Different Deployment Models for UEFI Firmware on Intel Platforms

Presented by Brian Richardson

Intel Corporation

presented by

UEFI Plugfest – October 2014

Agenda

Our “Standard” Firmware Model

Why Add New Models?

Various Options for Intel Platforms

Summary / Q&A

Our “Standard” Firmware Model

For Intel Architecture (IA), the common model involves the platform OEM/ODM working with an independent BIOS vendor (IBV) to generate firmware.

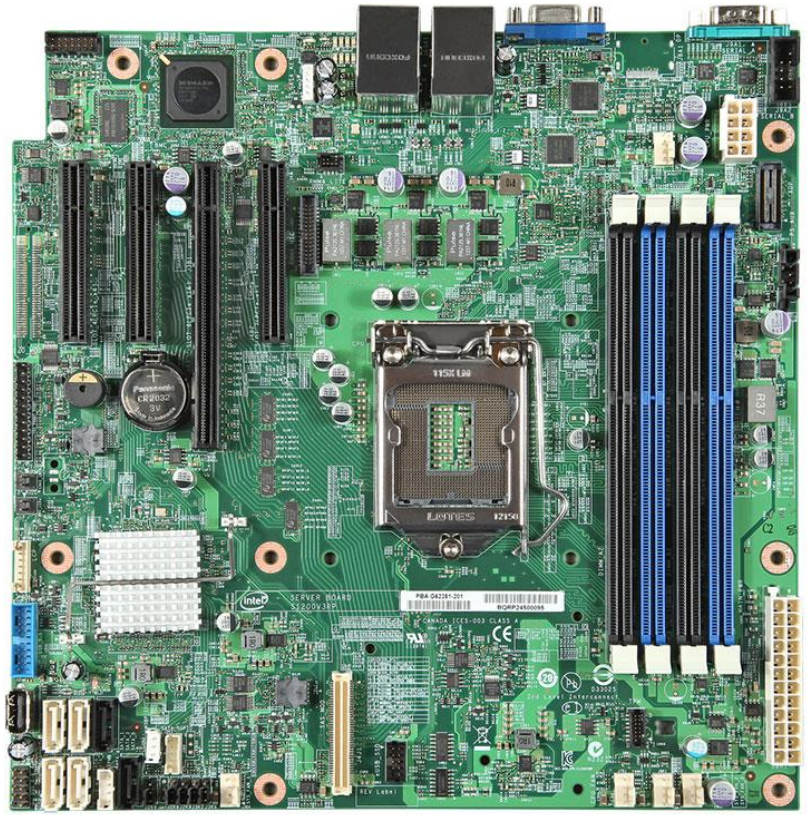
- License a codebase or use turnkey development.
- Allows heavy customization & differentiation.
- Option to contract resources as needed.

