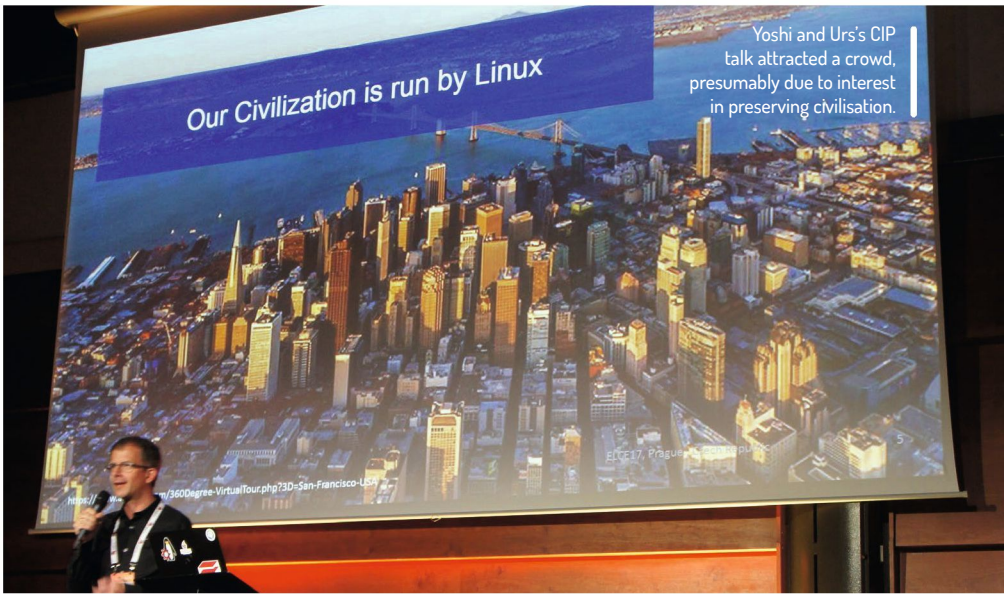
CIVILISATION RUNS ON LINUX

Jonni Bidwell always suspected Linux would save the world. Industry experts **Yoshitake Kobayashi** and **Urs Gleim** all but confirmed his hunch...



Yoshi and Urs's CIP talk attracted a crowd, presumably due to interest in preserving civilisation.

The Civil Infrastructure Platform (CIP) is a Linux Foundation initiative. It aims to establish a base layer of industrial-grade software to power critical services such as energy, water, transportation and communications – the lifeblood of today's civilisation.

Many of these projects run on open source software, and many more will do so in the future. Yet it's completely unfeasible to update the software running these things every five years (the current life-span of LTS distros), and many of these systems are looking at life-spans beyond 50 years. So the CIP introduces the idea of a super long-term support (SLTS) kernel.

Linux Format's Jonni Bidwell caught up with Toshiba's Yoshi Kobayashi and Urs Gleim, head of the Central Smart Embedded Systems Group at Siemens AG at the Linux Foundation's Open Source Summit in Prague in October 2017. There, he got the lowdown on how the CIP hopes to keep its kernel and base layer "industrial-grade". Since then, there have been a number of key developments, so we've summarised those, too.

Linux Format: Linux is running in all kinds of places, and lives depend on some of those applications. What is the Civil Infrastructure Platform and how is it going to help civilisation going forward?

Yoshi Kobayashi (YK) and Urs Gleim: (UG): Yes, early on in our presentation we have a slide entitled Our Civilization is Run by Linux, and it's not an exaggeration. Things like railway infrastructure, healthcare and industrial automation, these all have long-running systems. We're talking between 10 and 40 years, maybe even longer.

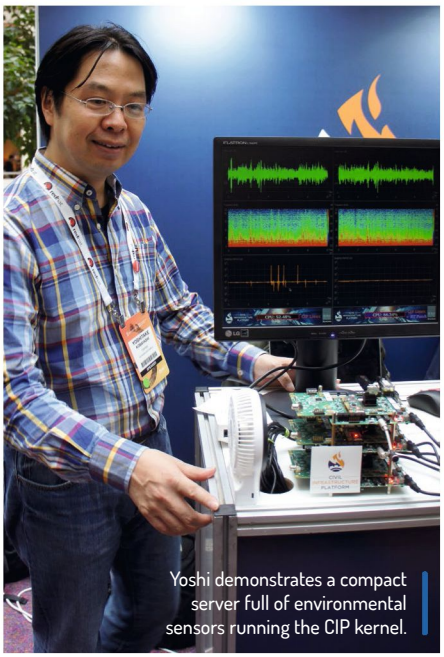
So we can't afford to change the software, say, every two years (as a cautious desktop user might). This is especially the case where safety certifications are involved – transport networks and power generation for example. Here, it can take close to two decades just to put a new system into service. So a better strategy is to apply security patches and small updates.

