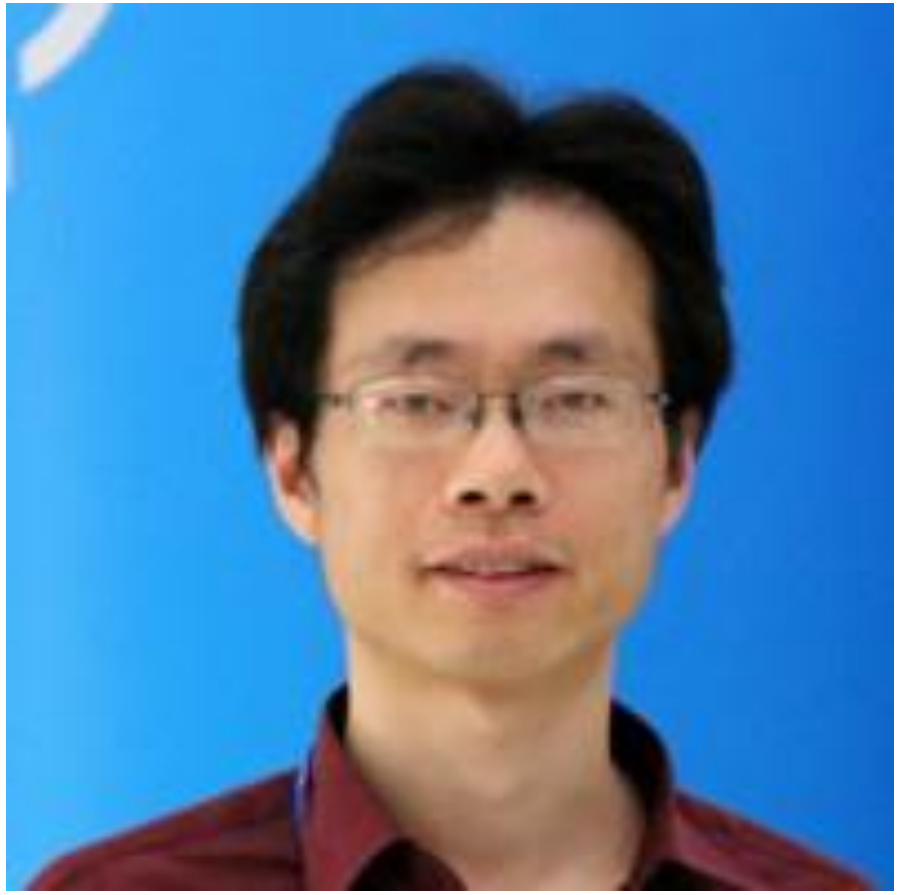
Virtual Firmware for Intel® Trust Domain Extensions

