



UEFI & EDK II Training

UEFI AND PLATFORM INITIALIZATION (PI) BOOT FLOW &
OVERVIEW

tianocore.org

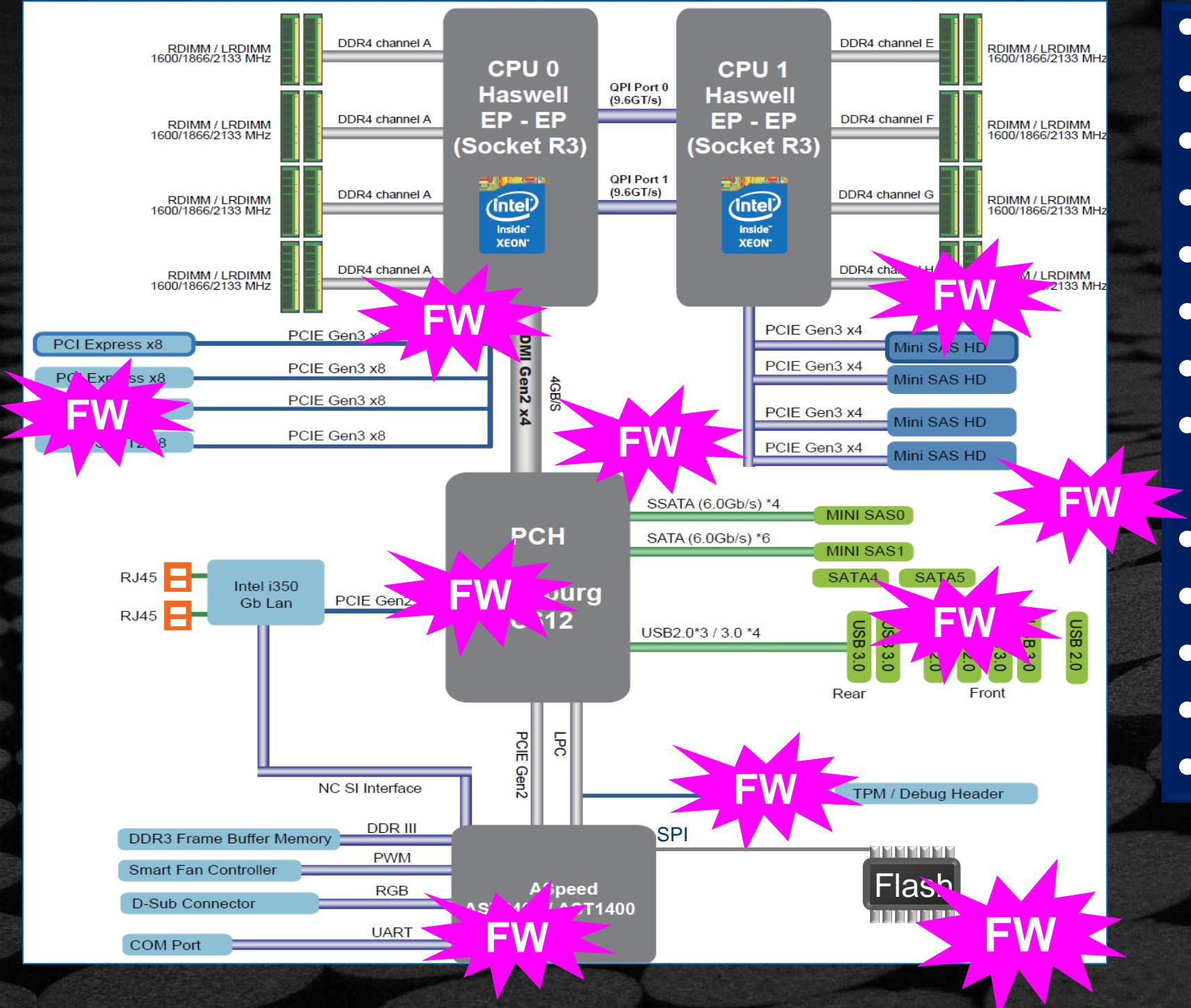
LESSON OBJECTIVE

- ★ Review PI and UEFI Boot Process
- ★ Answer web-based training related questions
- ★ Answer: Where does Intel® FSP Fit?
- ★ What's new in UEFI.org

WHERE IS THE FIRMWARE

Where is the UEFI Firmware on a platform

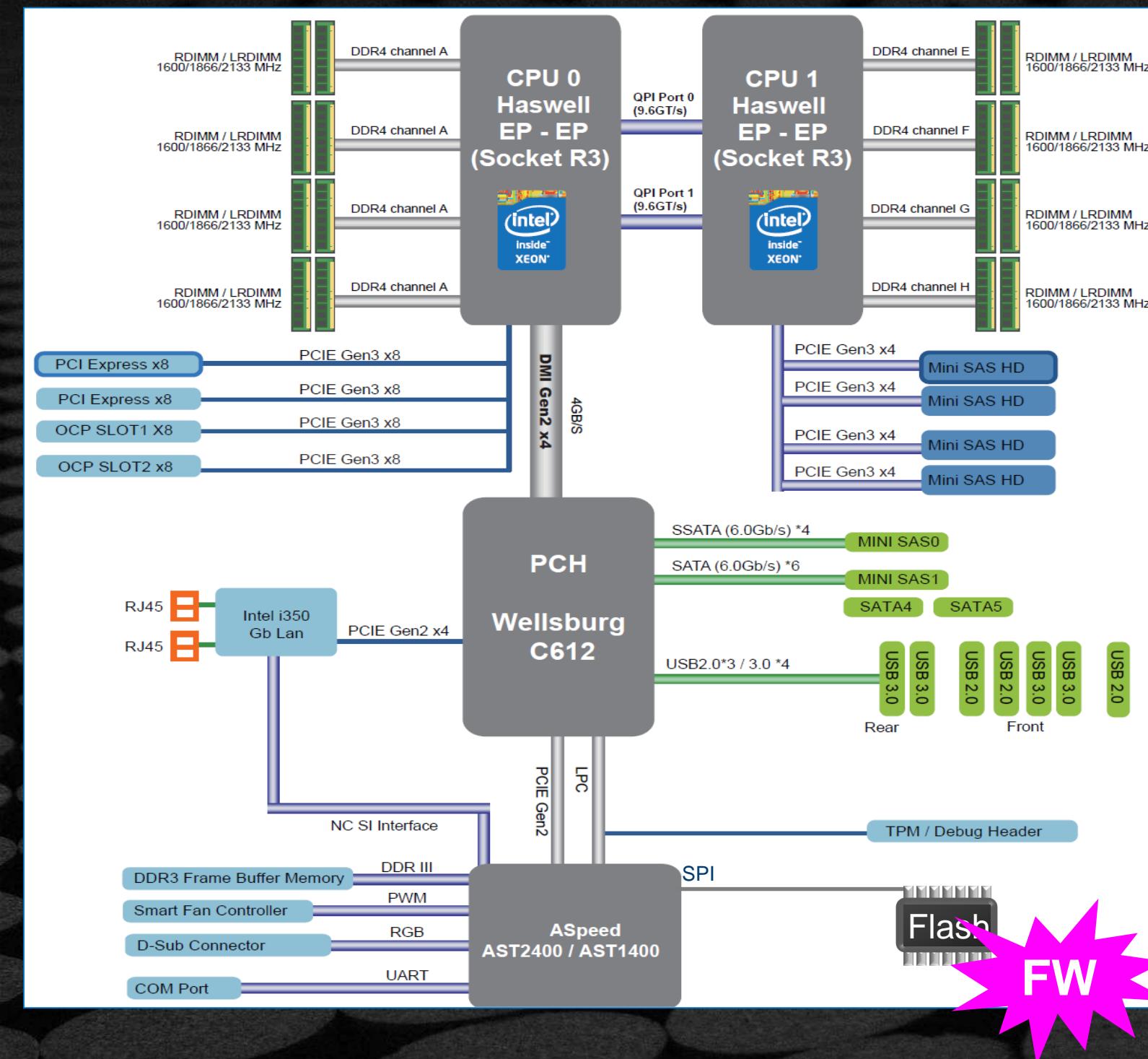
Firmware is Everywhere



- GBe NIC, WiFi, Bluetooth, WiGig
- Baseband (3G, LTE) Modems
- Sensor Hubs
- NFC, GPS Controllers
- HDD/SSD
- Keyboard and Embedded Controllers
- Battery Gauge
- Baseboard Management Controllers (BMC)
- Graphics/Video
- USB Thumb Drives, keyboards/mice
- Chargers, adapters
- TPM, security coprocessors
- Routers, network appliances

Main system firmware (BIOS, UEFI firmware, coreboot)

Firmware is Everywhere

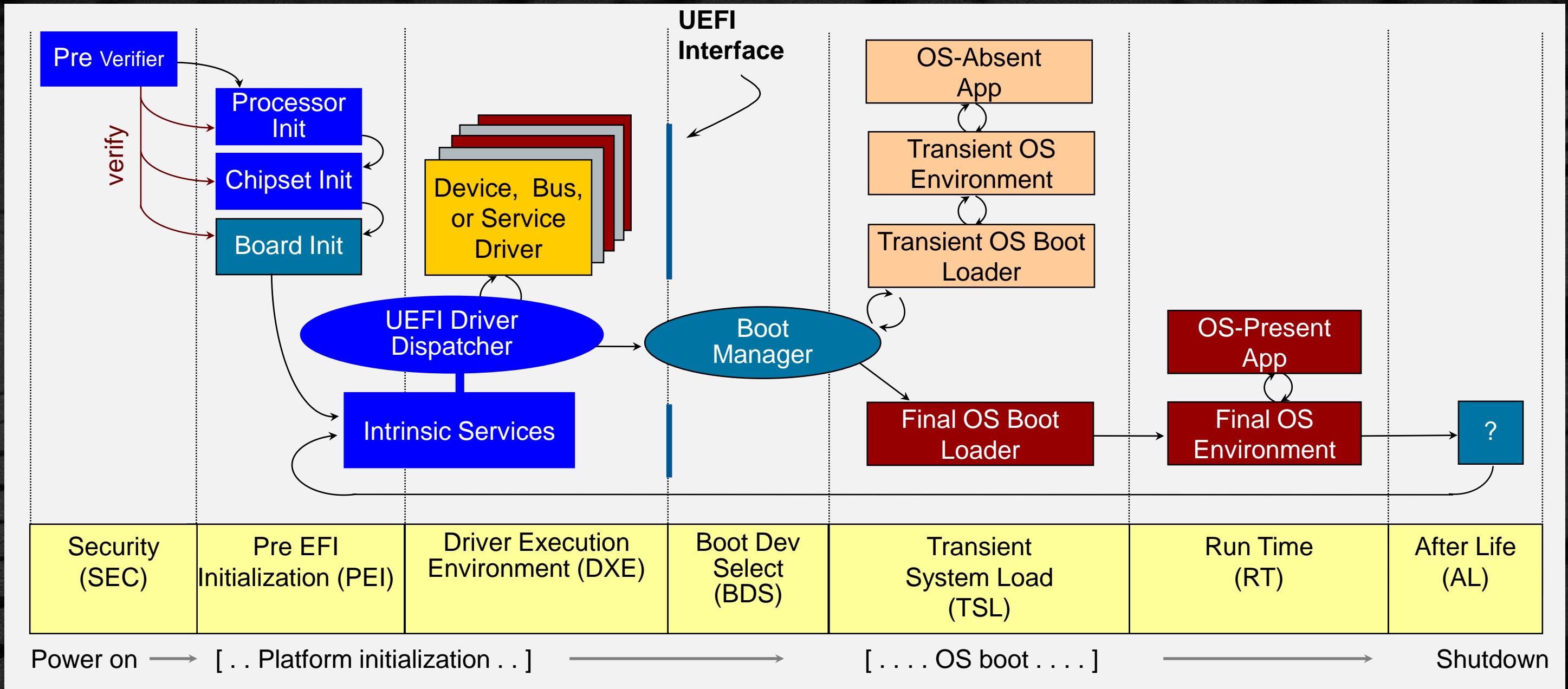


Main system firmware (BIOS,
UEFI firmware, coreboot)

UEFI BOOT EXECUTION FLOW

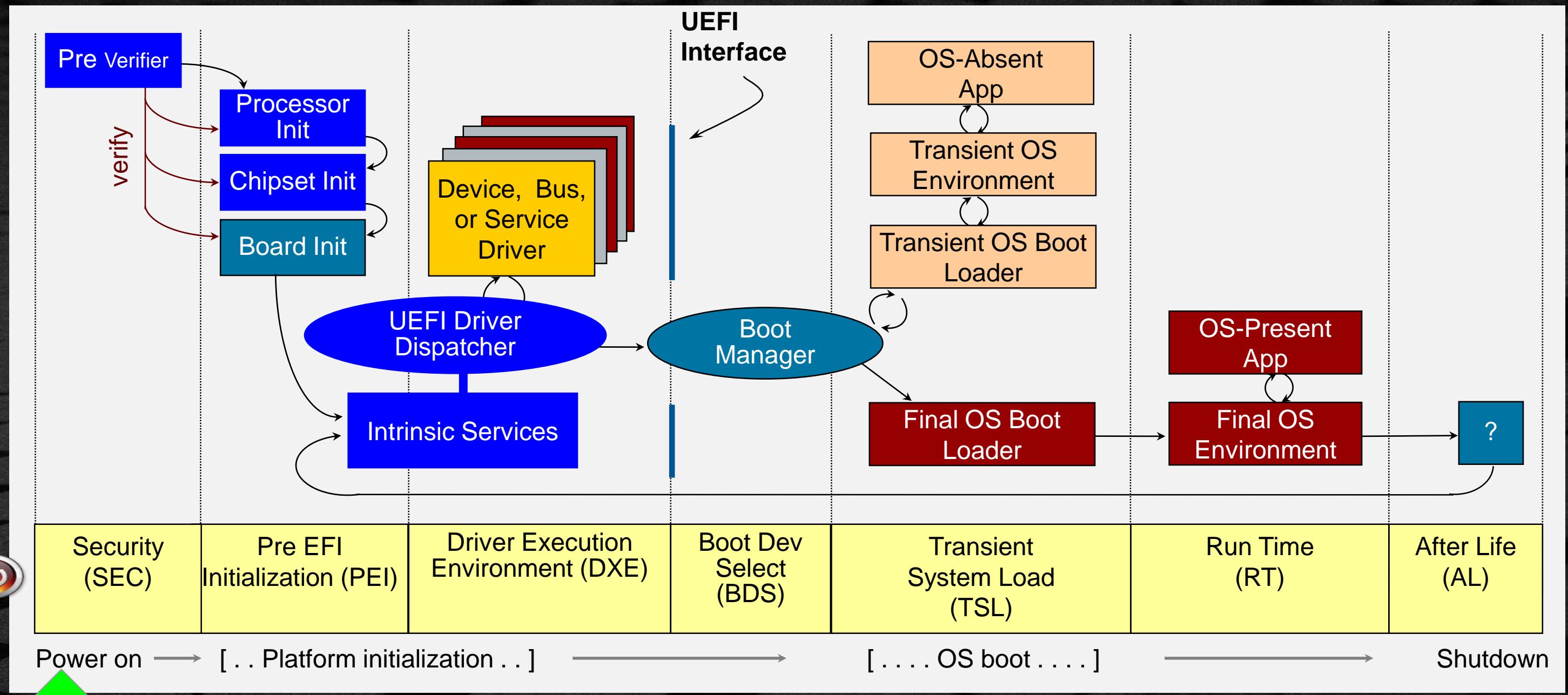
Starting at the processor reset vector

UEFI – PI & EDK II BOOT FLOW



The following Slides show UEFI and Platform Initialization (PI) Boot Execution Flow