For CIP the idea is to stick with one version of the Linux kernel and maintain it for as long as we can. The CIP kernel is much more focused on embedded devices than other long-term initiatives. So we have support for all the embedded board ports, but in most cases don't support desktop PCs or servers. The systems we're dealing with all run on dedicated hardware and embedded chipsets, so that's what we support.

Many of these kind of systems have already been using Linux for some time, and individual companies have already been working on their own super long-term maintenance. What we want is everyone working on the same platform, which will avoid some duplication of effort. But most importantly, we want this work to be done collaboratively with the upstream communities, not locally.

LXF: When did the project start and who were the initial industrial backers?

YK: CIP started in April 2016 and the initial supporters were Hitachi, Siemens and Toshiba. Since then, a number of other


Yoshi demonstrates a compact server full of environmental sensors running the CIP kernel.

» WHAT'S NEW?

This interview happened in October 2017, so here's a summary of some developments that have happened since then...

A new company, Moxa, joined the CIP in January. It's an edge-to-cloud connectivity provider that provides solutions for factory automation, smart cities and monitoring. Also in January you'll probably remember the Spectre and Meltdown attacks – which affect embedded ARM and Intel hardware as well as desktop CPUs – were disclosed. So the relevant patches are being backported to the CIP SLTS kernel. Some hardening features from the Kernel Self-Protection Project have been added, too.

We've also had a few CIP kernel releases. In April Ben Hutchings released the latest one: 4.4.126-cip22. Kernel 4.4 was released in January 2016, and was initially earmarked for long-term support until February 2018. That was extended in September 2017 to February 2022. It's possible that someone else will maintain it after this period (as happened with Ben Hutchings' tour de force support of the 3.2 kernel, which after six years came to an end in May). This would in some sense make the CIP team's lives easier, but is far from a free ride.

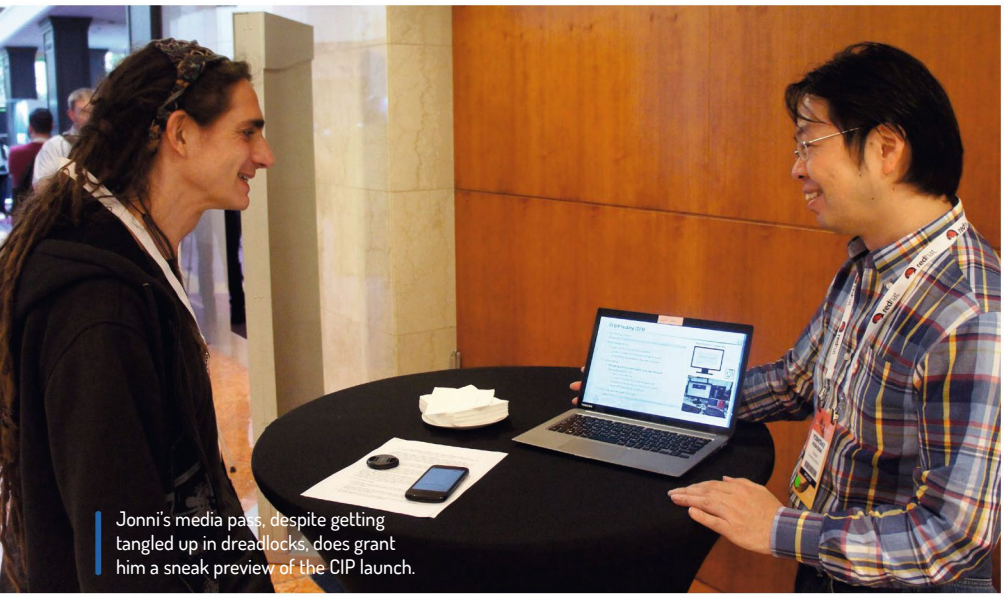
Keep up with developments at the Linux Civil Infrastructure Platform by visiting **www.cip-project.org**.

partners have come on board: Renesas Electronics, Codethink and Plat'Home. They support CIP by contributing directly to upstream projects and funding work related to the CIP's goals.

LXF: How do you go about testing and certifying different boards? This sounds like an awful lot of work...

YK: These systems run on lots of different hardware platforms, so each one needs to be individually tested and certified. But there's a lot of similarity across the tests that each board needs to run. So for example, the kernel has to be tested on every board, and there's a common software stack that runs these tests. So yes, we have to test all the different hardware, but by sharing these common parts, the workload is reduced.

