

STORAGE DEVELOPER CONFERENCE



Fremont, CA  
September 12-15, 2022

*BY Developers FOR Developers*

A **SNIA** Event

# SSD Firmware Resilience: Approaches and Challenges to Protect Against Emerging Threats



Gamil Cain  
Principal Engineer & Lead Product Security Architect  
[gamil.cain@solidigm.com](mailto:gamil.cain@solidigm.com)

Winson Yung  
Client SSD Firmware Architect  
[winson.yung@solidigm.com](mailto:winson.yung@solidigm.com)

# The New Paradigm of Solid-State Storage

\$8B+ Revenue\*

\*Solidigm + SK Revenue

3 NAND Factories

Global Organization,  
HQ in California

- Solid-State Innovation Since 1987

intel + SK hynix

- Leadership in Enterprise, Cloud & Client
- Pace-setting innovation across Floating Gate & Charge Trap, TLC & QLC

# Agenda

- Platform Resiliency Overview (as defined by NIST SP800-193)
- Resiliency Architectural Overview (as it relates to storage devices)
- Call to Action

# Platform Resiliency Overview



**NIST**  
Information Technology Laboratory  
**COMPUTER SECURITY RESOURCE CENTER**

**PUBLICATIONS**

**SP 800-193**

**Platform Firmware Resiliency Guidelines**

f t

Date Published: May 2018

Protect -> Detect -> Recover

# Brief Platform Resilience Taxonomy

---

- **Platform:** Comprised of hardware and firmware necessary to boot the system to a point at which software, or an operating system, can be loaded
- **Platform Device:** Typically contain mutable firmware, and are covered by the intended scope of the security guidelines in SP800-193
- **Code:** Firmware code is the set of instructions used by any device's processing unit to perform the operations required by the device
- **Critical Data (properties):**
  - It must be in a valid state for the proper booting and run-time operation of the device;
  - It persists across power cycles (e.g. stored in non-volatile memory)
  - It modifies the behavior or function of the device
  - It must be in a valid state to support protection, detection and/or recovery of platform firmware and associated data.

*For this talk: Firmware Resilience refers to firmware of an SSD that must be recoverable*

# Resilience: Not a New Concept for Storage Devices

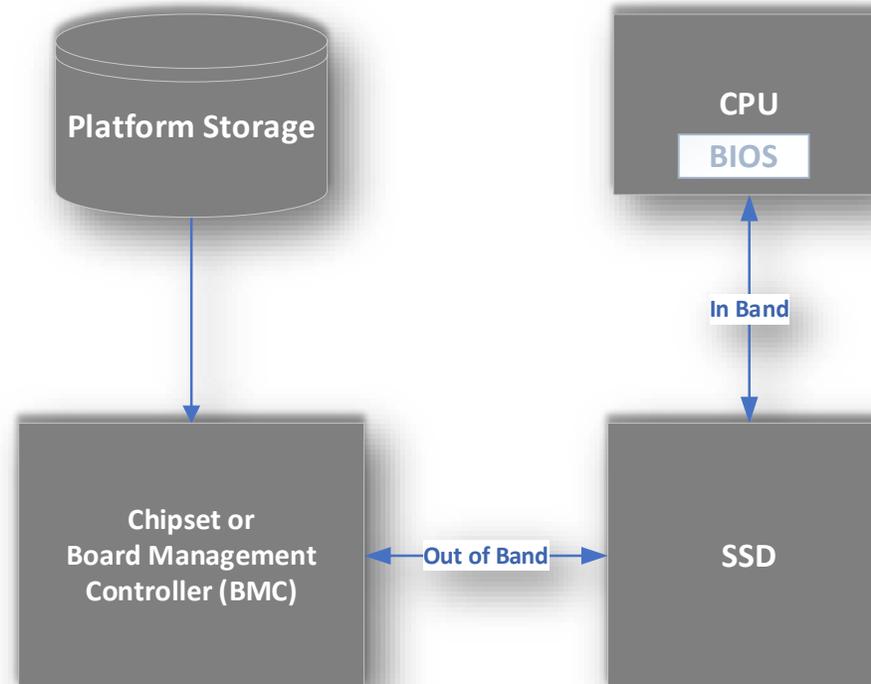
Existing Resilience Capabilities*	Description
Multiple Firmware Slots	Drive will automatically attempt to load device mutable code from an alternate slot if current slot fails
Multiple Firmware Copies / slot	Each slot contains redundant copies of Firmware
OCP Error Recovery Mechanisms	Enables Host to recover the device to a known security state
Read Only Mode	Enables the host to evacuate (read) user data when the drive reaches End of Life (EOL)
Telemetry	Provides forensic data for identifying failures
SMART Data	

