Universal Scalable Firmware (USF)

Simplifying and Scaling Firmware Development from Edge to Cloud

USF overview

Universal Scalable Firmware (USF) is Intel's next-generation firmware initiative to simplify and scale firmware development from edge to cloud and across all layers of the firmware stack from silicon and system-on-chip (SoC) to platform, bootloaders and OS payloads. USF's modular and flexible architecture helps enable developers and technology partners to accelerate the integration of new silicon and platform technologies, and more efficiently deploy breakthrough products and solutions at scale.

USF architecture is evolutionary and aspires to be comprehensive and scalable across compute architectures -- CPU and XPUs, such as the discrete graphics processing unit (dPGU) and the infrastructure processing unit (IPU) -- and different bootloaders and OS payloads, such as UEFI, LinuxBoot and ACRN. USF builds on existing firmware standards, such as UEFI and ACPI, and plans are to enable key open-source community bootloader projects, such as tianocore, coreboot and slim bootloader, as well as new programming and configuration languages, such as Rust and YAML.

USF simplifies firmware development by defining abstraction layers and domain boundaries between SoC (IP HW and IP FW), platform and operating system with verifiable interfaces and APIs between the firmware modules (see Figure 1):

- **Universal Payload API** (Platform Orchestration Layer ⇔ OS payload I/F)
- **FSP API** (SOC abstraction layer I/F ⇔ Platform Orchestration Layer)
- **IP API** (IP FW ⇔SOC abstraction layer I/F)
- **HW API** (IP HW ⇔ IP FW I/F)

USF also improves firmware security by enabling consistent support for firmware attestation, authentication, measurement and secure updates.

Intel's USF specifications consist of two parts – an Intel internal specification that covers the SOC construction and its internal interfaces (IP HW and IP FW); and an external industry specification that covers the interfaces to SOC, platform, and OS payloads, as well as building and managing full firmware products & solutions (i.e., how to initialize, configure, integrate, boot, update, and maintain). The external specification is open for active feedback and collaboration by the industry and technology partners. The external spec version at initial launch is purposely starting at a draft revision for technology partners to have the opportunity to help enhance its content & direction before v1.0 finalization.

Motivation for USF: Addressing Complexity and Enabling Flexibility & Scalability

Firmware is a critical part of the software foundation to unleash silicon innovation and expose platform technologies to high-level software stacks. Over the years, firmware has gotten more complex as the number of architectures and devices keep growing. SOC has increased in complexity with more IPs being added on a single SOC; and more firmware images added on every platform. SOC and platform code has become intertwined creating complex interactions. As size and complexity of SoCs and platform capabilities have grown, so too has firmware. Generally, firmware size increases nearly an order of magnitude every decade. In 40 years of PC/Servers, it is estimated to have grown from ~ 64KB (in 1980) to ~16-32MB in 2020.

To effectively support the industry's accelerated digital transformation, infrastructure and SOCs/Platforms designs should be able to scale and support diverse market segments requirements from edge to cloud including providing consistent interfaces across CPUs and XPUs. In the boot firmware space, there are several industry open-source bootloaders – such as tianocore, coreboot, and slim bootloader; and OS payloads – such as UEFI and LinuxBoot that have become popular with various developers' communities

across market segments. It is imperative that any modern firmware architecture & infrastructure be scalable and flexible to effectively address these different market segments' needs.

Addressing firmware developer community needs has always been a priority for Intel. As a decades-long pioneer in the firmware architecture space, Intel has history in driving firmware standards and related technologies from ACPI (advanced configuration and power interface) to UEFI (unified extensible firmware interface). With USF, we hope to help address the industry firmware development pain points, reduce complexity, accelerate innovation, improve firmware quality & security, and enable firmware flexibility & scalability across CPU and XPUs in the future.

Key USF features

Some of the USF features and plans for future advancements include:

- **Universal Payload I/F –** universal API for different OS payloads (i.e. UEFI, Linuxboot, ACRN embedded hypervisor); support for various bootloaders (i.e. tianocore/EDKII, coreboot, slim bootloader).
- **Platform Orchestration Layer (POL)** – simplified ACPI support; common libraries for various bootloaders and Rust safe language; standard binary configuration mechanism through YAML; support for firmware attestation, authentication, measurement, and modern secure update.
- **SOC/Firmware Support Package (FSP)** – FSP support for 64-bit reset vector, SMM encapsulation, authentication, unified configuration, and SOC level validation.

Key USF benefits

Future potential benefits from the USF architecture include:

- **Reduce industry firmware development cost:**
    - Modular and bootloader-agnostic architecture with consistent interfaces enable code reuse gen to gen.
    - API-based configurable modules help reduce complexity (and minimizes need for source code manipulation).
- **Improve firmware quality and security:**
    - Better quality through verifiable IP FW modules.
    - Better security infrastructure through support of firmware attestation, authentication, modern secure update, and measurement.
- **Expand access to new markets from edge to cloud:**
    - Universal payload support helps enable FW modules reuse across bootloaders (i.e., EDKII, coreboot, slim bootloader) and payloads (i.e., UEFI, LinuxBoot, ACRN) – Flexibility across market segments from edge to cloud.
- **Accelerate innovation:**
    - Gen to gen code reuse and ease of integration help developers focus on innovation and deploying value-add features quicker.

