# Generic IP independent BIOS Signing and Parsing

Gahan Saraiya

Institute of Technology
Nirma University

December 20, 2019

# Outline

# Major Components I

## TianoCore

The community supporting an open source implementation of the Unified Extensible Firmware Interface (UEFI).

## UEFI [a]

[a] Unified Extensible Firmware Interface

Specifies the layer between an operating system and the platform firmware.

## EDK II

A modern, feature-rich, cross-platform firmware development environment for the UEFI and UEFI Platform Initialization (PI) specifications.

# Major Components II

## ACPI [a] Component Architecture

[a] Advanced Configuration and Power Interface

A major goal of the architecture is to isolate all operating system dependencies to a relatively small translation or conversion layer (the OS Services Layer) so that the bulk of the ACPICA code is independent of any individual operating system.

## PCIe [a]

[a] Peripheral Component Interconnect Express

A major goal of the architecture is to isolate all operating system dependencies to a relatively small translation or conversion layer (the OS Services Layer) so that the bulk of the ACPICA code is independent of any individual operating system.

# PC Architecture of Intel

- Every Intel architecture hardware platform includes 2 major components:
  - Microprocessor Chip
  - Companion Chip (PCH)
- Previously, Intel Processors were paired with 2 companion chips: North Bridge & South Bridge
- Now, the functions of the north bridge are usually included in the processor itself.
- South Bridge is replaced by the PCH (Platform Control Hub)

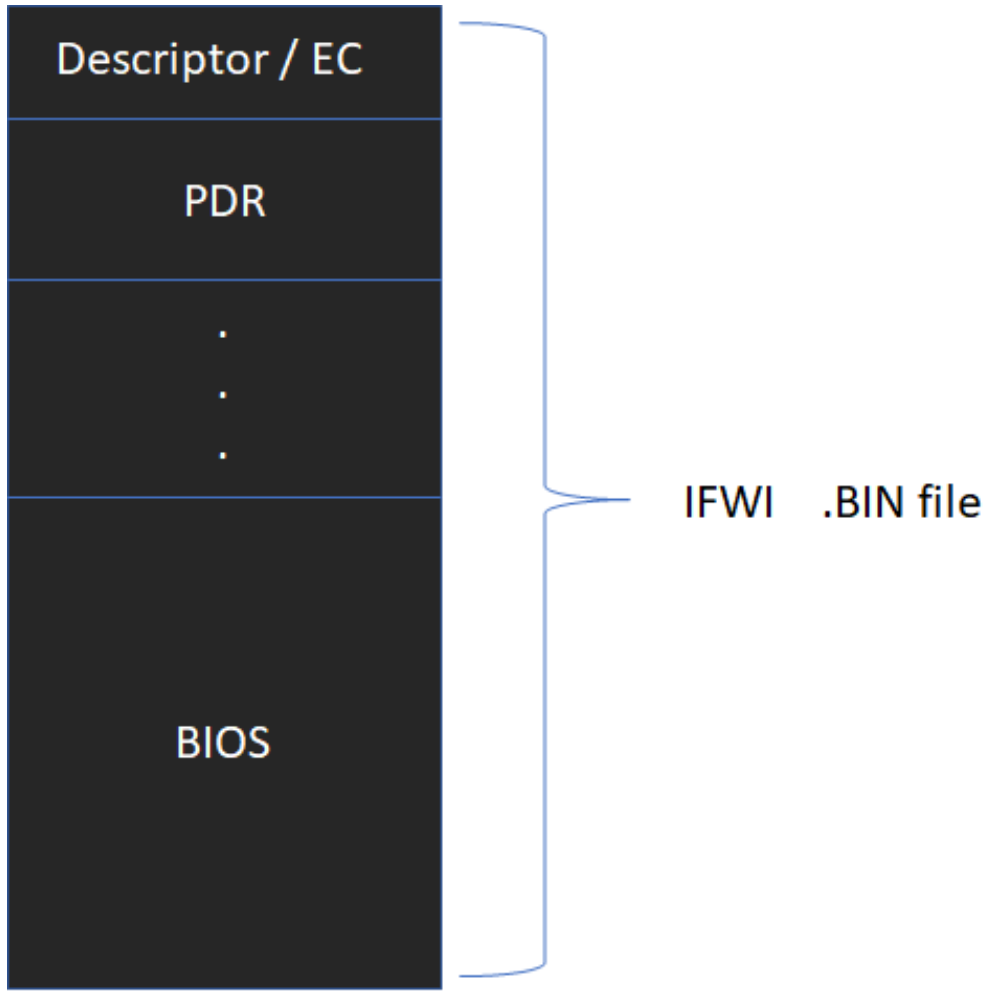
# Outline

# BIOS [1]

- Set of Software Routines
  - Initialize and test hardware on start
  - Provides the OS with a generic hardware abstraction
- the BIOS must do its job before your computer can load its operating system and applications