**What do you do when NONE of the above mechanisms are successful?**

*\* The number and capability of resilience capabilities may vary across storage vendors*

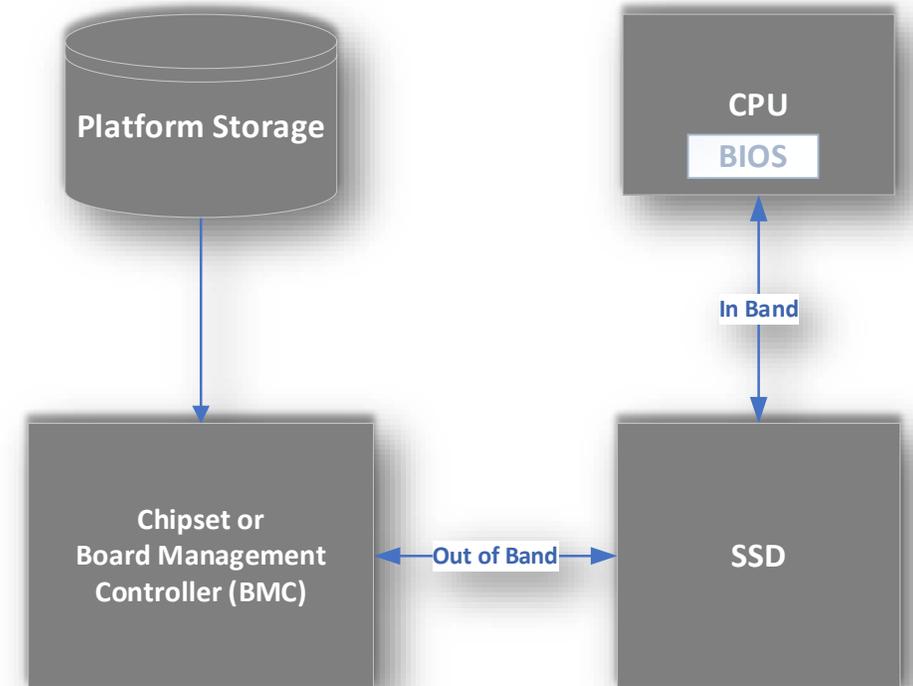
# Firmware Resilience Architecture

## Architectural Components for Resilience



# Elements of Firmware Resilience Architecture

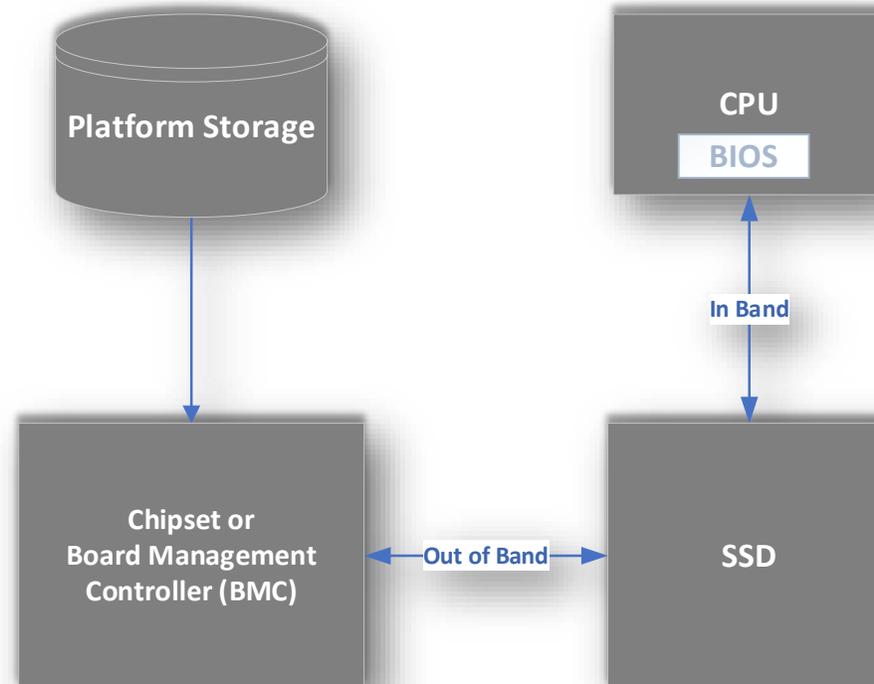
- **SSD Recovery State**
- **Interface(s) for Recovery**
- **Location of Recovery Firmware**
- **SSD Recovery Capabilities**
- **Support for Host Initiated Recovery**



