OSS Activities through EMLinux Development

Hiraku Toyooka
Cybertrust Japan Co., Ltd.

2021.05.27

Outline

- Security updates for embedded Linux
- What is EMLinux?
- OSS contributions from Cybertrust
- Future work

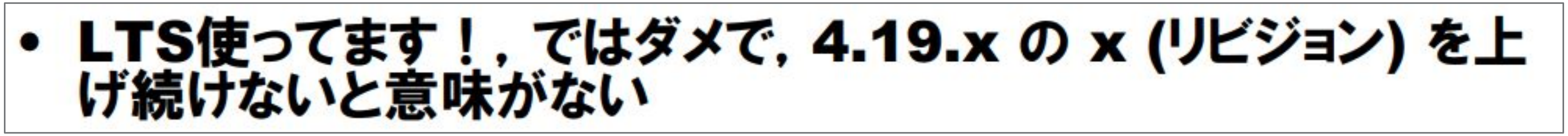
$ whoami

Hiraku Toyooka <hiraku.toyooka@miraclelinux.com>

- Engineering Management for EMLinux @ Cybertrust Japan
- Contributing to CIP Testing WG, meta-debian, etc.
- Maintainer of meta-emlinux, meta-debian-extended

Security updates for embedded Linux

- Security updates are becoming recognized as essential, but...
- Some difficulties to deliver the update for final products
  - There are massive out-of-tree patches, which make it difficult to backport community’s fixes
    - Most of these typically come from SoC vendor’s BSP
    - + your own code for a custom board
  - QA process with every update is costly



• LTS使ってます！，ではダメで，4.19.x の x (リビジョン) を上げ続けないと意味がない

Shinsuke Kato, “Linux Kernel のバージョンとLongterm Stable Kernel (LTS)”, Japan Technical Jamboree 70 (2019)

“There are massive out-of-tree patches”

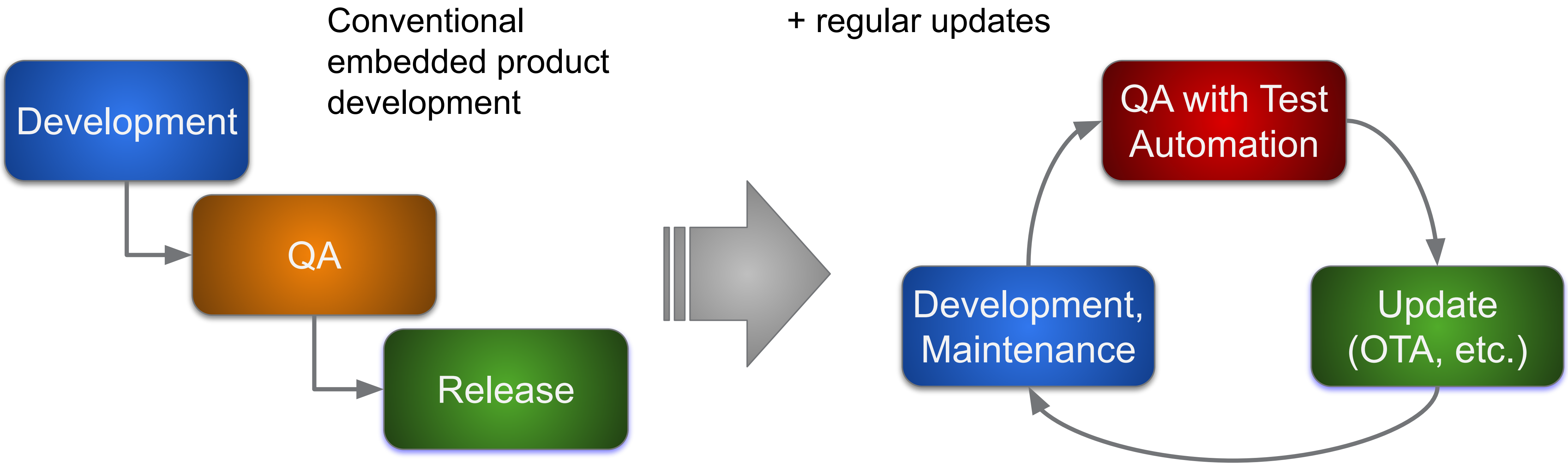
No perfect solution, but we might be able to …

