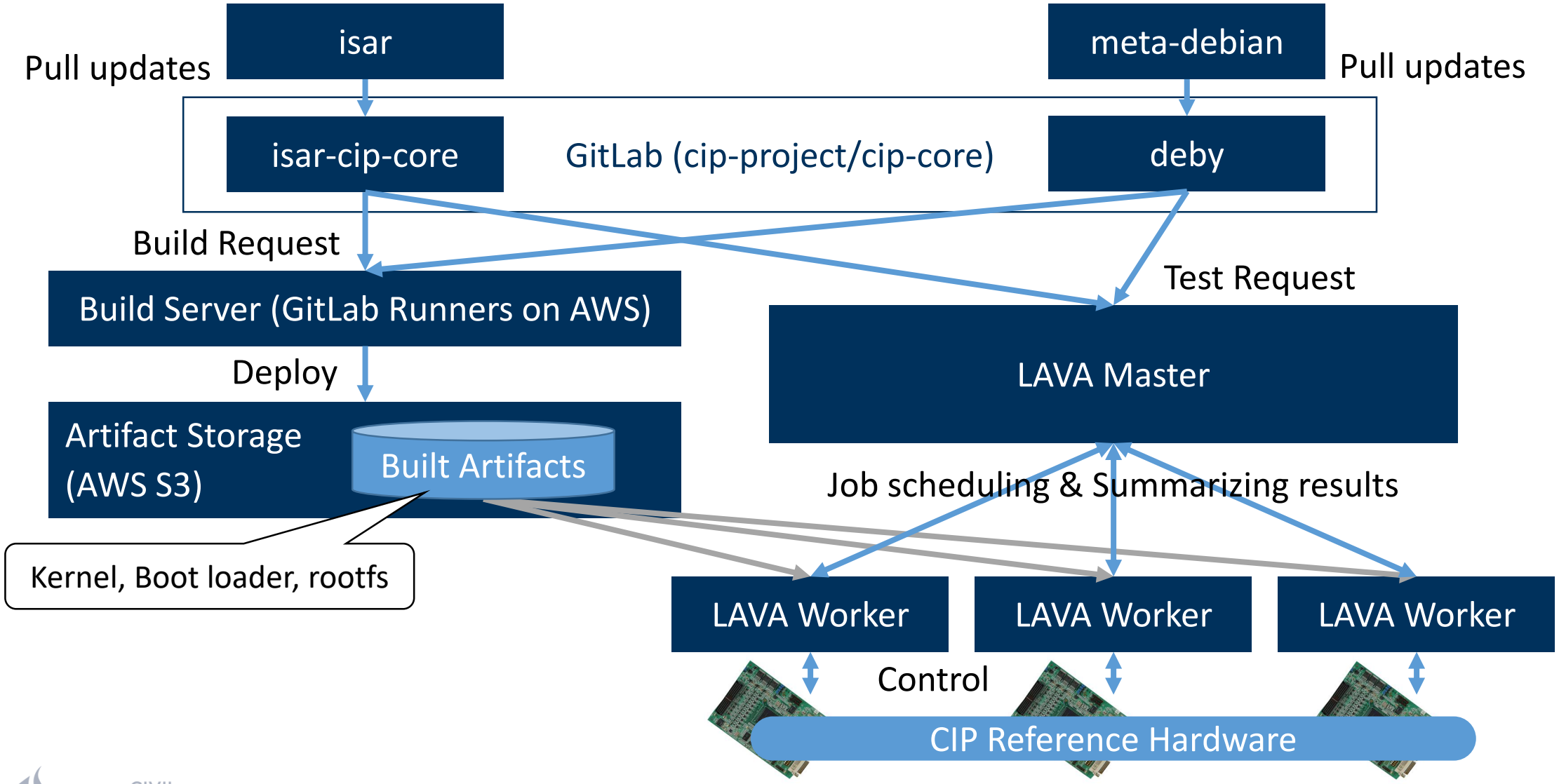
- Tiny profile (Deby)

```
$ git clone https://gitlab.com/cip-project/cip-core/deby && cd deby
$ ./scripts/setup-kas-docker.sh
$ ./kas-docker build kas-bbb.yml
```