*Some variations in Architectural Elements between Client and Datacenter storage devices*

# Firmware Resilience Architecture

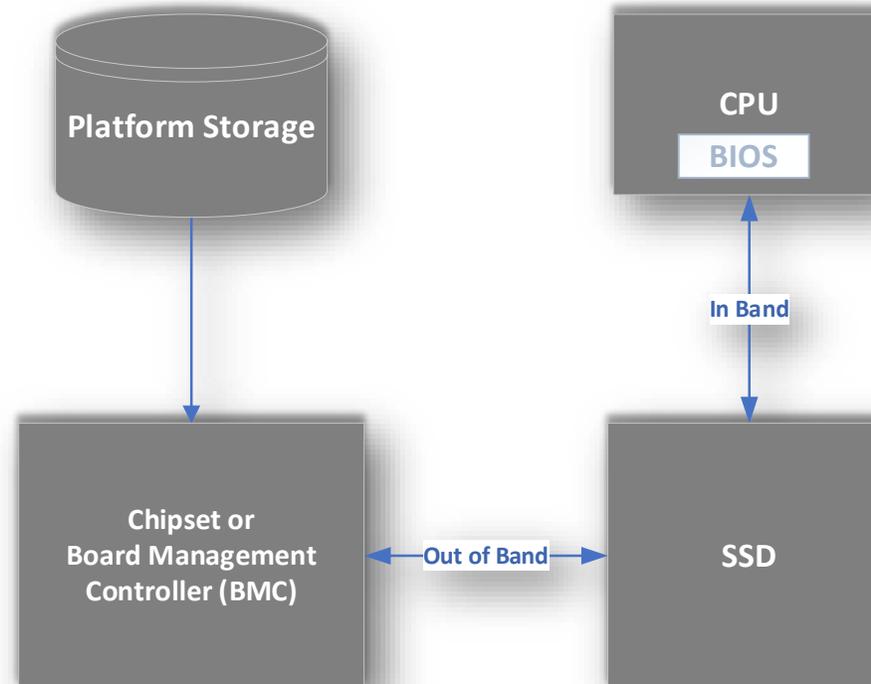
## SSD Recovery State



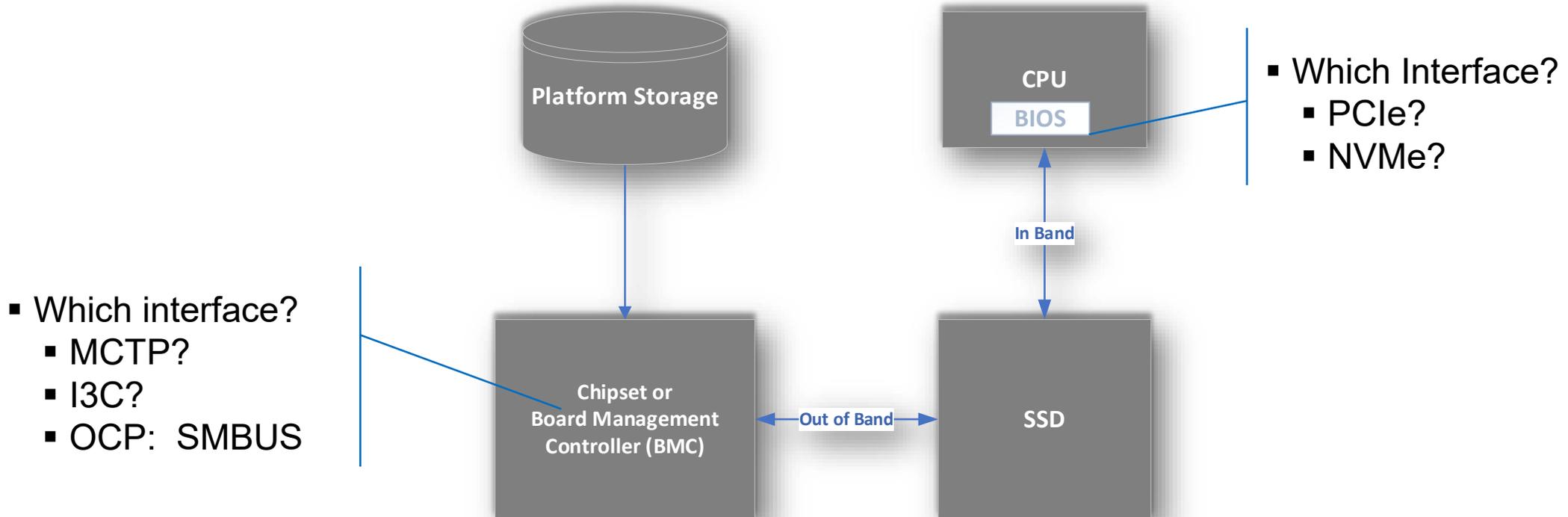


# Firmware Resilience Architecture

## Interface(s) for