S-RFUP: Secure Remote Firmware Update Protocol

Rakesh Podder
Colorado State University

Tyler Rios
Colorado State University

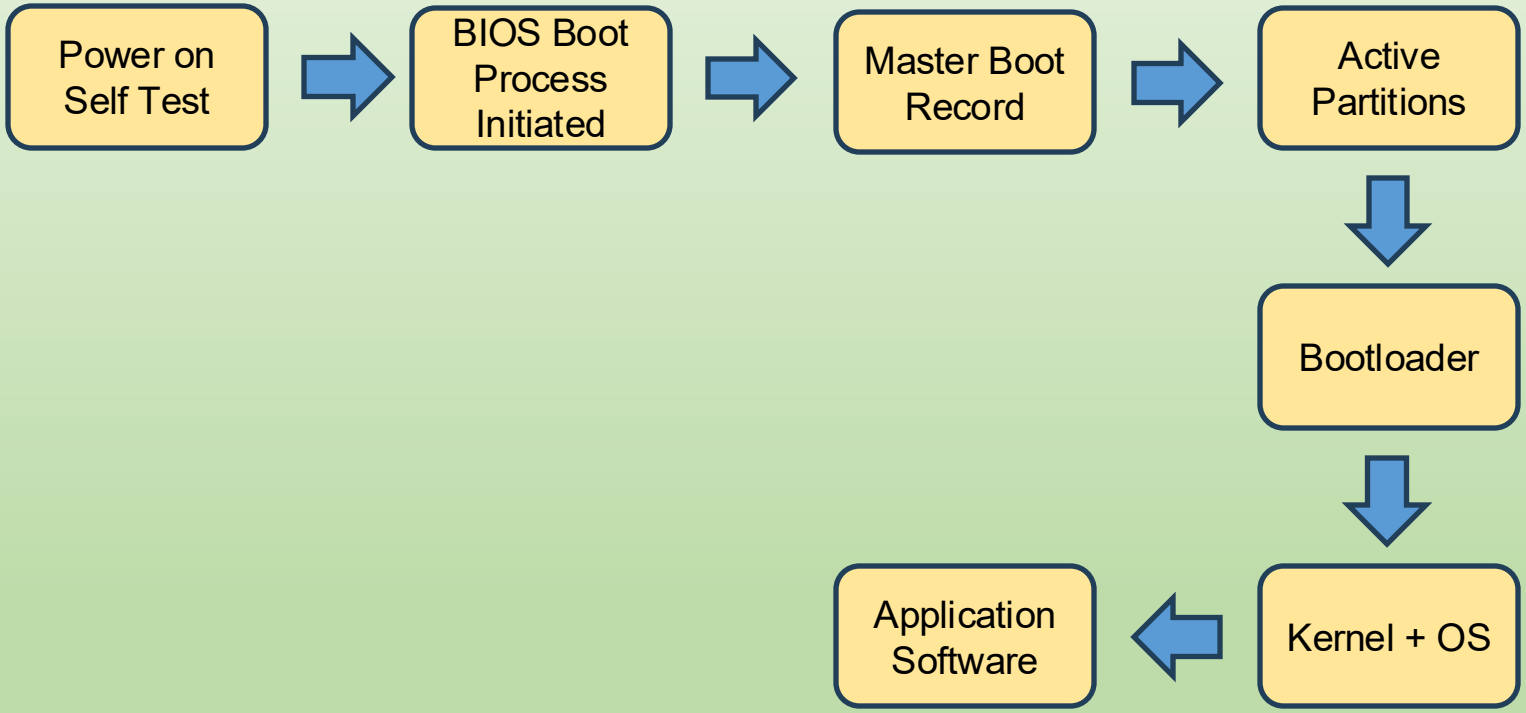
Indrajit Ray
Colorado State University

Presanna Raman
AMI US Holding, Inc.

