Contribution ID: **370** Type: **not specified**

Firmware ABI stability

Contemporary embedded systems increasingly come with bootloaders and firmware which expose some sort of ABI toward the Linux kernel, and the Linux kernel depends on such ABI to start other CPU cores, configure clock, power domains, pin multiplexing and other vital parts of the system.

With existing firmware interfaces like ACPI, the ABI stability is strictly enforced and ABI breakage seldom occurs. With newcomer firmware interfaces on contemporary embedded systems, commitment to ABI stability is left to vendors and ABI breakage is much more common. This leads to problems during kernel updates of such systems, which may mandate either a bootloader update, or special-case kernel fixes.

The gravity of this problem is further exacerbated by a growing number of closed source implementations of such firmware ABI provider implementations, either in the form of firmware running on secure coprocessor, or similar.

In this proposal, the speaker explains the firmware ABI stability problem in detail, including examples present on contemporary SoCs, and a grim prediction of where the firmware ABI development is likely to evolve. The talk will then offer multiple possible solutions how to address the problem. Once such option would be to use open source bootloader, like U-Boot, to start the additional firmware components from a fitImage bundle together with Linux kernel, and possibly use U-Boot SPL to minimize the security sensitive code footprint. Another alternative would be improved introspection of firmware ABI interfaces and proper versioning, which would allow Linux kernel to work around bugs in specific firmware versions.

Proper discussion and feedback from audience on this topic would be highly appreciated.

Primary author: VASUT, Marek

Presenter: VASUT, Marek

Session Classification: Embedded & Internet of Things MC

Track Classification: Embedded & Internet of Things MC