- minimize out-of-tree code
  - select SoC/device which its support code is merged into upstream enough
  - use upstream code (or do upstreaming the code), if possible
- rebase out-of-tree code onto community’s latest release
  - Option 1: rebase onto latest version/revision (rolling update)
    - Fixes will be available earliest
    - Latest version includes feature changes, which may require changes to the product code
    - Some SoC vendors provide BSP upgrades every 1~ year
  - Option 2: rebase onto latest LTS branch, if available
    - LTS branch only accepts bug fixes, little impact on product code
    - You need to upgrade to latest version after the LTS period is over

“QA process with every update is costly”

We should start automating some part of the QA process

- Test automation would be the first candidate



What is EMLinux?

- Embedded Linux environment using Yocto build system
  - (There are Community Edition and Product Edition)

- Purpose
  - continuously deliver security-fixes and bug-fixes
- How?
  - based on LTS model
  - leverage CIP SLTS kernel, meta-debian, Debian source pkgs
  - with Test Automation
  - upstreaming bug-fixes

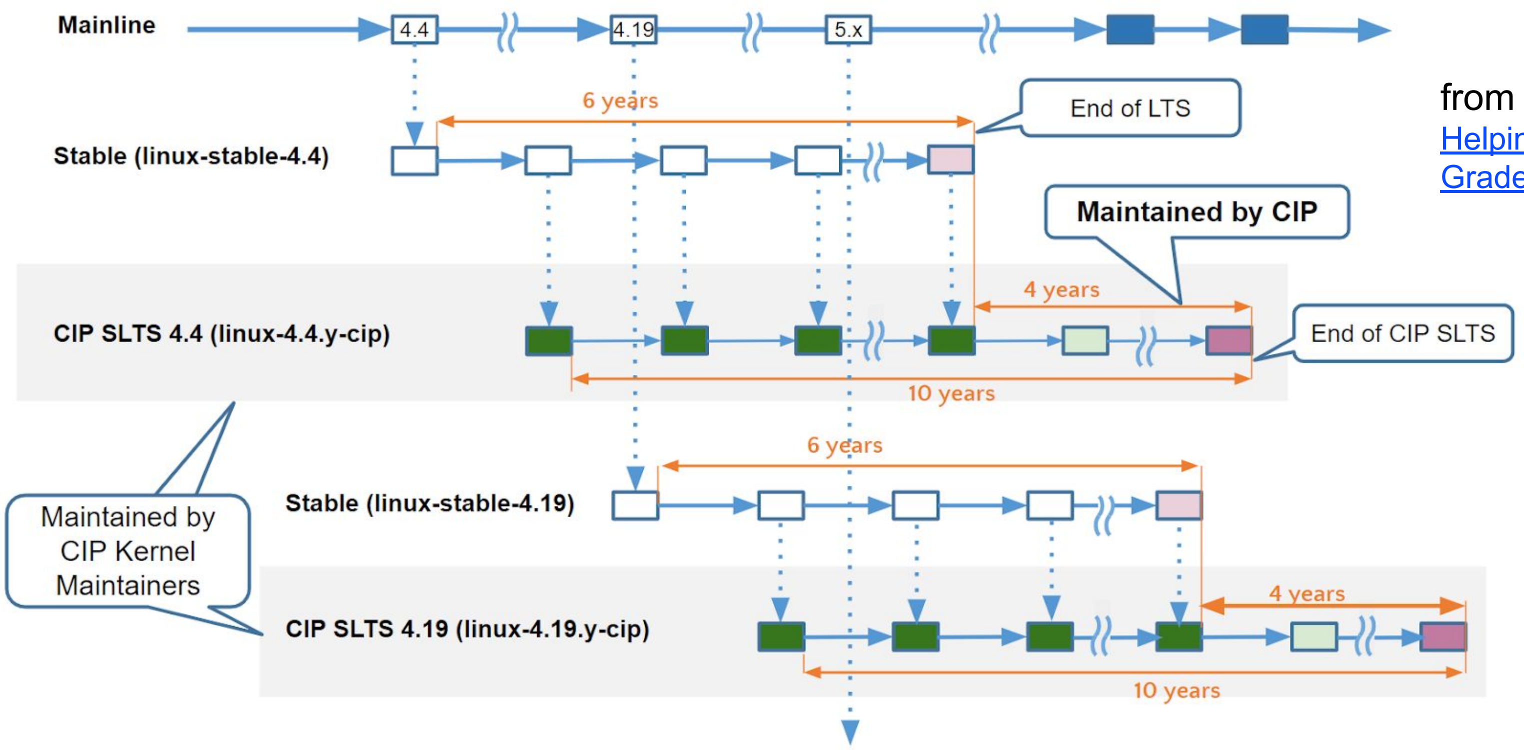
EMLinux: Why we choose LTS model?