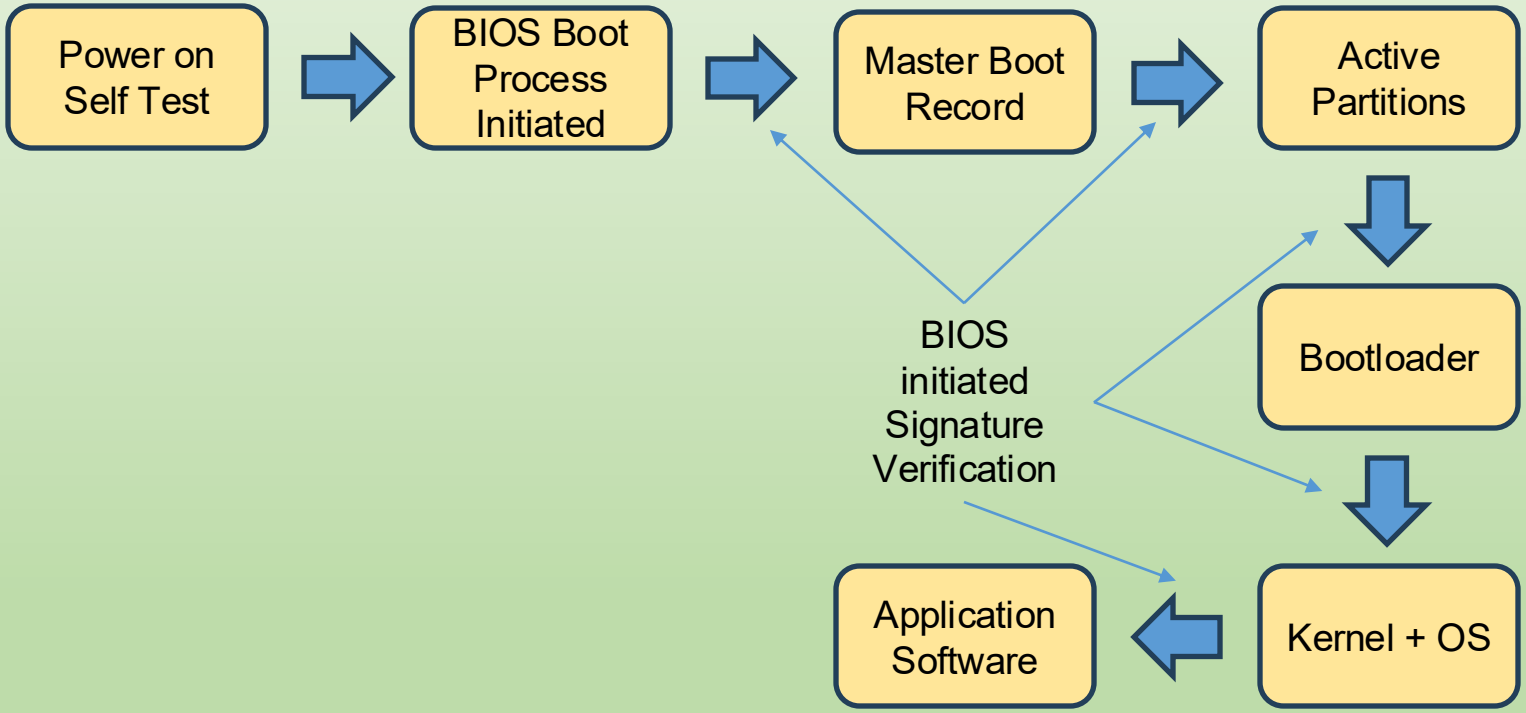
Stefano Righi
AMI US Holding, Inc.

