Open Source Host Firmware Directions

Vincent Zimmer

Email: vincent.zimmer@intel.com

Twitter: @vincentzimmer

Platform Security Summit - May 23, 2018

Disclaimer

- Intel has official documentation that is highly comprehensive and should be used to make technical decisions related to Intel's technologies. To have a high-level of accuracy for such documentation, lots of reviews are performed

- The accuracy of this talk can't be compared and should not be used to compare with official documentation. We are going to discuss directions, strategies and initiatives being proposed, giving recommendations to OEMs, researchers and customers but everything should be treated as my opinion instead of official statements

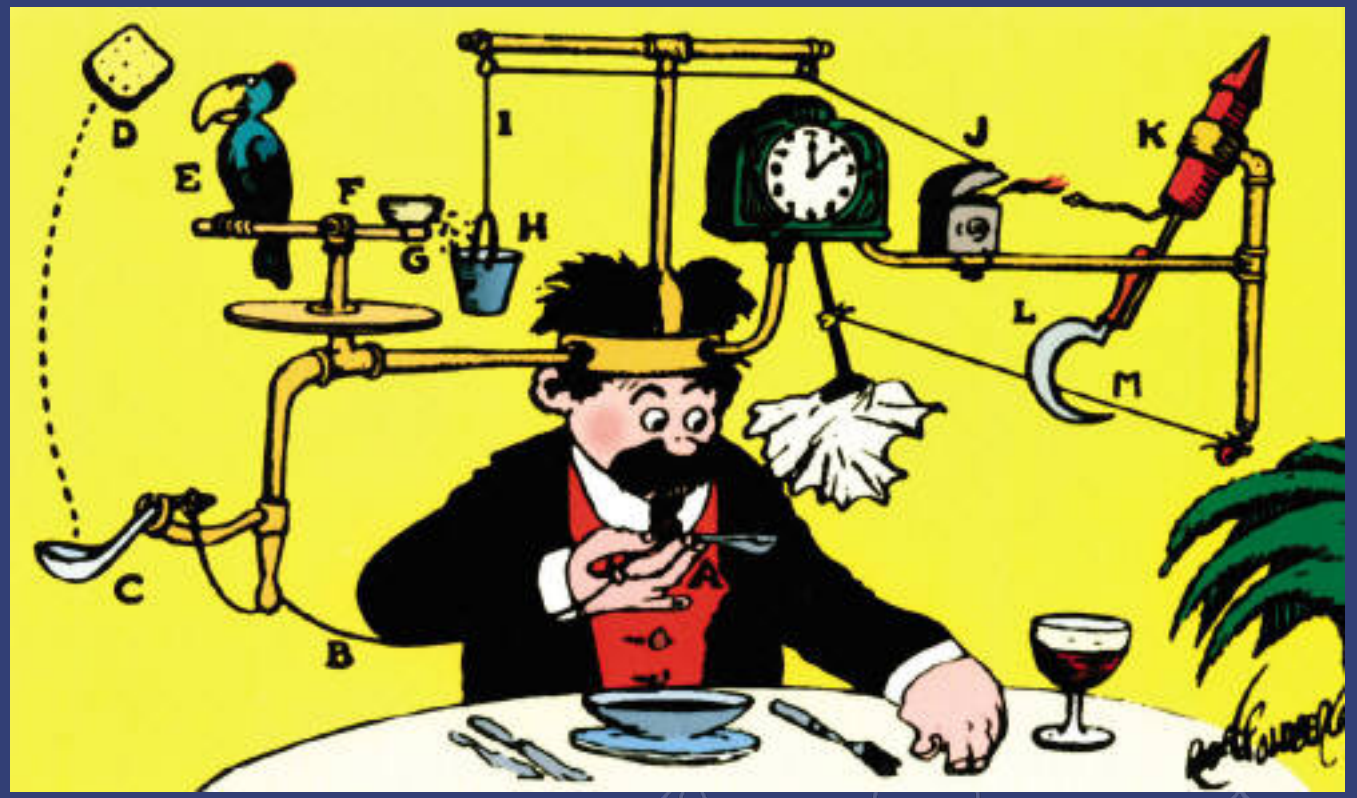
- There possibly are other initiatives and focus areas that we are not at liberty of talk about or that we are even unaware of, so this should not be considered the full scope of the problem, but instead, *OUR* vision of it and our opinion.

Agenda

- **History**
- Progress
- Challenges
- Call to action

BIOS

Blame It On Software



From http://www.thinglink.com/scene/498556018750390274

In the beginning….

19XX

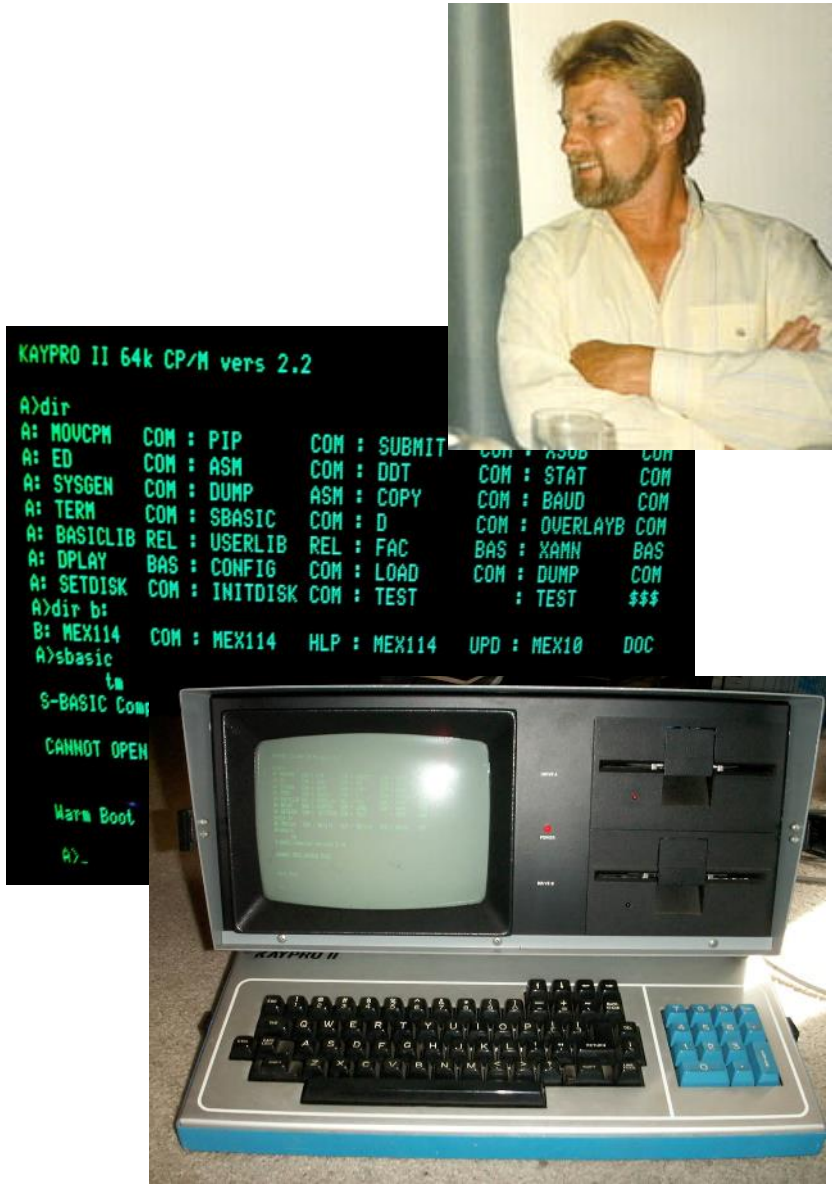
Pioneer

CP/M

BIOS
(machine specific CP/M)