- LTS model is easy for most users to start the security update process
  - including test automation
- Less changes in features or interfaces (than rolling update model)
  - Less impacts on product-specific code
  - (Automated) test cases are re-usable for a long time
  - Verification of the changes is easier

Leveraging CIP kernel, meta-debian, Debian source pkgs

■ CIP Super Long-term Stable (SLTS) kernel

- maintained by Civil Infrastructure Platform Project for 10+ years
- Upstream first policy. All LTS commits are merged.
- Twice a month release for 4.19.y-cip


Mainline
4.4
4.19
5.x
6 years
End of LTS
Stable (linux-stable-4.4)
Maintained by CIP
4 years
CIP SLTS 4.4 (linux-4.4.y-cip)
End of CIP SLTS
10 years
6 years
Maintained by CIP Kernel Maintainers
Stable (linux-stable-4.19)
4 years
CIP SLTS 4.19 (linux-4.19.y-cip)
10 years

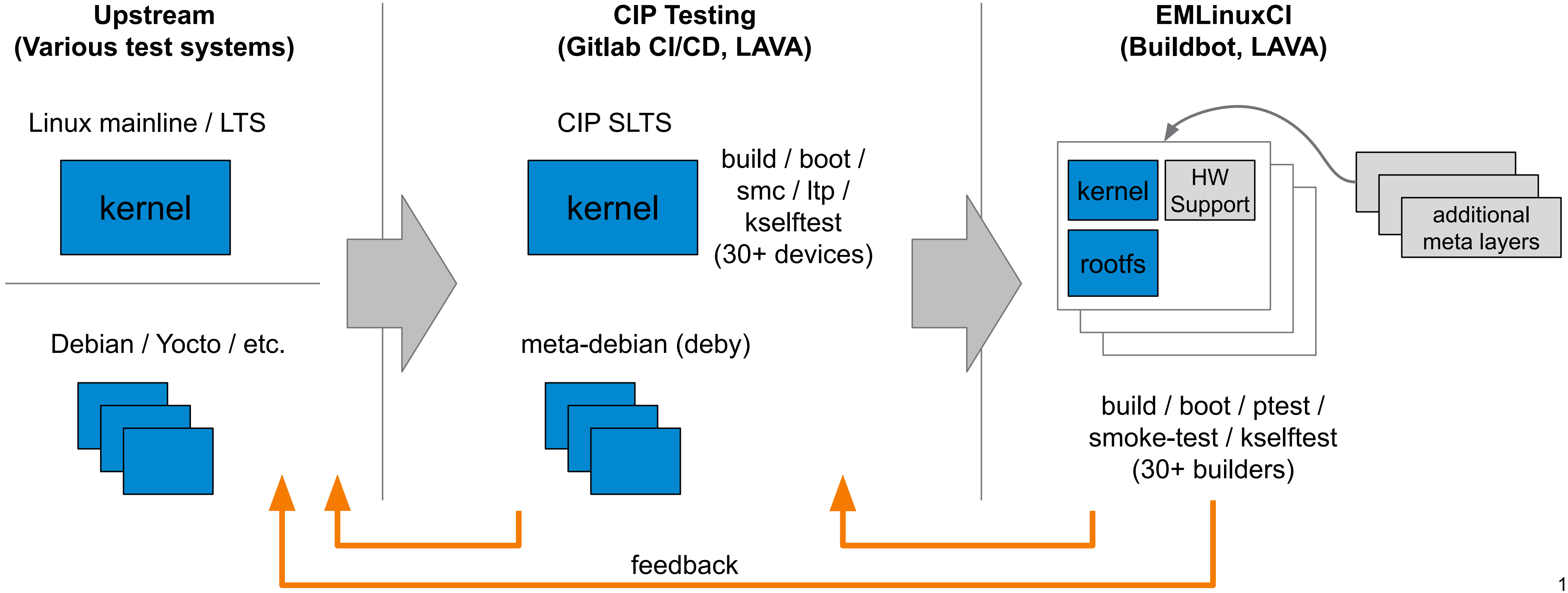
from CIP Kernel Team: Helping CIP Sustain Industrial Grade Systems