Preferred Use Cases

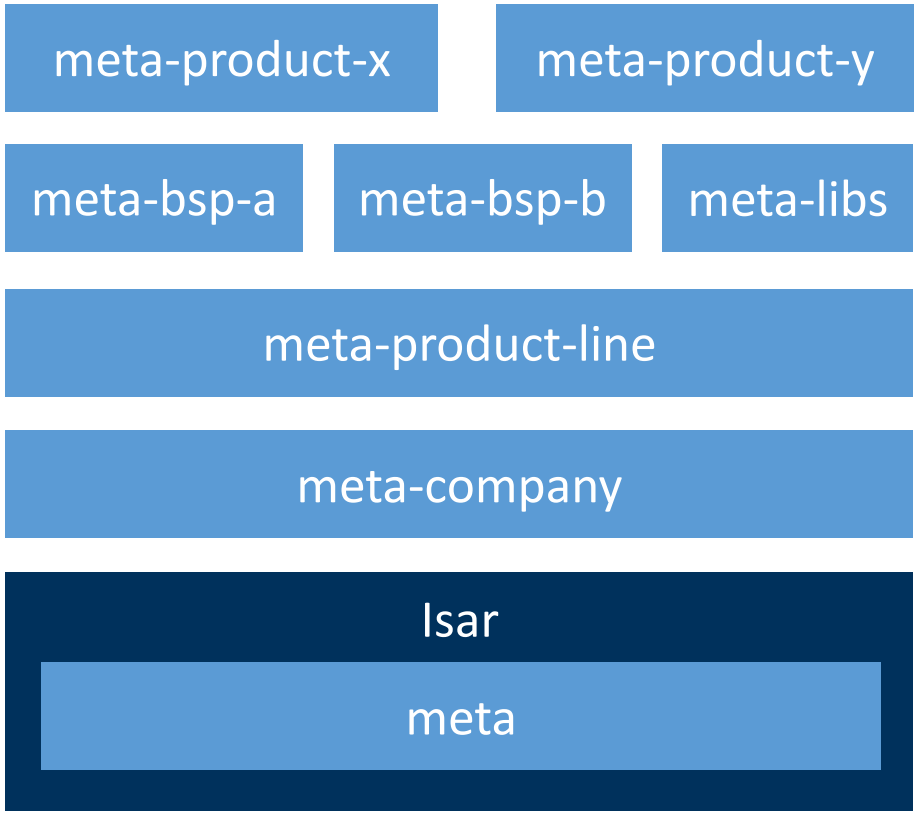
	Isar (Generic Profile)	Deby (Tiny Profile)
Available Packages	= Debian	App. 50 (+ Yocto Extension)
Footprint	> 100MB	2MB - 100MB
Compatibility	Debian (Binary packages)	Yocto Project (Recipes)
Required skill set	Debian (Packaging) bitbake	Yocto Project
Build time (minimal image)	Around 10min	Around 1h
Customization needs	Selected packages	Up to toolchain settings
Fitted systems (Examples)	IoT gateways, edge devices, industrial controllers …	Small IoT devices …

CIP Core: Testing Architecture



Building Products on Top of CIP Core

- Requirements
  - Customize base layer
  - Add product applications
- BitBake layering
  - Append changes to lower layers easily
  - Reuse company / product common elements
  - Available in both CIP Core profiles
- Current approaches in CIP members
  - Building products using this pattern on upstream projects (Isar, Deby)
  - Quicker integration of CIP results desired



(Example of using “Isar”)

Future Plans for CIP Core Implementation

- Enable direct use in product development
  - Regular releases of tested layer with dependencies
  - Mirroring of source & binary dependencies
- Provide image corresponding to CIP package list
- Integrate and test results of other CIP workgroups
  - Robust system update (Software update WG)
  - Functions to meet cybersecurity standard requirements (Security WG)

Summary

- CIP provides long-term maintained Open Source base layer, consisting of kernel and essential packages
- CIP Core defines package set and ensures integration
- Two implementation flavors available
  - Deby for smaller, Yocto/OE-compatible projects
  - isar-cip-core for medium to larger, Debian-compatible projects
- More product-ready features to come, from software update to security hardening

Questions

CIVIL
INFRASTRUCTURE
PLATFORM