20th International Conference on Information Systems Security

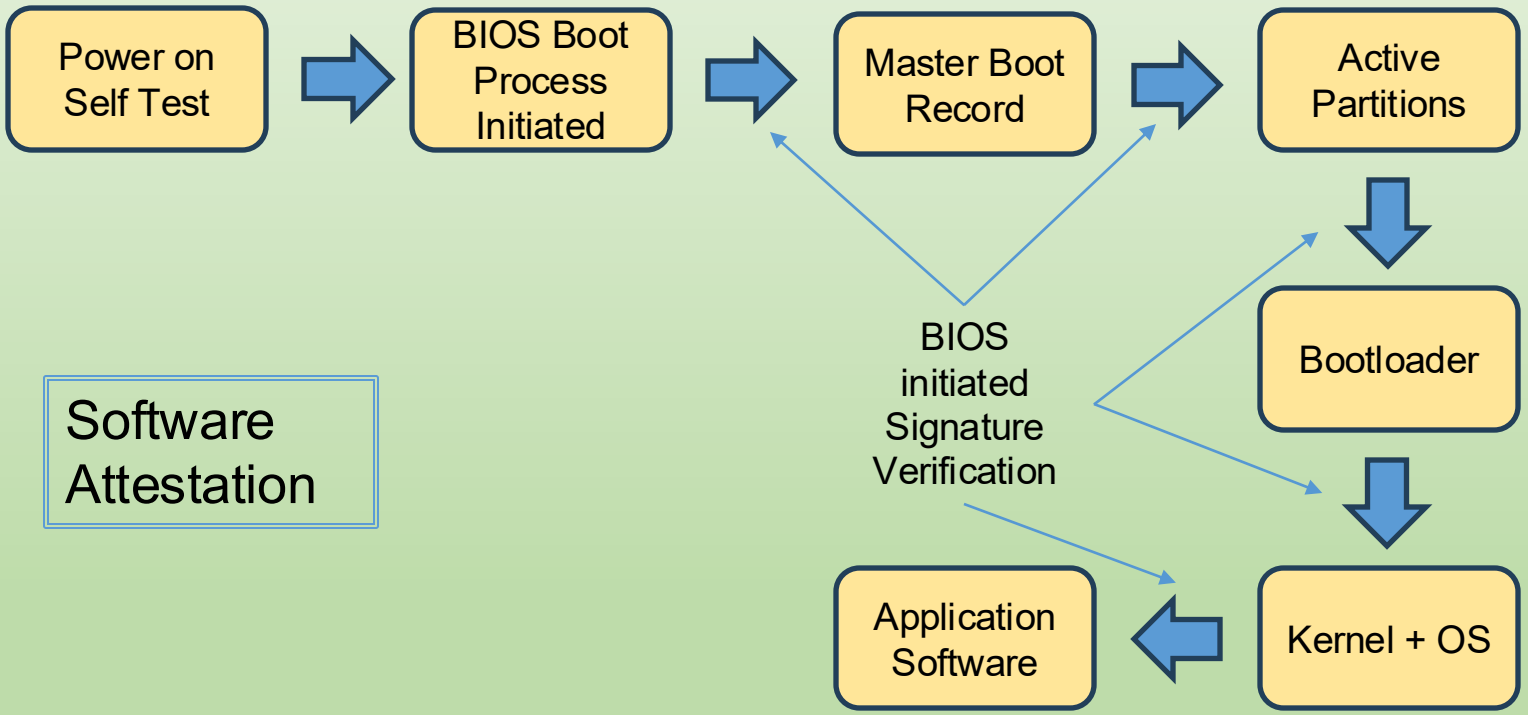
Background – Conventional Device Boot Process (Simplified Version)



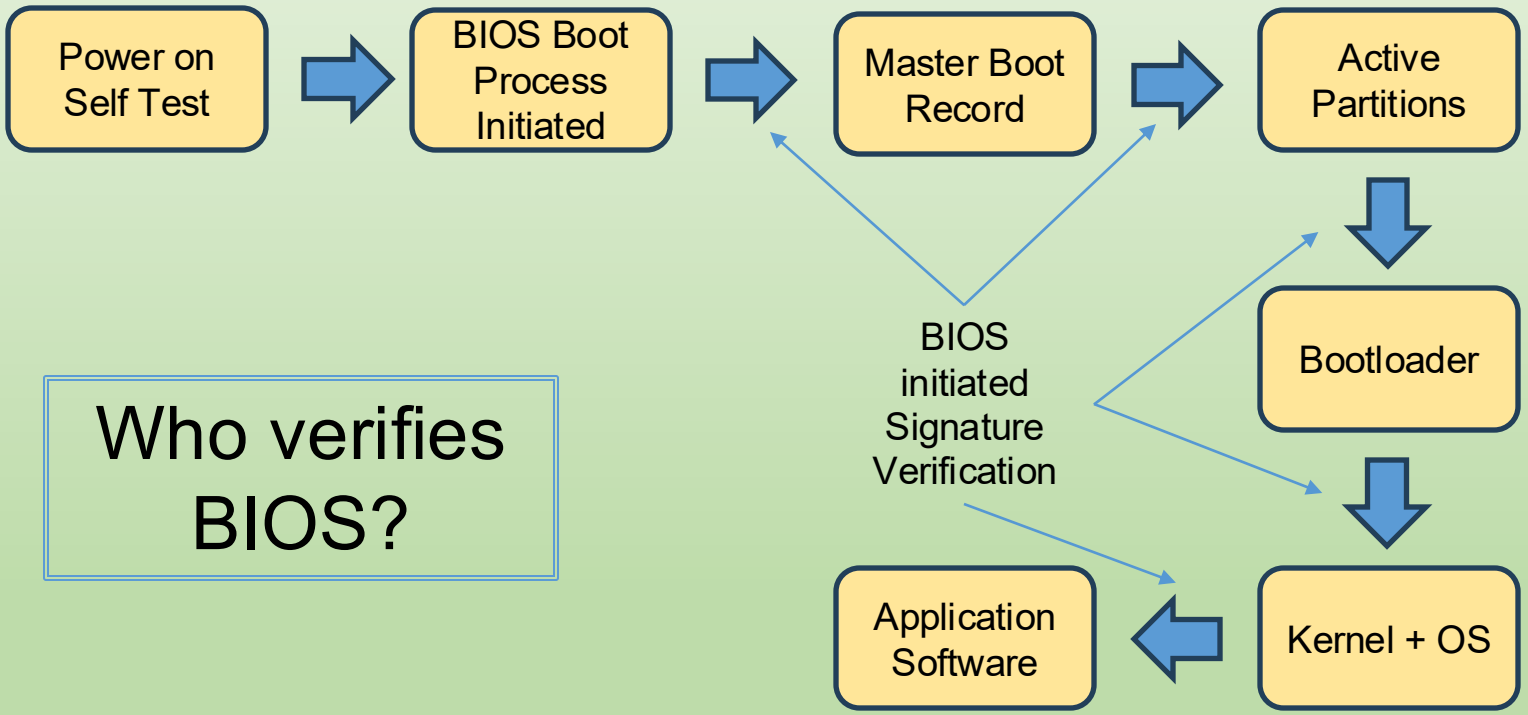
Background – Secure Boot Process with Code Signing