SPEAKER

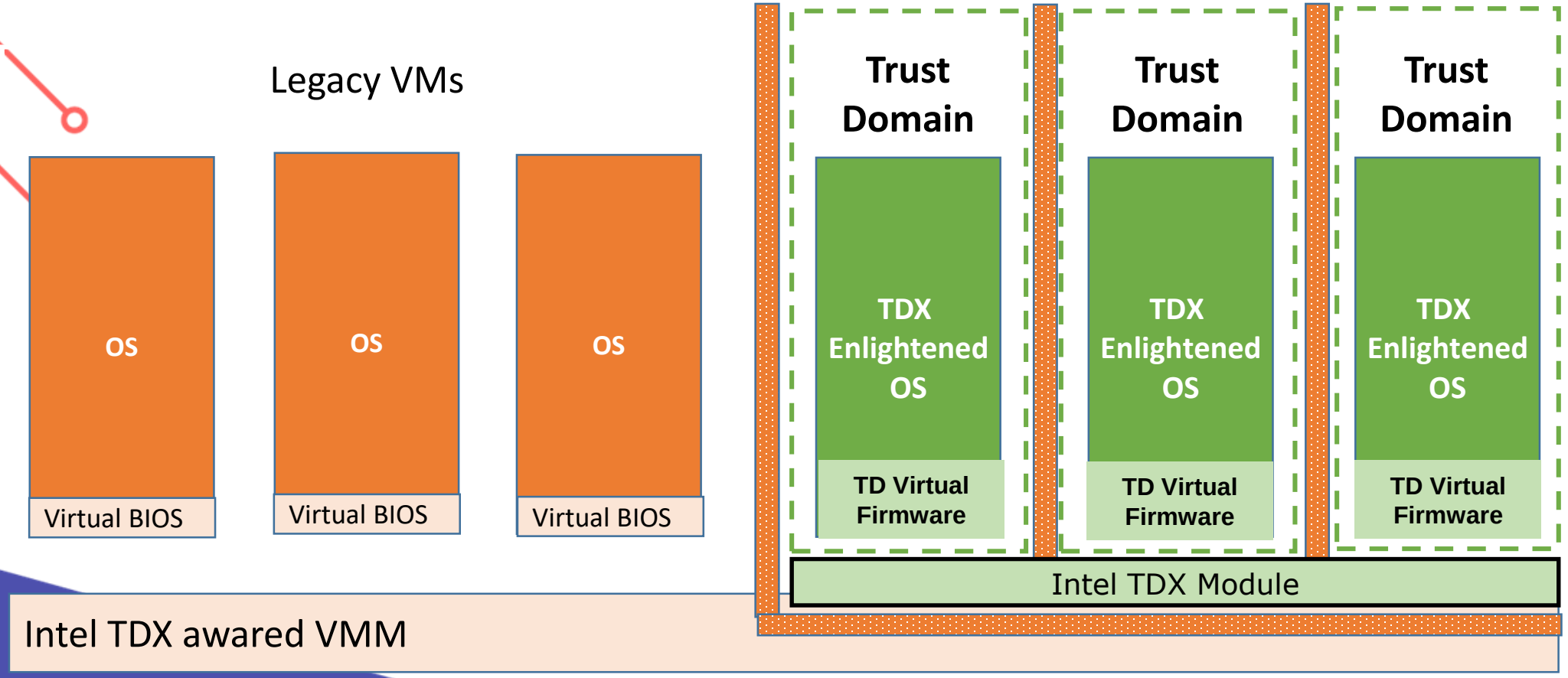
Jiewen Yao, Principal Engineer, Intel

Jiewen Yao

- **Jiewen Yao** is a principal engineer in the Intel Architecture, Graphics, and Software Group. He has been engaged as a firmware developer for over 15 years. He is a member of the UEFI Security sub team, and the TCG PC Client sub working group.
- He is the architect of TDX Virtual Firmware.



Intel® Trust Domain Extensions (TDX)



TD Virtual Firmware (TDVF)

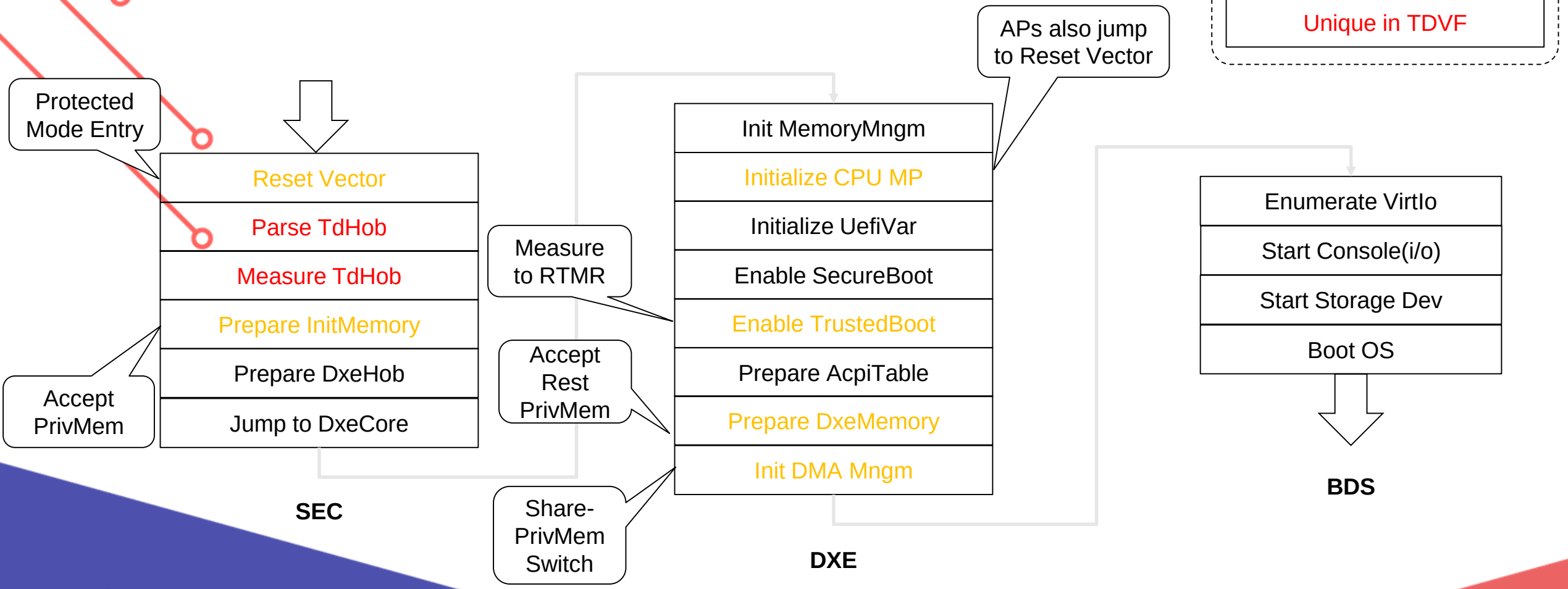
Responsibility:

