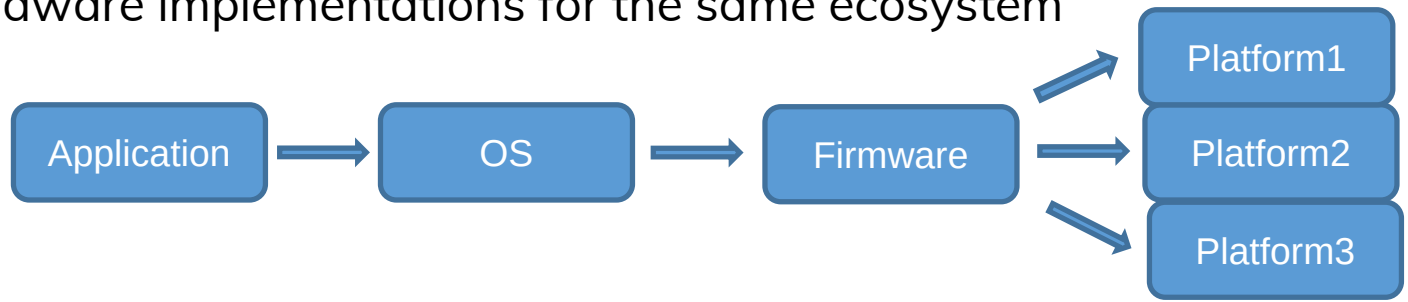
Community Driven UEFI Open Source Project

SAN19-508

Linaro connect
San Diego 2019

Firmware Open Source Motivation

We have experienced very successful open source development for Linux operating system. But in the firmware area, most of the developments are carried out by some major organizations with limited participation of community. Application driven open source development enables a software system ecosystem which can adopt various hardware components, including different architectures, and let the vendors deferential and create tangible results. Open source firmware and Standard firmware interface is critical to enable different hardware implementations for the same ecosystem



Open Source Firmware Development Consideration

- General System Firmware not specifically targeted at phone, client, server or cloud
- Universal interface supporting multiple OSes including Linux and Windows
- Adaptive to various silicon especially silicon provided by member companies
- **Community driven open source development model**
- **Encourage open technology and early standard implementation**
- **Code first path finding model**
- Primarily use permissive free software licenses
- **Long Term Stable instead of product driven**

We may not consider

- Find an open source solution for an existing platform design
- Reverse engineering existing code
- Do not force to use GPL free software license