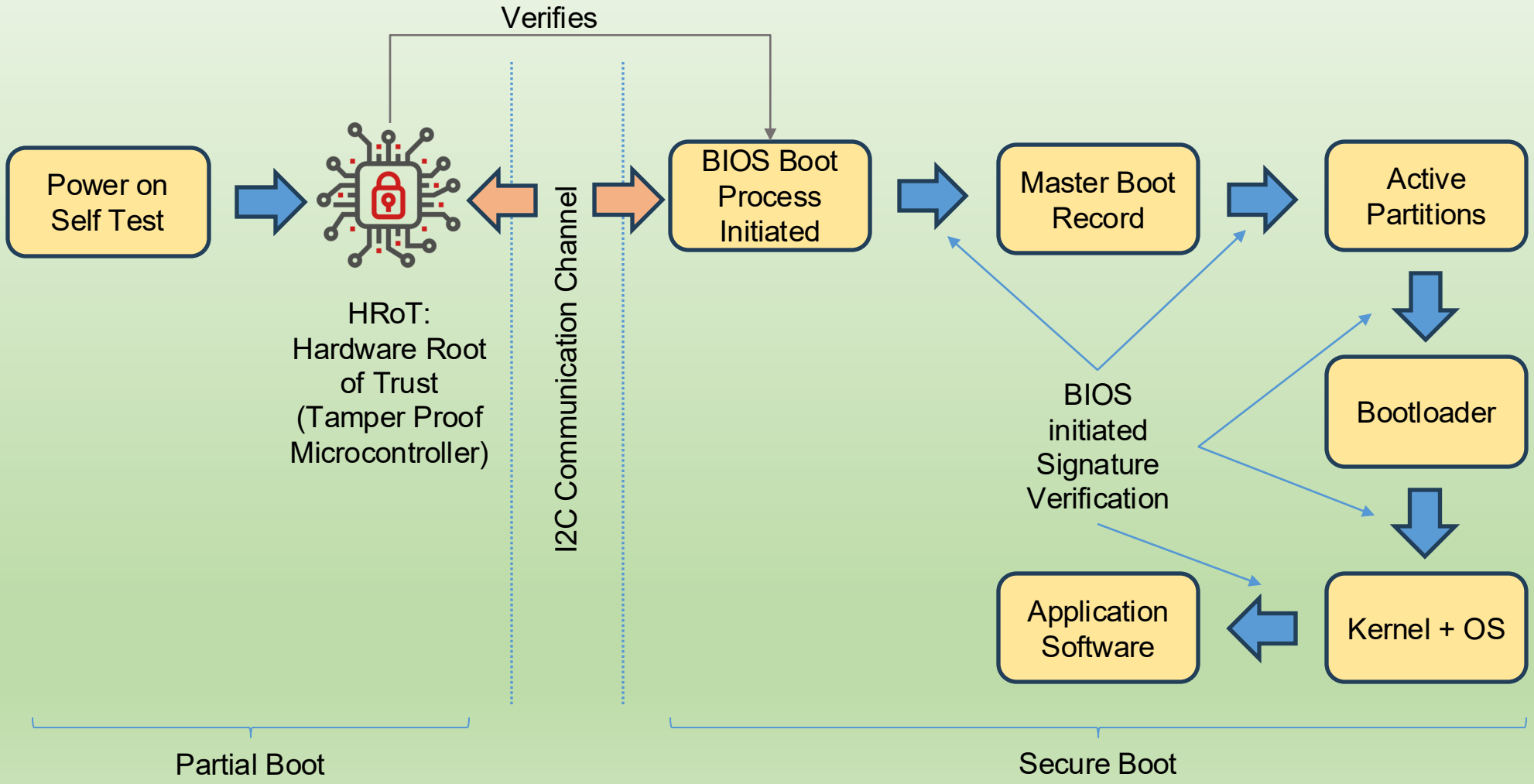
Background – Secure Boot Process with Code Signing