8080/Z80

1974 Basic I/O (Sub) System
by Gary Kildall in CP/M



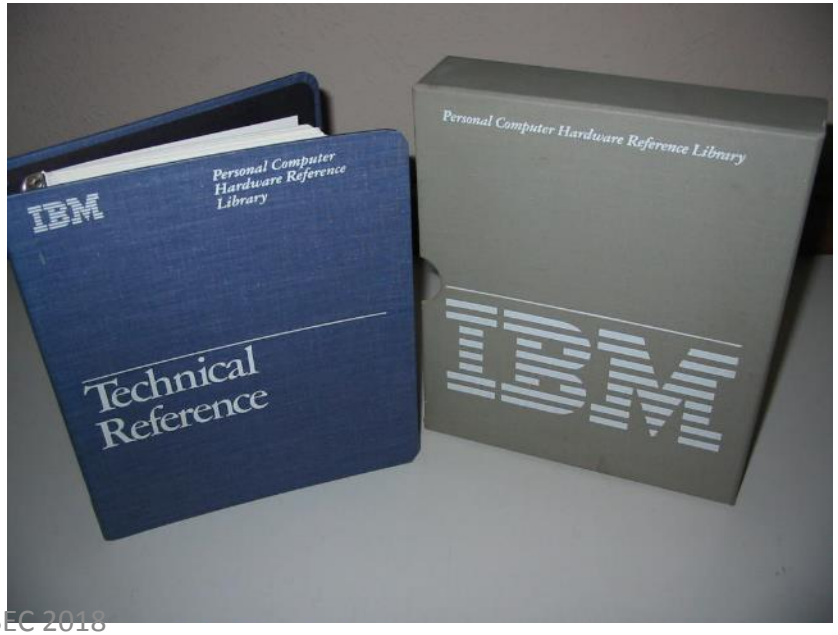
PC/AT BIOS

DOS

BIOS
(de facto standard)

8088

1981 IBM PC



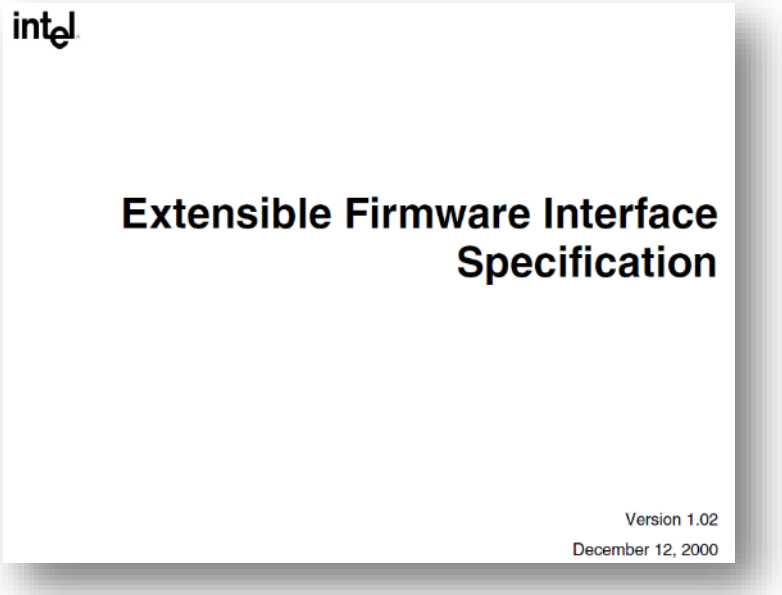
PC/AT BIOS -> EFI

IPF Windows/Linux

EFI

(Intel Standard)

IPF (Merced)



2000 Extensible Firmware Interface

Intel/HP IPF

Industry Transition

Pre-2000

All Platforms BIOS were proprietary

2000

Intel invented the Extensible Firmware Interface (EFI) and provided sample implementation under free BSD terms

2004

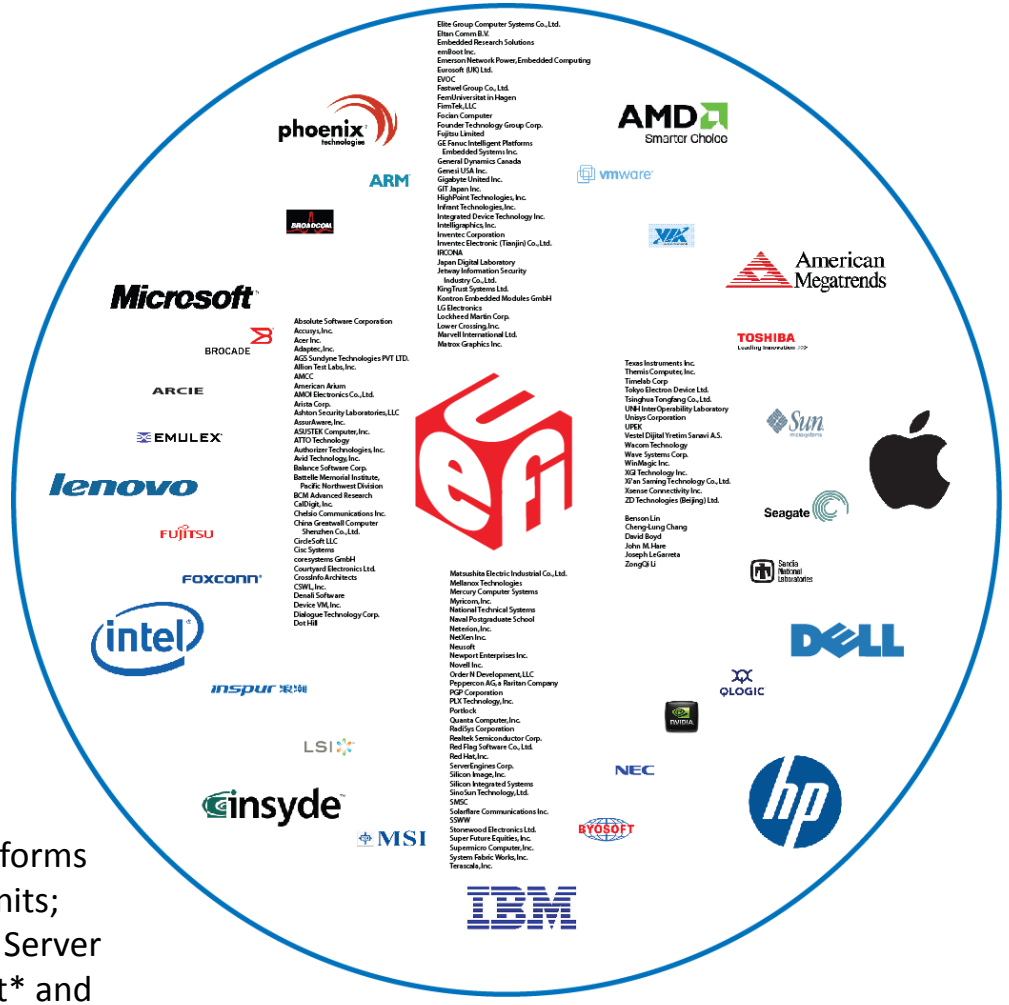
tianocore.org, open source EFI community launched