Of course, there are exceptions

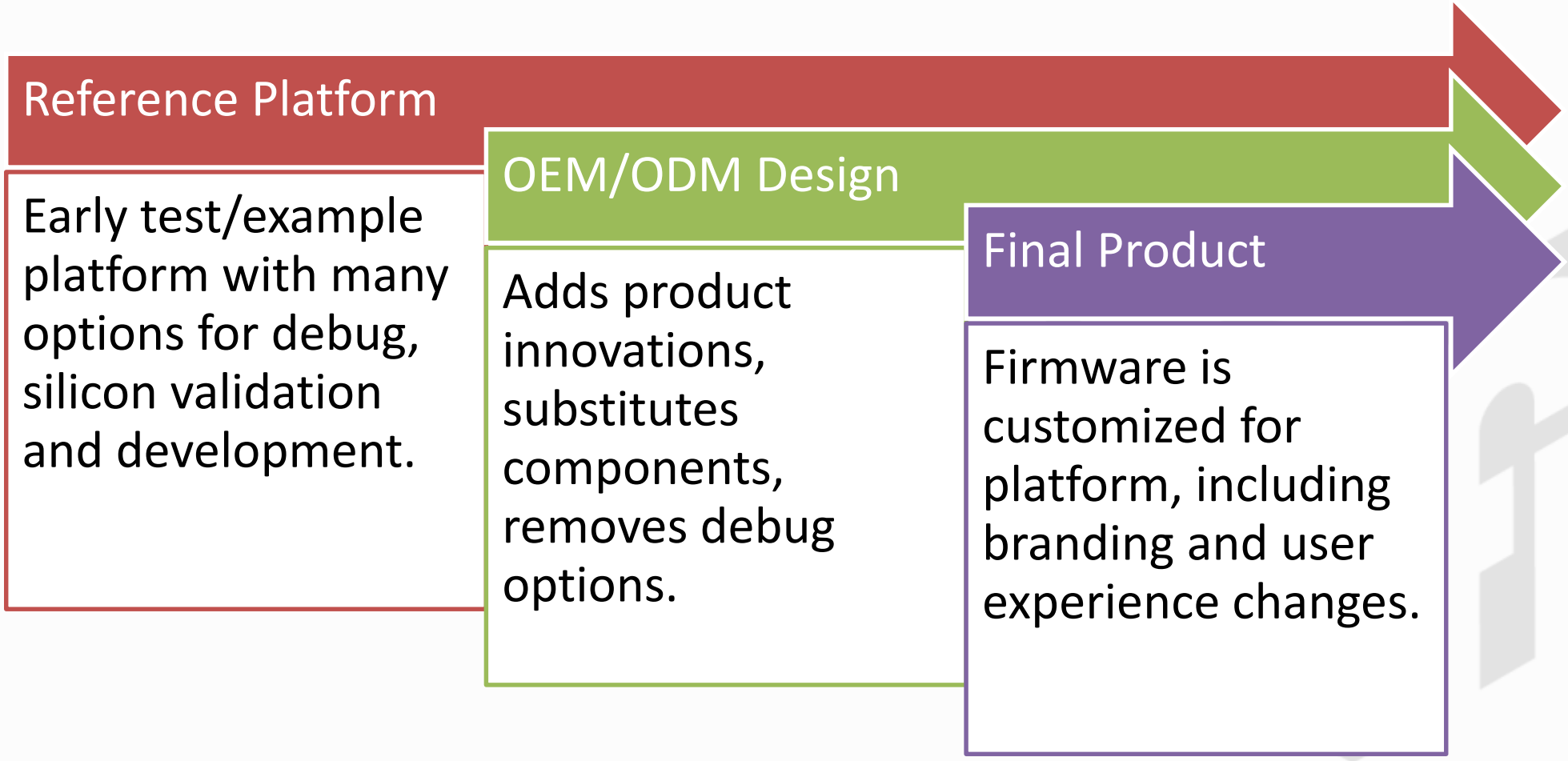
Reference Platforms come with UEFI images from Intel to test and validate silicon.

UEFI Development Kits add the latest UEFI firmware to stable Intel platforms for OS and peripheral testing.

Other UEFI members have similar validation programs.



Firmware from Reference to Product



New Markets Bring New Requirements

Landscape has changed since the PC/AT BIOS was introduced.

New products call for different solutions.

Simple derivatives

- Product w/o complex firmware requirements

Open Hardware & Maker Products

- Relies heavily on open source solutions

New IA developers

- New products, new business models, different ecosystems and customer requirements …

So … why tell the plugfest audience?

Most of the developers at a UEFI Plugfest know and understand the existing IA firmware ecosystem.

Those developers should also understand any new options in Intel's enabling ecosystem …