[1] Basic Input Output System

# Firmware Image



# Software Requirements

- Visual Studio C/C++ IDE
- Python 3
- Visual Studio Code
- EDK-II by TianoCore
- DediProg Engineering Tool
- Tera Term

# Outline

# Goal

Major Goal of the proposed work is to speedup Development cycle by reducing the iteration time of the build and deployment of the feature.

## Processing Debug/Unsigned BIOS

1. Applying changes directly to the sut
2. Applying changes on to the BIOS binary

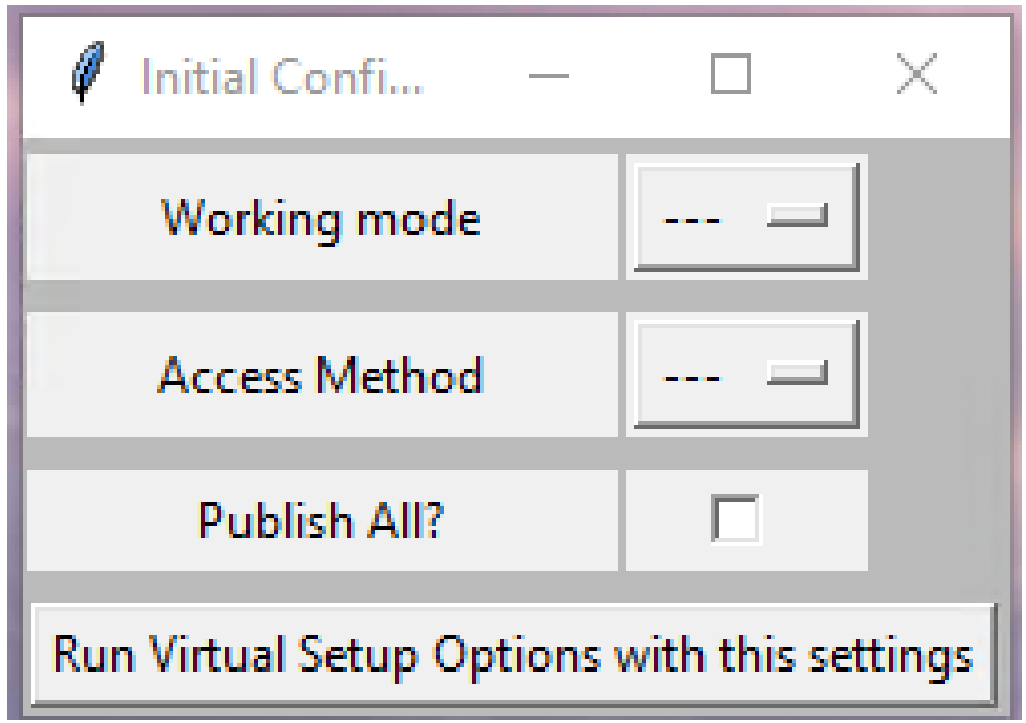
## Processing Firmware individually

Apply the whole firmware changes individually for BIOS

# Basic requirements to be fulfilled I

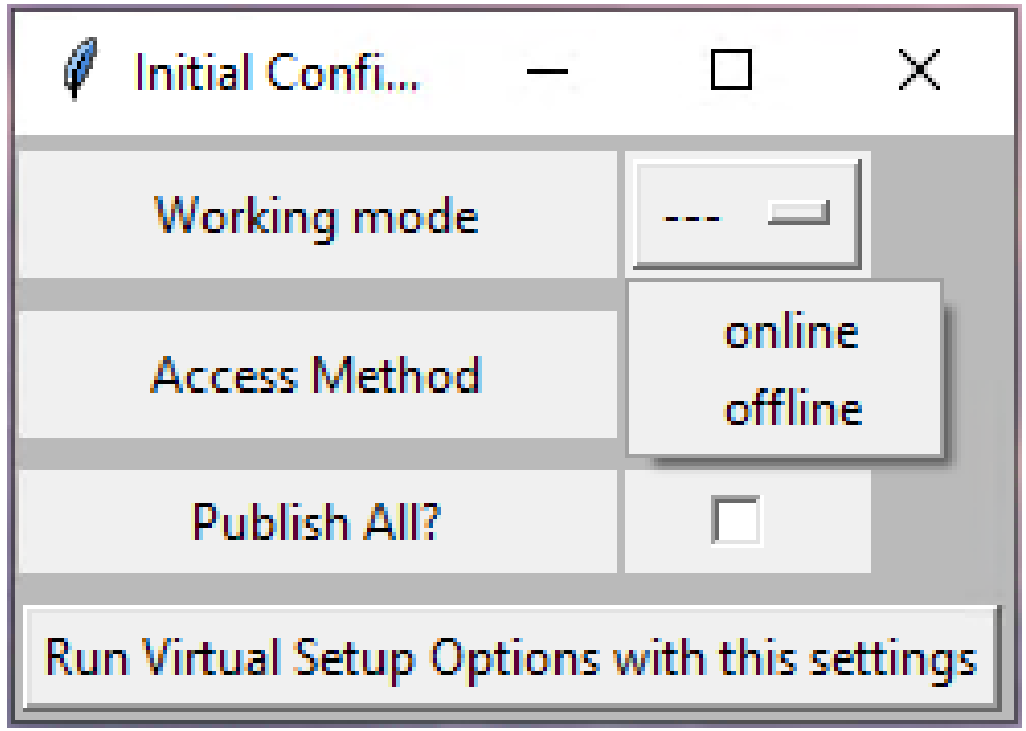
- Provide a solution which can work across all the platform binary and sut
- Provide a driver in BIOS firmware to aid the framework run directly on sut
- Provide a generic solution for both classification listed in 7
- Parsing the information from system/bin and simulate it to the framework
- Applying changes performed while simulation of framework
- Integration of new features and support for any new modules should be seamless