- Own 1st instruction of a TD (reset vector)
- Provide service to a TD operating system (TD-OS)
- Build chain-of-trust from Intel TDX Module to TD-OS.

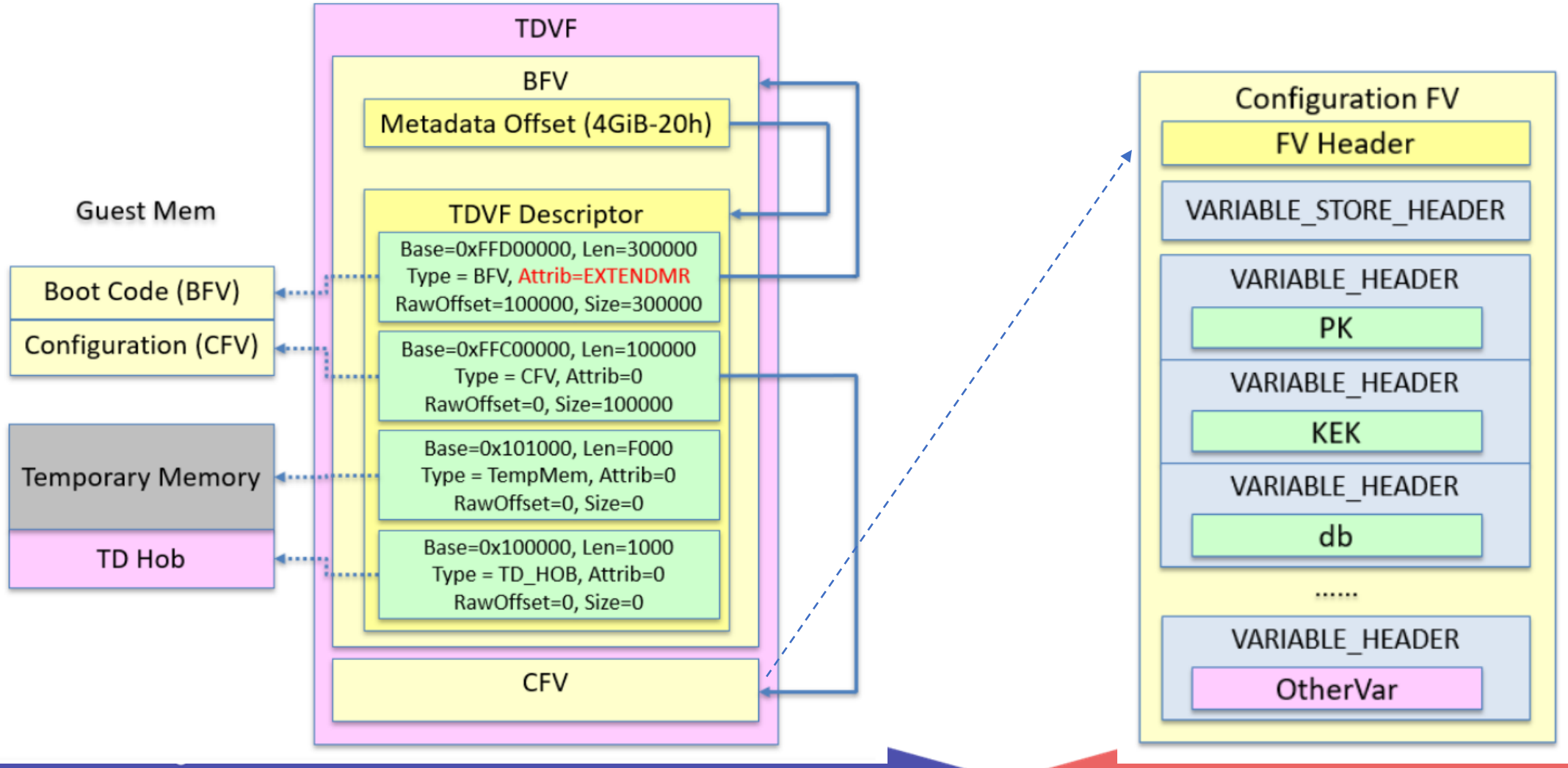
Implementation:

- Based upon EDKII Open Virtual Machine Firmware (OVMF)
- Simplified boot flow. (No PEI phase)

General Boot Flow – TDVF



TDVF Binary Layout



Launch State

- Reset Vector:
  - Protected Mode reset vector (0xFFFFFFF0)

- General Purpose Register:
  - **RBX**: Guest Physical Address Width (GPAW), 48 or 52
  - **RCX/R8**: hold a pointer of TD Hob. TD Hob contains the TD information, such as memory information, MMIO/IO information, which is passed from VMM.
  - **RSI**: Virtual CPU (VCPU) Index (0 ~ N-1)

Launch State

- Multi Processor Support
  - All CPUs jump to reset vector at same time.
  - VCPU 0 is selected as Bootstrap Processor (BSP)
  - VCPUs (1~N-1) are Application Processors (APs), parking and waiting to be waken up by BSP.

- TDCALL[TDG.VP.INFO]
  - R8: NUM_VCPUS

TDCALL

TDCALL	Usage	Comment
TDG.VP.VMCALL	Invoke service from the VMM	(See next page)
TDG.VP.INFO	Get TD information	GPAW, NUM_CPUS
TDG.MR.RTMR.EXTEND	Extend to TD runtime measurement register (RTMR)	SHA384 hash
TDG.VP.VEINFO.GET	Get #VE information	Exit Reason, Instruction Information
TDG.MR.REPORT	Get TD_REPORT	Measurement of the TD, TD configuration, Intel TDX module, etc.
TDG.VP.CPUIDVE.SET	Control unconditional #VE on CPUID	Supervisor mode, user mode.
TDG.MEM.PAGE.ACCEPT	Accept a pending, private page.	Guest physical address/size

TDCALL[TDG.VP.VMCALL]

TDG.VP.VMCALL	Usage
GetTdVmCallInfo	Enumerate VMCALL capabilities
MapGPA	Request VMM to map a GPA range as private or shared memory
GetQuote	Request a Quote-Enclave to sign the TD_REPORT to a TD_QUOTE
ReportFatalError	Report fatal error in TD.
SetupEventNotifyInterrupt	Request VMM specify which interrupt vector to use as an event notify vector.
Instruction.CPUID	Request VMM to emulate CPUID instruction
#VE.RequestMMIO	Request VMM to emulate the MMIO access
Instruction.HLT	Request VMM to emulate HLT instruction
Instruction.IO	Request VMM to emulate IO instruction
Instruction.RDMSR	Request VMM to emulate RDMSR instruction
Instruction.WRMSR	Request VMM to emulate WRMSR instruction
Instruction.PCONFIG	Request VMM to emulate PCONFIG instruction