UEFI - PI & EDK II BOOT FLOW - SEC



PRE-MEMORY INIT

Address space

0xFFFFFFFF

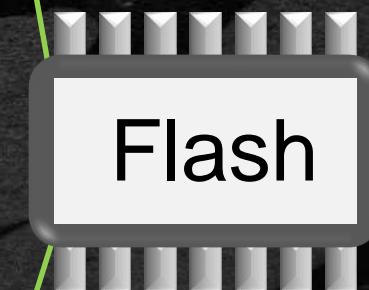
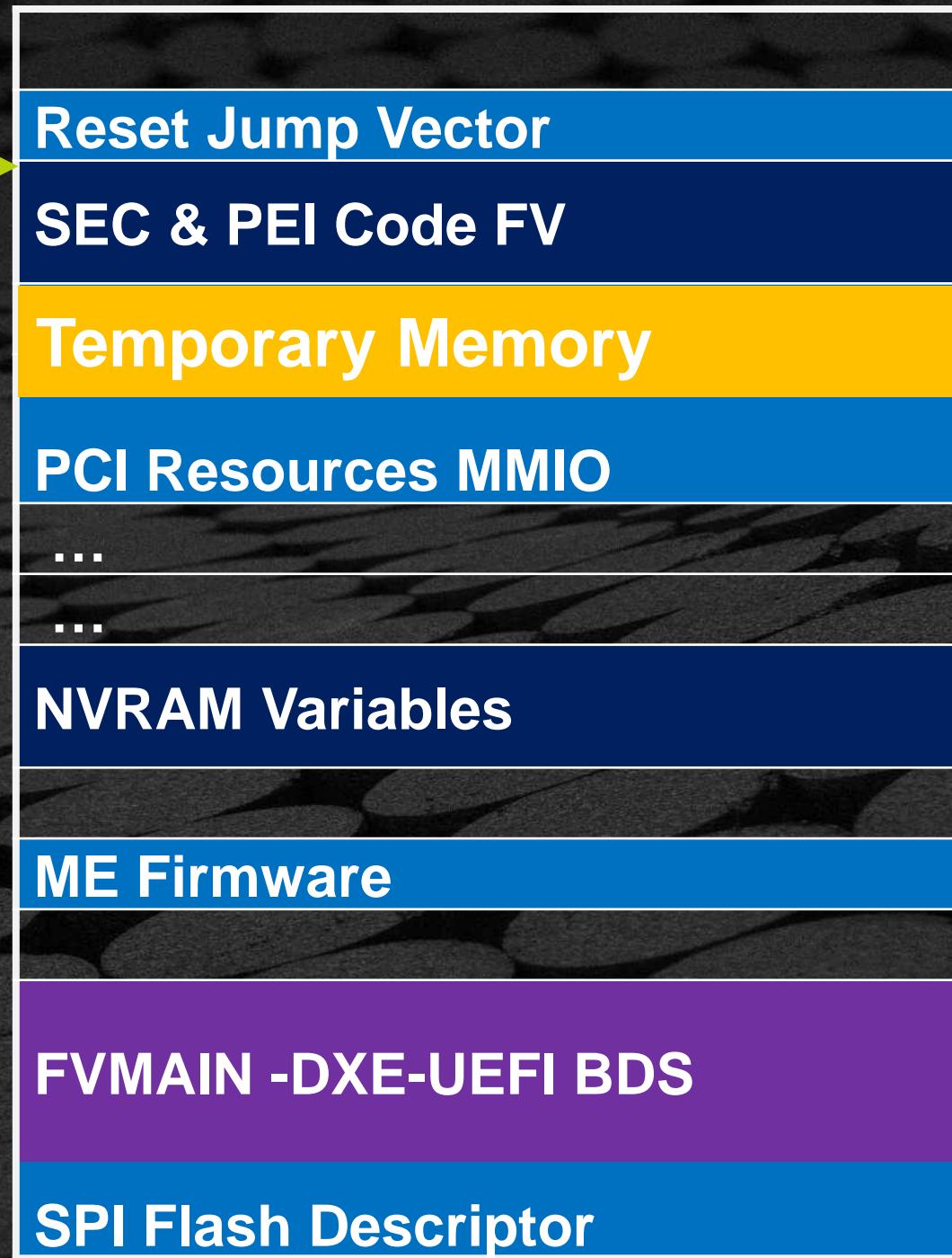
0xFFFFFFF0