Recovery



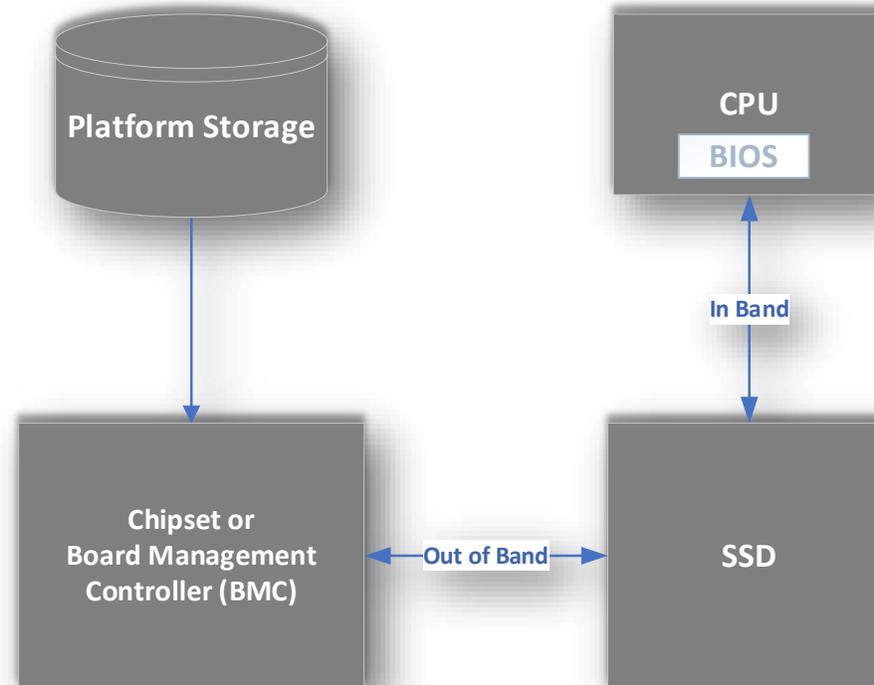
# Interface(s) for Recovery



- ✓ SSD SHALL expose an interface when no bootable firmware available
- ✓ Interface SHALL be available regardless of Firmware state
- ✓ Interface SHALL only accept verifiable Recovery Firmware