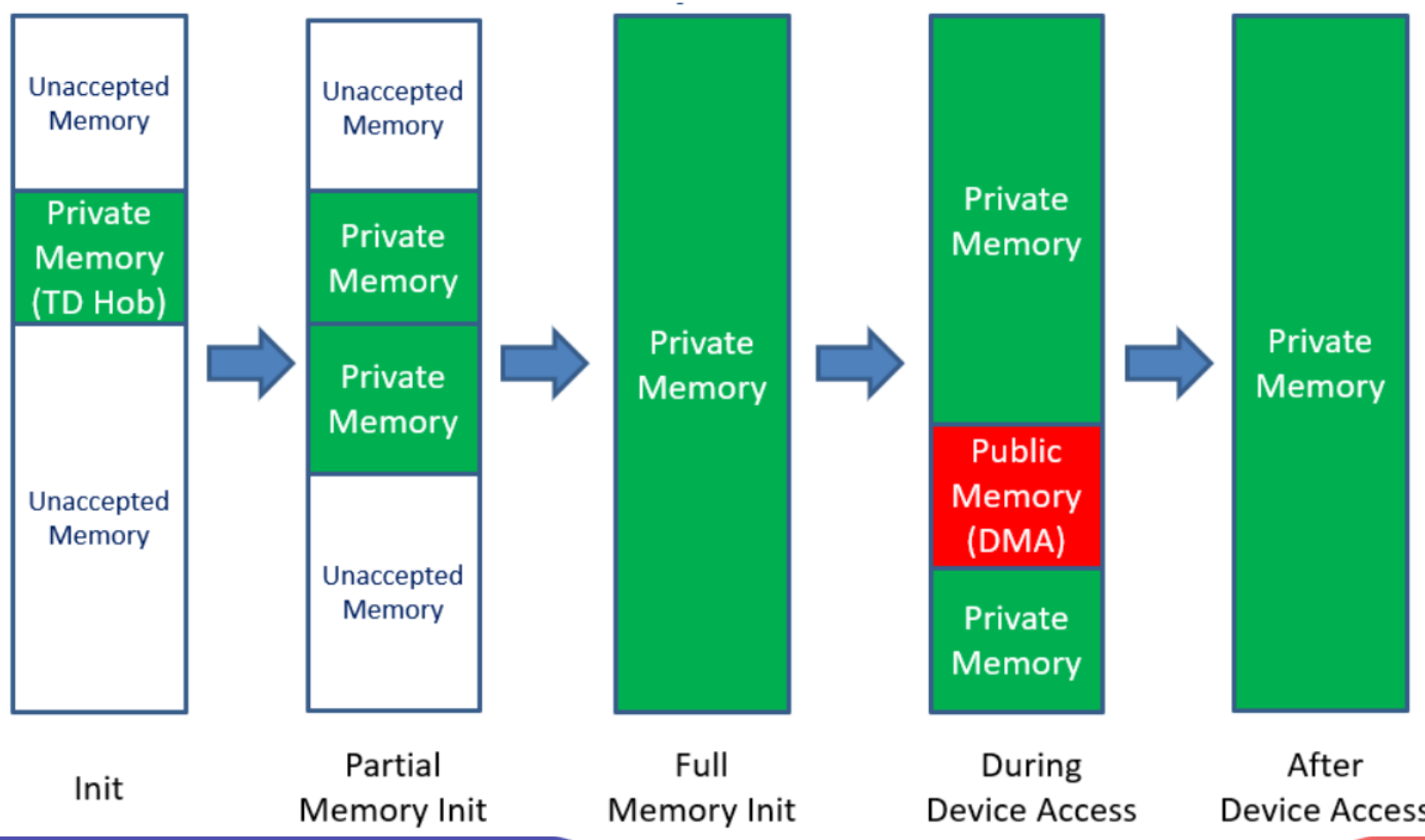
Memory Management

Type	Usage	Setup	Guest Page Table	Access
Private Memory	Default	1)**VMM:**SEAMCALL[TDH.MEM.PAGE.ADD] 2)**VMM:**SEAMCALL[TDH.MEM.PAGE.AUG] **TD:**TDCALL[TDG.MEM.PAGE.ACCEPT]	S-bit cleared	Direct Access
Shared Memory	Hypervisor communication buffer, Virtual device DMA buffer	Same as private memory	S-bit set	Direct Access
Unaccepted Memory	Private memory, not accepted yet.	**VMM:**SEAMCALL[TDH.MEM.PAGE.AUG]	N/A	N/A
MMIO	MMIO emulation	N/A	N/A	TDCALL[TDG.VP.VMCALL]<#VE.REQUESTMMIO>