0xFFFF8000

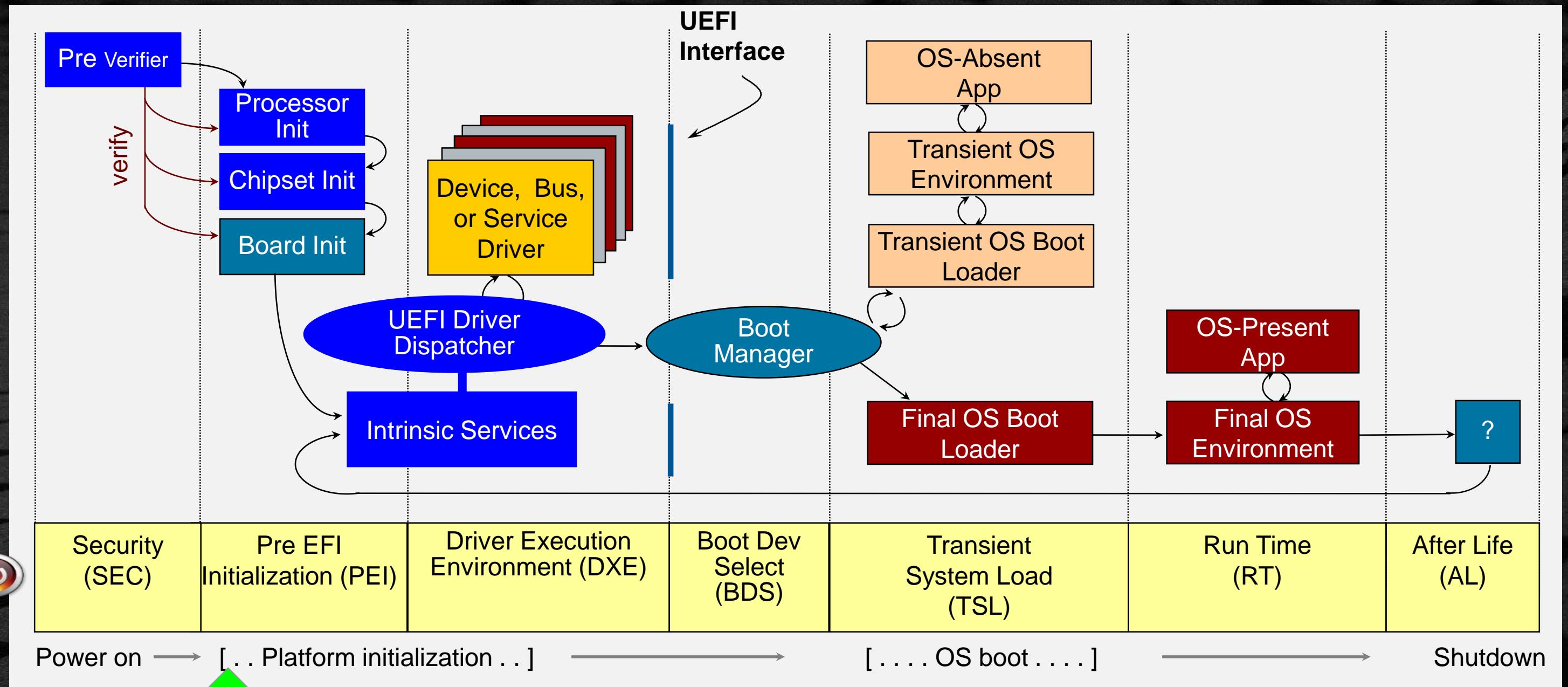
0xFFFF7000

0xFFFF6000

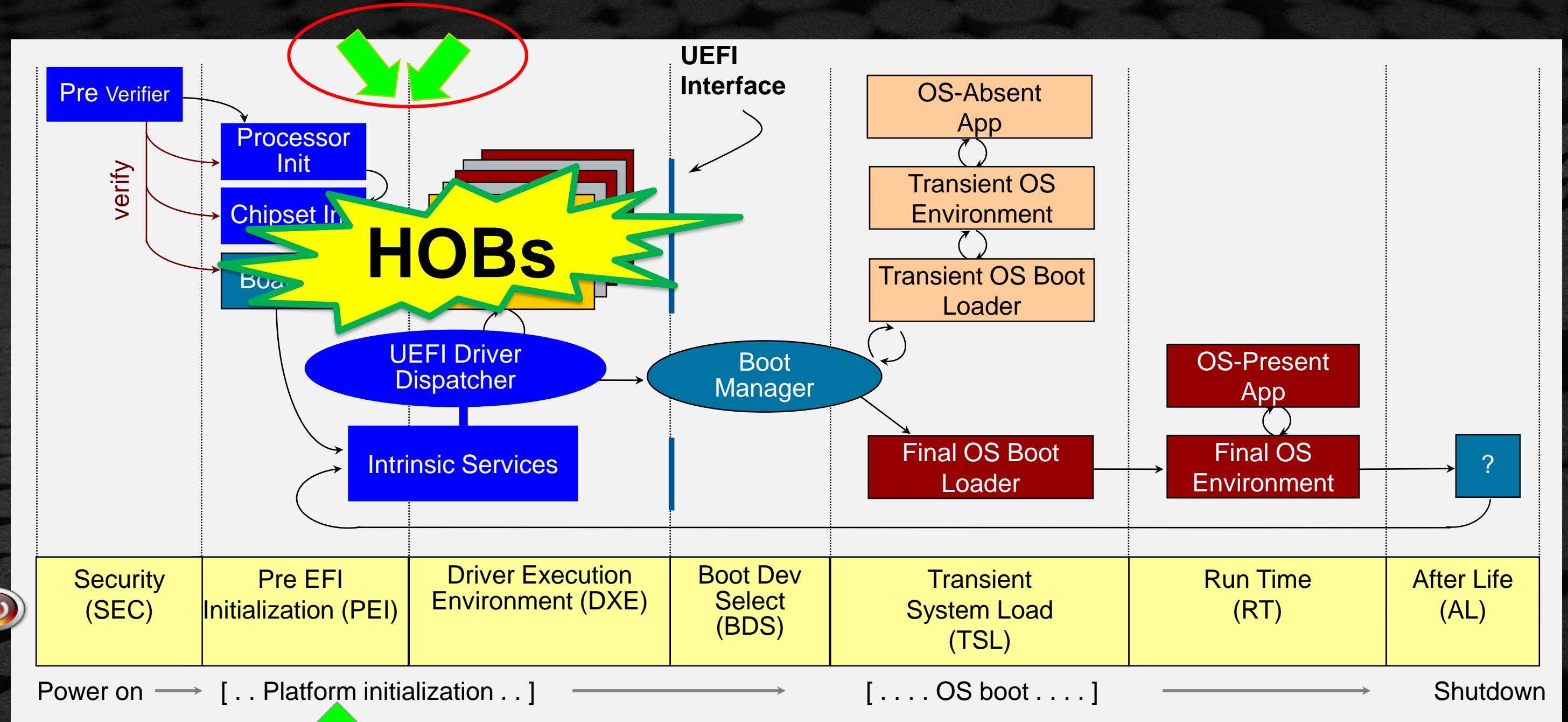
0xFFFF0000



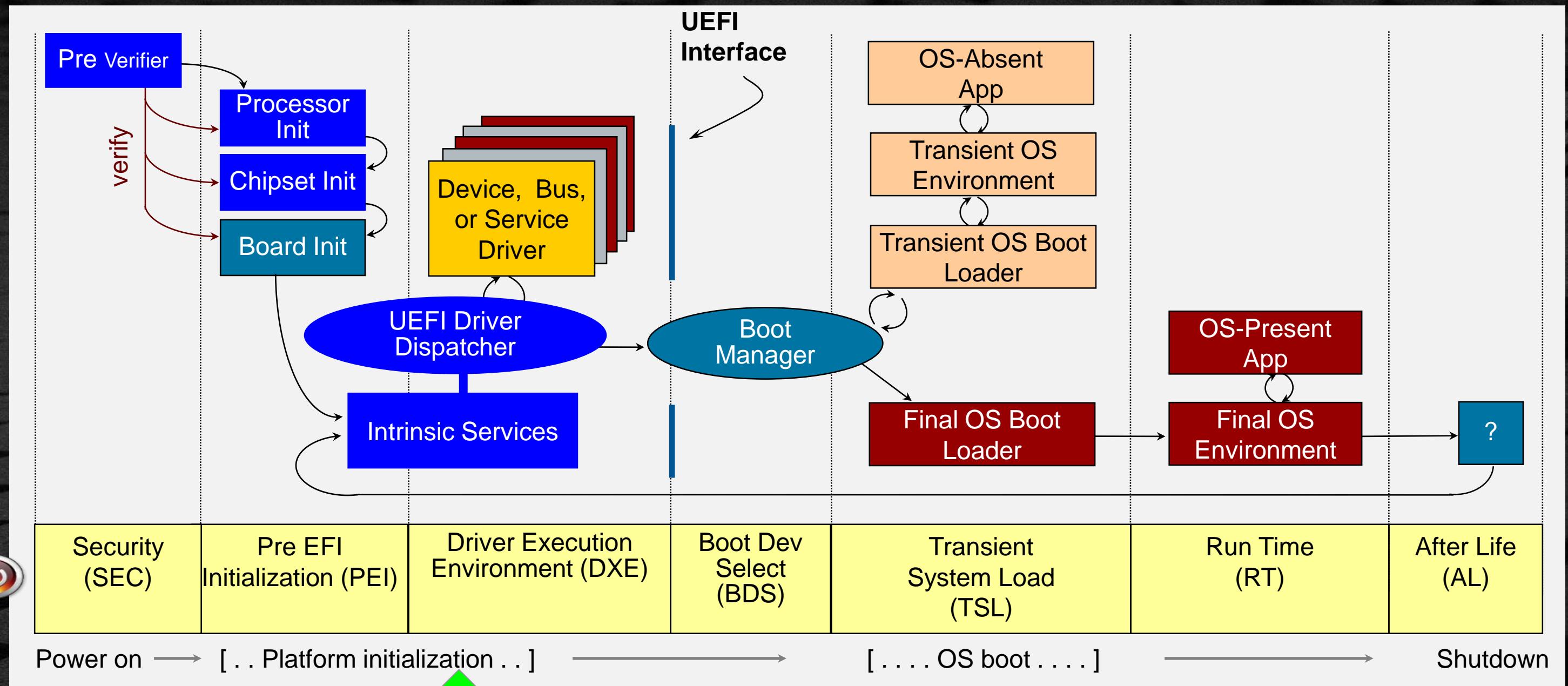
UEFI – PI & EDK II BOOT FLOW - PEI



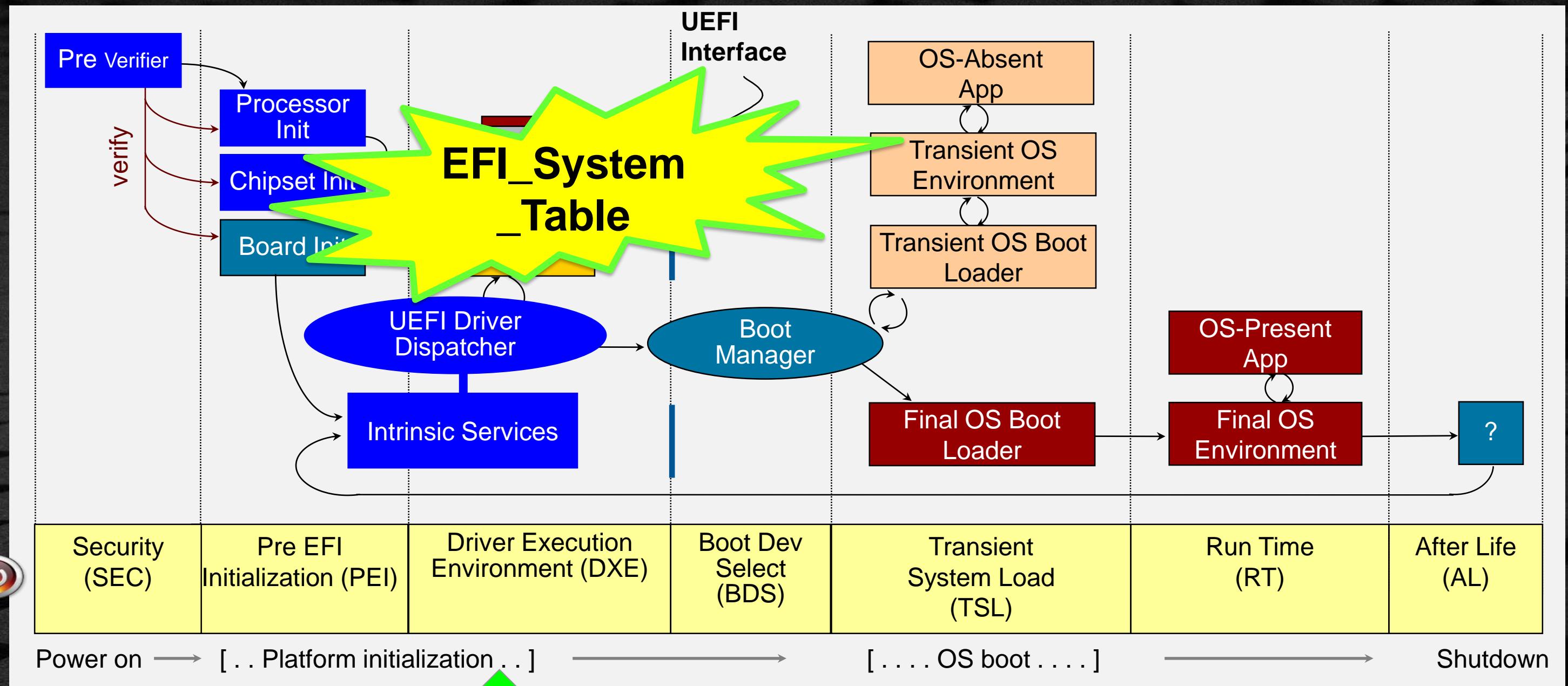
UEFI - PI & EDK II BOOT FLOW - DXE IPL



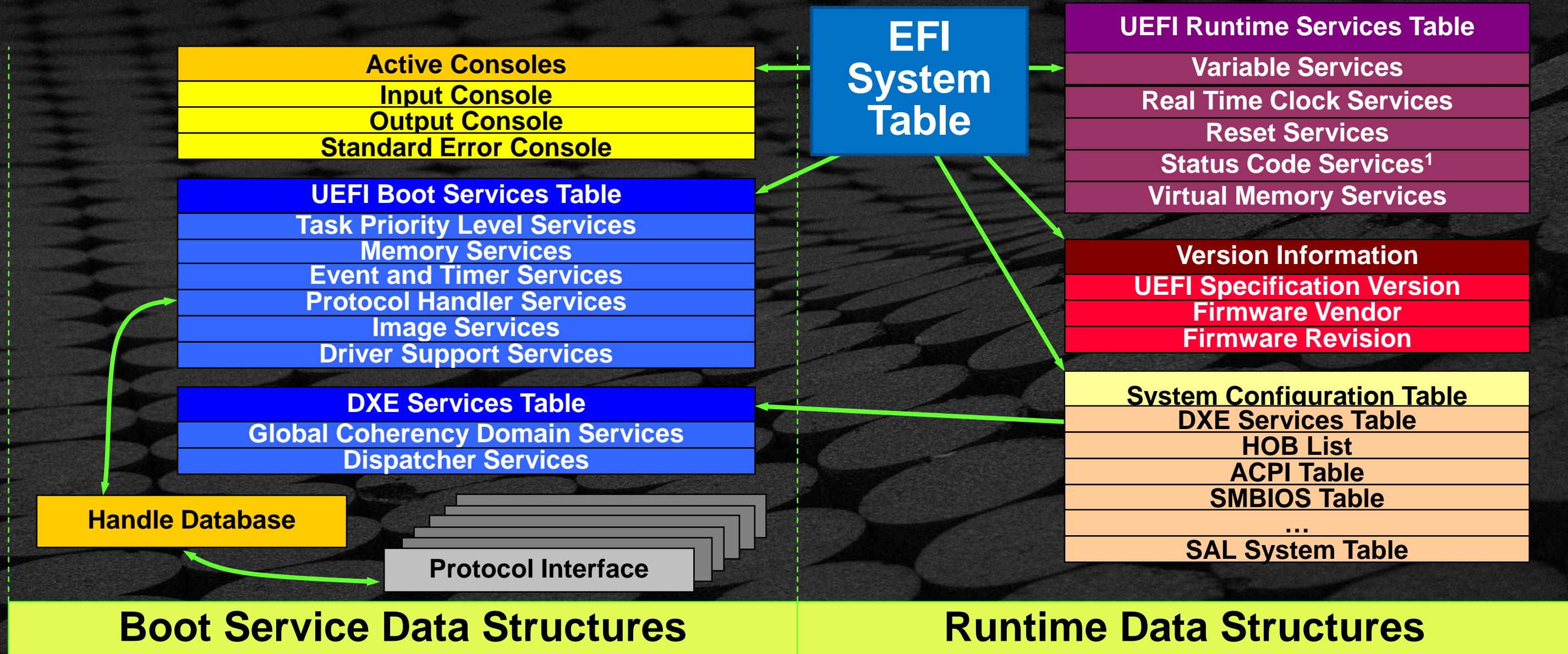
UEFI – PI & EDK II BOOT FLOW – DXE



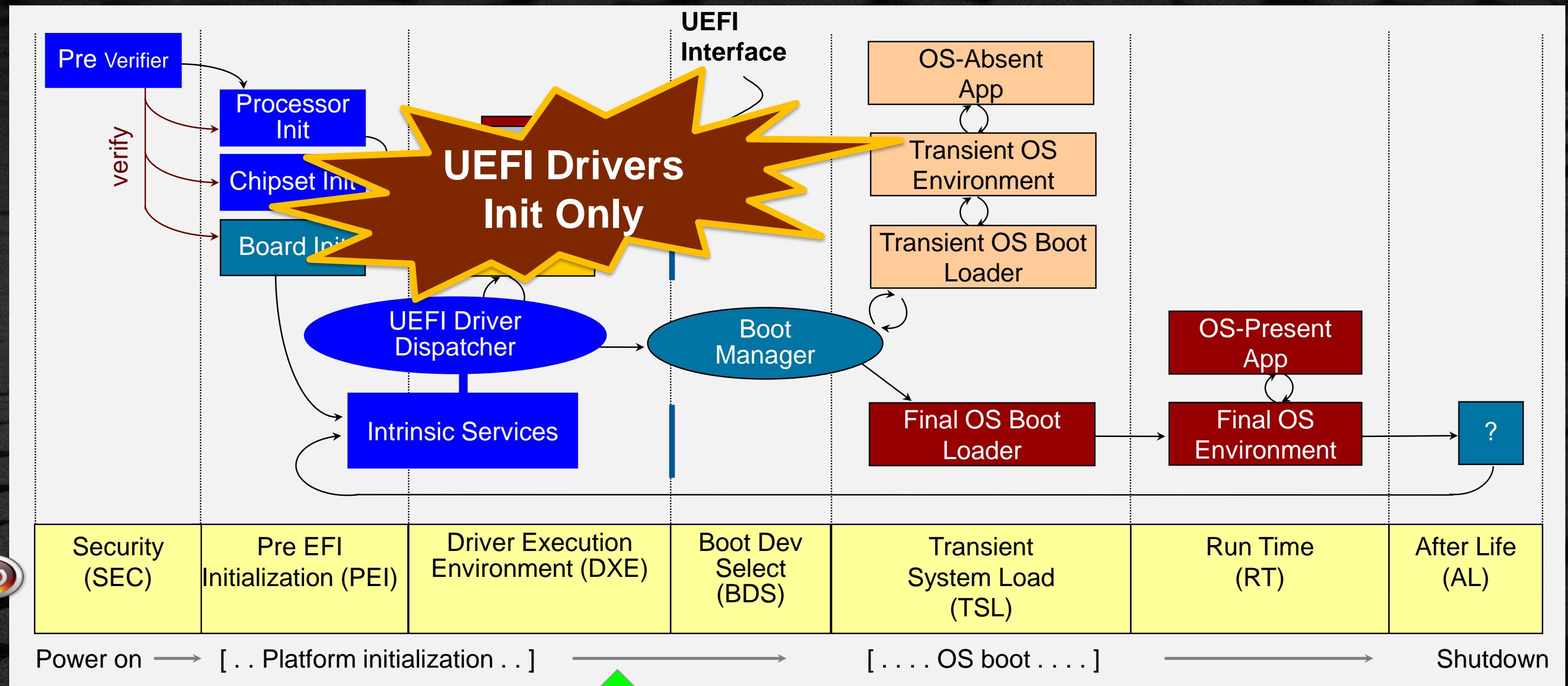
UEFI – PI & EDK II BOOT FLOW – DXE



