Staying Smart: Open Source's Role in Smart City Evolution

Yoshitake Kobayashi, Toshiba Corp., CIP TSC Chair
Urs Gleim, Siemens AG, CIP Board Chair
Embedded World 2020, February 26, 2020
Civil Infrastructure and its Challenges
IoT today – connecting systems

**Connected Cars**
Find and rent cars via smartphone.
Monitor fleets and provide service.

**Industry**
Collect data to improve processes (cost, quality, speed).
Minimize downtimes by predictive maintenance.

**Smart City**
Multimodal transportation, intelligent traffic control, smart energy management, emergency management, ...
<table>
<thead>
<tr>
<th>Transport</th>
<th>Energy</th>
<th>Others</th>
<th>Industry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rail automation</td>
<td>Power Generation</td>
<td>Building automation</td>
<td>Industry automation</td>
</tr>
<tr>
<td>Vehicle control</td>
<td>Turbine Control</td>
<td>Broadcasting</td>
<td>CNC control</td>
</tr>
<tr>
<td>Automatic ticket gates</td>
<td>Turbine Control</td>
<td>Healthcare</td>
<td>Industrial communication</td>
</tr>
</tbody>
</table>
Smart Cities combine consumer & industrial IoT

**Consumer IoT**
End user interfaces and comfort features
- e.g. car
- Permanent cloud connection required.
- Quality and availability: Best effort
- Low-cost / high volume

**Industrial (grade) IoT**
Digital backbone of connected systems
- e.g. traffic control
- Complex systems: local intelligence + centralized intelligence
- 24/7 operation even with no connection to backend.
- Guaranteed latency, throughput, and responsiveness.

Operator companies
- Single or few
- End user

Backend eco-systems
- Multiple users
- with different roles
- at different levels

Controlled network zone
Smart Cities need a Smart Infrastructure
IoT technology to be applied to industrial systems
A Power Plant System:

25-60 years products life-cycle

Very reluctant to perform product updates and upgrades of hardware and base software platform
Security
...for millions of devices
The key challenges

• Apply IoT concepts to industrial systems.

• Ensure quality and longevity of products.

• Keep millions of connected systems secure.

Industrial gradeness

• Reliability
• Functional Safety
• Real-time capabilities

Sustainability

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

Security

• Security & vulnerability management
• Firmware updates
• Minimize risk of regressions
Solving the Key Challenges
Speed and efficiency: focus on differentiating parts

Handling increasing complexity with constant development resources

Join forces by leveraging commodity components, partnering, and adapting open source software.

Open source software ensures long-term availability, flexibility, and maintainability without vendor lock-in.

Differentiating
Why to buy the product

Commodity
Invisible for customers

Proprietary application

2000 – 2015

Up to 2000

Proprietary application, proprietary operating system

Operating System

2016 and beyond

Mobile

HMI frameworks
Augmented reality platforms

Device

App isolation
Data collection
Network

Cloud

IoT backend
Stream processing

Cloud orchestration
Cloud frameworks

Middleware
Communication
Domain-specific frameworks

Enterprise IT interfaces

Network
Monitoring
Virtualization

Data collection
Operating System

Stream processing

Civil Infrastructure Platform
Facts and Issues: Smart City uses Commodity Software

**Facts**
- Millions or trillions smart devices
- Similar software components (e.g. Linux)
- Industrial IoT requirements
  - Security
  - Sustainability
  - Industrial-grade

**Issues**
- A lot of products have to meet IIoT requirements
- Same development and maintenance efforts spent by many companies or even business units
- **No common solution** for base building blocks
CIP is the 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
What is “Open Source Base Layer (OSBL)”?

Layered Linux distribution for industrial products, utilizing and influencing the relevant Open Source projects:

- **CIP Core packages**
  - (tens)
  - (10+ years maintenance, based on LTS kernels)
- **Additional packages**
  - (hundreds)
- **company-specific middleware and applications**
Mapping CIP into the company

Up to 70% effort reduction achievable for OSS license clearing and vulnerability monitoring, kernel and package maintenance, application adaptation and testing for an individual product.

