Backporting is so 1993

Ricardo Salveti - ricardo@foundries.io

Michael Scott - mike@foundries.io

Embedded Linux Conference & OpenIoT Summit - Edinburgh

Introduction

Contents

1. Connected Products Requirements
2. Linux LTS Releases
3. Zephyr
4. Why latest software
5. Working with latest (microPlatforms)

Connected Embedded Products

FOUNDRIES.IO

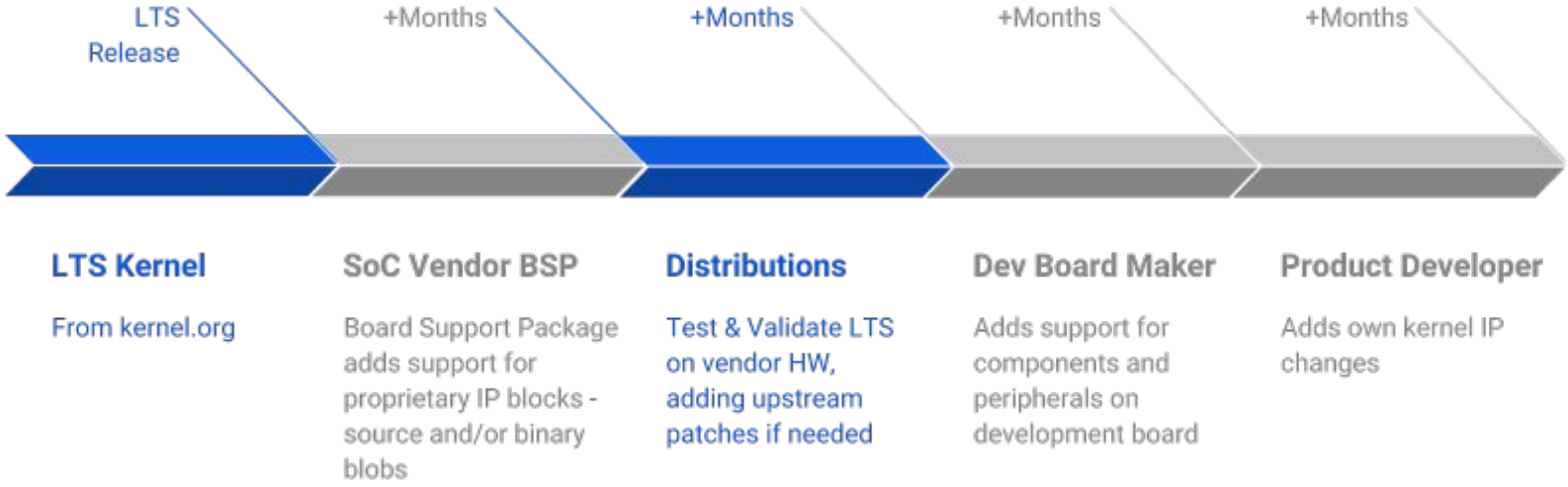
- Long Product Lifetime (+10 Years)
- Always on / connected
- Large attack surface
- Continuous Maintenance
- Secure over-the-air software updates

Linux Kernel Long Term Support

- New release approximately once a year
- Maintenance for 6 or more years
- Maintained by the Linux community
- Usually same base release as used by distributions

Embedded Devices Today

FOUNDRIES.IO



Complex and long supply chain, hard to maintain and apply updates

Stable Maintenance Is Not Easy

- How to identify bug fixes that should be backported?
  - Which fixes are security related?
- Complex maintenance chain across SoC/Board vendors
- New features might be desired
  - E.g. Kernel Self Protection
- Backport can also introduce new bugs and regressions

Stable Maintenance Is Not Easy

USN-3741-1 introduced mitigations in the Linux kernel for Ubuntu 14.04 LTS to address L1 Terminal Fault (L1TF) vulnerabilities (CVE-2018-3620, CVE-2018-3646). Unfortunately, the update introduced regressions that caused kernel panics when booting in some environments as well as preventing Java applications from starting. This update fixes the problems.

Stable Maintenance Is Not Easy

USN-3522-1 fixed a vulnerability in the Linux kernel to address Meltdown (CVE-2017-5754). Unfortunately, that update introduced a regression where a few systems failed to boot successfully. This update fixes the problem.

Stable Maintenance Is Not Easy

CVE-2017-18344 - https://seclists.org/oss-sec/2018/q3/76

Bug introduced in 3.10 and fixed in 4.15-rc4.