Memory State Transition

Transition	Usage	Action
Unaccepted -> Private	Lazy memory init	TDCALL[TDG.MEM.PAGE.ACCEPT]
Private -> Shared	Communication buffer setup	Set S-bit in page table. TDCALL[TDG.VP.VMCALL]<MAPGPA>
Shared -> Private	Communication buffer reclaim	Clear S-bit in page table. TDCALL[TDG.VP.VMCALL]<MAPGPA> TDCALL[TDG.MEM.PAGE.ACCEPT]

Memory State Transition



UEFI/PI Memory Indicator

Type	UEFI Memory Map	PI GCD	PI Hob	ACPI E820	ACPI ASL
Private Memory	Normal UEFI Memory Type	EfiGcdMemoryType SystemMemory	EFI_RESOURCE_SYSTEM_MEMORY (EFI_RESOURCE_ATTRIBUYTE_ENCRYPTED)	Normal Memory Range	N/A
Shared Memory	Normal UEFI Memory Type	EfiGcdMemoryType SystemMemory	EFI_RESOURCE_SYSTEM_MEMORY	Normal Memory Range	N/A
Unaccepted Memory	EfiUnaccepted Memory (*)	EfiGcdMemoryType Unaccepted (*)	EFI_RESOURCE_SYSTEM_MEMORY (EFI_RESOURCE_ATTRIBUYTE_UNACCEPTED) (*)	AddressRangeUnaccepted (*)	N/A
MMIO	N/A	EfiGcdMemoryType MemoryMappedIo	EFI_RESOURCE_MEMORY_MAPPED_IO	N/A	Memory

ACPI – MP Wakeup

- AP init in OS
  - All APs are reported via MADT ACPI table.
  - A new MPWK structure is defined to describe a 4KiB mailbox.
    - APs loop to check the vector in the mailbox.
    - OS fills the AP Wakeup vector, then AP jumps to the Wakeup vector

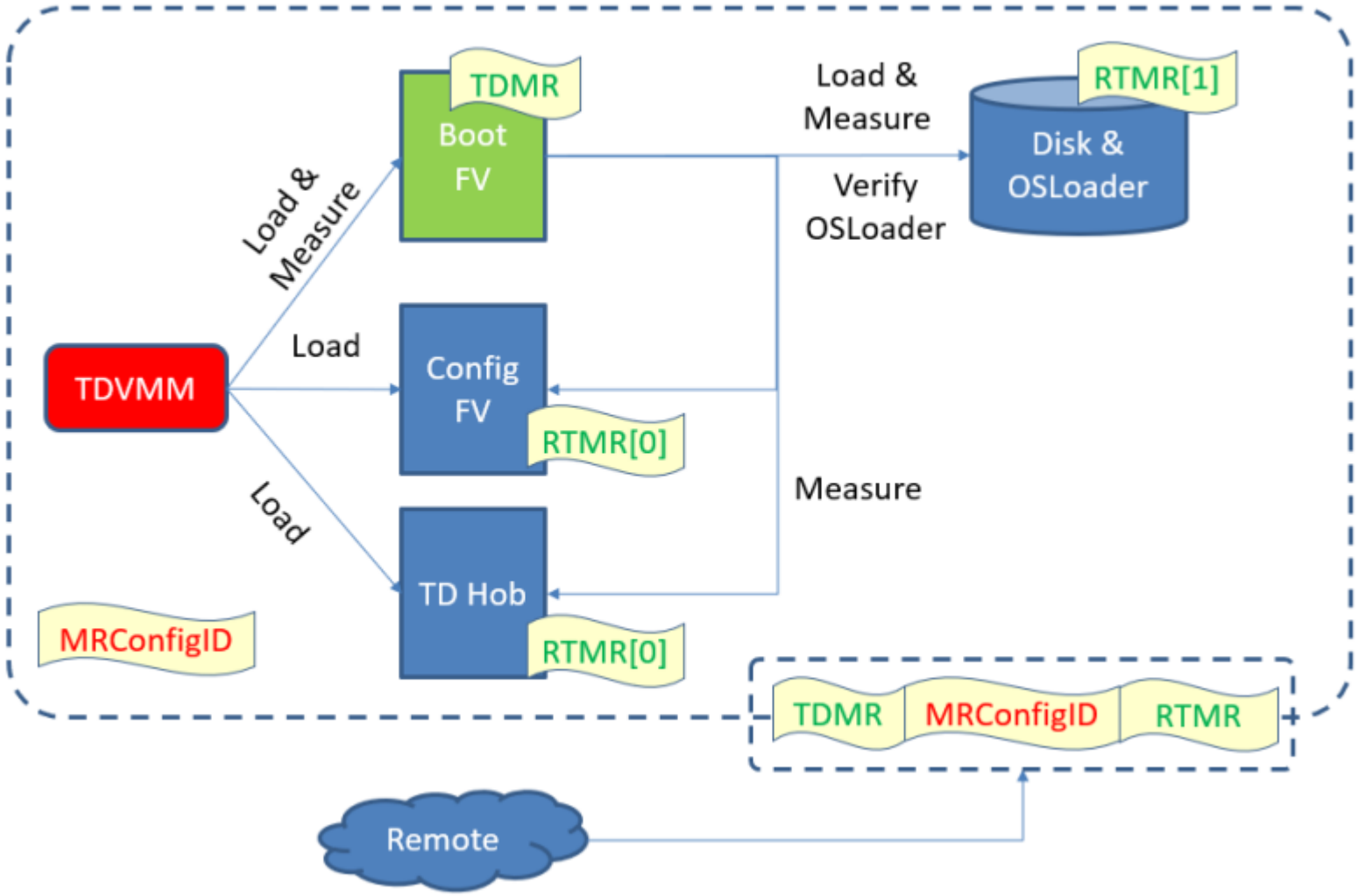
```
typedef struct {
  UINT8                                 Type;
  UINT8                                 Length;
  UINT16                                MailBoxVersion;
  UINT32                                Reserved2;
  UINT64                                MailBoxAddress;
} ACPI_MADT_MPWK_STRUCT;

typedef struct {
  UINT32                                Command;
  UINT32                                ApicId;
  UINT64                                WakeupVector;
  UINT8                                 OsReserved[SIZE_TO_2K];
  UINT8
FirmwareReserved[SIZE_TO_4K];
} ACPI_MPWK_MAIL_BOX;
```