**CIP Kernel**
- (10+ years maintenance)

**CIP Core packages**
- (tens)

**Additional packages**
- (hundreds)

- Corporate team/central project
- Business Units/Products
- Companies/Divisions

**“distribution”**
- Domain-specific extensions
- Domain-specific extensions
- ...

- Firmware Update
- Security Hardening
- Container Runtime
- ...

Base packages, SDK, Build chain, QA

Kernel

OSS  Open Source Software  QA  quality assurance  SDK  software development kit
The backbone of CIP are the member companies.
CIP: add. HW support & extended maintenance

- **Applications**
  - Additional packages *(hundreds)*
  - CIP Core packages *(tens)*
  - CIP Kernel
  - out-of-tree drivers

- **Support Term**
  - From Debian *(3 years)*
  - From Debian LTS *(+2 years)*
  - From Debian ELTS *(+x years)*

- **CIP support** *(10 years + more)*

- **TBD**
  - Work with Debian community

- **Depend on SOC vendors**
  - 5 years
  - 10 years

**Support Term**
CIP lays the Foundation for Sustainable Smart Cities
Scope of activities

- **Domain Specific communication (e.g. OPC UA)**
- **Shared config. & logging**
- **Multimedia**

**Middleware/Libraries**

- **CIP Core Packages**
- **Safe & Secure Update**
- **Monitoring**
- **Real-time / safe virtualization**

**Tools**

- **Build environment** (e.g. bitbake, dpkg)
- **Test automation**
- **Tracing & reporting tools**
- **Configuration management**
- **Device management** (update, download)
- **Application life-cycle management**

**Concepts**

- **Functional safety architecture/strategy**, including compliance w/standards (e.g., NERC CIP, IEC61508)
- **Long-term support Strategy**: security patch management
- **Standardization**: collaborative effort with others
- **License clearing**
- **Export Control Classification**

**On-device software stack**

- **Product development and maintenance**

**Super Long Term Supported Kernel (STLS)**

**Real-time support**

**App container infrastructure** (mid-term)

**App Framework** (optionally, mid-term)
CIP governance structure and projects

Governing Board (GB)

Technical Steering Committee (TSC)

CIP Projects and its scopes

1. SLTS kernel
   - ✔

2. Real-time
   - ✔

3. CIP Core
   - ✔

4. Testing
   - ✔

5. Security WG(*)
   - ✔

6. Software Update WG
   - ✔

(*): Workgroup

- Industrial grade
- Sustainability
- Security

SLTS

Kernel

CIP Core

Testing

Security WG

Software Update WG

CIP Projects and its scopes
Collaborative development with other OSS projects

Contribute, Collaborate and use by CIP

- mainline
- LTS
- debian
- LAVA
- Reproducible Builds
- yocto project
- Real-Time LINUX
- KernelCI
- EDGEXFOUNDRY

Upstream Projects

1. Upstream first

Use the upstream code

2. Use the upstream code

Integrate

3. Integrate

CIP Open Source Base Layer (OSBL)

Contributing by CIP members as future candidates

- Jailhouse
- Fuego
- hawkBit
- fossology
- 360

Upstream first

1. Upstream first
Collaborative development with other OSS projects

Contribute, Collaborate and use by CIP

- mainline
- LTS
- Reproducible Builds
- yocto project
- KBase
- EDGEXFOUNDRY

Upstream Projects

- Use the upstream code

CIP Open Source Base Layer (OSBL)

Contributing by CIP members as future candidates

- Jailhouse
- Fuego
- hawkBit
- fossology

Upstream first

Integration

Upstream first
CIP SLTS kernel development (Upstream first development)