System Firmware Components in a typical solution

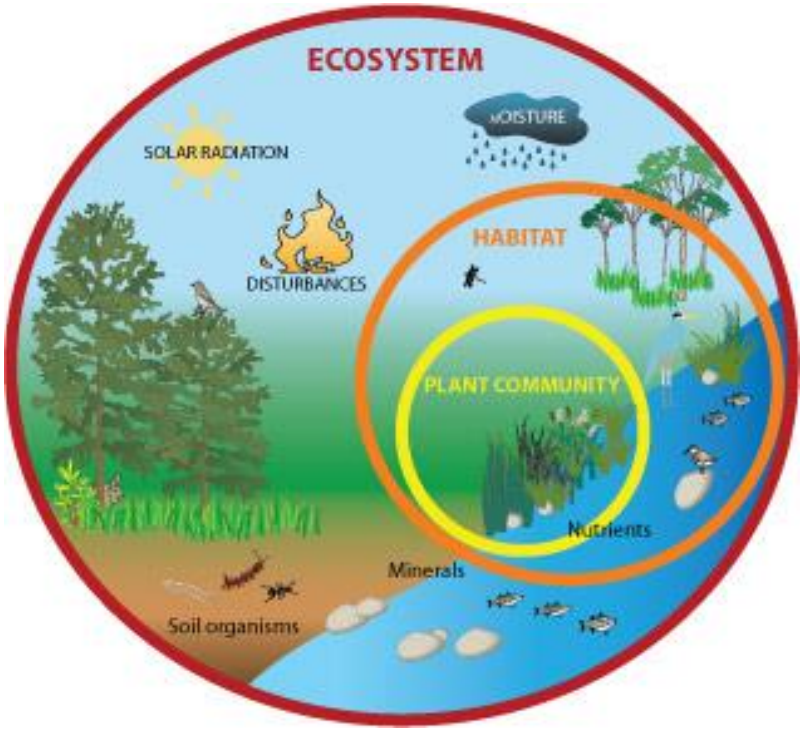
tianocore

arm

Core	Silicon	Platform
• Generic Framework code • Standard services lib • Industry Standard such as PCIe support • Easy to open source	• CPU architecture specific code • Silicon Vendor driver code • Third party support module • Very much vendor specific	• Platform configuration • Data tables • Advanced feature implementation such as RAS • Should be as open as possible such that people can utilize them

Keep Healthy Ecosystem

- Silicon vendor
  - SoC
- IP provider
  - PCIe
- IBV
- ODM
- OEM
- Cloud Provider
- Software Service provider
- OS vendors
- Open source application
- Commercial software



Protect silicon vendor IP

- Open source firmware has been driven by cloud providers such as Google and Facebook
- Due to the complexity of hardware such as memory controller, the successful Linux OS approach is not able to meet the requirement at firmware level.
- We would like to separate the kernel code for EDK implementation and the silicon code for EDK such that the silicon code can be more flexible to accepting 3rd party open source code with different license model.
- Linaro can setup the architecture and only take care of the kernel, but treats the silicon code or binary module as plugin with known interfaces
- May include some mature silicon modules such as SBSA compliant modules for CI test.

- Modular structure and BSD license is preferred

LTS UEFI Firmware for ERP

- Enterprise companies has a lot of hardware products and the life cycle of these systems could be more than 10 years, such as telecom equipment's. Linux OS has been used widely in this market. Linux kernel has LTS management such it can well matches the long term support requests.
- UEFI forum defines the interfaces for hardware and software using UEFI and ACPI standards. The implementation of the firmware is flexible but dominated with PI specification. These type of firmware source code are hosted by the tianocore project in github and is referenced as EDK implementation.
- If we compare EDK code with Linux code, we can see many differences. The most obvious one is that the long term stable label is not there. Current EDK has released “stable label” quarterly , but there is no LTS concept.
- Due to the multiple vendor nature, it will greatly benefit the industry if we have LTS label for ARM based platform.

Firmware Interfaces

Many vendor specific interfaces are widely used and they are keeping evolve.

Example:

IFWI: Intel firmware interface

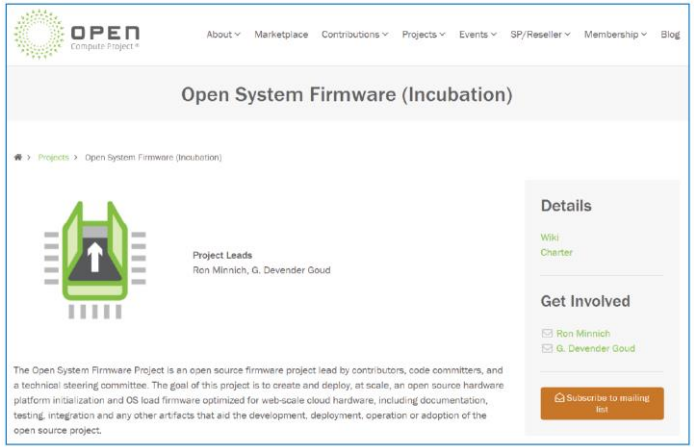
FSP API: Intel Firmware support package

ARM: PSCI, SCMI, SPCI

It will be nice to use Industry standard interfaces or detectable open source interfaces