# Firmware Resilience Architecture

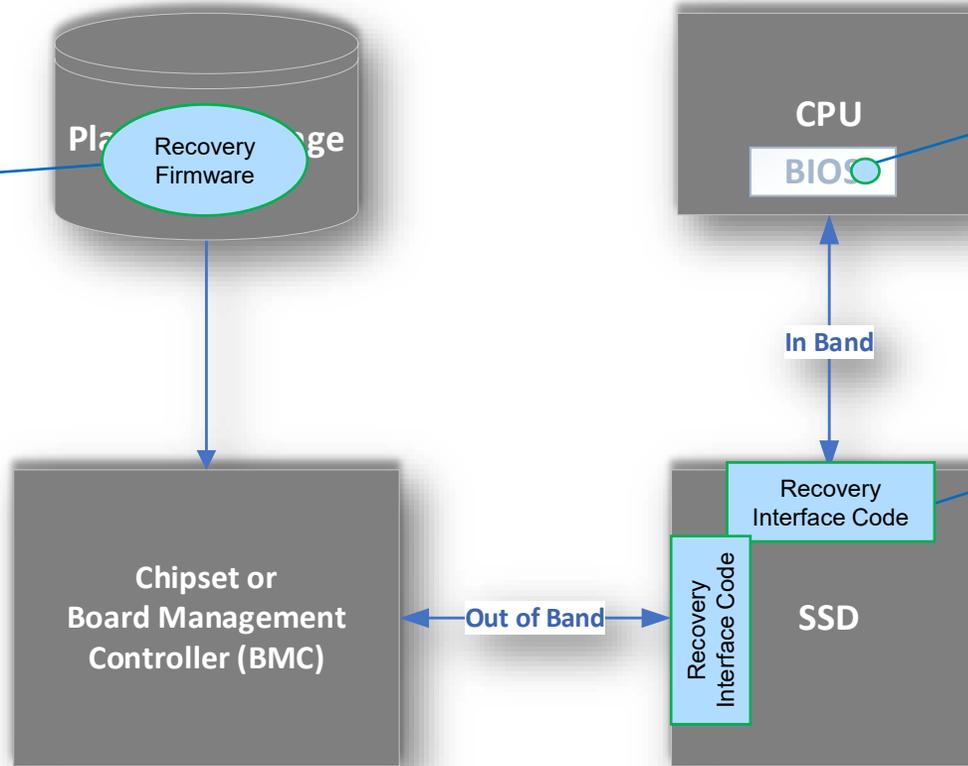
## Location of Recovery Firmware



# Location of Recovery Firmware

## Recovery Firmware

- Fully functional.
- Integrity Protected.
- Updatable at any time.



## Recovery Firmware

- Fully functional.
- Integrity Protected.
- Updatable at any time.

## Recovery Interface Code

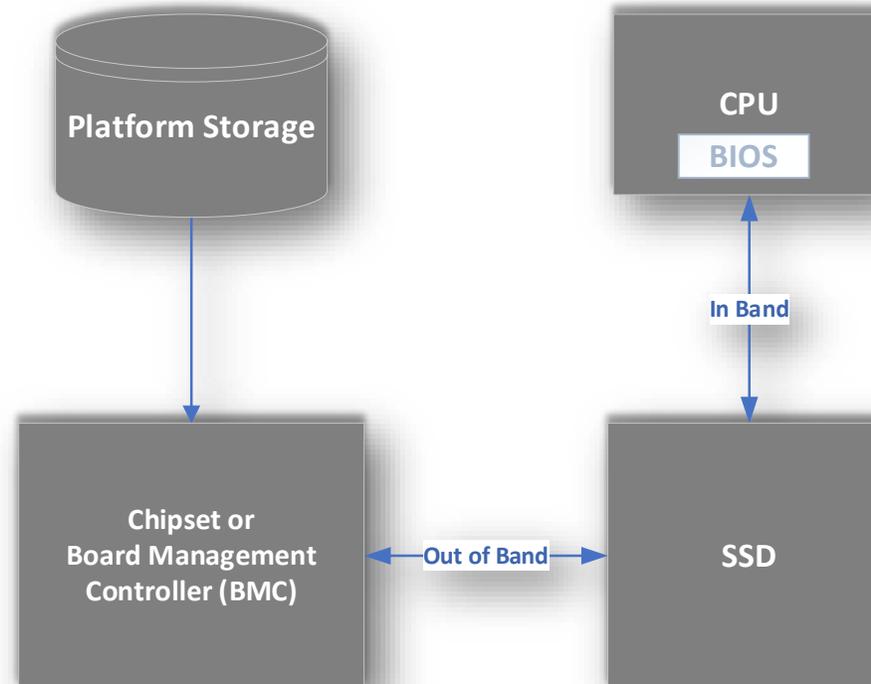
- Immutable (recommended).
- Automatically run if no bootable Firmware
- Interface commands limited to Recovery