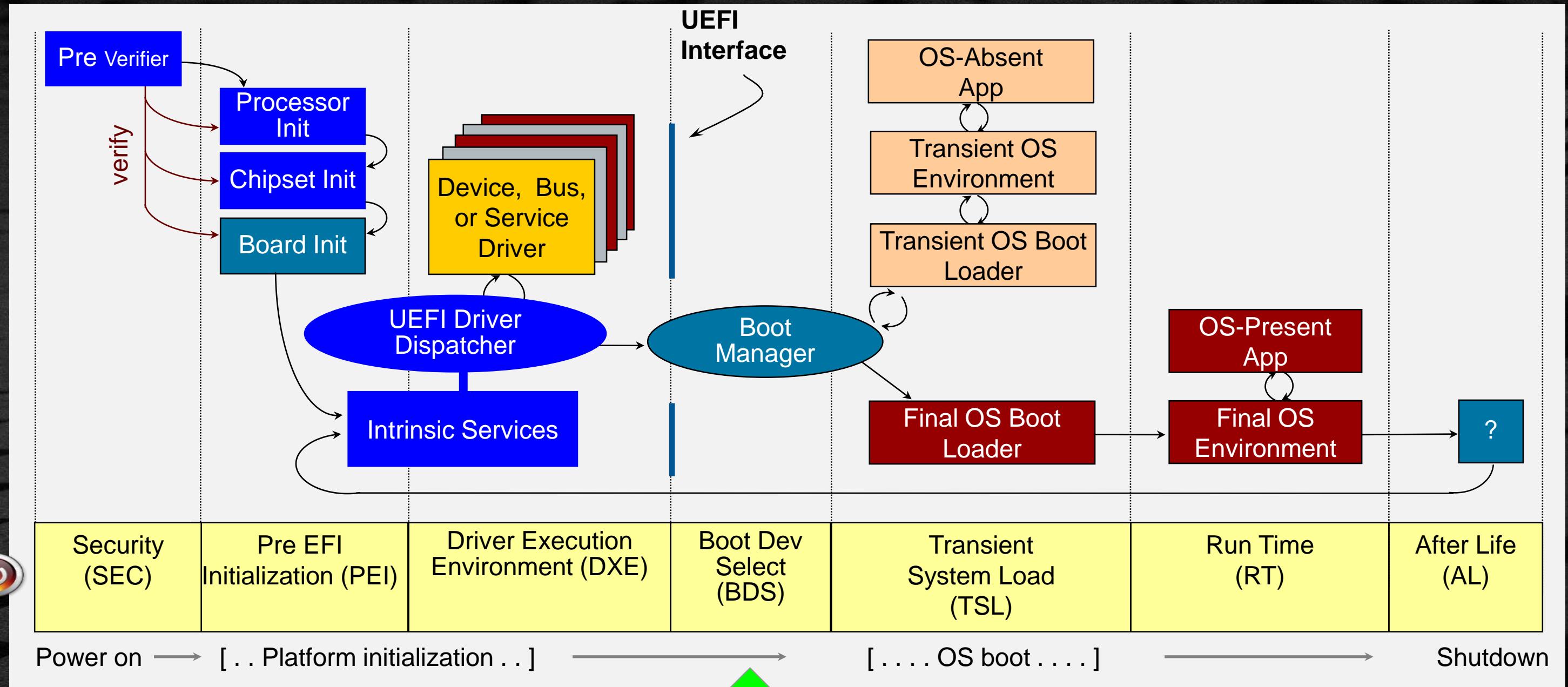
UEFI SYSTEM TABLE



UEFI – PI & EDK II BOOT FLOW – DXE UEFI



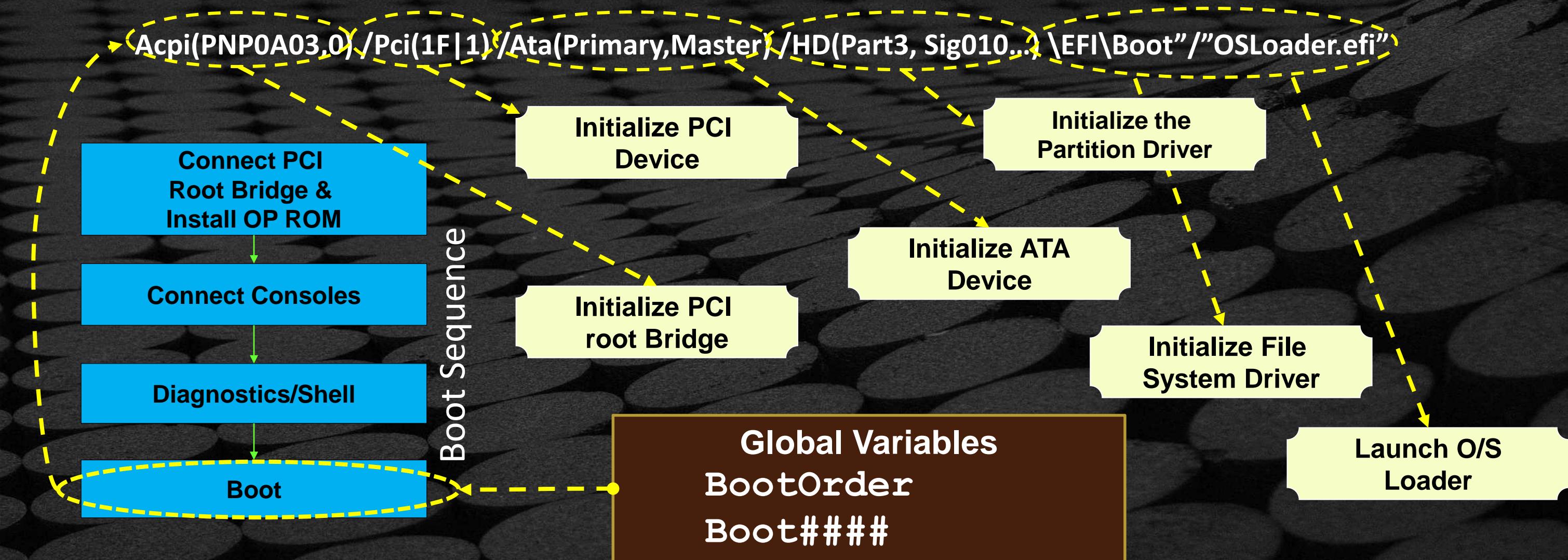
UEFI – PI & EDK II BOOT FLOW – BDS



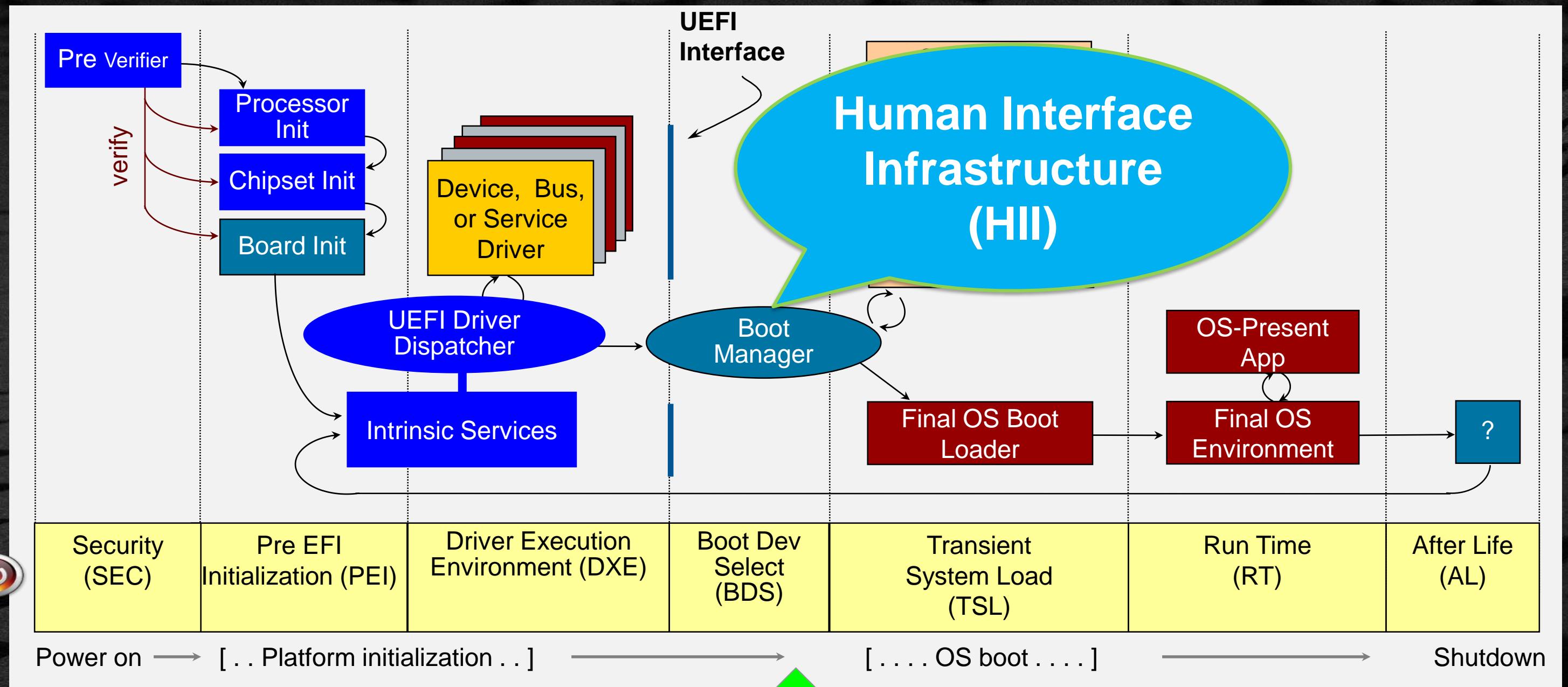
UEFI DEVICE PATH AND GLOBAL VARIABLES

The UEFI Device Path describes a boot target

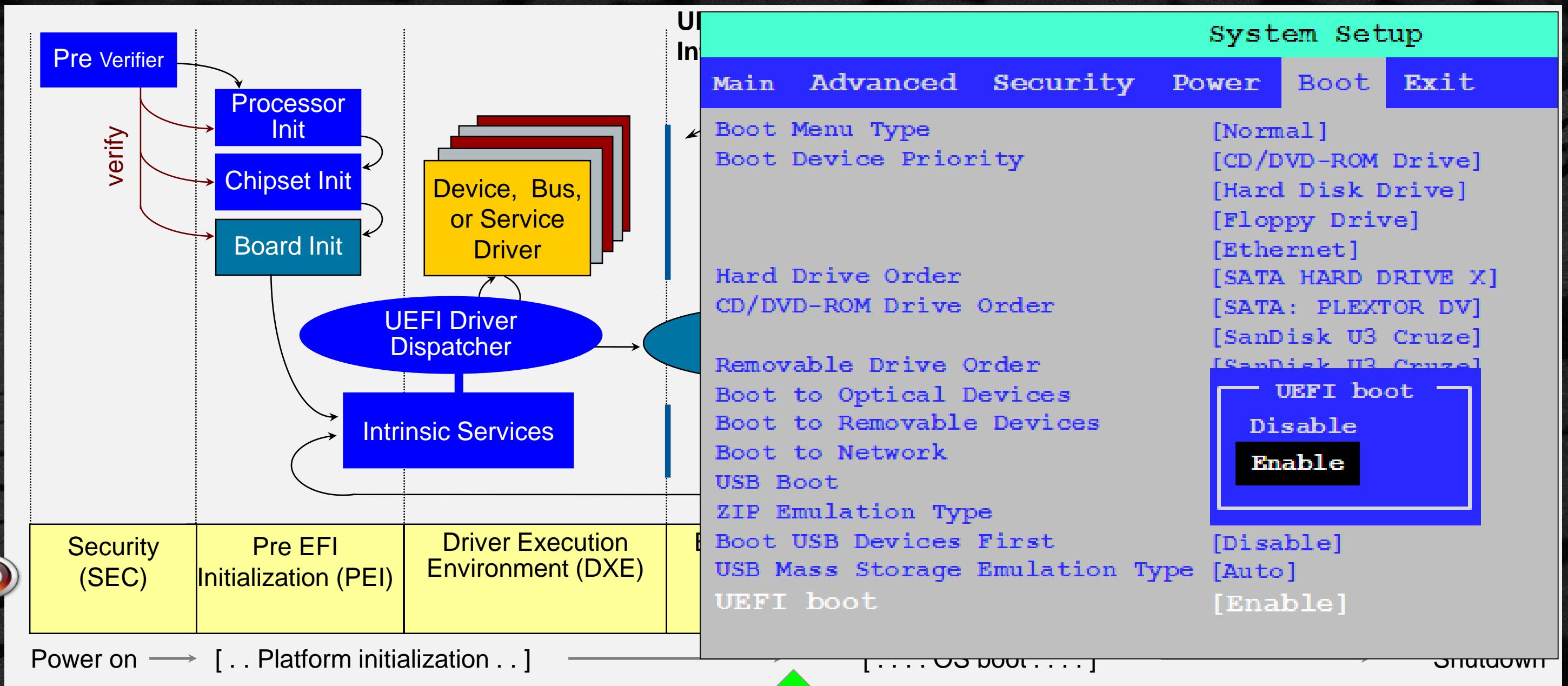
- Binary description of the physical location of a specific target