- Nov 30, 2017 - bug reported by syzbot (4.15)
- Dec 15, 2017 - fix committed upstream (4.15-rc4)
- Feb 17, 2018 - fix backported to the 4.4 LTS
- Mar 15, 2018 - fix added to the Ubuntu Xenial 4.4
- Jul 25, 2018 - CVE requested
- Aug 2, 2018 - notified linux-distros and oss-security

And now for something not Linux: The Zephyr™ Project

The Zephyr™ Project, is a Linux Foundation hosted Collaboration Project, an open source collaborative effort uniting leaders from across the industry to build a best-in-breed small, scalable, real-time operating system (RTOS) optimized for resource constrained devices, across multiple architectures.

Key components: Open Source, Secure, Modular and Connected

And now for something not Linux: The Zephyr™ Project

Let’s say you’re making a connected wearable and want to use Zephyr for a product. We started development last year planning for a release this October.

The Plan: Fork and forget! Right!? What could go wrong?

And now for something not Linux: The Zephyr™ Project

V1.10 released December 8th 2017: ~1800 commits

- Initial alpha-quality thread-level memory protection on x86, user-space and memory domains
- **Major overhaul to the build system and a switch from Kbuild to CMake.**
- HTTP API changed to use net-app API. Old HTTP API is deprecated.
- Deprecated ZoAP library in-favor of new CoAP library
- Various fixes for: TCP, RPL, ARP, DNS, LWM2M, Ethernet, net-app API, Network shell, and BSD socket API

And now for something not Linux: The Zephyr™ Project

V1.11 released March 9th 2018: ~1500 commits

- Thread-level memory protection on x86, ARC and Arm, userspace and memory domains
- Native development environment on Microsoft Windows.
- Thread support via integration with OpenThread.
- Lightweight flash storage layer for constrained devices.
- **LWM2M fixes and enhancements, CoAP fixes, TCP fixes, HTTP fixes, RPL fixes, Net-app API fixes, Net-shell fixes, BSD socket API fixes**

And now for something not Linux: The Zephyr™ Project

V1.12 released June 11th: ~1950 commits

- Support multiple concurrent filesystem devices, partitions, and FS types
- kernel/sched: Fix preemption logic
- **kernel: Scheduler rewrite**
- Add initial WiFi management API definitions.
- Network timeout fixes, ICMPv6 error check fixes, BSD socket sample application fixes.
- Multiple BLE Mesh bugfixes and improvements

And now for something not Linux: The Zephyr™ Project

V1.13 released Sept. 10th: ~1800 commits

- Support for TLS and DTLS using BSD socket API
- ADC: Introduced reworked API and updated Nordic, NXP, Atmel, and Synopsys DesignWare drivers
- **Bug fixes: Handle large IPv6 packets properly, IPv6 address lifetime fixes, IPv6 fragmentation fixes, DHCPv4 fixes, TCP retry, RST packet handling, and memory leak fixes, HTTP fix when sending the last chunk, MQTT fixes, LWM2M cleanups and fixes.**

And now for something not Linux: The Zephyr™ Project

Upcoming movement for v1.14:

- System Logger changed to Logger (including macro name changes)
- **Network APIs moving from linked list network buffer (net_app) APIs to POSIX socket APIs**
- Major rework of Timer APIs, and more!

Can you imagine shipping with Zephyr v1.9 from September 7th last year?

Our Belief

“There is no such thing as secure software since security is an arms race. Therefore, the latest software is the most secure.”

Latest Software

- Easier to report and fix issues
- Faster review & testing iteration
- Easier integration of new features
- Smaller supply chain

Working with Latest: Zephyr

How do you manage large amounts of churn in an OSS project like Zephyr?

Take it in bite-sized chunks!

- Bring in “small” batches of 100 to 200 commits at a time.
- Easier to bisect regressions.
- Easier to examine commits and understand major differences like API changes.
- Easier to integrate back into your code base.
- **Test, test and test some more.**

Working with Latest: Zephyr

Who tests the testers?

- Design a test plan with a goal in mind. **Don't just test things at random.**
- If you have specific use-cases, look for samples that do exactly that in the upstream code. Use those samples as tests. Example: your product needs HTTPS. Setup an automated test of the HTTP-client sample using HTTPS for every upstream commit. **If something breaks, you'll know about it as soon as it enters mainline!**
- Run "sanity" tests on QEMU targets as often as you can. No HW needed and these verify the general integrity of your software.
- If possible use a CI scheduler to coordinate real world tests on your hardware performing your use-cases.

Working with Latest: Zephyr

Understand the development cycle!

Read the Zephyr wiki program management page: https://github.com/zephyrproject-rtos/zephyr/wiki/Program-Management

There will be **A LOT** of “hazardous” commits going in during the first few weeks of new development. It then slows down and gets better as the RCs are released.