2005

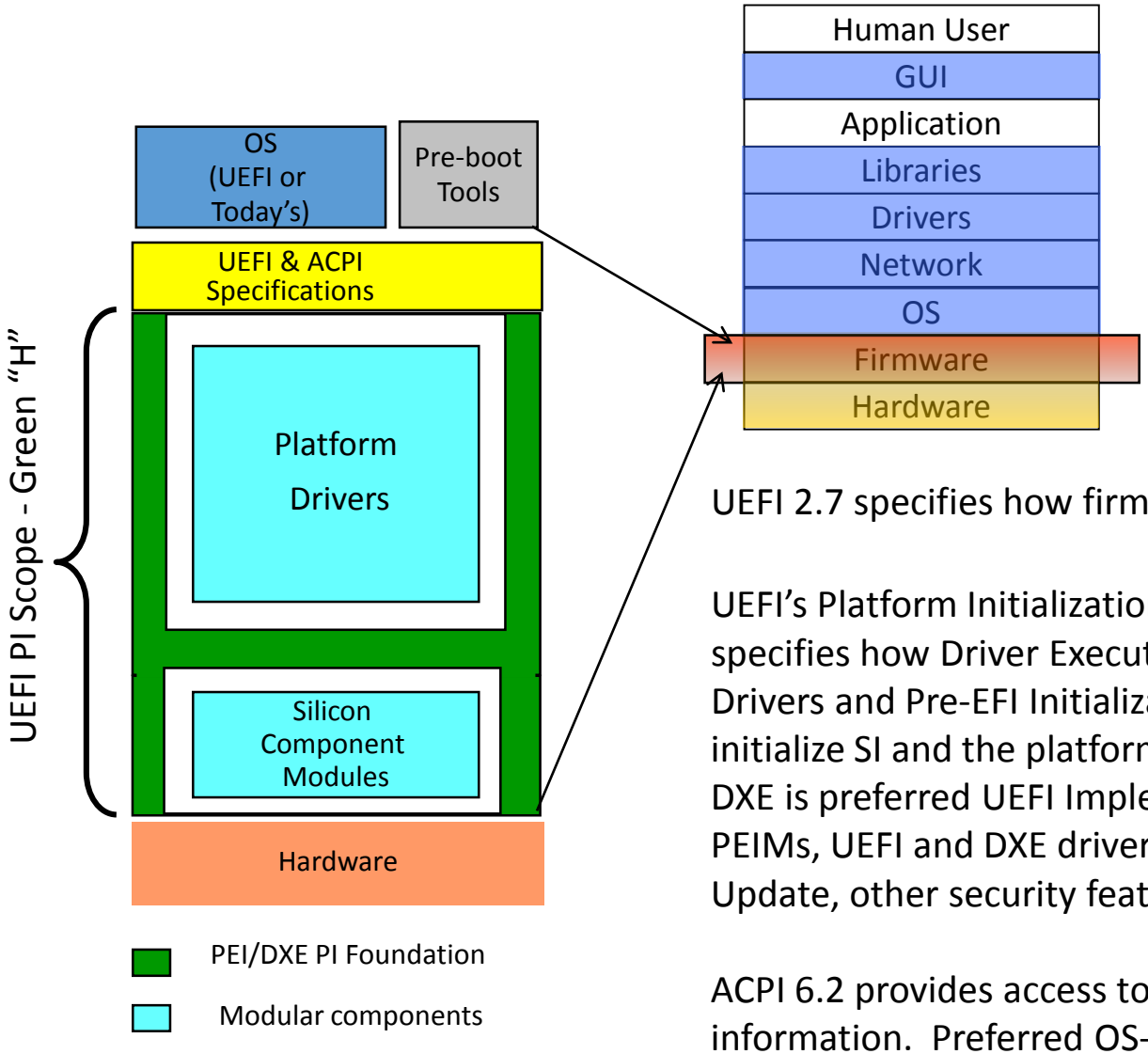
Unified EFI (UEFI)
Industry forum, with 11 members, was formed to standardize EFI

2018

240 members and growing!
Major MNCs shipping; UEFI platforms crossed most of IA worldwide units; Microsoft* UEFI x64 support in Server 2008, Vista* and Win7*; RedHat* and SuSEI* OS support. Mandatory for Windows 8 client. ARM 32 and 64 bit support. ACPI added.



