How to make Smart Cities stay smart with Open Source Projects

Yoshitake Kobayashi, Toshiba Corp., CIP TSC Chair
Open Source Summit China, Shanghai, June 25-26, 2019
IoT today – connecting systems

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

**Smart City**
Multimodal transportation, intelligent traffic control, smart energy management, emergency management, ...

**Industry**
Collect data to improve processes (cost, quality, speed). Minimize downtimes by predictive maintenance.
Challenges when applying IoT to Civil Infrastructure
## “Hidden” Industrial IoT Systems

<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></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
- Single or few operator companies
- End user

*Permanent cloud connection required.*
*Quality and availability: Best effort*
*Low-cost / high volume*

**Industrial (grade) IoT**
Digital backbone of connected systems

- traffic control
- Backend eco-systems
- Controlled network zone
- Multiple users with different roles at different levels

*Complex systems: local intelligence + centralized intelligence*
*24/7 operation even with no connection to backend.*
*Guaranteed latency, throughput, and responsiveness.*

Open Source Summit China 2019
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 grade**
- 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
Open Source
meets
Industrial Challenges
What is Open Source Software (OSS)

Source: Wikimedia Commons
(Creative Commons Attribution 2.0 Generic license)
Example: Linux Foundation
Example: Linux Foundation

The Linux Foundation is a Critical Part of the Tech Ecosystem

1320+ Members From 41 Countries
100% of Fortune 100 Tech & Telecom
30,000+ Developers Contributing Code
150+ Open Source Projects
$15.7B Shared Value

In 2018, the Linux Foundation added a new member every day.
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

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

Up to 2000
Proprietary application, proprietary operating system

2000 – 2015
Proprietary application
Operating System

2016 and beyond

Proprietary application
Middleware
Domain-specific frameworks
HMI frameworks
Augmented reality platforms
Mobile
Enterprise IT interfaces

Device
App isolation
Data collection
Operating System
Virtualization
Monitoring

Cloud
IoT backend
Stream processing
Cloud frameworks

Cloud orchestration
How to solve the Key Challenges
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)
- additional packages (hundreds)
- CIP kernel (10+ years maintenance, based on LTS kernels)
- company-specific middleware and applications
Mapping CIP into the company

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

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.
The backbone of CIP are the member companies

Open Source Projects (Upstream work)
CIP lays the Foundation for Sustainable Smart Cities
Scope of activities

User space

- App container infrastructure (mid-term)
  - Domain Specific communication (e.g. OPC UA)
  - Safe & Secure Update
  - Real-time support

- Middleware/Libraries
  - CIP Core Packages
  - Monitoring
  - Security
  - Real-time / safe virtualization

- Kernel space

- App Framework (optionally, mid-term)
  - Shared config. & logging

- Linux Kernel
  - Super Long Term Supported Kernel (STLS)

- 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

Open Source Summit China 2019
CIP governance structure and projects

Governing Board (GB)

Technical Steering Committee (TSC)

1. SLTS kernel
2. Real-time
3. Testing
4. CIP Core
5. Security WG(*)
6. Software update WG

(*) Workgroup

CIP Projects and its scopes

- Industrial grade
- Sustainability
- Security
Collaborative development with other OSS projects

Contribute, Collaborate and use by CIP

Upstream Projects
- mainline
- LTS
- Real-Time LINUX
- debian
- LAVA
- KernelCI
- Reproducible Builds
- yocto project
- EdgeX Foundry

Contributing by CIP members as future candidates
- Jailhouse
- Fuego
- hawkBit
- 360

1. Upstream first
2. Use the upstream code
3. Integrate

CIP Open Source Base Layer (OSBL)
CIP SLTS kernel development (Upstream first development)

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

• Status
  • CIP SLTS kernels has been released
    • Linux 4.4.166-cip29
    • Linux 4.19.13-cip1
    • https://git.kernel.org/pub/scm/linux/kernel/git/cip
  • CIP kernel team participate into LTS review process

Mainline / LTS

Feature mainlining
Review and test results/ Fixes

Mentor / Maintainer
Developers

CIP kernel team

LTS kernel

1 up-streaming
2 use
3 integrate
Real-time Linux development (PREEMPT_RT)

- **Goal**
  - Provide CIP SLTS kernel with real-time enhancement (RT_path)
  - Work with Real-time Linux Project to standardize Real-time enhancement

- **Status**
  - CIP SLTS RT kernels has been released
    - Linux 4.4.166-cip29-rt21
    - Linux 4.19.13-cip1-rt
    - [https://git.kernel.org/pub/scm/linux/kernel/git/cip](https://git.kernel.org/pub/scm/linux/kernel/git/cip)
  - Test results are available on CI-RT
    - [https://ci-rt.linutronix.de/RT-Test/](https://ci-rt.linutronix.de/RT-Test/)

Open Source Summit China 2019
3 CIP Testing

• Goal
  • Providing a test environment to test the CIP kernel and more
  • Single developer can test at their desk

• Current status
  • The first version B@D already released
  • [https://gitlab.com/cip-project/board-at-desk-single-dev](https://gitlab.com/cip-project/board-at-desk-single-dev)
• Goal
  • Provide a reference implementation with CIP core packages for testing
  • Following implementations will be provided
    • Tiny profile
      • eg. Small IoT devices
    • Debian profile
      • eg. IoT gateways

• Status
  • CIP Core Tiny profile has been released
    • https://gitlab.com/cip-project/cip-core
  • PoC implementation is available for Debian profile
    • https://gitlab.com/cip-playground/isar-cip-core
## Security working group

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

### Status
- Started for feasibility study
- A demonstration scheduled at OSS Japan in July

### Components

<table>
<thead>
<tr>
<th>Component</th>
<th>H/W</th>
<th>S/W</th>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>User equipment</td>
<td>Certified platform (e.g. Reference board)</td>
<td>User application</td>
<td>User manual</td>
</tr>
<tr>
<td>Middleware (e.g. Device driver)</td>
<td>Linux Kernel (CIP)</td>
<td>Application (sample)</td>
<td>Design document</td>
</tr>
<tr>
<td>Compliant environment for evaluation</td>
<td>Equipment for evaluation (TBD.)</td>
<td>Test cases</td>
<td>Evaluation document</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reference implementations</td>
<td></td>
</tr>
</tbody>
</table>

### CIP deliverables*

- Reference implementations

---

*: Noted that this image is under planning and for only illustrative purposes.
Software update working group

• Goal
  • Incorporate a common solution for software updates into CIP core

• Status
  • Just started
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, PREEMPT_RT, KernelCI, …)
  • Providing suitable tool chains
  • Ensuring in-depth tests

• **Contribution and collaboration with upstream projects are the key CIP activities**
Join us

CIP for sustainable Smart Cities with Open Source Software
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
Question?
Thanks for your attention!