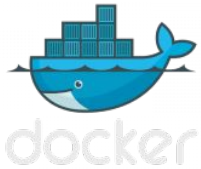
Working with Latest: Linux

- Separation of core platform from application code
  - Allow pieces to move independently
- Continuous build, testing and validation
  - Buildroot, Yocto / OE
  - KernelCI.org, LAVA
- Joint effort across IP, SoC and Hardware vendors for continuous upstream support

microPlatforms

What are microPlatforms?

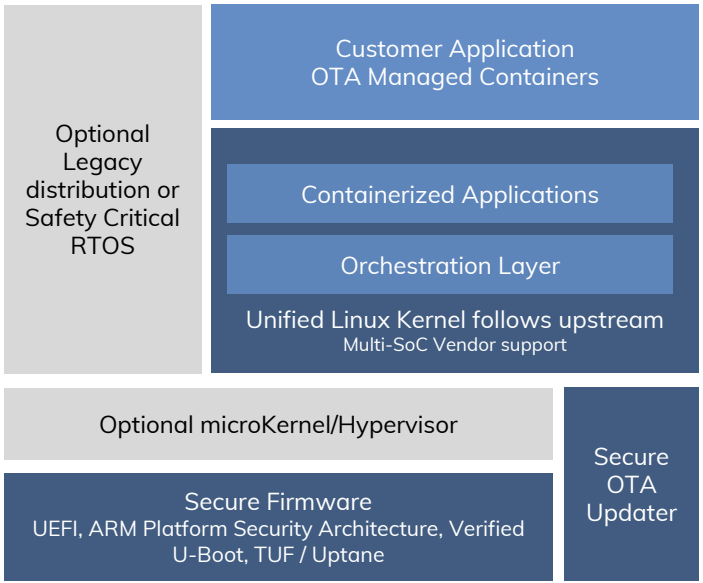
FOUNDRIES.IO



- Minimal
  - RO (Smaller attack surfaces)
  - Persistent RW space for user applications
  - Virtualization and Container Runtime
- Securable
  - Verified Boot
  - Trusted Execution Environments
  - Threat modeled
- Updatable
  - Fault tolerant update mechanisms built-in
  - Delta based incremental updates
  - TUF / Uptane compliant
- No proprietary lock-in
  - Open source from sensor to cloud

LmP Software Architecture

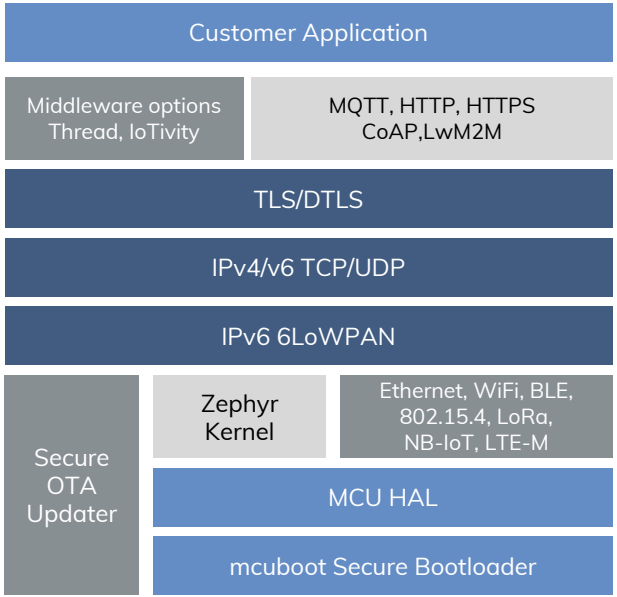
FOUNDRIES.IO



- Stabilized Upstream Software
  - U-Boot/UEFI
  - Linux
  - OE/Yocto
- Containers are First Class Citizens
  - Docker Runtime
  - Multi-Architecture Support
- OTA Updatable
  - Continuous Software Updates Provided via Subscription
  - OSTree + TUF/Uptane
- Tested
  - Automated sensor to cloud end to end testing done on every merge

Zephyr microPlatform

FOUNDRIES.IO



- Stabilized Upstream Software
  - Zephyr
  - MCUboot
- Implements Standards
  - LwM2M
  - HTTP/HTTPS
  - MQTT/MQTTS
- Connected
  - WiFi / BLE / 6LoWPAN / 802.15.4 / MESH / NB-IoT
- OTA Updatable
  - Continuous Updates provided via Subscription
- Tested
  - Automated end to end testing on every merge

Thank You!

FOUNDRIES.IO