Today's Stack



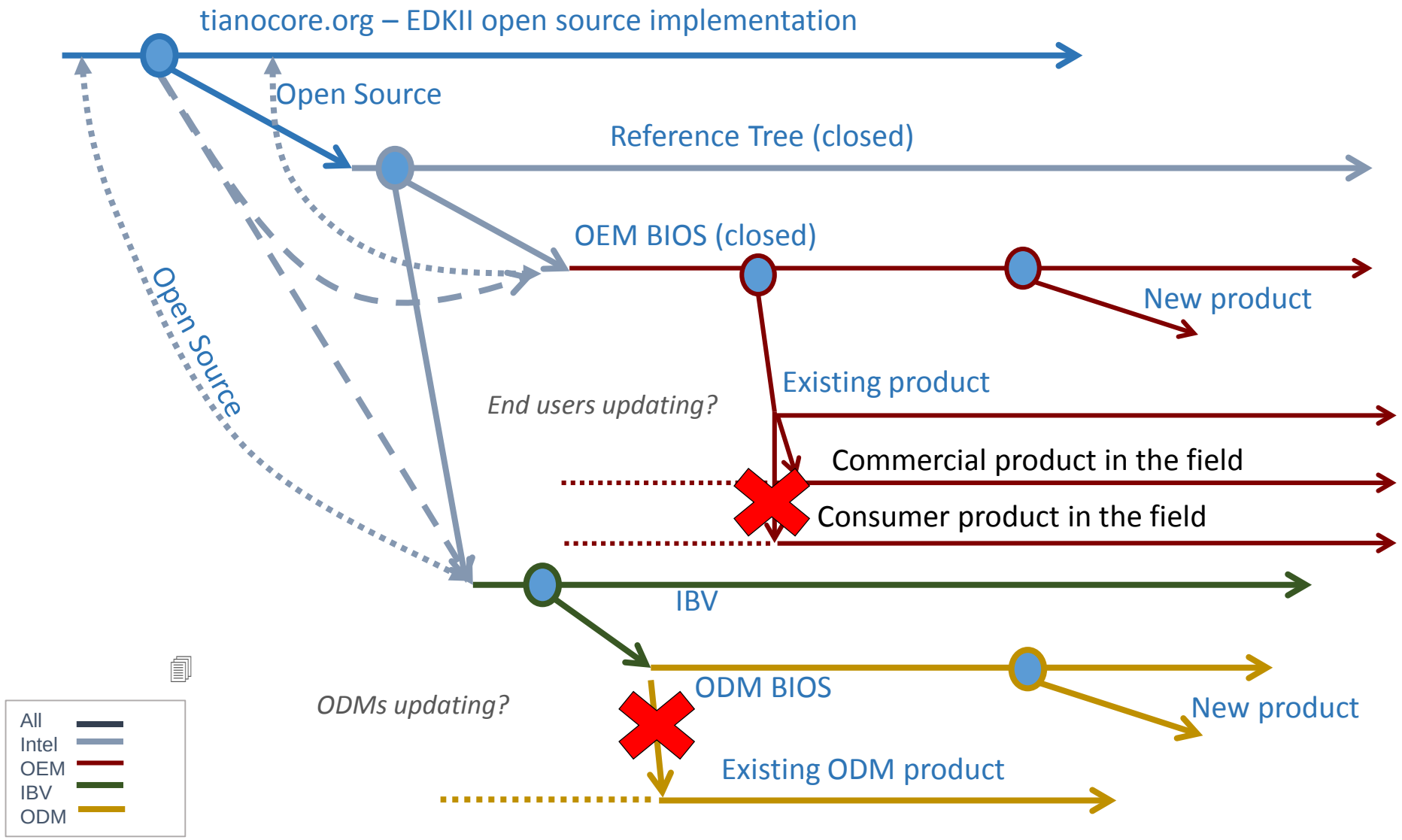
Full system stack
(user -> hardware)

UEFI 2.7 specifies how firmware boots OS loader

UEFI's Platform Initialization (PI) 1.6 Architecture specifies how Driver Execution Environment (DXE) Drivers and Pre-EFI Initialization (PEI) Modules (PEIMs) initialize SI and the platform
DXE is preferred UEFI Implementation
PEIMs, UEFI and DXE drivers implement networking, Update, other security features

ACPI 6.2 provides access to non-discoverable platform information. Preferred OS-to-platform runtime interface

Todays' ecosystem



Agenda

- History
- **Progress**
- Challenges
- Call to action

Firmware options

From https://en.wikipedia.org/wiki/Chinese_restaurant

Do others believe this?

The Future of Firmware

We have been witnessing an interesting phenomenon since the beginning of this century: open source projects are gaining momentum, led by companies such as Google and Facebook. Many legacy and proprietary software solutions are either disappearing or losing steam very quickly; open source solutions are becoming a primary interest of technologists at an amazing speed.

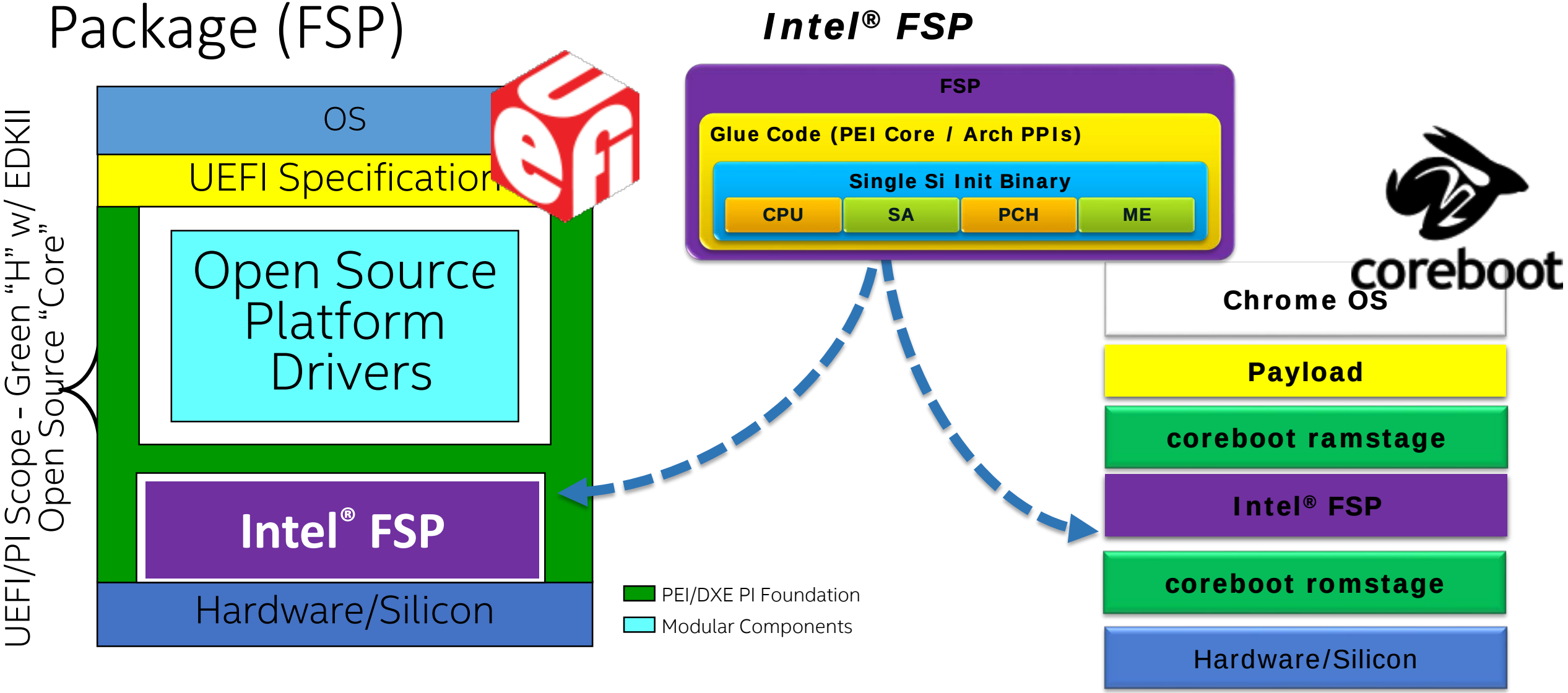
Even though this century is still young, we are riding on a fascinating wave that will make the 21st century a distinctly different century than any other. The phrase "open source" clearly connotes sharing and collaboration, in contrast to the waning business philosophy of