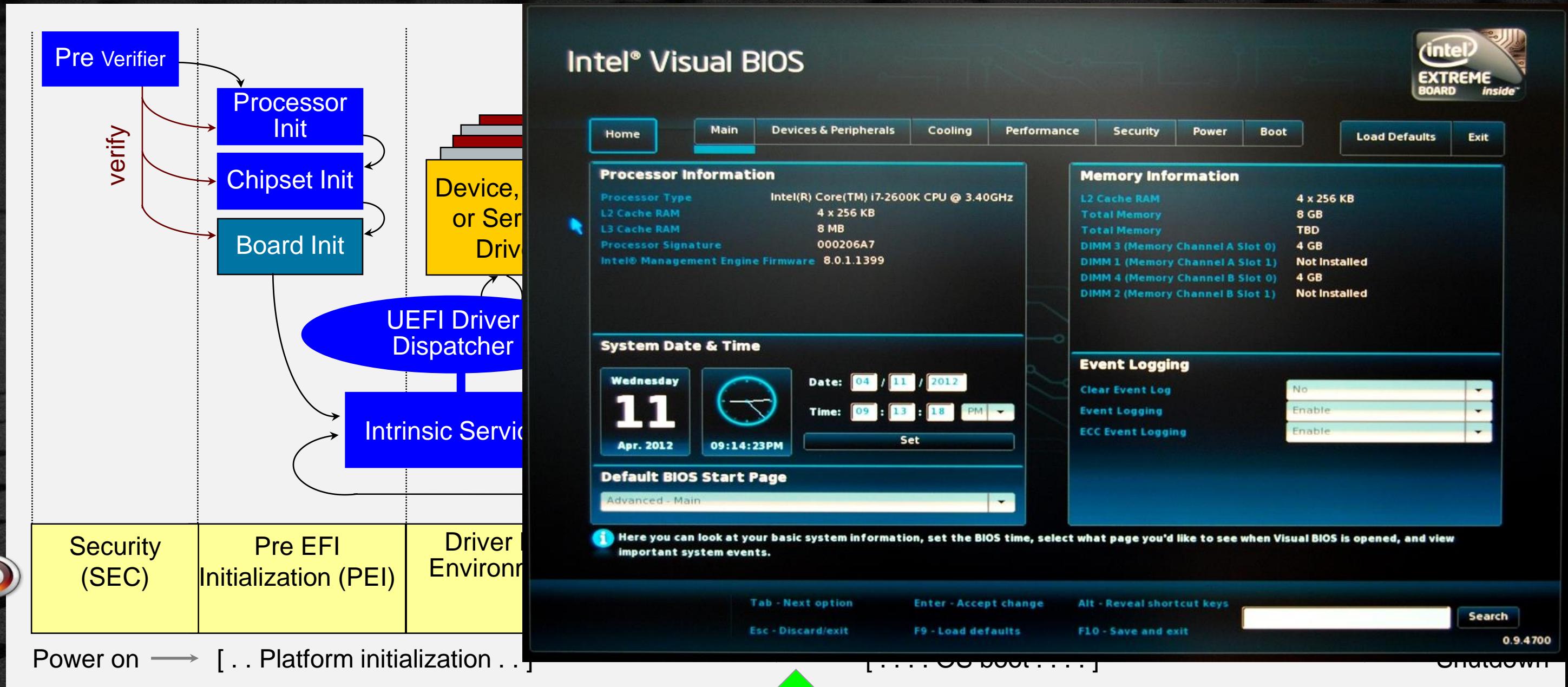
UEFI – PI & EDK II BOOT FLOW – HII



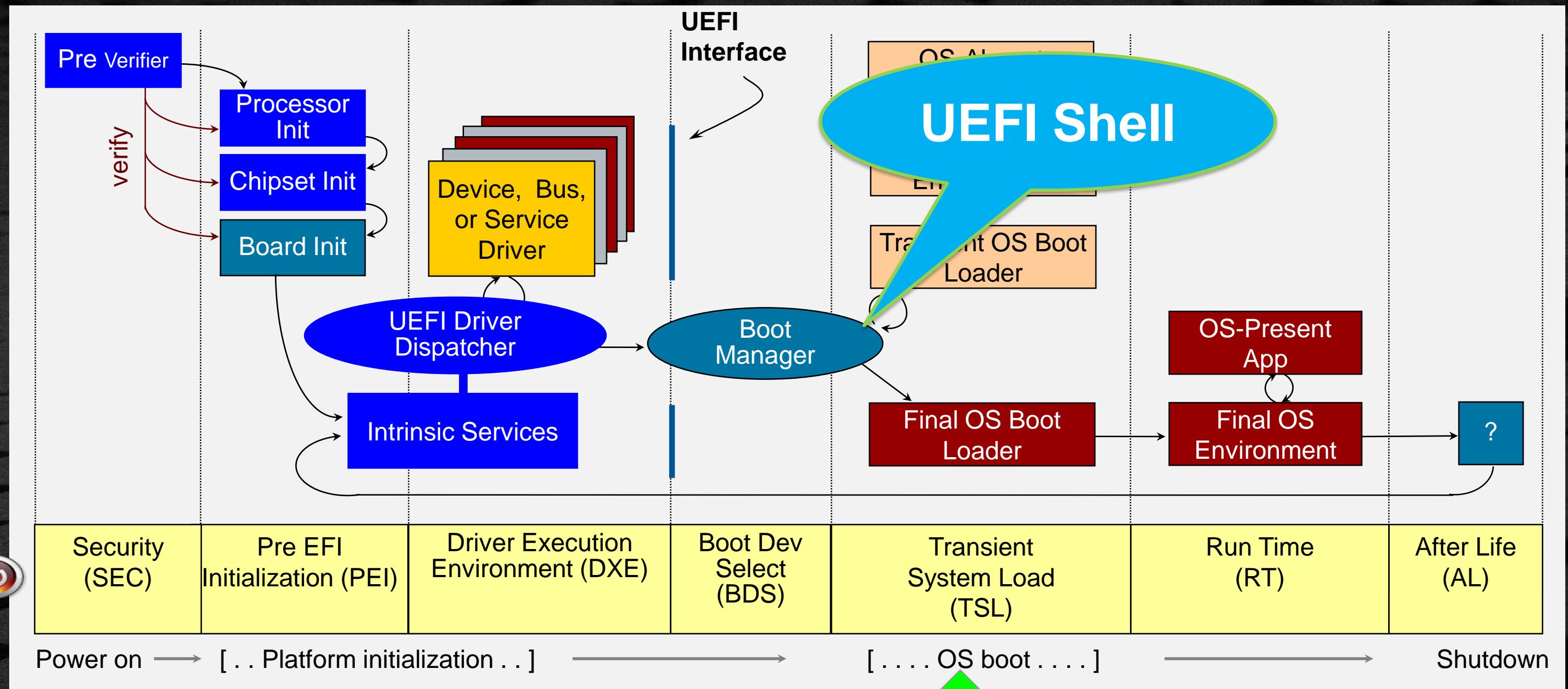
UEFI – PI & EDK II BOOT FLOW – HII



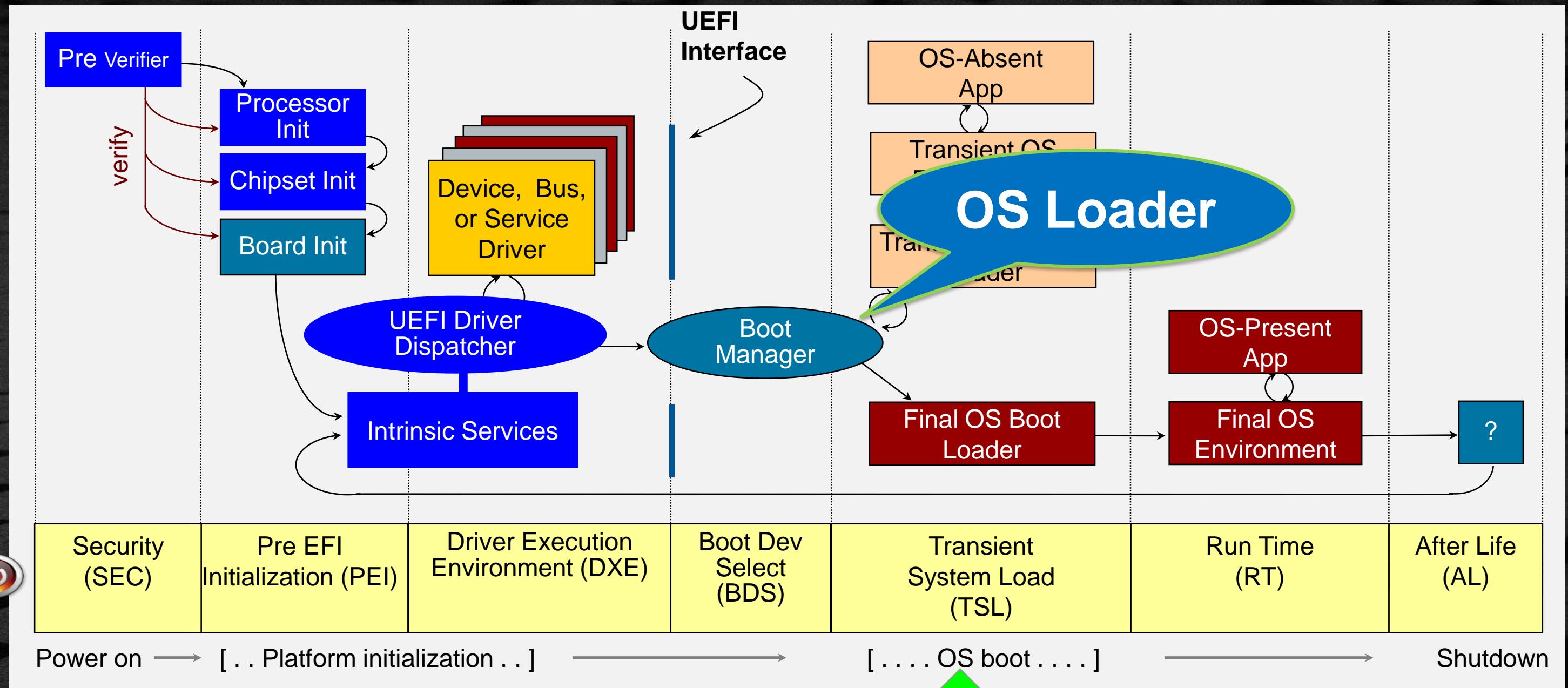
UEFI – PI & EDK II BOOT FLOW – HII



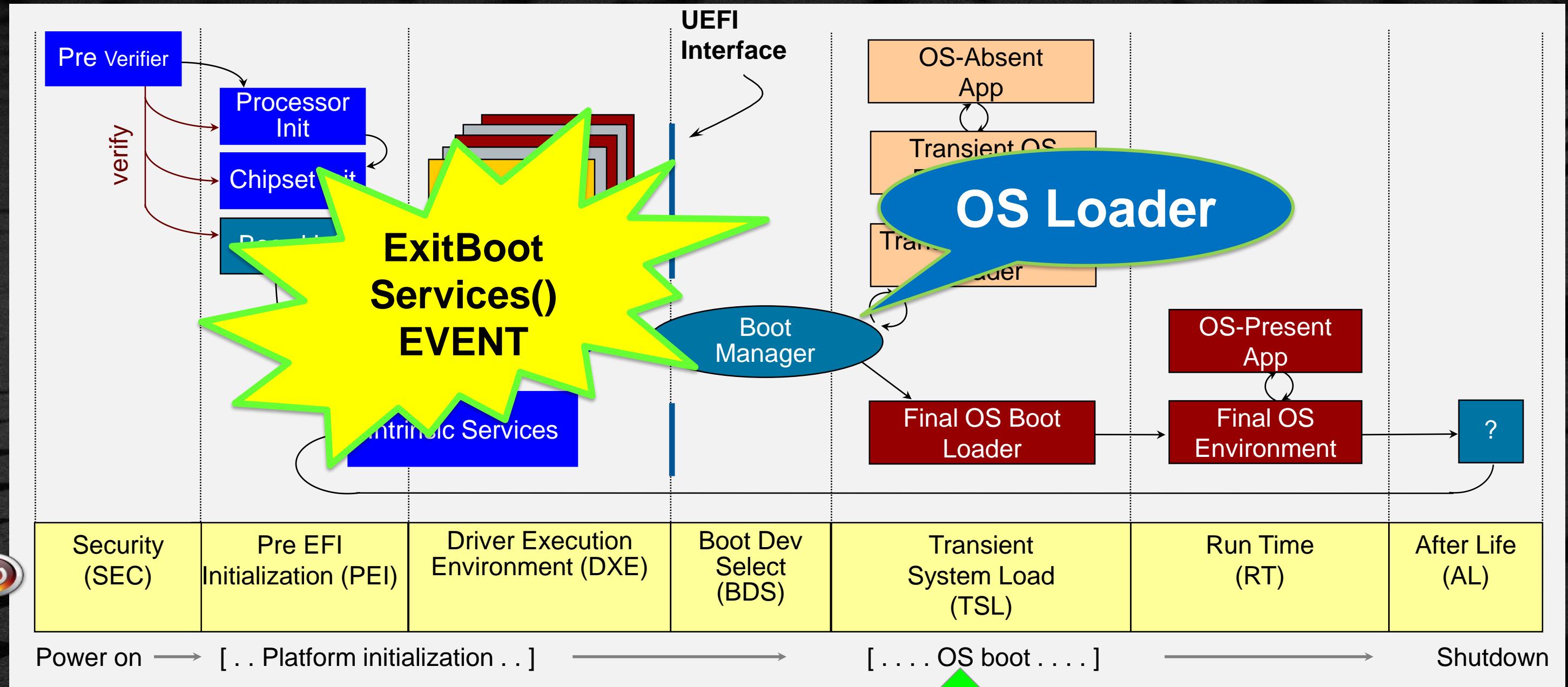
UEFI – PI & EDK II BOOT FLOW – TSL



UEFI – PI & EDK II BOOT FLOW – BOOT LOADER

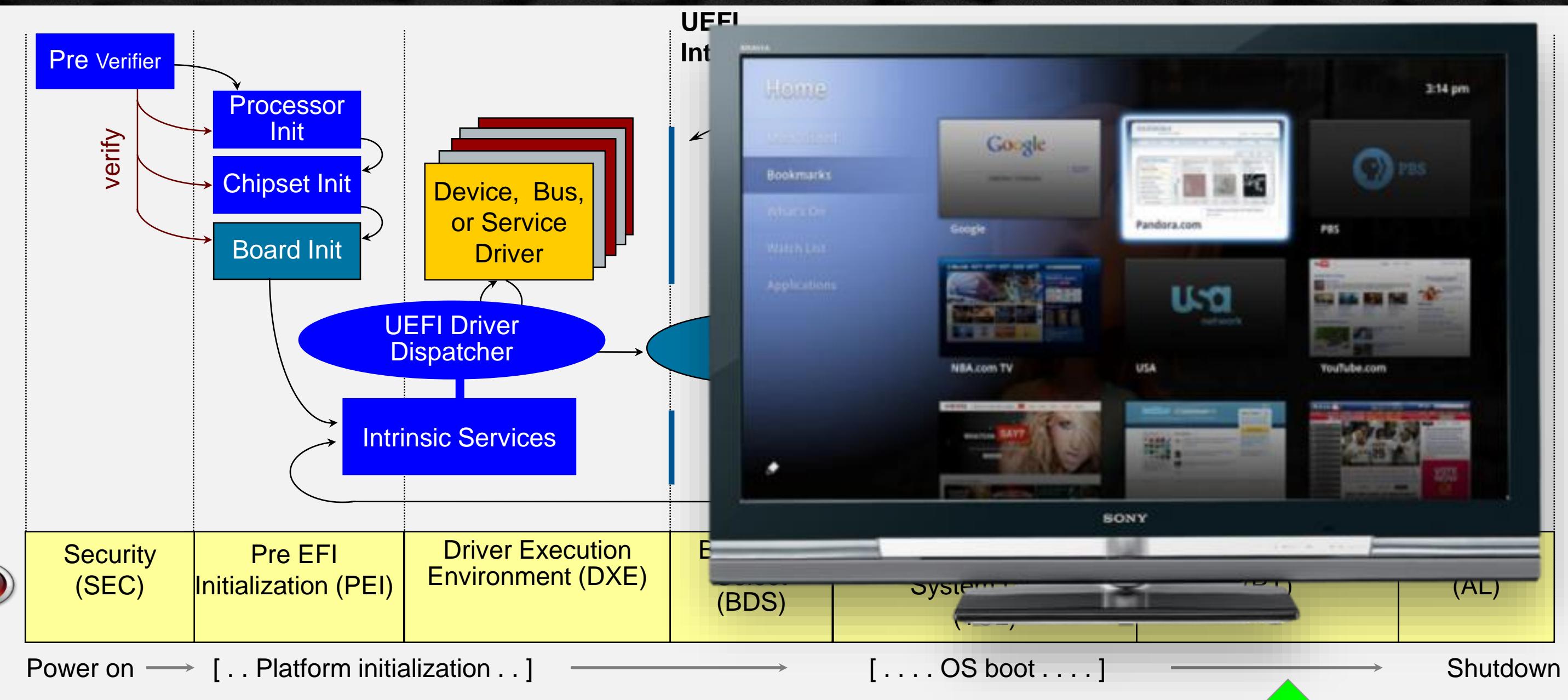


UEFI - PI & EDK II BOOT FLOW - EVENT

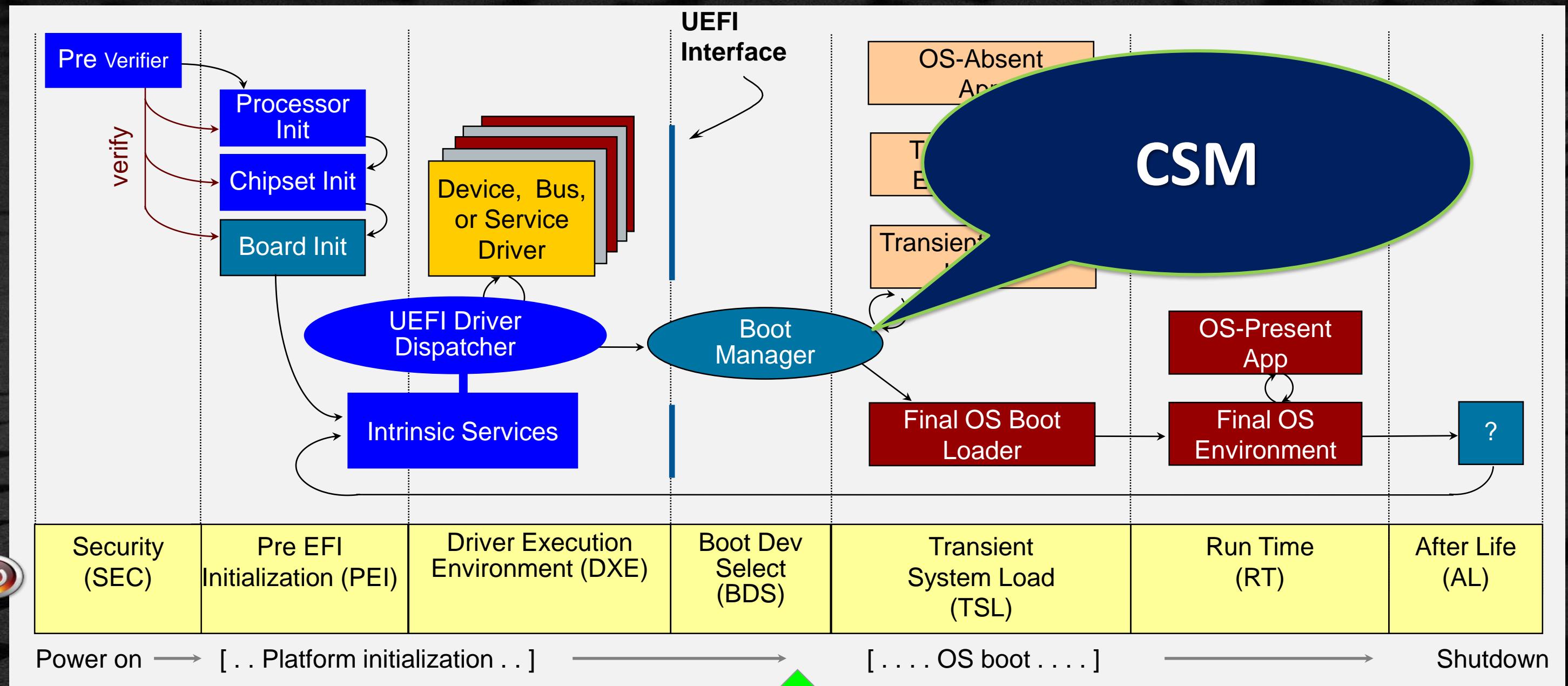


* tianocore

UEFI - PI & EDK II BOOT FLOW - BOOT UEFI OS



UEFI - PI & EDK II BOOT FLOW - BOOT LEGACY



UEFI - PI & EDK II BOOT FLOW - BOOT LEGACY



THE INTEL® FIRMWARE SUPPORT PACKAGE (INTEL® FSP)

INTEL® FSP - COMPONENTS