Trusted/Measured Boot (TDMR + 4 RTMR)

PCR	Typical Usage	TD Register	TD Reg Index	Extended by	Comment
0	Firmware Code	TDMR	0	**VMM**:SEAMCALL[TDH.MR.EXTEND]	VF code (BFV, initial page table)
1	Firmware Data	RTMR [0]	1	**TDVF**:TDCALL[TDG.MR.RTMR.EXTEND]	Dynamic Configuration Data (TD HOB, ACPI) Data from FW_CFG_IO_SELECTOR/ FW_CFG_IO_DATA
2	Option ROM code	N/A		N/A	
3	Option ROM data	N/A		N/A	
4	OS loader code	RTMR [1]	2	**TDVF**:TDCALL[TDG.MR.RTMR.EXTEND]	OS loader
5	Boot Configuration	RTMR [1]	2	**TDVF**:TDCALL[TDG.MR.RTMR.EXTEND]	GPT, Boot Variable
6	N/A	N/A		N/A	
7	Secure Boot Configuration	RTMR [0]	1	**TDVF**:TDCALL[TDG.MR.RTMR.EXTEND]	SecureBootConfig (CFV)
	TD OS App measurement	RTMR [2]	3	**TDOS**:TDCALL[TDG.MR.RTMR.EXTEND]	TD OS App. Done by TD OS.