- From https://www.apress.com/us/book/9781484200711

What are we doing

- Open development environment
  - Open source core
  - Open source platform code
  - IP protected initialization in well-defined binary blob
  - Open up all of the build tools

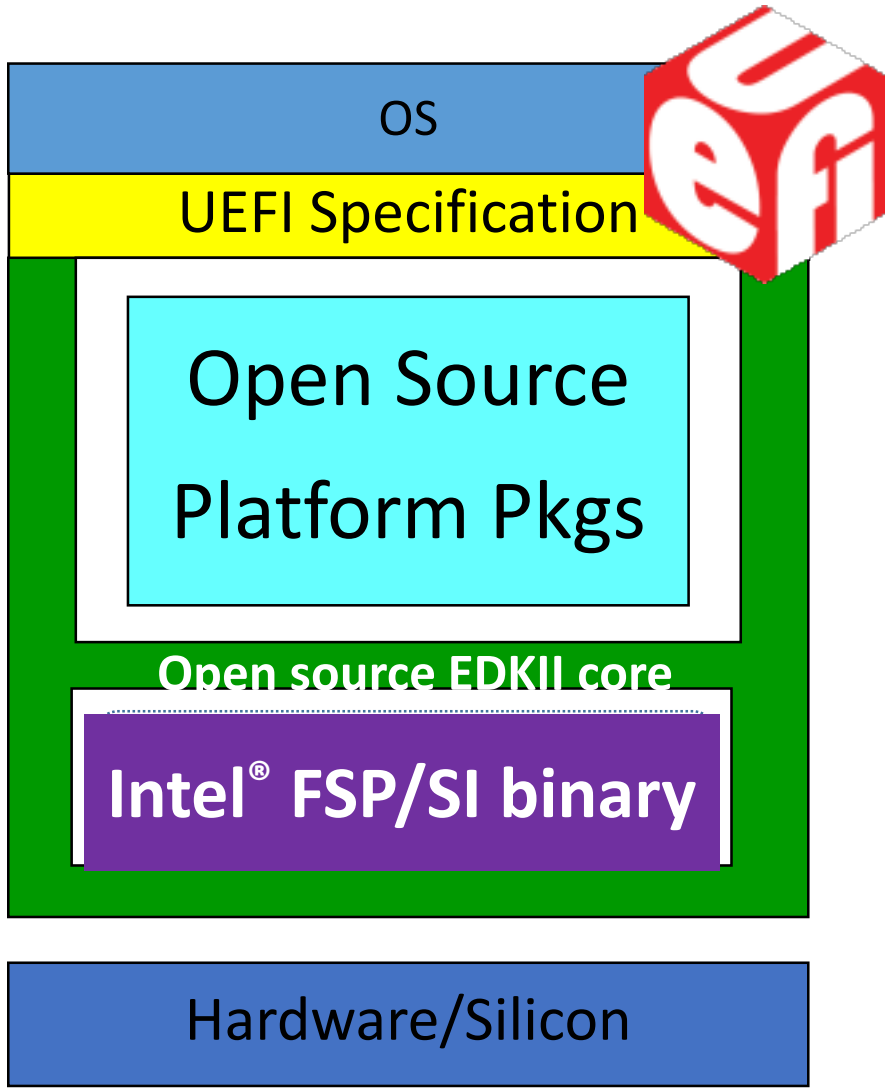
UEFI and coreboot with the Intel Firmware Support Package (FSP)



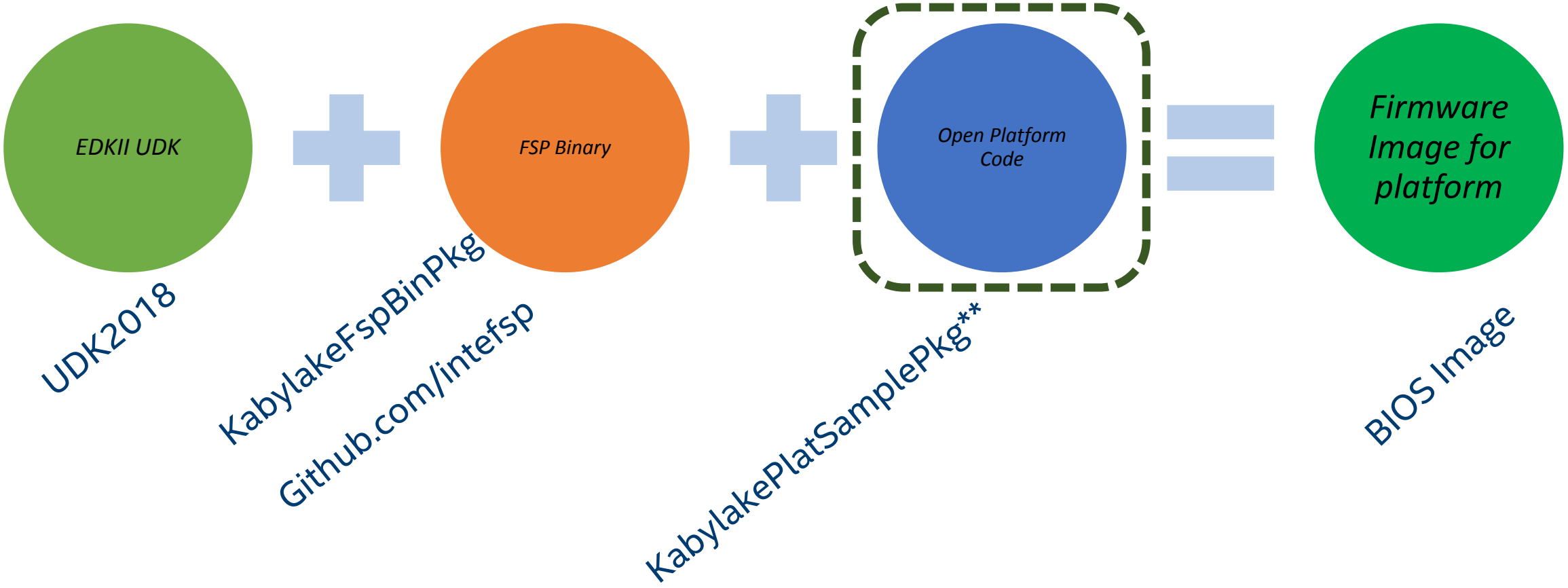
From https://firmware.intel.com/sites/default/files/resources/SF14_STTS001_102f.pdf

Internal mode of evolution

- **FSP / Binary FV's** - Evolution of the Intel ® Firmware Support Package (FSP) from 1.0 to 1.1(simplified boot flow), to 2.0 – Intel.com/fsp
- **Open Source platform code** – Simplified, product quality, open source capable platform package. Built on industry standards and EDK II technology for ease of porting. Upstream platform code. – tianocore.org
- EDKII – existing upstream/open source core
- MinTree – minimum open source core and platform code to boot shrinkwrap OS



Putting it all together



Why Intel FSP?