# Implementation Snaps I



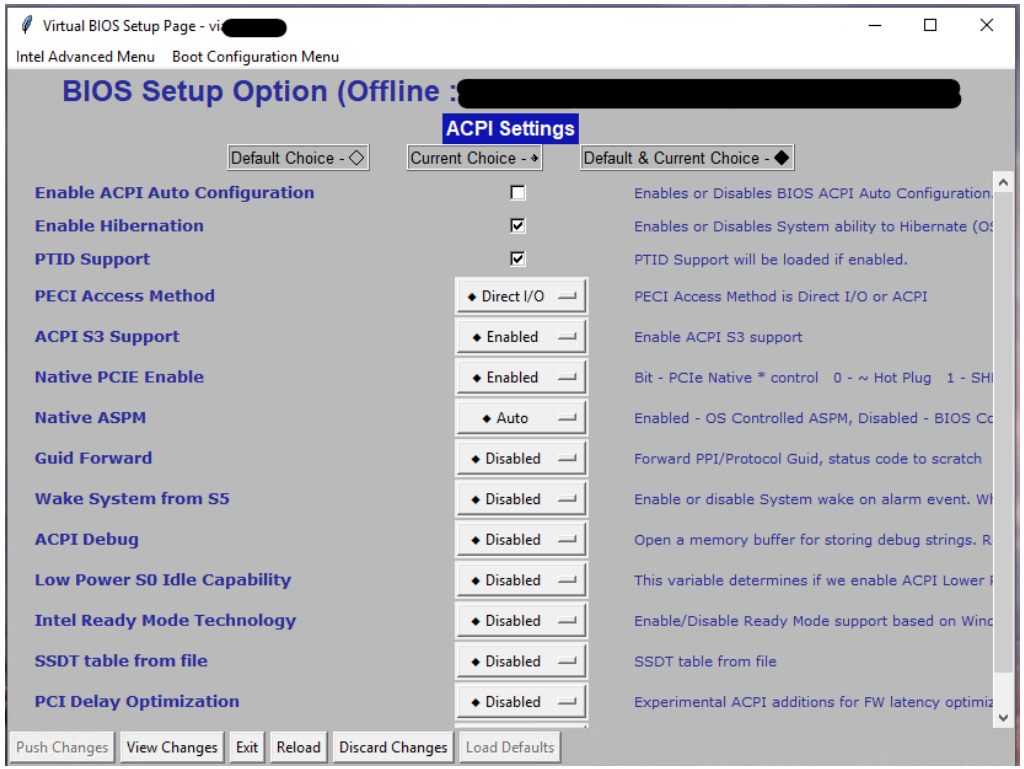

Menu to Select initial configuration for work

# Implementation Snaps II



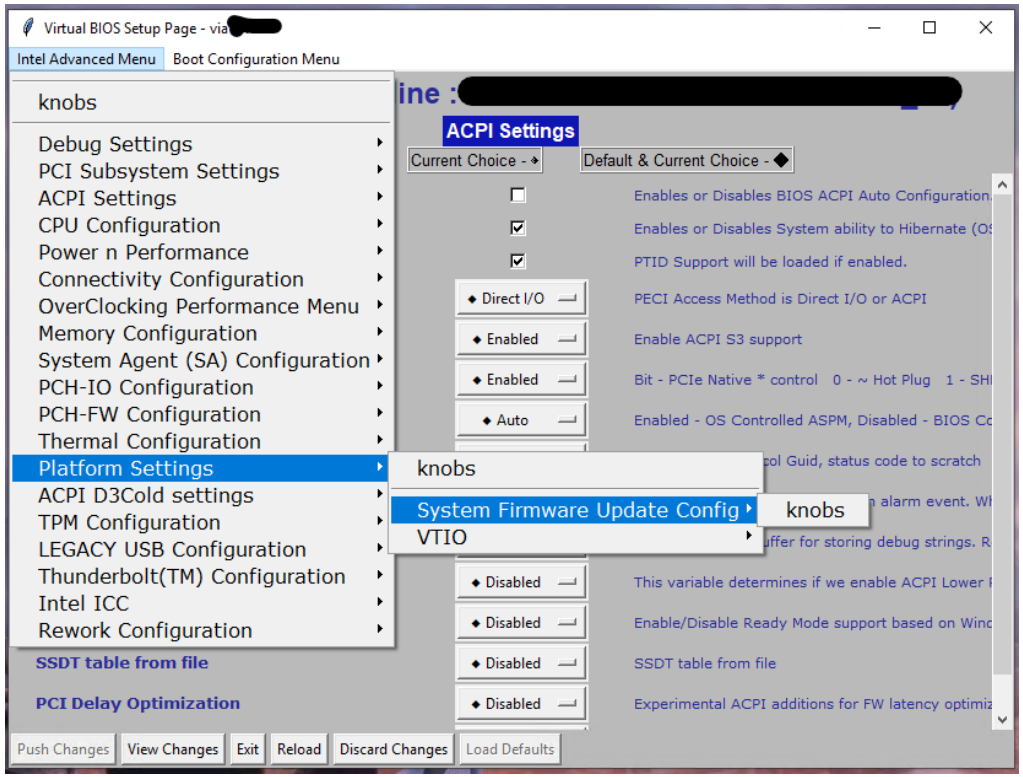

Available work mode for the system: Online and Offline

# Implementation Snaps III




Setup Options listed under ACPI Configurations

# Implementation Snaps IV

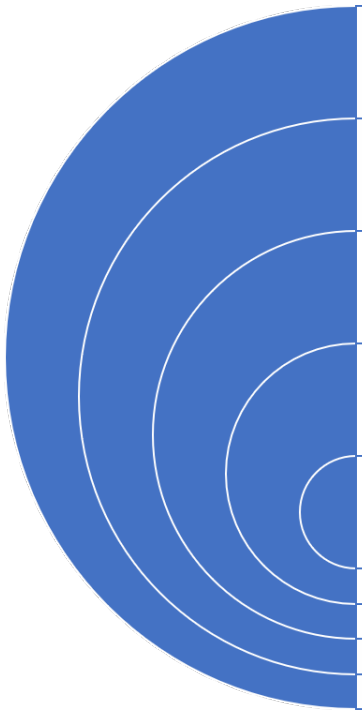



Navigating through BIOS setup page

# Primary Goals

- Remove other Intellectual Property's dependency (ip dependency) during firmware loading
- ip Subsystem :
  - Loader and Verifier
  - ip is always consumer
- Signature verification using sha hash algorithm and should be ease support for adding new algorithmic support as needed.
- Should support hardware based and software based verification support modifying memory requirements for given IP without impacting eco-system
- Prevent common security threats
- Allow easier OEM adoption and modification based on the respective design
- Reusability/Portability of design across many ips
- Generic design which supports any new IP integration