- New options for platform debug & testing.
- New options for enabling with UEFI

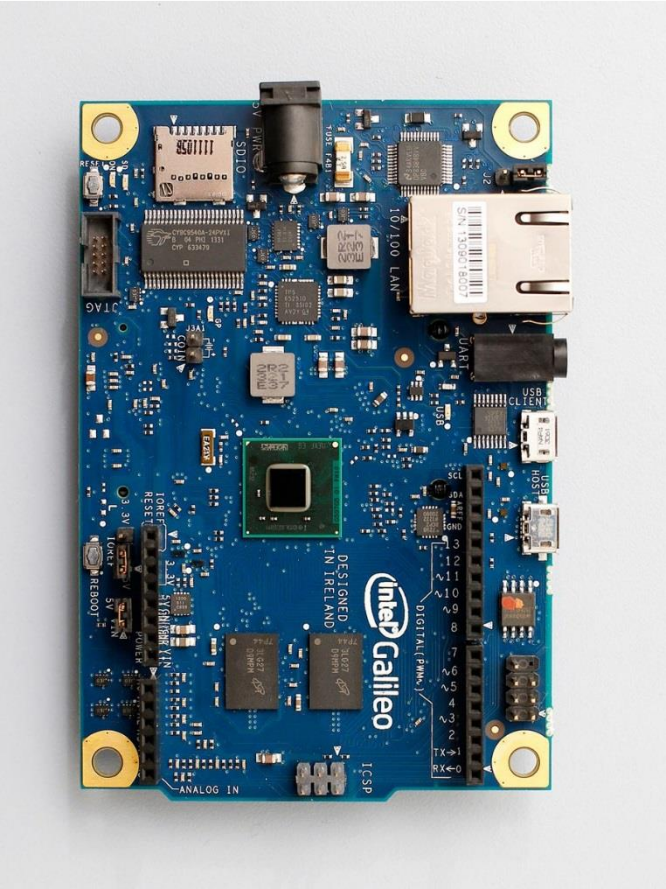
Various Options for Intel Platforms

These are *in addition to* the traditional ecosystem.

Breaks down into three major categories …

- Full Open Source
- Open Source + Binary Components
- Configurable Binary Components

Full Open Source



Intel is using open hardware designs in IoTG markets.

New maker/hobby products use open hardware schematics and open source reference code for UEFI & EDK II (no NDA required).

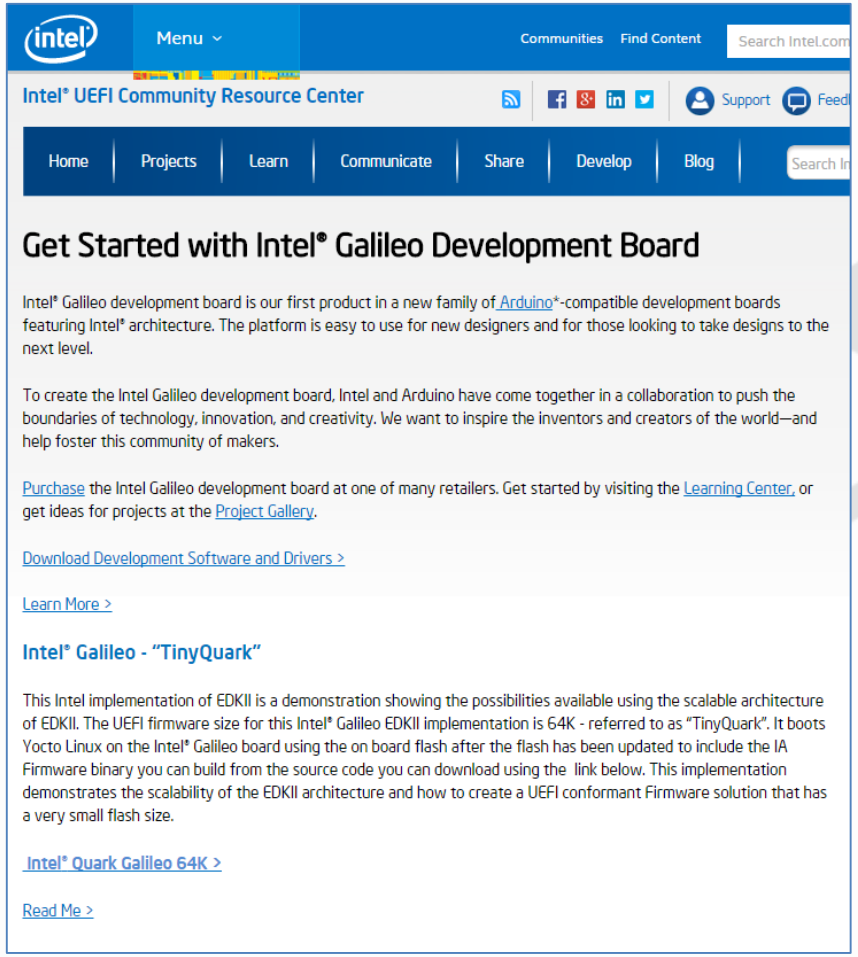
Example: Intel® Galileo Platform

Small footprint project w/ EDK II

Two firmware projects …

1. Standard firmware package, based on EDK II.
2. "TinyQuark" variant, 64KB UEFI firmware that directly boots to Yocto Linux in-flash

More information at uefidk.com



Open Source + Binary Components

Similar approach to maker product in regard to open hardware design.

However, some silicon components rely on IP that cannot be placed into open source.