- Lower threshold for IA adoption
  - Royalty-free binary with standard APIs to integrate (Easier to port)
  - Easier license obtaining (IP encapsulated in binary)
- Faster development cycles
  - Reuse pre-validated silicon code
  - Direct drop-in model for updates
- Better flexibility and scalability
  - Leave board specific init and silicon init configuration for bootloader
  - Can be configured dynamically (UPD) or statically (BCT)
- More engagement with the whole ecosystem

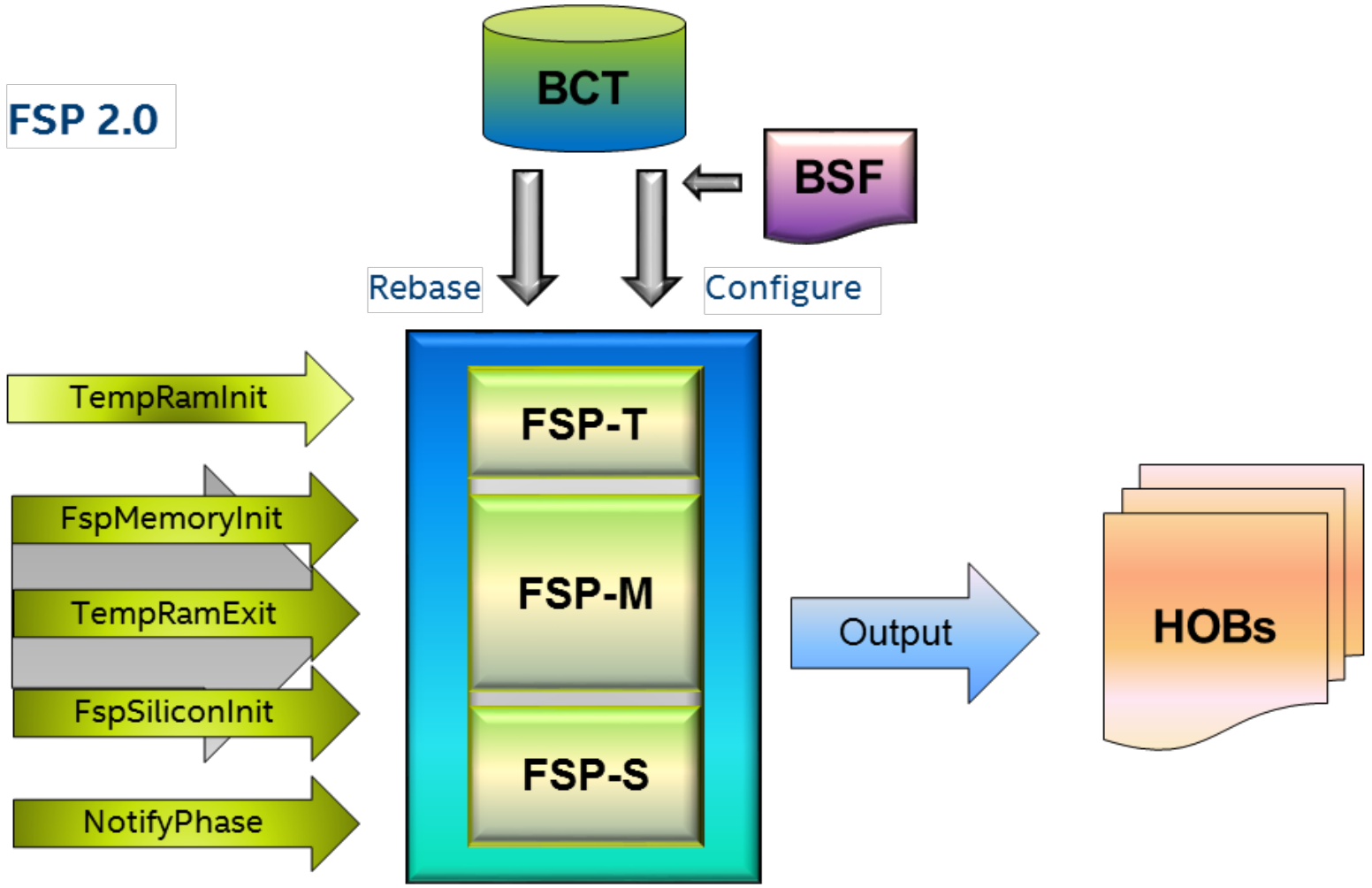
Intel FSP Journey

A Path to Simplicity and Flexibility

- Initial FSP 1.0 implementation prototype, consumed in OTM bootloader.

- Proof of Concept (Middle 2012)
- Intel® FSP 1.0 (Apr 2014)
  - Defined TempRamInit/FspInit/FspNotify APIs, HOB structures and boot flow.
- Intel® FSP 1.1 (Apr 2015) /1.1a (Nov 2015)
  - Split FspInit API into FspMemoryInit/TempRamExit/FspSiliconInit to allow more flexibility for initialization flow
- Intel® FSP 2.0 (May 2016)
  - Group APIs into FSP-T/M/S components to support boot from block devices