UEFI, ACP and PI are good examples

Industry Open Source Firmware Efforts



Scope

Supports all processor architectures found in the web-scale data center.
Support for cloud operating systems
Support for compute (GP & AI/ML/FPGA), storage, & network devices.
Development and deployment tools
Security feature

https://www.opencompute.org/projects/open-system-firmware

MISSION

Change the way of firmware development, collaboration with others and share knowledge

https://osfc.io

U-Boot

Architecture/SoC

ARM32: Aspeed, Altera, Allwinner, Atmel, Broadcom, Qemu, Qualcomm, Marvell, NXP, Rockchip, STM32, Tegra, TI, UniPhier, Xilinx

ARM64: Allwinner, Marvell, NXP, Rockchip, Tegra, UniPhier, Xilinx

X86 (Baytrail, Broadwell, Quark, etc)

ARC, M68K, MicroBlaze, MIPS, NDS32, NIOS2, PowerPC, RISCV, Sandbox, SuperH, Xtensa

Boards

185+ different board vendors
~1400 different boards

Language

C
C/C++ Header
Assembly
Python
make
Perl
Bourne Shell
C++
yacc
YAML
Glade
lex
NAnt script
Markdown
Bourne Again Shell
DOS Batch
CSS
Kermit
Tcl/Tk
sed
INI

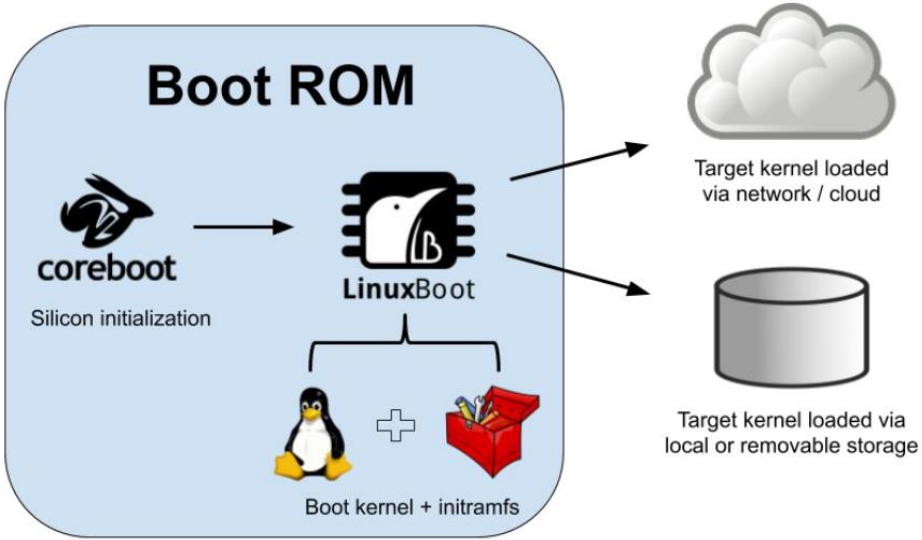
Reference: OSFC, 2019 State of U-Boot Development Report
by Jagan Teki, Amarula Solutions

Coreboot

Coreboot is an extended firmware platform that delivers a lightning fast and secure boot experience on modern computers and embedded systems. As an Open Source project it provides auditability and maximum control over technology.