UG: Part of the CIP's role is to produce this testing infrastructure too, so everyone is using the same test tools, the same test automation. It's important to harmonise all these testing infrastructures, because once that's in place it's not that much work to support additional dedicated boards. There are a few talks on this, one of the projects is called Board at Desk (B at D – see box, *below*), which enables you to easily set up a development environment on your local workstation, which is connected via serial link to a board. B at D can currently be tested on our reference board: the Beaglebone Black and the Renesas RZ/G1M.



Jonni's media pass, despite getting tangled up in dreadlocks, does grant him a sneak preview of the CIP launch.

LXF: Security is obviously a big issue for these kind of systems. How are you going to tackle this?

YK: We'll port all the security patches from the Linux kernel community to our CIP kernel and test them on the relevant hardware – hopefully that's not too onerous.

The other concern is userland, above the kernel layer. We have what we call the CIP base layer, which is our common software stack. We want as much of this stack to be shared across the hardware. This would be too much work for just one team, so collaboration is important here: collaboration with both Linux distributions and other projects.

UG: The most work in terms of security is done in the server area. Google and other big companies are the major contributors here. There are a few relevant projects, the Core Infrastructure Initiative (CII, see **www.coreinfrastructure.org**) and the Kernel Self Protection Project (see **https://kernsec.org/wiki/index.php/Kernel_Self_Protection_Project**) for example. We take things from those and backport them to the CIP kernel. The CIP kernel is based on Kernel 4.4, and we backport security patches and features from newer kernels to it.

LXF: Reproducible builds, particularly in Debian, have been getting a lot of attention. Might these be used in the CIP? What extra tooling and testing would be required?

YK: Reproducible builds is a future topic for the CIP Project. We haven't started work on this, yet but plan to soon. Currently, we're focusing on Debian activities and long-term support. In the next year, you'll see more collaboration and strategy with Debian and plans for both groups: work done in CIP context and/or supporting the Debian efforts on this.

LXF: The release of CIP Core was just announced at this conference [in October 2017]. Is this like a distribution for your kernel efforts?

» BOARD AT DESK (B AT D)

LXF: Tell us more about Board at Desk

YK: B at D is a VM that provides an easy way for developers to test Linux kernels from their local workstations. We don't create everything for this project. It uses Kernel CI and LAVA – both open source projects used by Linaro and the kernel community – which we put together on this Debian VM to create a test infrastructure.

Kernel CI is an automated system for the building of Linux kernels linked to large board farms. This enables each company to use a common test environment that can be used easily (they can test boards from their desks), and it's quite good for us. We can share the same test cases and also share results for the CIP kernel. Shared and trusted testing is an important philosophy for us. Not everyone has access to all boards, and we needed to get away from the centrally managed nature of **kernelci.org**. Furthermore, Kernel CI and LAVA are both powerful tools, but they're complicated beasts too. We hope that B at D will reduce the entry barrier to these two projects.

We released B at D v1.0 just before this conference. The main changes are that we've updated the VM from Debian Jessie to Stretch. Previously, we had some packages from Debian Testing in there, which led to some instabilities and inconsistencies. We have a new version of LAVA, too: 2017.7.

All of this updating work has been done by the CIP testing team at Codethink. We also support Windows as a host OS, but that probably doesn't interest you! What might interest you is that we now build the initramfs locally. This used to happen remotely on Linaro servers, which meant that we couldn't test without internet connectivity. Now we can, and we can do so faster.

UG: We're focused on the kernel first. The CIP Core sits on top of that and provides a minimal reference filesystem. It's not a proper distribution by any stretch of the imagination, mind. We have only the basic packages, a shell, some protocols (provided by OpenSSL), a bootloader, glibc, busybox, binutils and compilers. That's really it for the moment, so you can see it's not really comparable to Ubuntu or something.

The CIP core is the lowest common denominator; individual systems will add whatever they need. The release is like a first milestone to create our software stack. This gives us our kernel and candidates for userland packages. It's based on Debian source code and binaries. We decided to collaborate with the Debian community because they already provide long-term support.

Note that the community provides support for five years (through the Debian LTS effort), whereas CIP needs to be supported for at least ten years. We need to work with the community to bridge that support gap. Once we gain experience with this kind of support longevity, future kernels will have even longer support periods. CIP is a platinum-level supporter of Debian LTS.

LXF: Other Internet of Things projects have been talking about doing updates atomically, rather than per-package. Can you say if this approach be adopted by CIP Core?

YK: At the moment that sort of thing would have to be implemented on top of the CIP base layer, so we're not strongly focused on that at the moment. But IoT use cases, and package/firmware updates in general, are important, so we may introduce something like this in future.