- CPU, memory controller, and chipset initialization functions as a binary package
- Provides silicon initialization ingredients
- Plugs into existing firmware frameworks
- Integration guide, includes API documentation

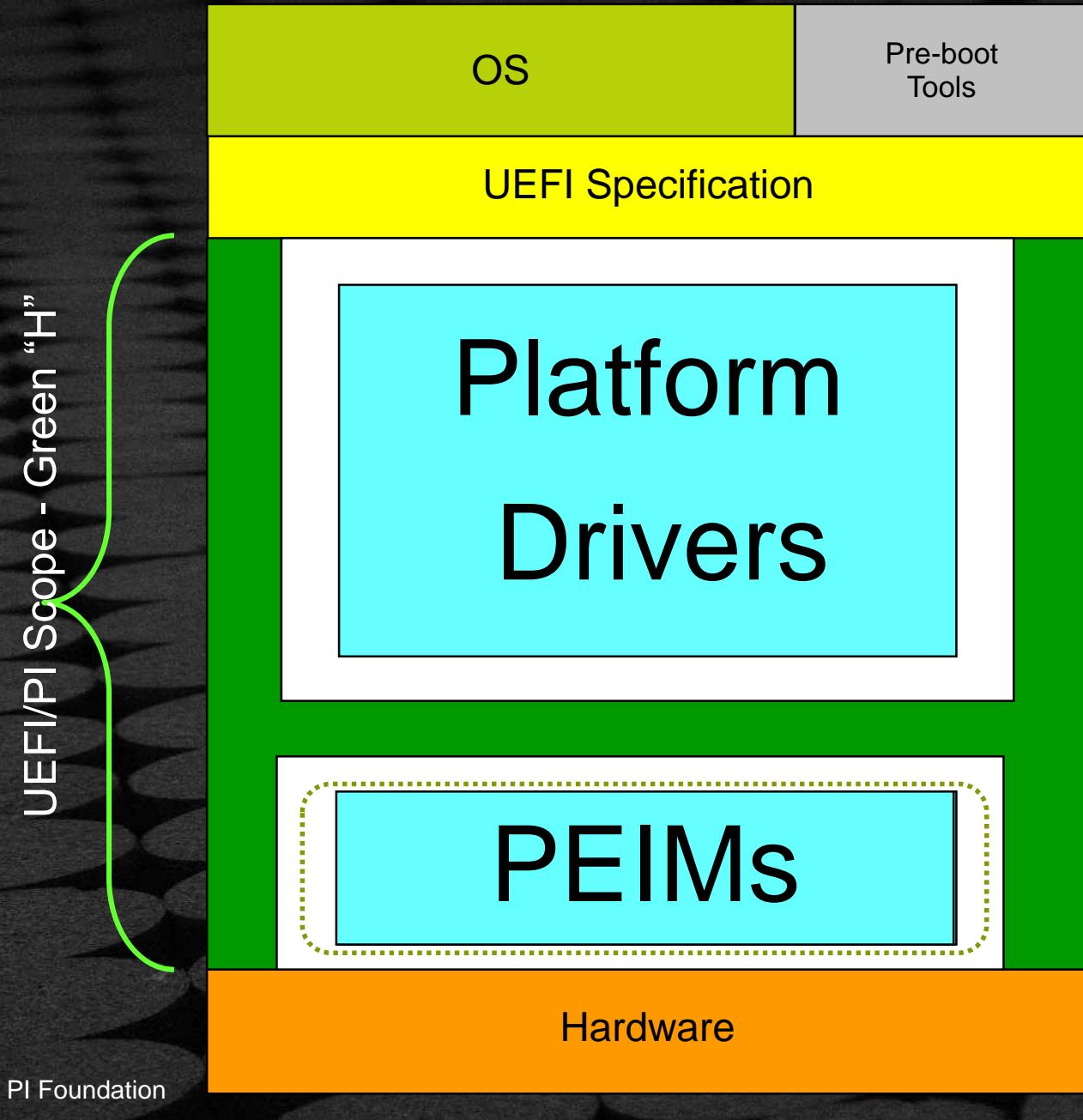
Intel FSP is currently available for the many Intel hardware-producing divisions

See: [About Intel FSP](#)

White Paper Example: [Open Braswell - Design and Porting Guide](#)

Intel® FSP is NOT a stand-alone boot-loader

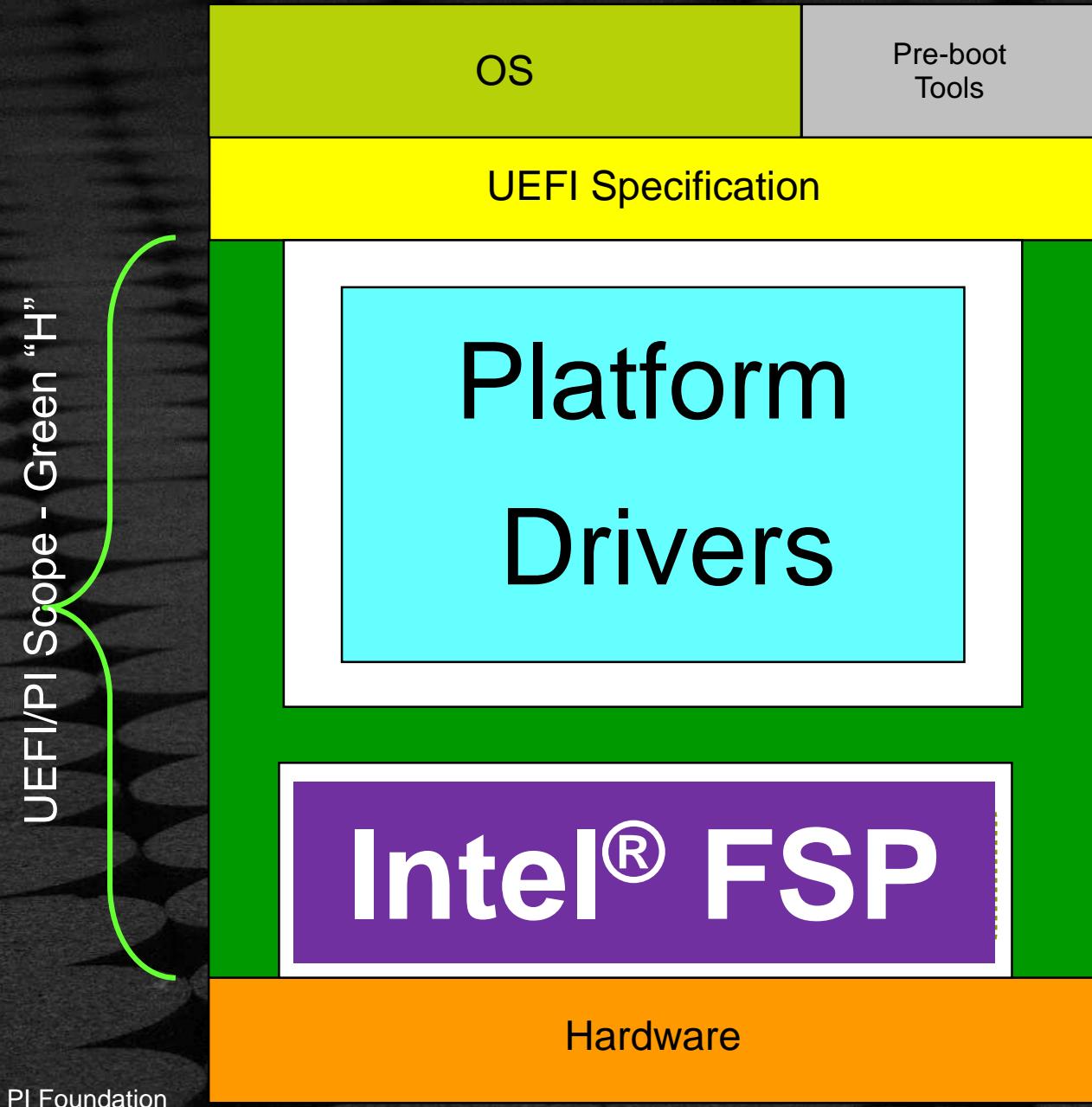
INTEL® FSP TO OPEN SOURCE EDK II



EDK II provides the framework (“Green H”)

Intel® Firmware Support Package (Intel® FSP) provides low level of silicon initialization