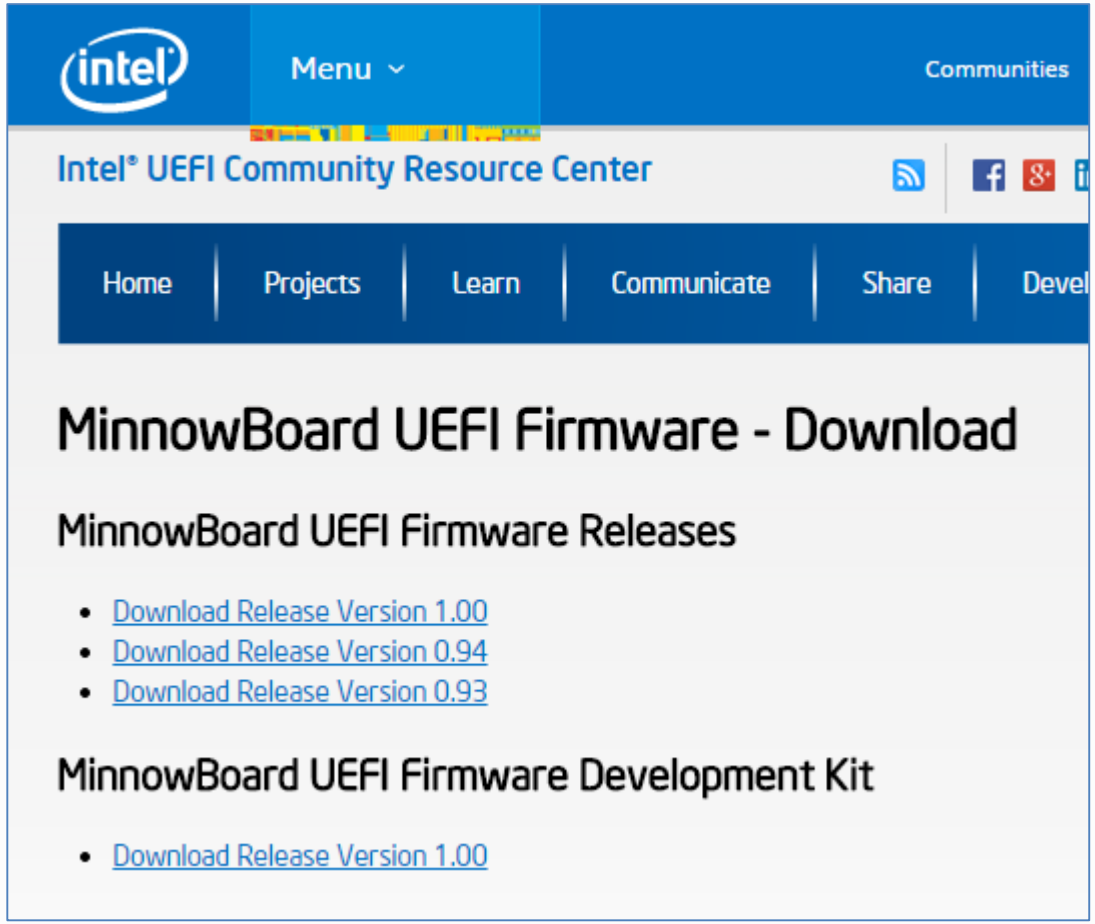
Example: MinnowBoard/MinnowBoard Max

Firmware for this open hardware design has two primary options:

1. EDK II + binary modules
2. EDK II + Intel® Firmware Support Package (FSP)

Accommodates need to keep some IP protected in firmware.