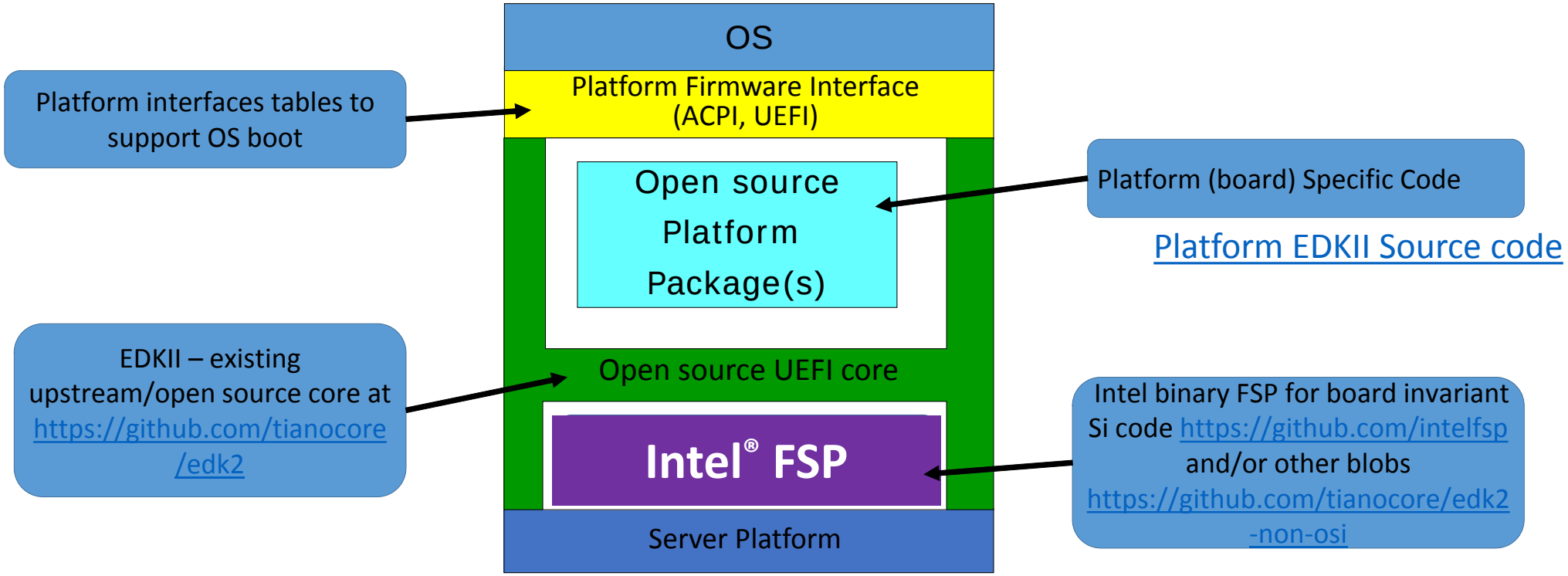
Intel FSP Journey in pictures



From the Open Compute Conference 2018

From https://www.youtube.com/watch?v=Dh6N7Pj1CL

UEFI-based Open Firmware
(for Intel-based Server Platforms)



Mt. Olympus Xeon-based platform with UEFI-based open firmware available

What has Intel released for the Purley platform?

- In March 2018, Intel released an Open Source UEFI Firmware implementation for the Intel XSP Motherboard, based on the Intel® Xeon® Scalable Processor family (formerly codenamed "Purley"). This platform is part of Microsoft's Project Olympus, a next generation rack-level solution open-sourced through the Open Compute Project (OCP)
- This tree follows a "minimum platform" (MinPlatform) philosophy Min Platform Design, providing boot to a UEFI compliant operating system using a minimalist approach to managing features, code, complexity, and developer effort
- Have an open substrate to collaborate with parties in the OCP Open System Firmware (OSF) project
  - Create and deploy, at scale, an open source hardware platform initialization and OS load firmware optimized for web-scale cloud hardware, including documentation, testing, integration and any other artifacts that aid the development, deployment, operation or adoption of the open source project. [from OSF Charter]

Min Server Background

- The Purley project uses binaries in the edk2-non-osi repository for platform silicon initialization. These binaries are built from the existing Intel silicon support UEFI firmware modules delivered to customers under NDA
  - Goal is to be more transparent with the silicon support code, providing more as open source in the future
- Expect to have mixed source and binary solutions for supporting silicon products for the foreseeable future
- The Purley project binaries are not Intel® FSP compliant. These binaries are UEFI PI Architecture Firmware Volumes containing UEFI PI Architecture PEIM and drivers

Why release this firmware implementation in open source?

- Released the Purley MinPlatform UEFI firmware project in TianoCore as part of the support of OCP. Projects like OCP have incorporated openness into their core tenants:

- *The Open Compute Project Foundation is a rapidly growing, global community whose mission is to design, use, and enable mainstream delivery of the most efficient designs for scalable computing. We believe that openly sharing ideas, specifications, and other intellectual property is the key to maximizing innovation and reducing complexity in tech components.*

-- *http://www.opencompute.org/about/*

- From the OCP perspective, this open development approach extends through the entire software stack

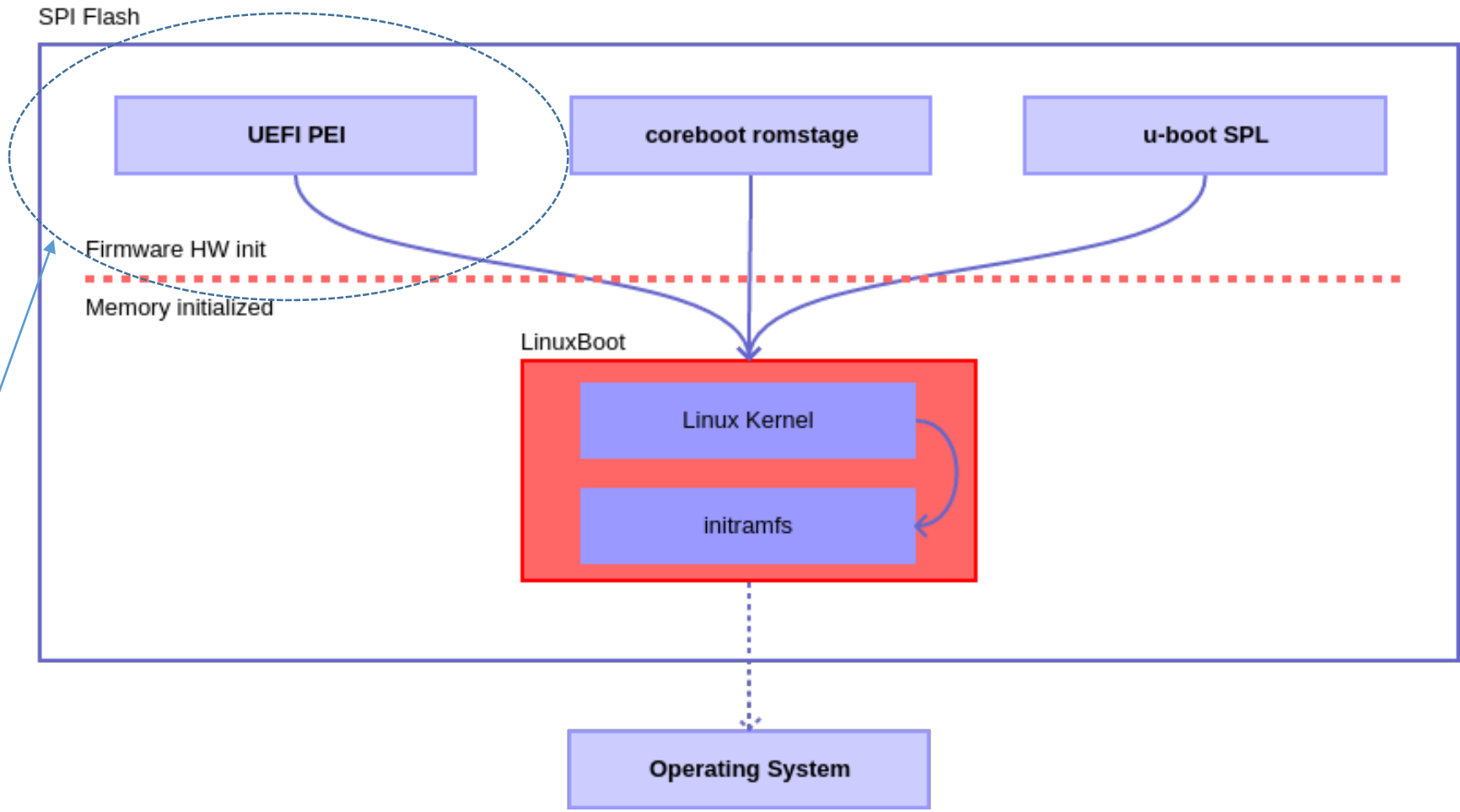
- Intend to use the MinPlatform to demonstrate best practices for things like simplified firmware implementations, fast boot times, legacy removal, and demonstrate firmware features for base platforms

What are the capabilities available in a MinPlatform firmware tree?