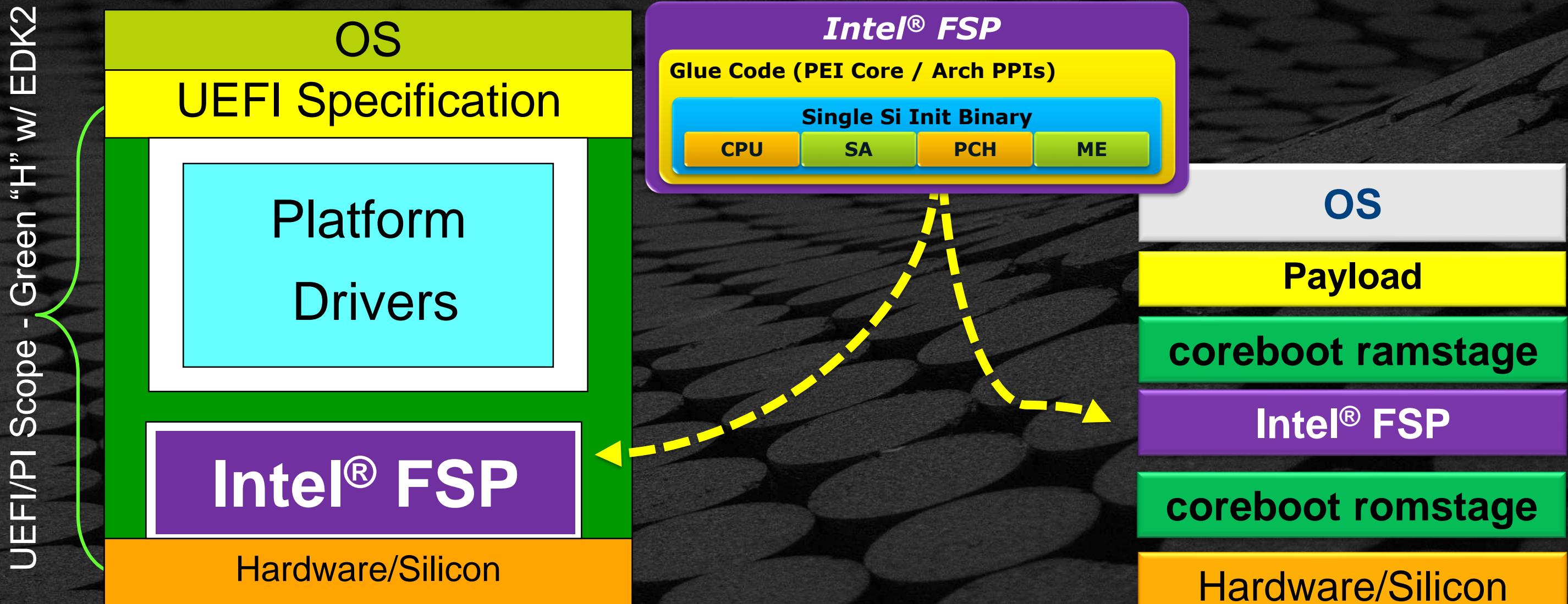
INTEL® FSP TO OPEN SOURCE EDK II



EDK II provides the framework (“Green H”)

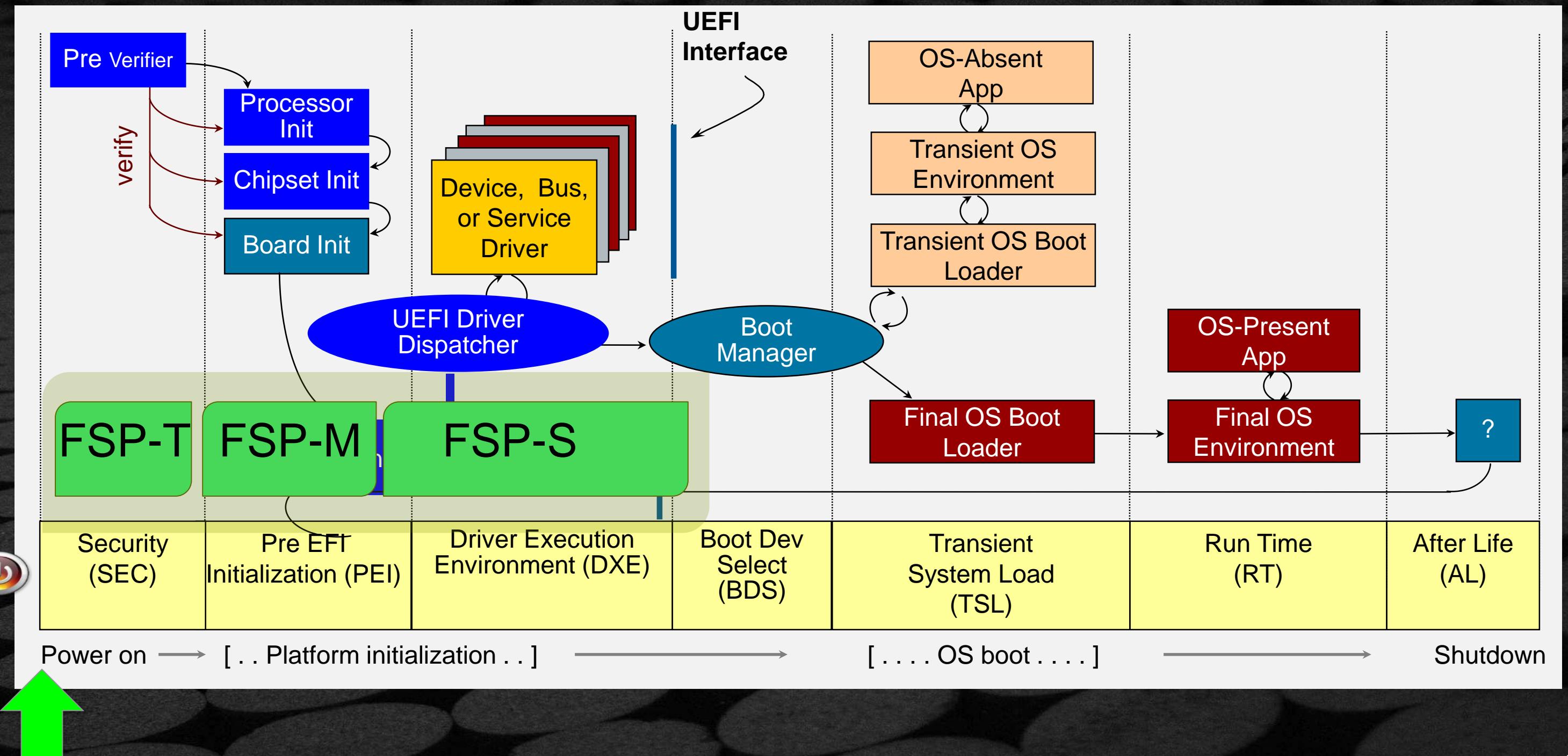
Intel® Firmware Support Package (Intel® FSP) provides low level of silicon initialization

Intel® FSP "Produced" to "Consuming" Intel® Architecture Firmware

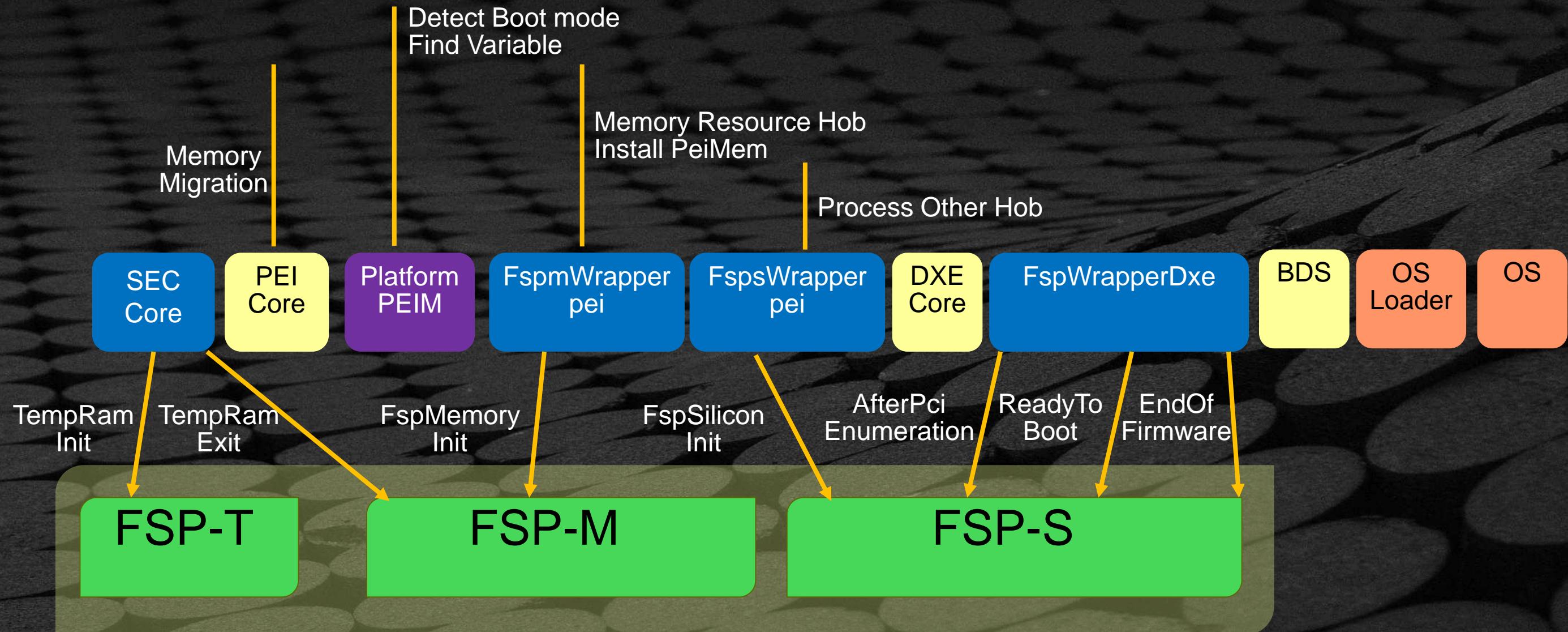


Intel FSP is independent of the bootloader solutions

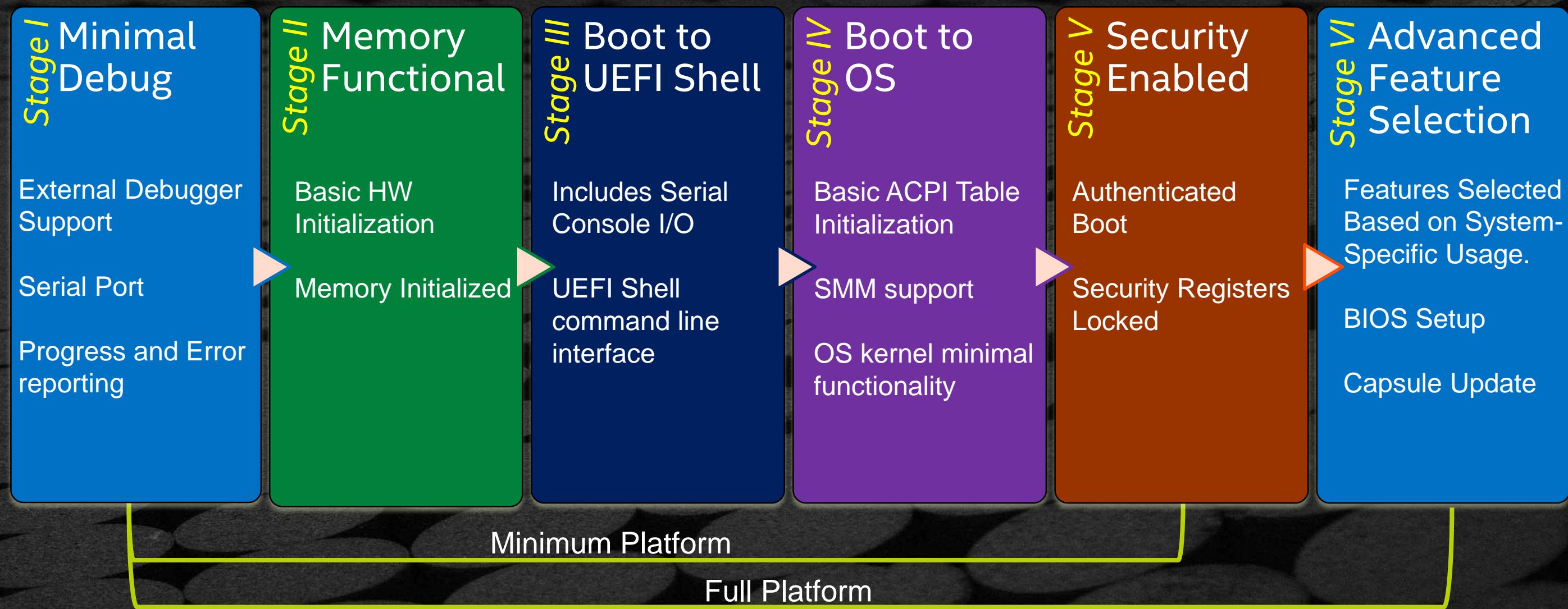
UEFI - PI & EDK II BOOT FLOW – FSP



Boot Flow with UEFI & Intel® FSP



MinPlatform + Intel® FSP Boot Flow - Staged Approach



INTEL® FSP PRODUCER

- Examples of binary instances on <http://www.intel.com/fsp> with integration guides
 - This includes hardware initialization code that is EDK II based PEI Modules (PEIM's)
 - Modules are encapsulated as a UEFI PI firmware volume w/ extra header
 - Configure w/Vital Product Data (VPD)-style Platform Configuration Data (PCD) externalized from the modules
 - Resultant output state reported via UEFI Platform Initialization (PI) Hand Off Block (HOB)

[Intel® Firmware Support Package \(Intel® FSP\) External Architecture Specification \(EAS\) v2.1](#)

Resource: <https://firmware.intel.com/blog/open-source-platforms-edkii-using-intel-fsp>

SOURCE FOR INTEL® FSP PRODUCER CODE

- CPU and chipset-specific code for PEIM's inside of the Intel FSP can be open or closed, code at [Intel FSP-repo](#)
- PEI core and infrastructure code at [tianocore.org/edk2](#)
 - [/MdePkg](#)
 - [/MdeModulePkg](#)
- Code to interface Intel FSP to EDK II can be found at :
 - [/IntelFsp2Pkg](#) and Wrapper at: [/IntelFsp2WrapperPkg](#)

Intel FSP can encapsulate IP protected initialization code PRODUCED by Intel business units

WHAT'S NEW IN THE UEFI SPECIFICATIONS?