Example #1: EDK II + Binary Modules

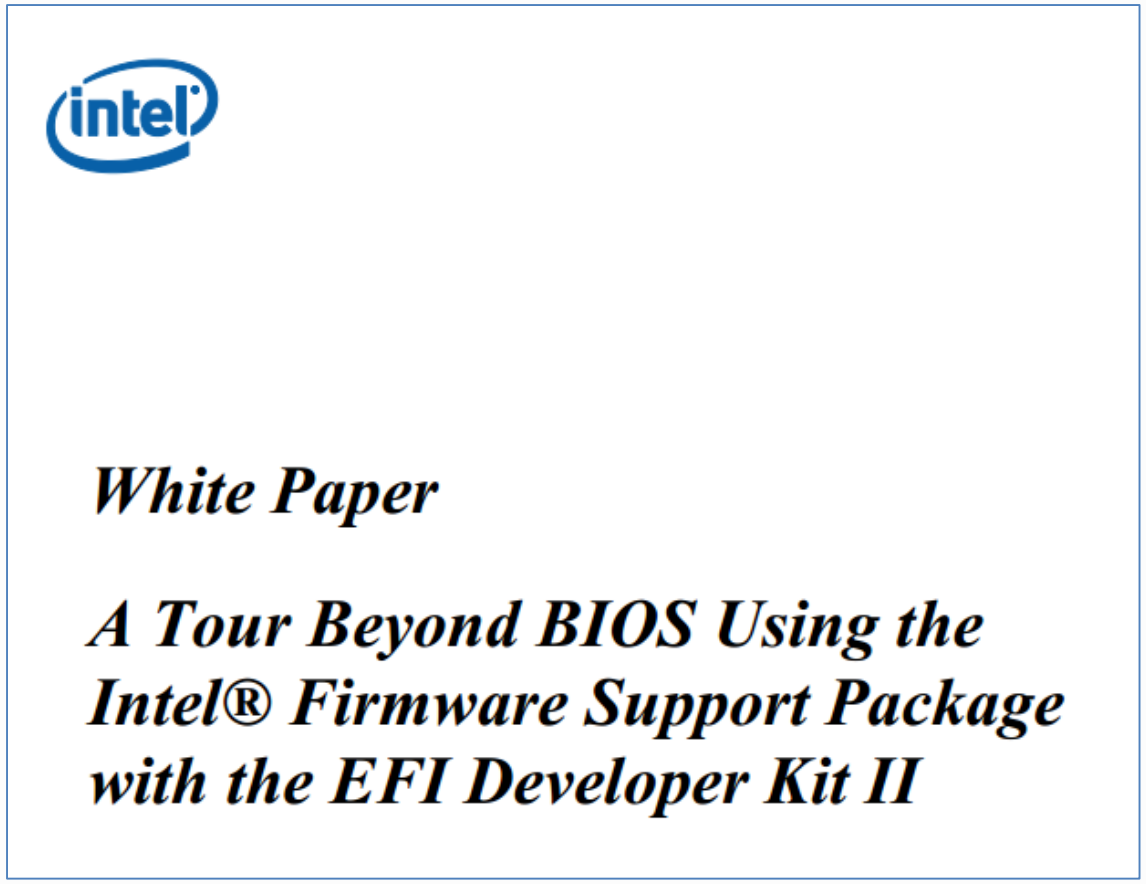


Firmware development kit combines EDK II code with select processor/chipset init modules.

Allows deeper development and debug w/o 100% open source.

Binary components are not configurable by the developer.

Example #2: EDK II + Intel® FSP



[Whitepaper available at uefidk.com]

Intel® FSP is an initialization binary, based on UEFI PI specs.

Developers use tools to set how Intel FSP configures hardware.