https://www.coreboot.org

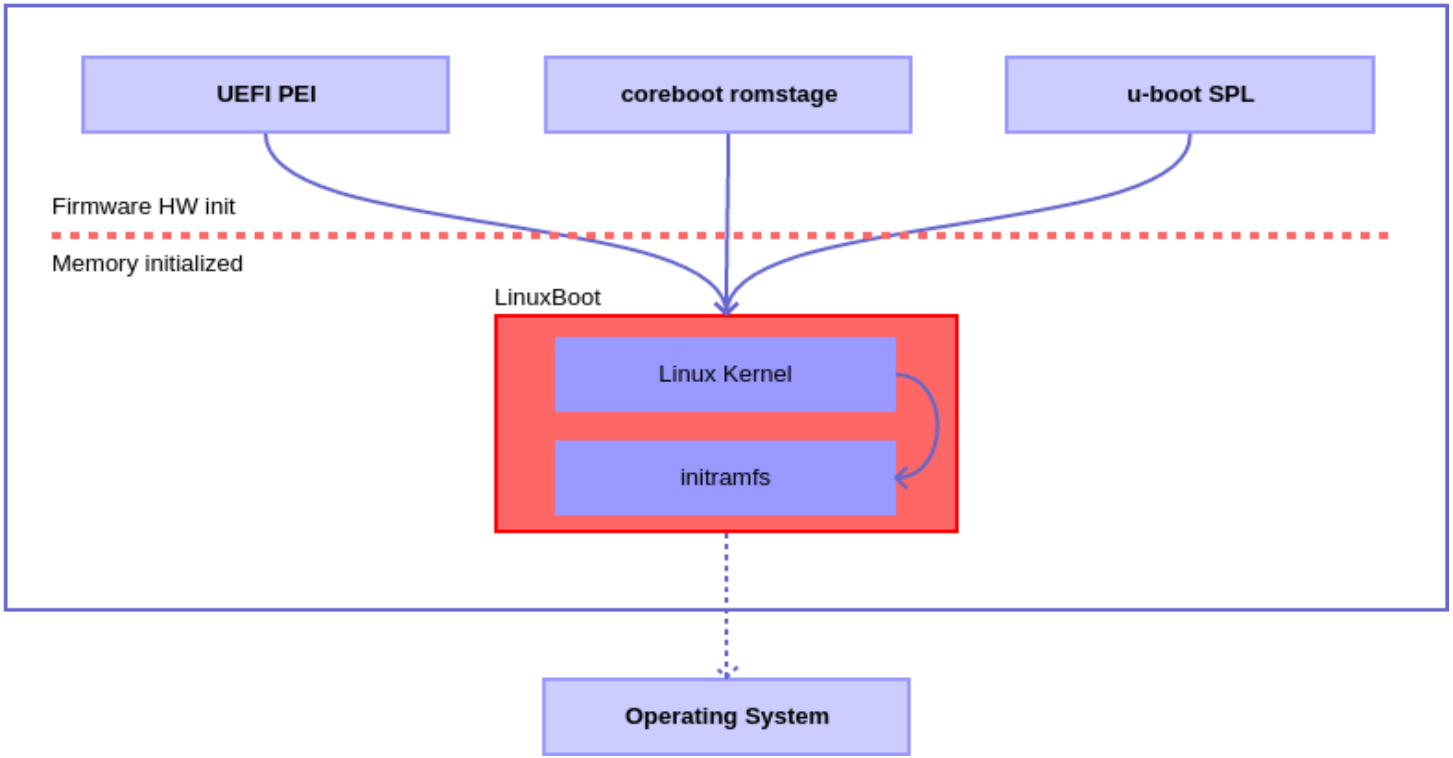
Facebook OSF Solution Using Coreboot



- **Debuggability**
- **Manageability**
- **Workload requirement**
- **Security**
- **Innovation**
- **Green**

Reference: OSFC 2019 Build coreboot/linuxboot firmware for Facebook OCP platform
By Jonathan Zhang and Morgan Jang