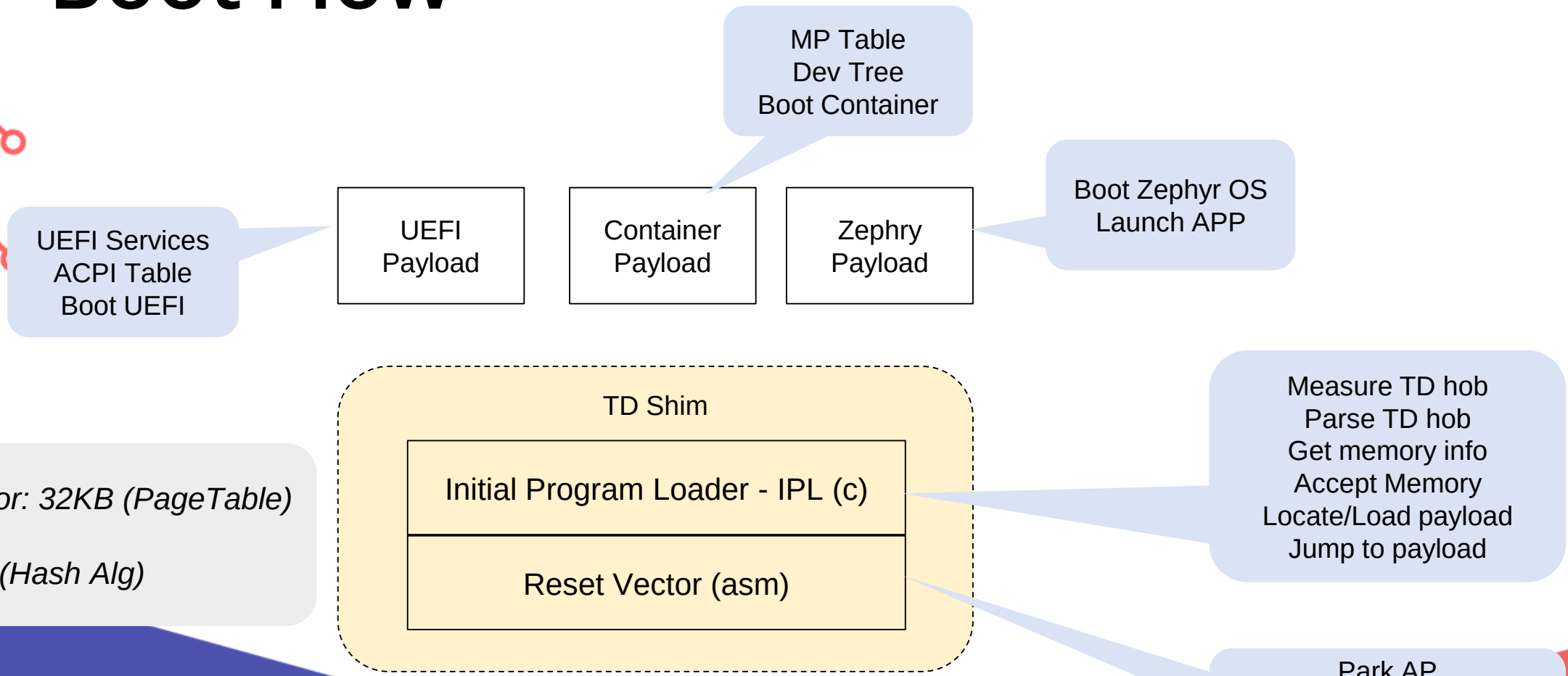
TDVF measurement



TD Shim – A tiny TDVF

	TDVF	TD Shim
Reset Vector	YES	YES
SEC (Initial Program loader - IPL)	IPL to boot a UEFI Core	IPL to boot a payload (UEFI, Container, Zephyr, etc)
UEFI Core	UEFI Services	NO
Device Driver	Virtio, PCI, etc	NO
ACPI Table (MultiProcessor)	MADT / DSDT	Static ACPI table only. Or MP table extension
Memory Map	UEFI Memory Map	E820 table
Trusted Boot	TD Measurement + TD Event Log (ACPI)	TD Measurement + TD Event Log Table
Secure Boot	UEFI Secure Boot	NO

Boot Flow



ResetVector: 32KB (PageTable)

IPL: 25KB (Hash Alg)

Summary

Intel® TDX

- Supports memory and CPU state confidentiality and integrity
- Supprots measurement and remote attestation

TD Virtual Firmware (TDVF)

- Launch a TD-OS
- Build the chain of trust

Reference

Intel® TDX Specification and Whitepaper

- https://software.intel.com/content/www/us/en/develop/articles/intel-trust-domain-extensions.html
- Intel® TDX Virtual Firmware Design Guide
- Intel® TDX Guest-Hypervisor Communication Interface

TDVF Pre-Production Code

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

Questions

Thank you

OPEN SOURCE FIRMWARE CONFERENCE