EDK II can "consume" Intel FSP

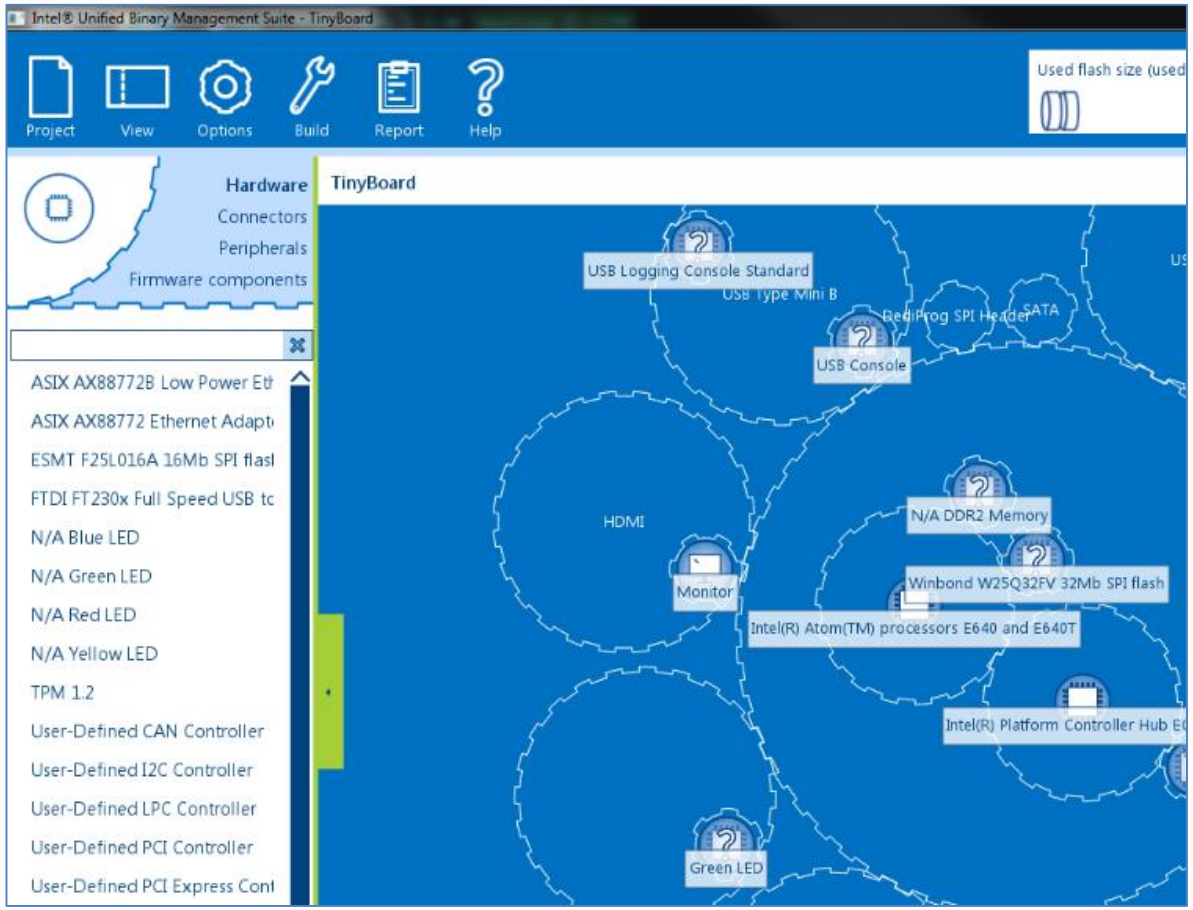
- *SecCore* hands off to Intel FSP
- FSP produces HOBs at end

Configurable Binary Components

Uses pre-compiled PEIM/DXE with configurable options for fast deployment.

Designed for platforms with simple firmware needs.

Example: Intel® Unified Binary Management Suite (Intel® UBMS)



So, which approach is the best?

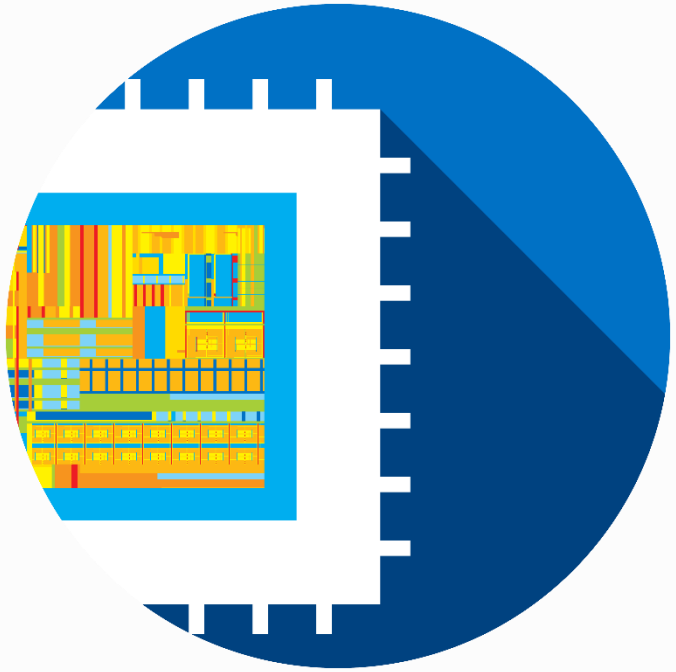
Depends on each market's business requirements.

Not every enabling model fits into every market segment.

Some markets are still enabled using the IBV ecosystem, while other markets add new options.

- *Does the end-user need to modify the firmware?*
- *Do you need to protect any IP in the platform firmware?*
- *Will customers use the firmware to make derivative products based on this design?*
- *What level of firmware debugging is required?*
- *Which option bets fits the platform's security needs?*

Summary / Q&A



Intel is introducing new firmware options to support UEFI.

Incorporating configurable binary components and open source options.

Allows developers to pick the best firmware option for their IA product.

For more information on the Unified EFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by