LinuxBoot



https://www.linuxboot.org/

Innovations with LinuxBoot

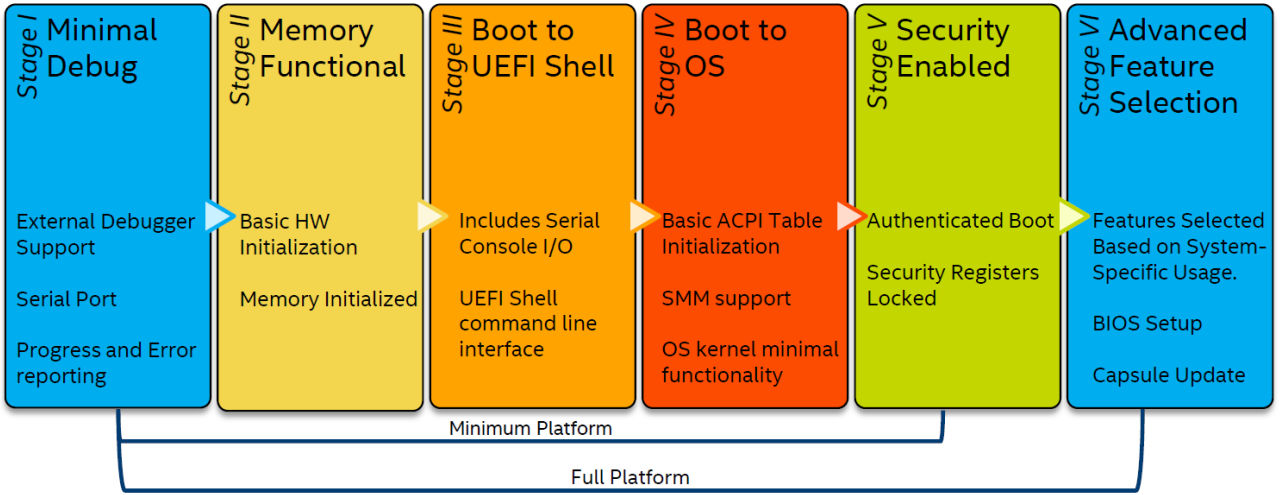
LinuxBoos enable people with fundamental computer OS knowledge to explore various possibilities

Webboot

Intern Presentation
Interns: Louis Murerwa, Urvisha Patel
Hosts: Ron Minnich and Gan Shun Lim