### ➤ What if Recovery Interface Code can't be immutable?

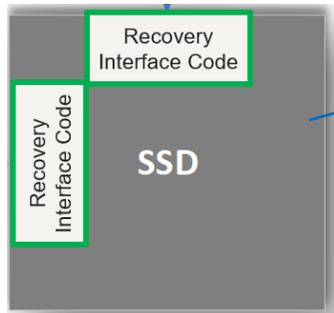
- Designate a Firmware slot to store interface code as "read-only" Firmware
  - "Read Only" firmware may encounter same issues that made recovery necessary

# Firmware Resilience Architecture

## SSD Recovery Capabilities



# SSD Recovery Capabilities



## Recovery Interface Code

- Automatically runs when no bootable firmware found
- Discovery capabilities, such as...
  - OCP: Secure Boot Failure
  - Boot failures recorded in log pages
  - Etc.

**Not Addressed in any recognized industry**

**interface specifications**

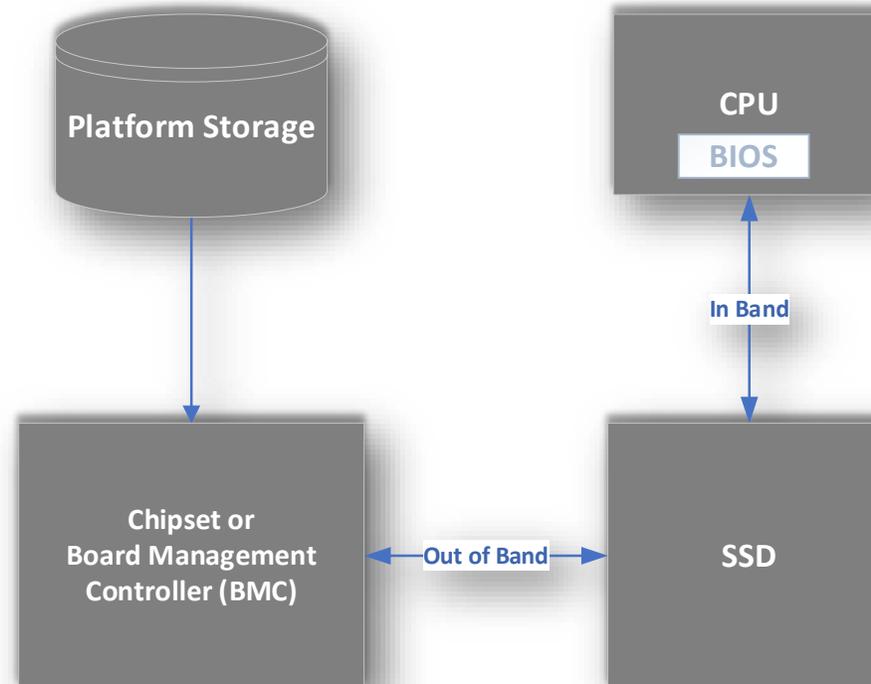
**(PCIe, DMTF, NVMe, SMBUS, etc.)**

- Standard method for creating/retrieving/recovery image
  - SHALL only accept verifiable recovery images
- Recovery (or its construction) of Critical Data
  - MAY imply some level of device formatting
  - MAY imply full recovery in a secure manufacturing environment
    - E.g., to restore cryptographic keys