USF technology and Intel products

USF covers multiple layers of the firmware stack (SOC, Platform, OS boot). Intel products' implementation and integration progress will vary depending on the layer as well as silicon & platform generation. USF will be deployed over several product generations to ensure the latest technology advancements and converted IPs are quickly available to technology partners as they are ready at each layer. Please work with your Intel customer product teams to learn the latest plan of record status as it varies from platform to platform.

Call for collaboration

Delivering on the USF vision cannot be achieved by Intel alone. Success comes hand in hand through collaboration with developers, technology partners, and firmware ecosystem communities. Please join our github USF spec community and collaborate in shaping the specification to address your firmware development pain points. Let's accelerate innovation and time to value together!

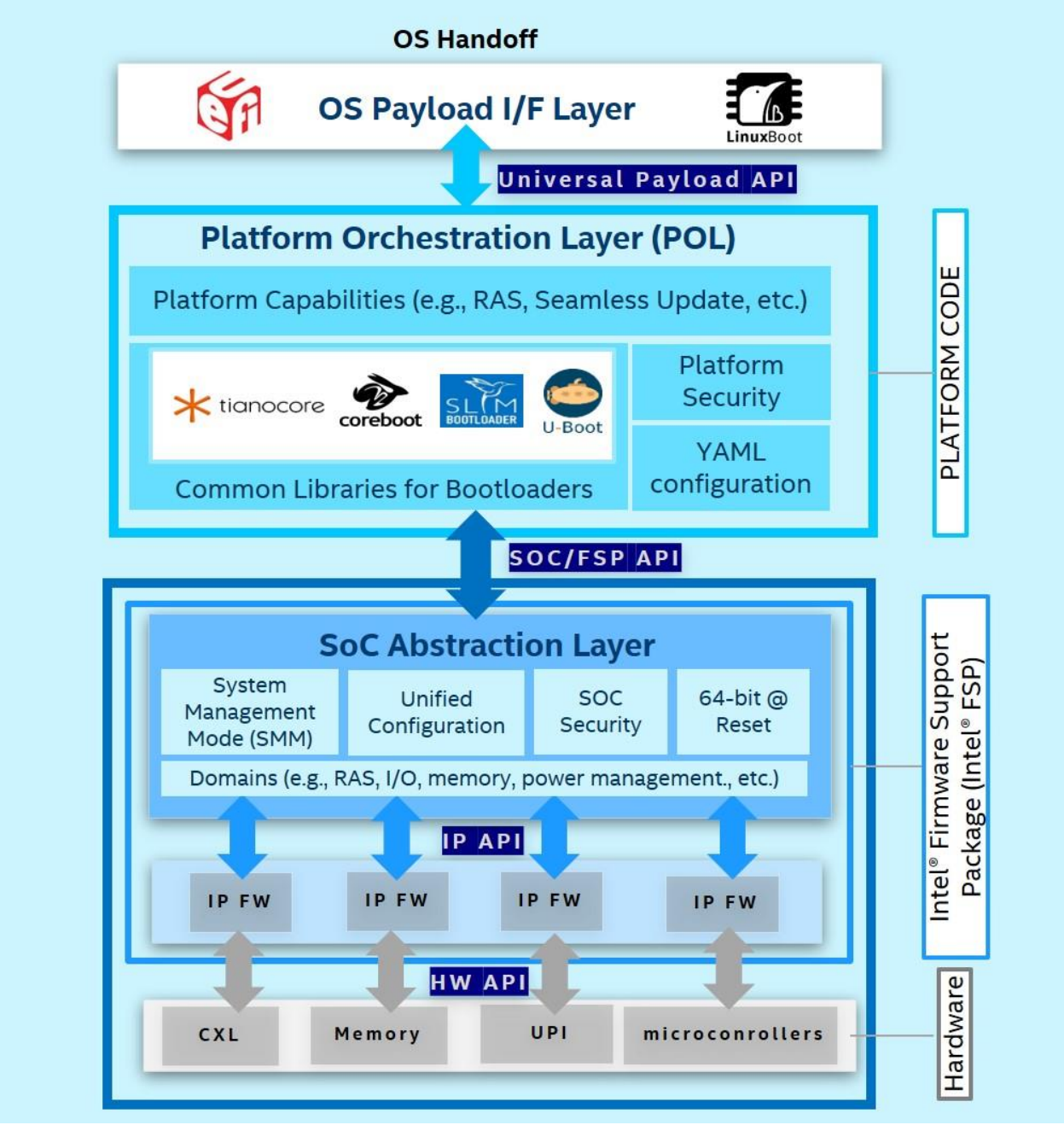

Figure 1 – Universal Scalable Firmware Architecture