Minimum platform Approach

Staged implementation to address one size fit all problem

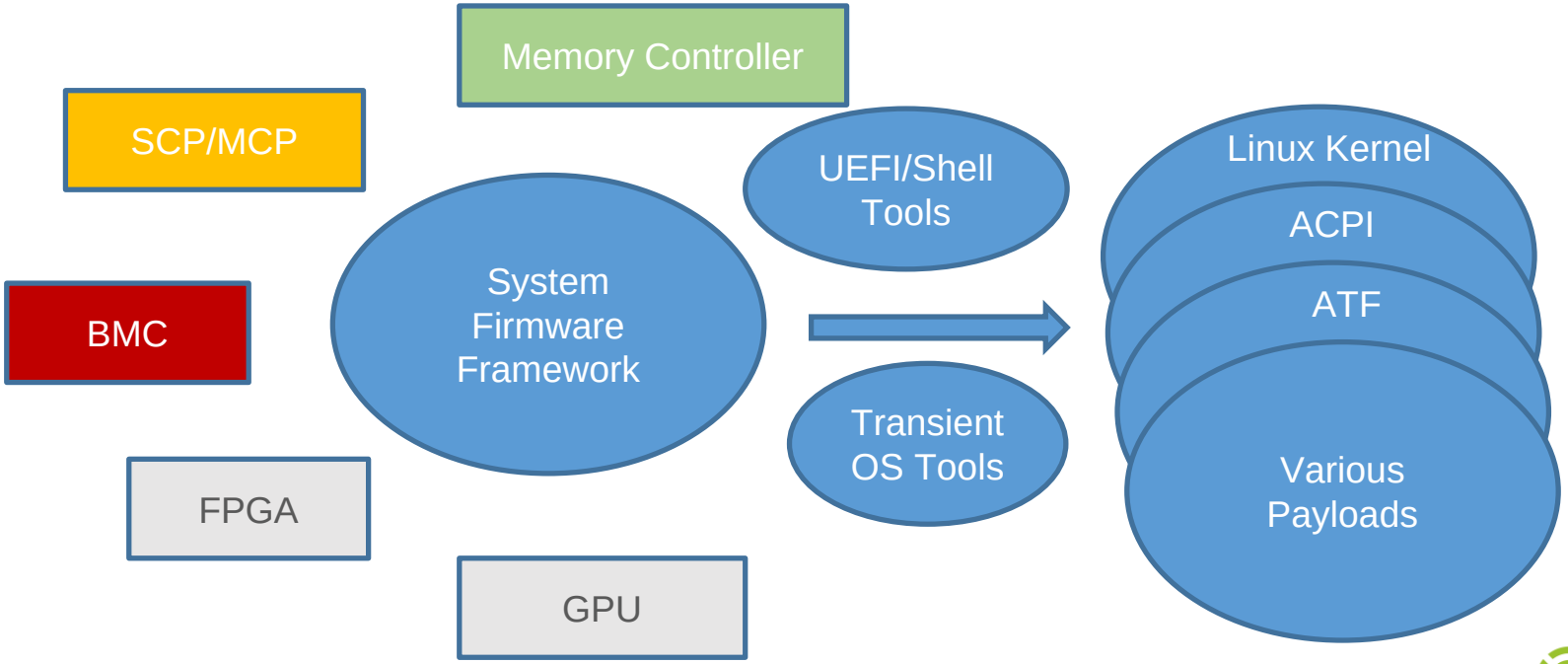


Reference: OSFC 2019 Minimum Platform: Open Source UEFI Firmware for Intel Based Platforms
By Michael Kubacki

https://edk2-docs.gitbooks.io/edk-ii-minimum-platform-specification/

Heterogeneous Approach

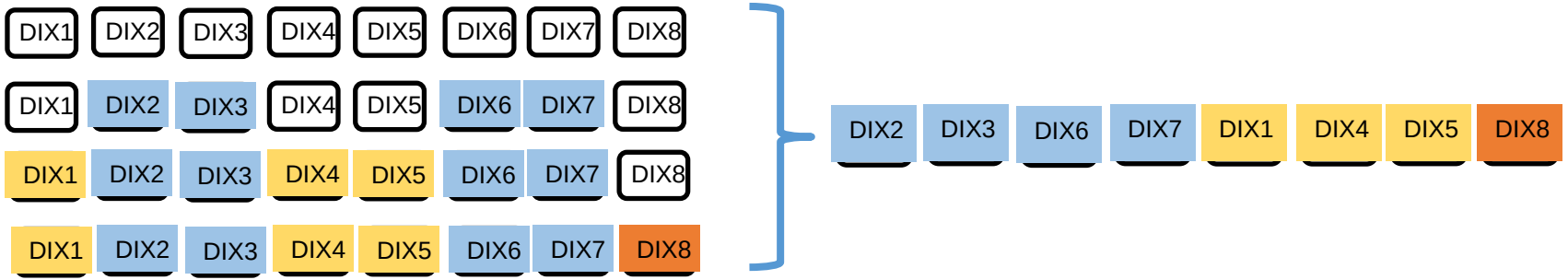
Allow multiple vendors to contribute their own solution in one framework