LATEST UEFI SPECIFICATIONS



Unified Extensible Firmware Interface Forum

[Http://uefi.org](http://uefi.org)



<http://www.uefi.org/specsandtesttools>

What's New in the UEFI Specifications



Resources Presented at Events from [UEFI Forum Education Link](#)

UDK2018: Key Features - Q2 2018

UEFI Development Kit (UDK) releases are stable, validated snapshots of EDK II

- Industry Standards & Public Specifications
 - UEFI 2.7
 - UEFI PI 1.6
 - ACPI 6.2
- Centralized Config Management
- IOMMU-based DMA Protection
- Stack Guard, Heap Guard and NULL Pointer Detection
- Compilers / Tools
 - Microsoft Visual Studio 2017 tool chain
 - Hash-based incremental build
 - Build time improvement using multi-threading in GenFds to generate FFS files
- More Info: [TianoCore Wiki UDK2018](#)

Include Helper .chm files for different Packages: [UDK2018 Documentation](#)

EDK II - Open Source

Community Development

- Stable Tag Releases- cycle of releasing stable versions of EDK II Firmware
- Adding UEFI Spec updates and new key features and bug fixes
- Three phases of development
 - Development phase
 - Soft Feature Freeze
 - Hard Feature Freeze

More Information on Stable Tag Releases:

[TianoCore Wiki](#)



Tag: edk2-stable201908 Features:
[edk2 releases Stable tag](#)

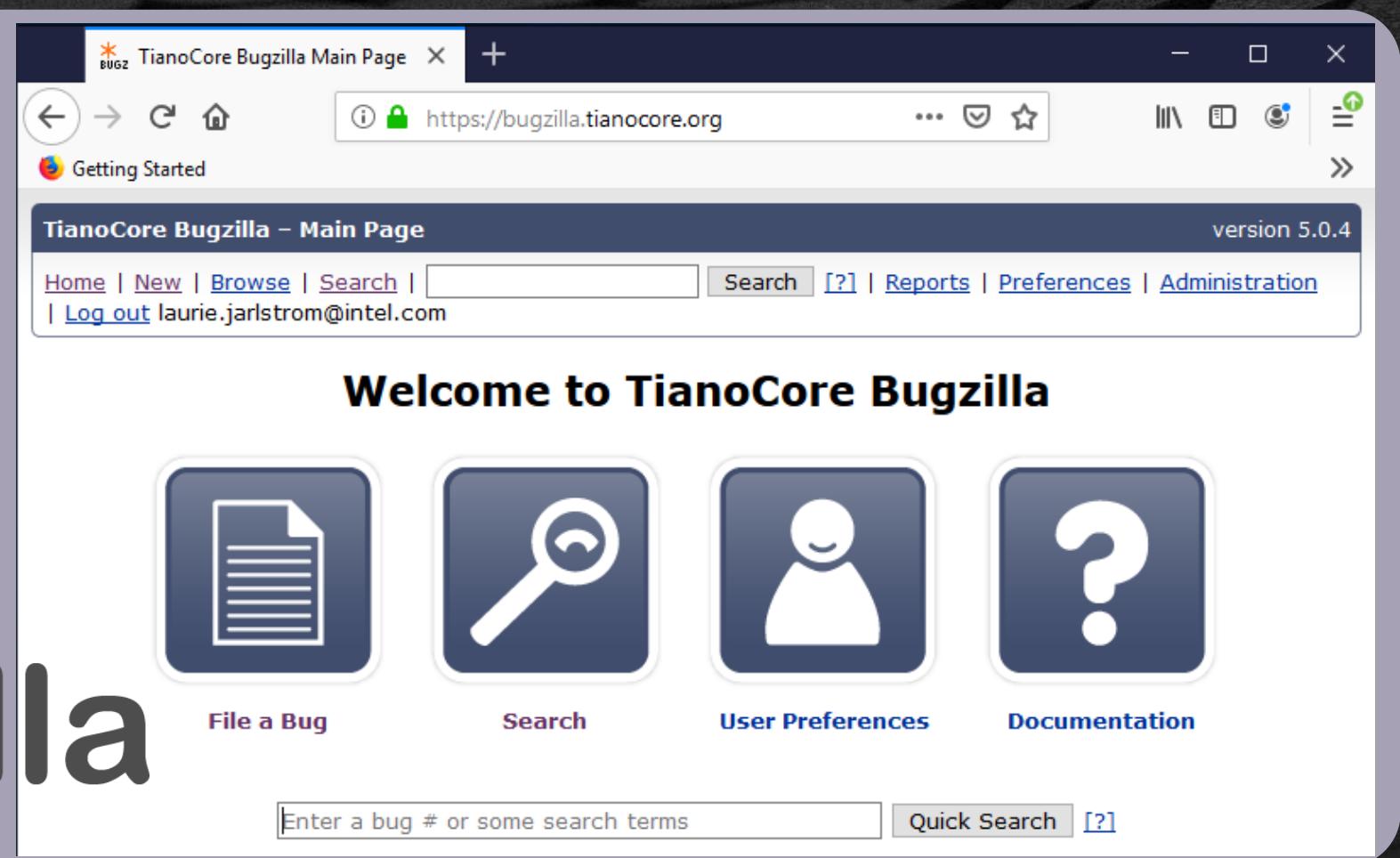
Report a bug on Bugzilla



Create a user account <https://bugzilla.tianocore.org/>

Search if bug “already” reported

File [New Report](#) – Pick a product – fill out form for the bug



The screenshot shows a web browser window displaying the TianoCore Bugzilla Main Page. The title bar reads "BUGZ TianoCore Bugzilla Main Page". The address bar shows the URL "https://bugzilla.tianocore.org". The page header includes the TianoCore Bugzilla logo, the text "TianoCore Bugzilla – Main Page", "version 5.0.4", and a user session with the name "laurie.jarlstrom@intel.com". Below the header is a navigation menu with links for "Home", "New", "Browse", "Search", "Log out", and "Administration". The main content area features a "Welcome to TianoCore Bugzilla" message and four large blue buttons with white icons: "File a Bug" (document), "Search" (magnifying glass), "User Preferences" (person), and "Documentation" (question mark). At the bottom, there is a search bar with the placeholder "Enter a bug # or some search terms" and a "Quick Search" button.

Bugzilla

SUMMARY

- ★ Review PI and UEFI Boot Process
- ★ Answer web-based training related questions
- ★ Answer: Where does Intel® FSP Fit?
- ★ What's new in UEFI.org

Questions?

Answers!!!?



Return to Main Training Page



Return to Training Table of contents for next presentation [link](#)



tianocore



BACKUP

What is Intel® Firmware Support Package?

- Intel® Firmware Support Package (Intel® FSP) includes:
 - A binary file
 - An integration guide
 - A rebasing tool
 - An IDE configuration tool / Boot Setting File (BSF)
- Provide silicon initialization code:
 - Initializes processor core, chipset as explained in BIOS Writers' Guide
 - Is relocatable in ROM
 - Can be configured for platform customization
- Boot loader agnostic and can be easily integrated with many options:
 - Open source boot loaders: UEFI –EDK II, Coreboot, U-boot, etc.
 - RTOS
 - Others

Intel® FSP V2.0 Boot Flow

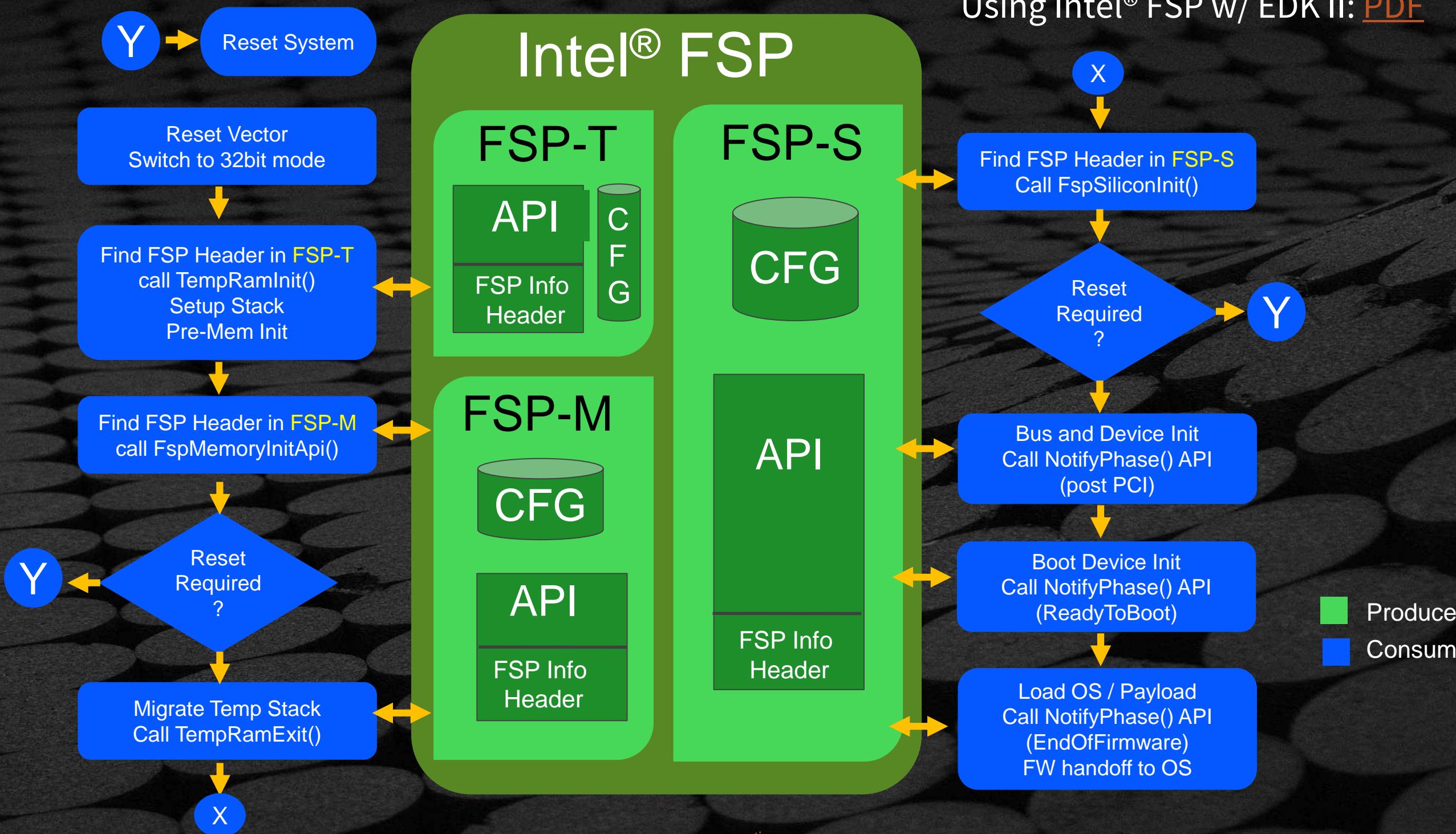


Diagram illustrates the high level boot flow

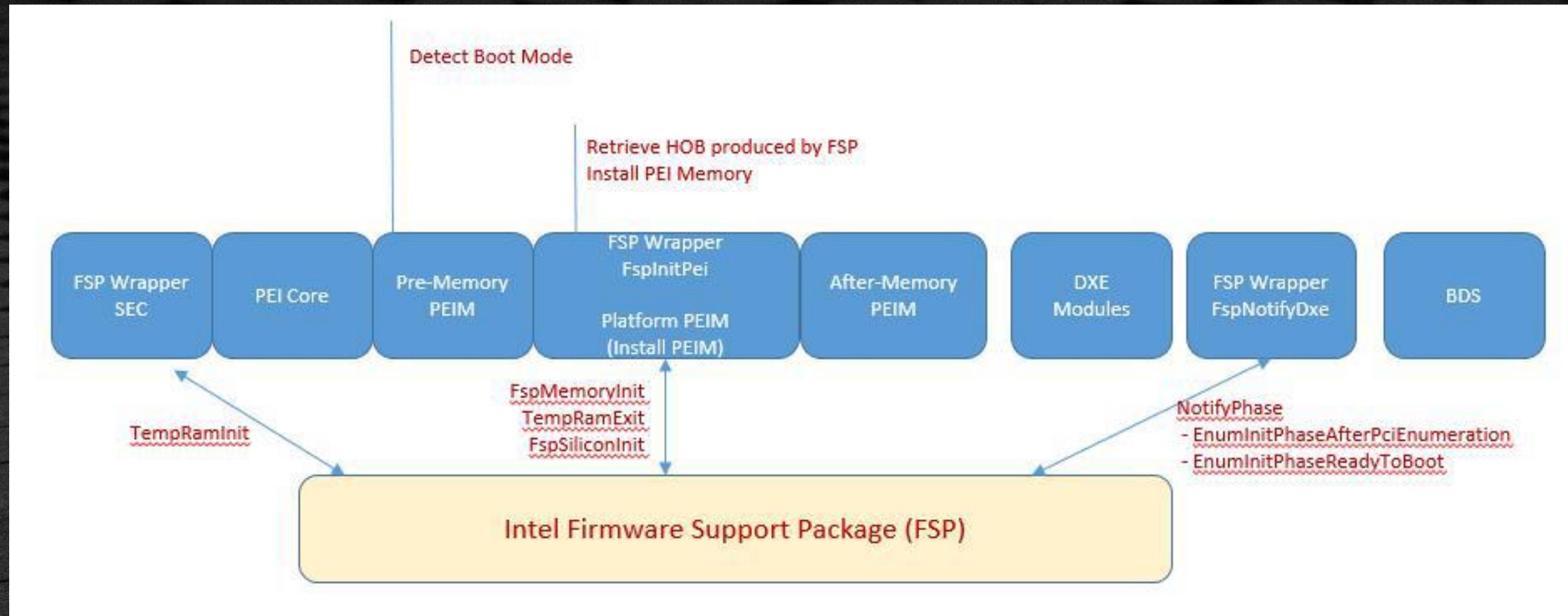


Image Source: Open Braswell Platform Designing Porting Guide [PDF](#)

Placement:

Once the Intel FSP binary is ready for integration, the bootloader build process needs to be configured to place the Intel FSP binary at the proper base address.

Rebase:

The Intel FSP is not Position Independent Code (PIC) and the whole Intel FAP has to be rebased with the Binary Configuration Tool (BCT) if it is placed at a location that is different from the default base address of the Intel FSP.

Interface:

The bootloader needs to add code to setup the Operating environment for the Intel FSP, call Intel FSP with the correct parameters and parse the Intel FSP output to retrieve the necessary information returned by the Intel FSP.

Customization:

The static Intel FSP configuration parameters/features are part of the Intel FSP binary and can be customized with BCT

<https://www.intel.com/content/www/us/en/intelligent-systems/intel-firmware-support-package/fsp-firmware-solutions-iot-video.html> - at -41:00 secs into

video



tianocore



WAY – WAY - BACK - BACKUP

UEFI Specification & EDK II Reference Implementation Timeline