Leveraging CIP kernel, meta-debian, Debian source pkgs

- meta-debian
  - “Yocto Project extension for using Debian source packages”
  - Created by Toshiba people
  - Debian source packages:
    - Stable version typically accepts only bug-fixes
    - 3 + 2(LTS) years maintenance period
  - Packages are updated with each Debian’s point release
  - Some packages will be maintained as CIP Core Packages for 10 years
- meta-debian-extended
  - Additional packages for meta-debian (Same recipe format)
  - Created by Cybertrust

Test Automation

- SLTS kernel and meta-debian(deby) are tested in CIP Testing
- EMLinux is tested with in-house EMLinuxCI (+ periodic manual tests)
- Found bugs / issues are fed back to upstream



OSS contribution from Cybertrust

Kernel

- Maintenance of CIP SLTS kernel
  - Current Kernel Team Chair is from Cybertrust
- CIP Testing WG
  - Operating LAVA lab (lab-cip-cybertrust) in CIP LAVA
    - also used from KernelCI: contributing upstream work
  - Contributing new features and bug-fixes to some projects
    - CIP Testing: Kselftest integration is in-progress
    - KernelCI: kernelci-core, kernelci-docker, lava-docker, etc.
    - LAVA: xilinx-zcu102 (re-)support

OSS contribution from Cybertrust

meta-debian

- many contributions
  - Cybertrust people made 142 of 525 commits in warrior branch
  - package addition
  - cve-check feature with Debian Security Bug Tracker
  - recipe updates on each Debian point release
  - bug-fixes

OSS contribution from Cybertrust

meta-debian

- cve-check feature with Debian Security Bug Tracker
  - Yocto cve-check refers NVD DB, which uses version numbers to determine whether the vulnerability is included or not

```
PACKAGE NAME: openssl-native
PACKAGE VERSION: 1.1.1d
CVE: CVE-2021-23841
CVE STATUS: Patched
CVE SUMMARY: The OpenSSL public API function X509_issuer_and_serial_hash() attempts to ...
```

  - For Debian source packages, we need additional considerations
    - Security bugs are fixed (backported) in the same version like:
    - 1.1.1d-0+deb10u1 + (fix in 1.1.1e+) -> 1.1.1d-0+deb10u2
    - False positives happen only with NVD DB
  - We complement that information by using Debian Security Bug Tracker -> merged.

OSS contribution from Cybertrust

Others

- Yocto (poky)
  - Some features and bug-fixes
- OpenEmbedded
  - License corrections
- util-linux
  - bug-fixes in a test case
- Buildbot
  - support git-repo '--submodules' option

Future Work

- Expansion of test cases
  - continue to integrate kselftests into CIP Testing
  - ptest enablement in meta-debian{-extended}
- Expansion of KernelCI collaboration
  - support xilinx-zcu102
- Direct contributions to Linux Kernel (mainline, LTS), Debian