**Goal**
- Providing CIP kernels with more than 10 years maintenance period
  - Super Long Time Stable kernel

**Mentor**: Ben Hutchings

**Maintainers**: Nobuhiro Iwamatsu, Pavel Machek

**Status**
- CIP SLTS kernels has been released
  - V4.19.78-cip12 (October 12th)
  - V4.4.196-cip38 (October 12th)
- Created CVE tracker
- Participate to LTS review process

**Resources**
- [https://git.kernel.org/pub/scm/linux/kernel/git/cip](https://git.kernel.org/pub/scm/linux/kernel/git/cip)
- [https://gitlab.com/cip-project/cip-kernel/cip-kernel-sec](https://gitlab.com/cip-project/cip-kernel/cip-kernel-sec)
Real-time Linux development (PREEMPT_RT)

Goal
- CIP joins RT Linux project as Gold member to Work with them to upstream Real-time enhancement
- Provide CIP SLTS kernel with real-time enhancement by using RT patch

Current status
- CIP SLTS RT kernels has been released
  - v4.19.72-cip10-rt3 (October 2\textsuperscript{nd})
  - 4.4.190-cip36-rt25 (October 3\textsuperscript{rd})
  - https://git.kernel.org/pub/scm/Linux/kernel/git/cip
Goal

- Provide a reference implementation with CIP core packages for testing
- Following implementations are provided
  - Tiny profile ➤ E.g. Small IoT devices
  - Generic profile ➤ E.g. IoT gateways

Status

- CIP Core profiles are available
  - https://gitlab.com/cip-project/cip-core
CIP Testing

Goal

- Providing a test environment to test the CIP kernel and CIP Core

Current status

- Moved to distributed testing environment on AWS with LAVA
- Integrated with GitLab-CI

Diagram:

1. Publish results (Planned)
2. Integration
3. CIP Testing project

Steps:

1. up-streaming
2. use
3. integrate
CIP Testing (Architecture with CIP Core)

Profiles

- Generic
  - isar-cip-core
  - CIP Kernel
  - Debian
  - deby

- Tiny

Implementations

Build and Test

Artifact Storage (AWS S3)

Built Artifacts

GitLab Runners (Image builder)

LAVA Worker

LAVA Master

CIP Reference Hardware
Security working group

Goal

- Provide guidelines and reference implementations to help developers to meet cybersecurity standard requirements (IEC 62443)

Status

- Started for feasibility study

* this image represents the planning and is for illustrative purpose only
6 Software update working group

Goal

- Incorporate a common solution for software updates into CIP core
  - Device management
  - Deployment
  - Safe update

Status

- Selected OSS update tools
Summary

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 very long-term maintenance including the required build tool chains
• **Security**: Improving to have security features and to follow Cyber Security Standard
• **Software update**: Incorporate a common solution for software updates into CIP core
• **Collaboration**: Linux, Debian/Debian-LTS, Real Time Linux, Reproducible Builds, 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, Real-time Linux, Reproducible builds, KernelCI, …)
  - Providing suitable tool chains
  - Ensuring in-depth tests

- Contribution and collaboration with upstream projects are the key CIP activities to make sustainable infrastructure
Join us

CIP for sustainable Smart Cities with Open Source Software

CIVIL INFRASTRUCTURE PLATFORM

RENESAS  SIEMENS  TOSHIBA

cybertrust  HITACHI

MOXA

Plat’Home
Contact Information and Resources

To get the latest information, please contact:

- CIP Mailing list: cip-dev@lists.cip-project.org

Other resources

- Twitter: @cip_project
- CIP web site: https://www.cip-project.org
- CIP news: https://www.cip-project.org/news/in-the-news
- CIP wiki: https://wiki.linuxfoundation.org/civilinfrastructureplatform/
- CIP source code
  - CIP GitLab: https://gitlab.com/cip-project
  - CIP kernel: git://git.kernel.org/pub/scm/linux/kernel/git/cip/linux-cip.git
Thank you