# Structure of Module



| | |
|---|---|
| Header | •Name<br>•Size<br>•Offset<br>•Checksum |
| Pointer(s) to sub structure (manifest(s), metadata(s) and binaries) | •Name<br>•Size<br>•Offset |
| Manifest(s) | •Size<br>•Type<br>•Signature<br>•Hash |
| Metadata(s) | •Name<br>•Size<br>•Hash |
| Binaries/firmware | •Modules |
| | |
| | |
| | |
| | |

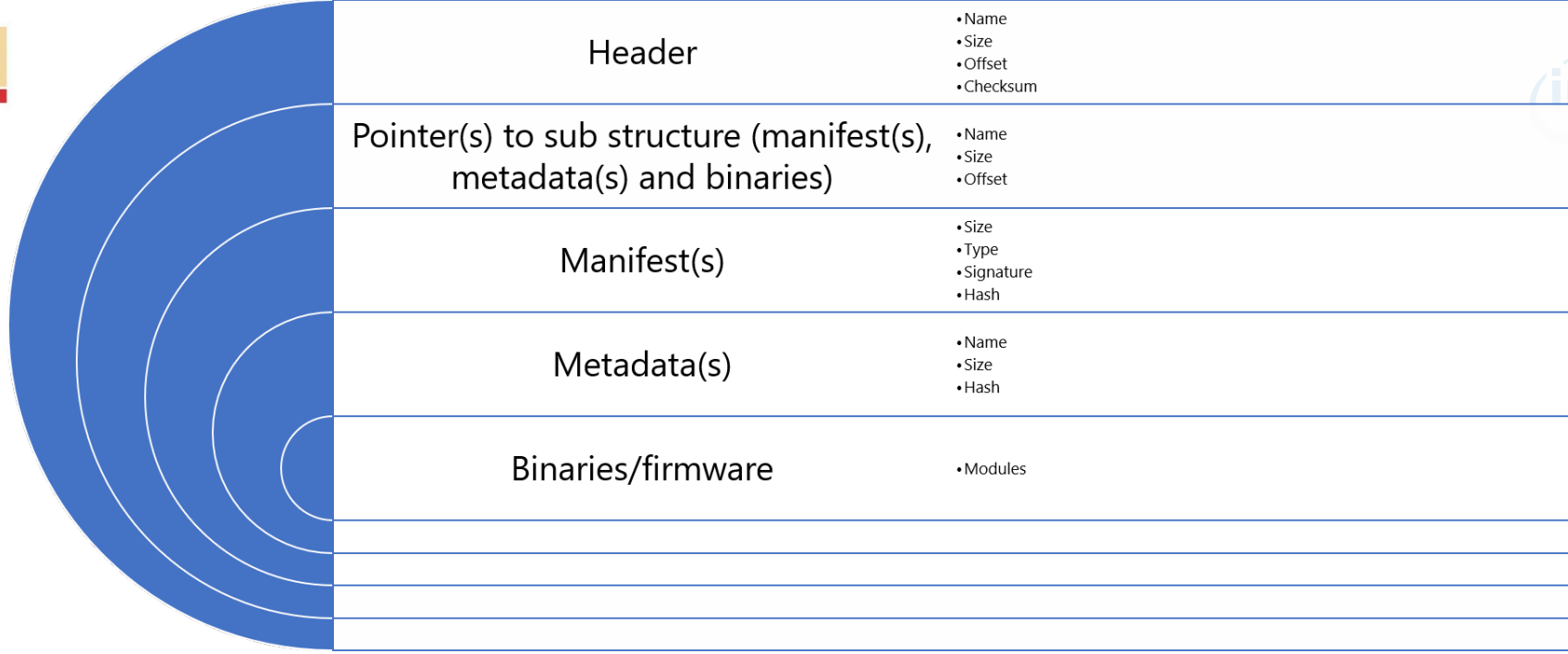
Proposed Structure for firmware signing

# Outline

# Future Scope

- Study of existing hotspots for automation
- Analyzing and gathering requirements of automation to hotspot
- Implementing and managing platform to keep up-to-date the user base of the framework

# Outline



| | |
|---|---|
| Header | •Name<br>•Size<br>•Offset<br>•Checksum |
| Pointer(s) to sub structure (manifest(s), metadata(s) and binaries) | •Name<br>•Size<br>•Offset |
| Manifest(s) | •Size<br>•Type<br>•Signature<br>•Hash |
| Metadata(s) | •Name<br>•Size<br>•Hash |
| Binaries/firmware | •Modules |

Proposed Structure for firmware signing

# Thank You!