*\* NIST SP800-193: Critical data is mutable data which persists across power cycles and must be in a valid state for the booting of the platform to securely and correctly proceed.*

# Firmware Resilience Architecture

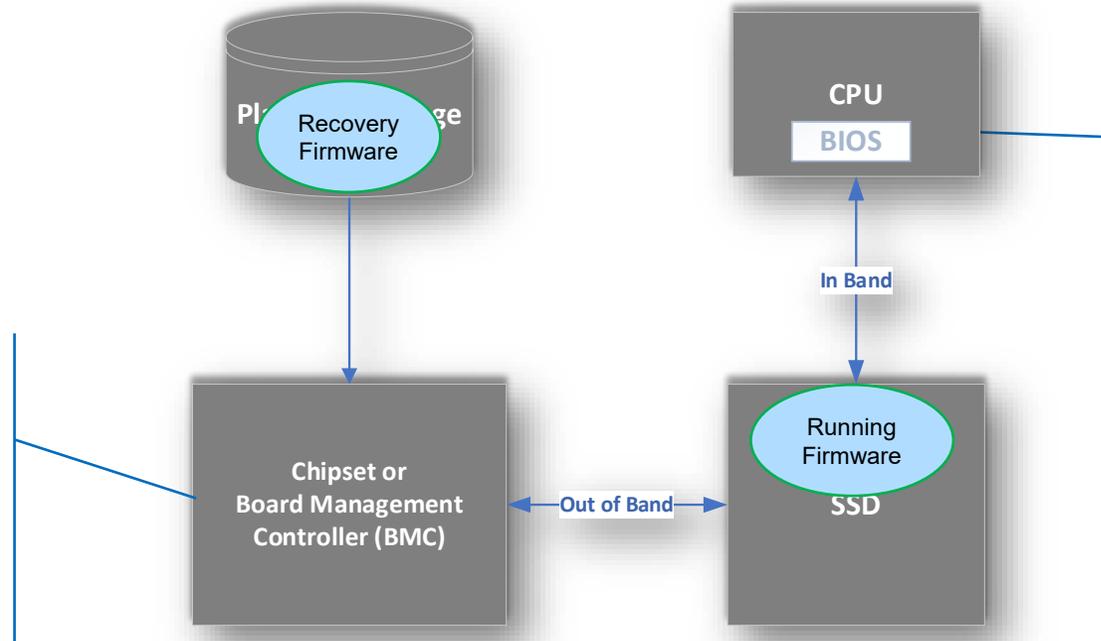
## Support for Host Initiated Recovery



# Host Initiated Recovery

## Host Initiated Recovery

- Example: BMC unable to successfully perform Attestation, triggering a Recovery Policy



## Host Initiated Recovery

- Example: Host software detects instability and wants to restore SSD to factory state.

## How is Trust established for Host Initiated Recovery?

- Physical Presence examples
  - GPIO
  - VU command with TCG PSID\*

\* <https://trustedcomputinggroup.org/resource/tcg-storage-opal-feature-set-psid/>

---

# Call to Action

# Call to Action

- Industry partnership to align on Recovery usage models and functional requirements
  - OCP putting a stake in the ground. Broader industry participation encouraged
- Help drive industry interface specification(s) for Recovery
  - Driven by platform usage models / constraints (e.g., client vs. datacenter platforms)

References	Description
<a href="#">NIST SP800-193 - Platform Firmware Resiliency Guidelines</a>	Establishes a framework for Protection, Detection and <b>Recovery</b>
<a href="#">OCP Recovery Specification</a>	Industry led specification. Defines a protocol for recovery based on SMBUS. Not yet ratified by OCP
	Industry Interface Specification(s) that define the interface semantics for Recovery. May require multiple interface definitions to support all variant use cases for Recovery

---

# Thank You!





# Please take a moment to rate this session.

Your feedback is important to us.