- Developers using MinPlatform can build a functional firmware image from TianoCore content in GitHub:
- Upstream open source EDK II (https://github.com/tianocore/edk2)
- Platform code, including SMBIOS and ACPI (https://github.com/tianocore/edk2-platforms)
- Closed source binaries for silicon initialization code (https://github.com/tianocore/edk2-non-osi)
- This firmware image boot can boot shrink-wrap UEFI OS from local media (NVMe) or network devices (PXE). Additional features include UEFI Secure Boot and TPM support.
  - Future additions include Capsule Update, additional platforms
- Customers who require features beyond the MinPlatform implementation can work with their third-party firmware vendors to develop advanced platform features and custom solutions
- Allow for composing other boot solutions, such as ….

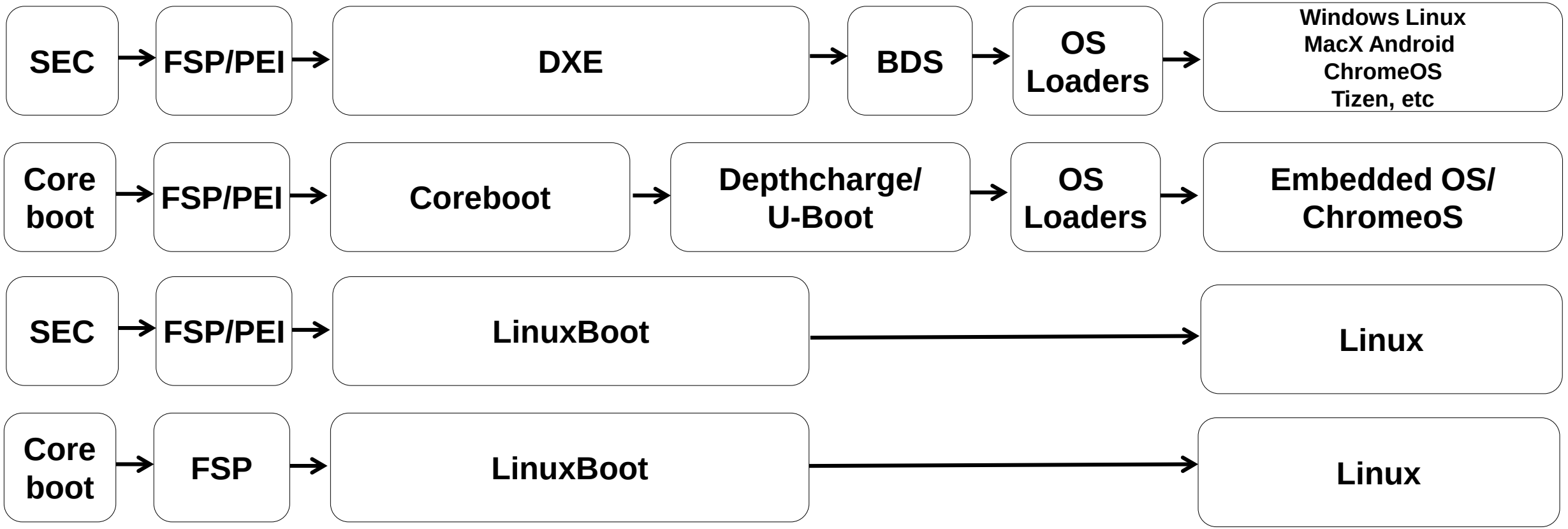
A richer world of booting

- Extend the concept of Payload to have a full OS – Linuxboot https://www.linuxboot.org/
- Reuse Linux drivers instead of UEFI drivers
- Leverage the base infrastructure from the Min Server to compose other payloads (e.g., UEFI PEI above can be leveraged from Min Server, re-use EDKII build tools, etc)



From https://www.linuxboot.org/

Tying it all up



Status on open source

- Active work stream in Open Compute Conference (OCP) for Open Source http://www.opencompute.org/wiki/Open_System_Firmware

- Intel FSP 2.0 binaries for all client Atom and Core CPU's
- https://github.com/intelfsp and other opaque binaries at https://github.com/tianocore/edk2-non-osi/

- Open source EDKII platform code for IOT, client and server at https://github.com/tianocore/edk2-platforms

- UEFI EDKII core at https://github.com/tianocore/edk2

- Open source platforms for Atom, Core and Microserver at https://github.com/coreboot/coreboot

Agenda

- History
- Progress
- **Challenges**
- Call to action

Challenges

- Free up tools
  - Many SI tools are still closed
- Free up SI code
  - Intel FSP considered ‘soft’ lock down. Can go 2 paths – hard lock-down/boot-rom or liberate code and fully open source
- Documentation delay
  - Open source has to await public documents like EDS
- Debug of binaries

Agenda

- History
- Progress
- Challenges
- **Call to action**

Call to action

- Provide feedback on this direction
- Get involved in the various open source firmware and standards activities

More information

- http://www.uefi.org
- http://ww.tianocore.org
- https://github.com/tianocore/edk2
- https://github.com/tianocore/edk2-platforms
- https://github.com/tianocore/tianocore.github.io/wiki/EDK-II-white-papers
- https://github.com/IntelFsp/FSP
- http://www.intel.com/fsp
- http://firmware.intel.com
- http://www.coreboot.org
- http://opencompute.org/
- http://opencompute.org/projects/open-system-firmware/
- https://www.apress.com/us/book/9781484200711
- https://www.degruyter.com/view/product/484468
- https://www.degruyter.com/view/product/484477
- https://www.youtube.com/watch?v=Dh6N7Pj1CL
- https://cansecwest.com/slides/2015/UEFI%20open%20platforms_Vincent.pptx
- https://github.com/rrbranco/BlackHat2017/blob/master/BlackHat2017-BlackBIOS-v0.13-Published.pdf
- https://github.com/tianocore/edk2-platforms/blob/devel-MinPlatform/Platform/Intel/MinPlatformPkg/Docs/A_Tour_Beyond_BIOS_Open_Source_IA_Firmware_Platform_Design_Guide_in_EFI_Developer_Kit_II%20-%20V2.pdf
- https://firmware.intel.com/sites/default/files/A_Tour_Beyond_BIOS_Creating_the_Intel_Firmware_Support_Package_with_the_EFI_Developer_Kit_II_%28FSP2.0%29.pdf
- https://firmware.intel.com/sites/default/files/A_Tour_Beyond_BIOS_Using_the_Intel_Firmware_Support_Package_with_the_EFI_Developer_Kit_II_%28FSP2.0%29.pdf