Fast Boot Consideration

software should not be a barrier for speed

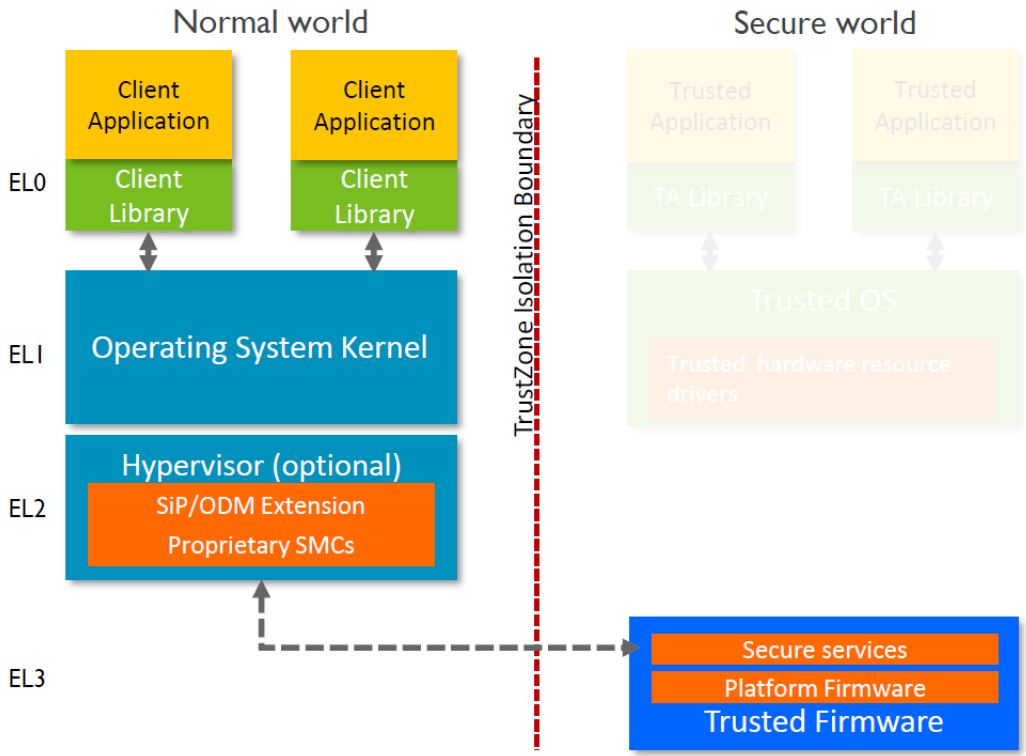
- Resume from previous configured data
  - S4 resume instead of regular boot
- Only initialize necessary components
  - Jump through non-IPL device
- Fixed initialization sequence



- Parallel program
  - Multi-processor/thread initialization

Coexisting with non-UEFI modules

- Arm Trusted Firmware is a key component in EDK2 firmware and other firmware solutions
- Due to its general usages, it may not be used as UEFI compatible extensions similar to Intel FSP
- RUN time services should be used conservatively after the OS has started
- Consider to implement UEFI PI compatible interface similar to FSP2.1



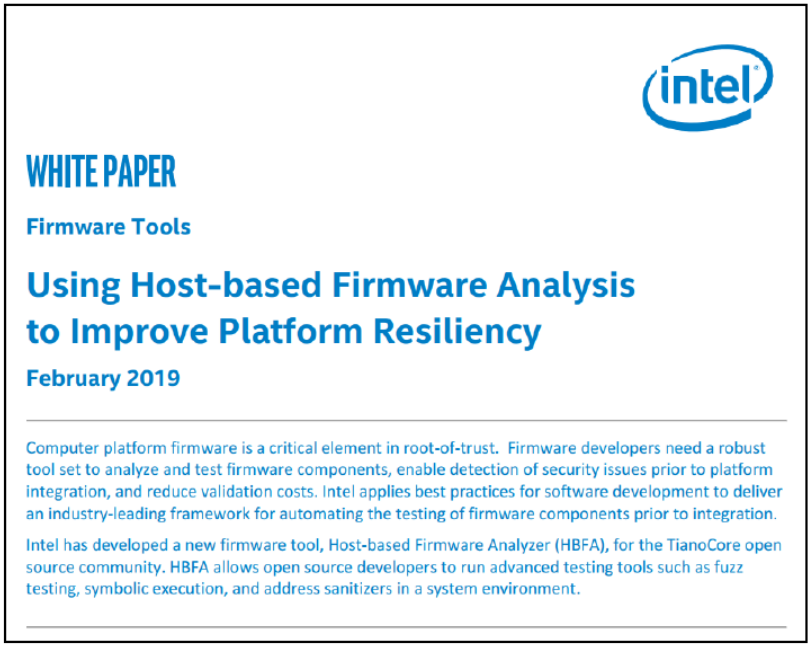
Effort to use kernel tools in EDK2

Utilize the experience from OS kernel

Host-based Firmware Analyzer is available now in edk2-staging

https://github.com/tianocore/edk2-staging/tree/HBFA

Download the whitepaper: *Using Host-based Firmware Analysis to Improve Platform Resiliency*



intel

WHITE PAPER

Firmware Tools

Using Host-based Firmware Analysis to Improve Platform Resiliency

February 2019

Computer platform firmware is a critical element in root-of-trust. Firmware developers need a robust tool set to analyze and test firmware components, enable detection of security issues prior to platform integration, and reduce validation costs. Intel applies best practices for software development to deliver an industry-leading framework for automating the testing of firmware components prior to integration.