UG: Yes, this will be an option for projects later on. Internally, in our company for example, we have different solutions depending on the projects: some based on packages, some based on binary updates with *bindiff*, and so on.

LXF: There's mention of real-time patches at http://bit.ly/cip-kernal-patchset. Can you elaborate on this? Is there an official real-time CIP kernel on the way?

UG: Yes, we have a real-time patch at **https://git.kernel.org/pub/scm/linux/kernel/git/wagi/linux-cip-rt.git**. Since the CIP kernel is based on Kernel 4.4 and there's a version of the PREEMPT_RT patches being maintained for 4.4, maintaining a variation of that 4.4 stable-rt patchset is possible without too much overhead. The CIP real-time patchset will be followed by the 4.4 stable-rt patchset as closely as possible. The stable-rt patchset won't gain new features (such as hrtimer rework, cpu hotplug rework and no_hz fixes) because backporting has a high risk of breaking stable-rt. Therefore, the stable-rt maintaining goals overlap with the cip-rt goals, which enables us to keep the variations of the real-time patchset smaller.

MAINTAINING PROJECT MOMENTUM

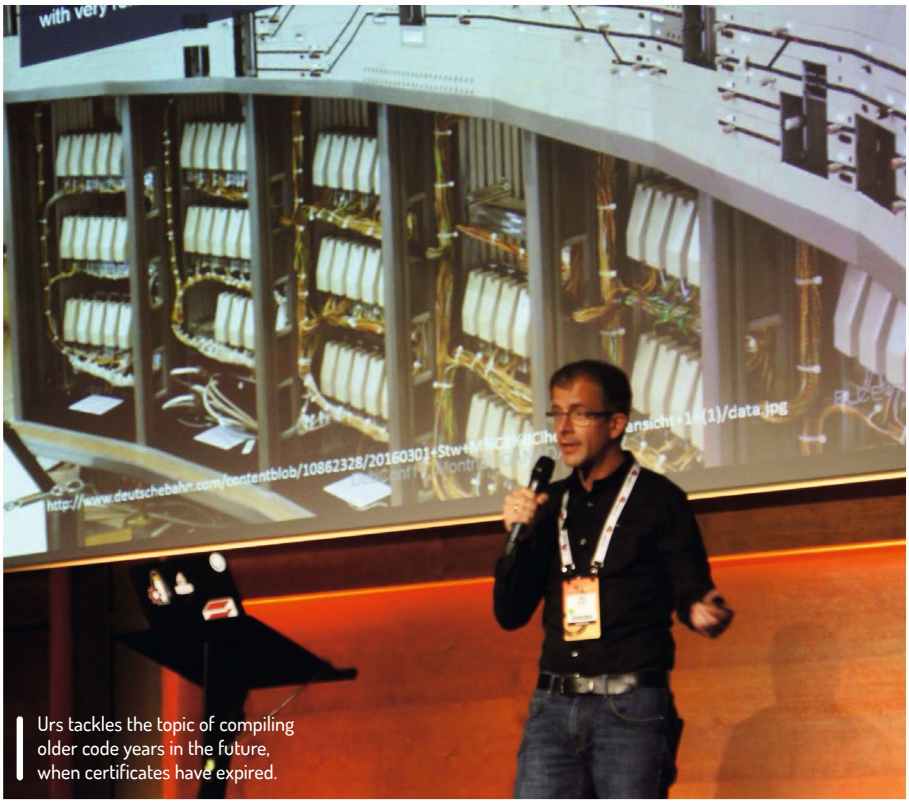
"When we have silicon vendors and their board support packagers on board, this really helps the platform get yet more distribution and support"

LXF: So you two are representing Toshiba and Siemens, both huge companies. Who else is on board here?

UG: We have Hitachi, Codetshink (a Manchester-based company that's very active in Linux development) and Plat'Home (a Japanese IoT company). The latest member is Renesas, which is the first silicon vendor to step in and so this is a very important step for CIP. When we have silicon vendors and their board support packagers on board, this really helps the platform get yet more distribution and support.

LXF: Renesas is very much involved with Automotive Grade Linux (AGL) too. The goal there is to provide a standard Linux base for use in automotive things, mainly in-vehicle infotainment (IVI) systems at the moment. Is it fair to say the CIP is trying to do a similar thing, to standardise the platform for infrastructure type things?

YK: In a way that's right. The main difference is that AGL is trying to create a big software stack as soon as possible and share it. It's not so much focused on the maintenance. We want to start using the CIP platform in actual products as soon as possible, so we need to be focused on reliability and stability. Super long-term support is something completely new, and there's a lot we need to figure out. LXF



Urs tackles the topic of compiling older code years in the future, when certificates have expired.