Intel has developed a new firmware tool, Host-based Firmware Analyzer (HBFA), for the TianoCore open source community. HBFA allows open source developers to run advanced testing tools such as fuzz testing, symbolic execution, and address sanitizers in a system environment.

Reference: OSFC 2019 Hardening Firmware Components with Host-based Analysis Tools
By Brian Richardson

Burden the tools

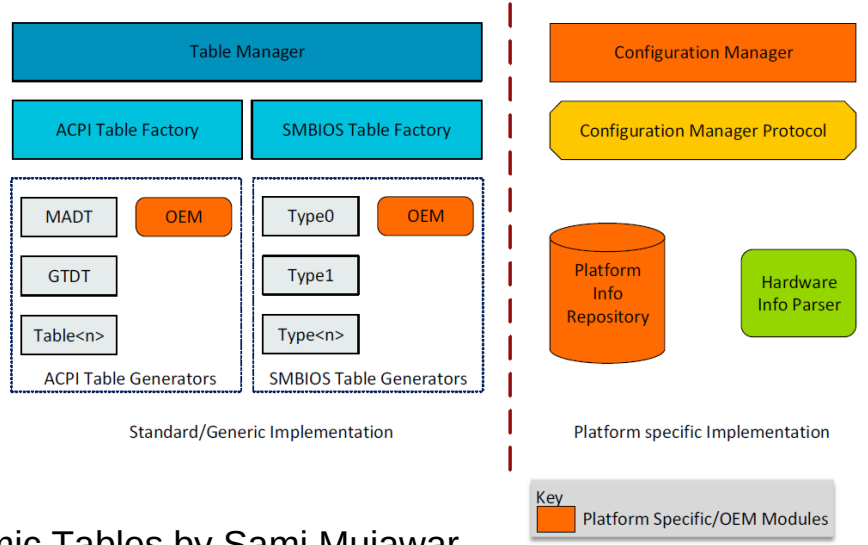
Automation making the initial development easier

- UEFI SCT - https://github.com/tianocore/edk2-test
- FWTS – https://wiki.ubuntu.com/FirmwareTestSuite

- **Dynamic Tables Framework**

ACPI
- AML Generation
- Device Tree Translation

SMBIOS



- **ACPIView**

Reference: OSFC 2019 An update on Dynamic Tables by Sami Mujawar

Call for Actions

A Linaro Recommended Approach

- Support large scale arm based system deployment from numerus vendors
  - Try to support complex server to simple IoT
- Leverage all the efforts we have made in the past a few years
  - SBSA QEMU Machine (https://connect.linaro.org/resources/yvr18/sessions/yvr18-511/)
    - The goal is to have a complete open platform with basic support for SBSA 3 features
  - SBSA QEMU Machine Firmware Prototype
  - CI for Running SBSA Test Suite on QEMU Machine
  - EDK2 Maintainer ship
- New efforts suggestions
  - UEFI LTS similar to UDK2018 but modified and verified with arm based platforms not necessary QEMU platform
  - Remove any advanced server features and make them installable packages for cloud, storage, edge and PC
  - Work with ATF and ACPI such that these are also become installable
  - Reducing the footprint for Mobile and IoT device similar to minimum platform

Thank you

Join Linaro to accelerate deployment of your Arm-based solutions through collaboration

contactus@linaro.org

kangkang.shen@futurewei.com

Linaro connect
San Diego 2019