Background – Secure Boot Process with Code Signing

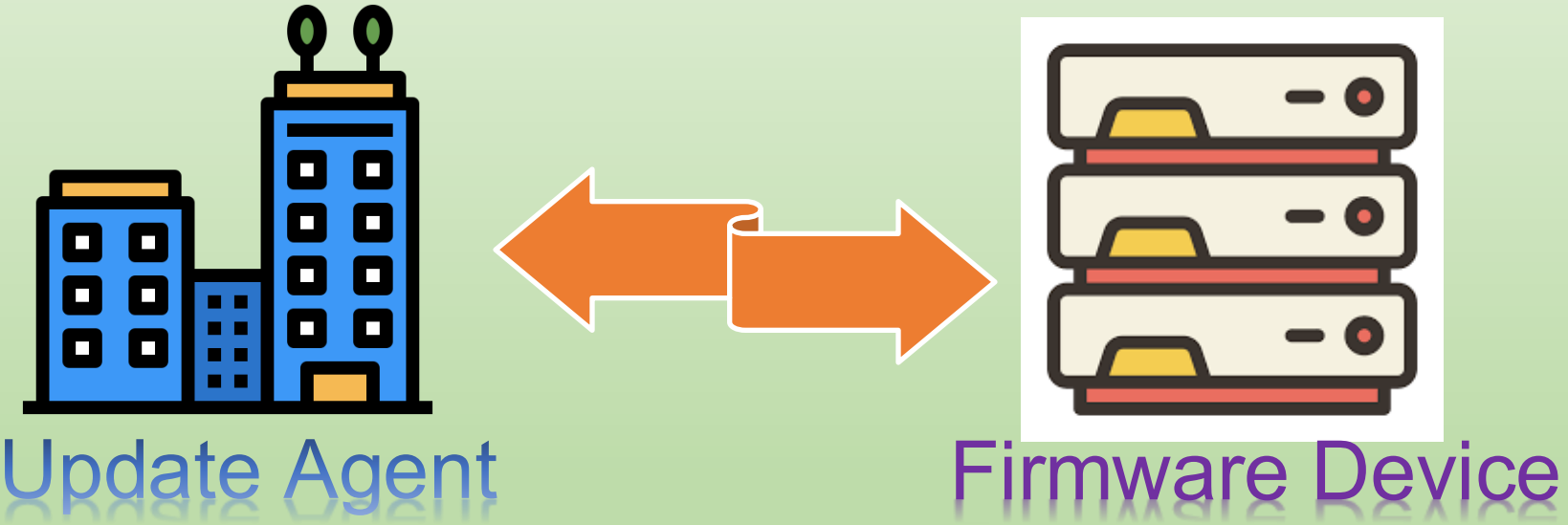


Background – Trusted Boot in PIT-Cerberus Protocol*



* Rakesh Podder et al, "The PIT-Cerberus Framework: Preventing Device Tampering During Transit," IEEE QRS 2024

Motivation – BIOS / Firmware Update



Motivation - BIOS / Firmware Update

Rollback Protection Scheme

Dual-bank Update Scheme

Delta Update Scheme

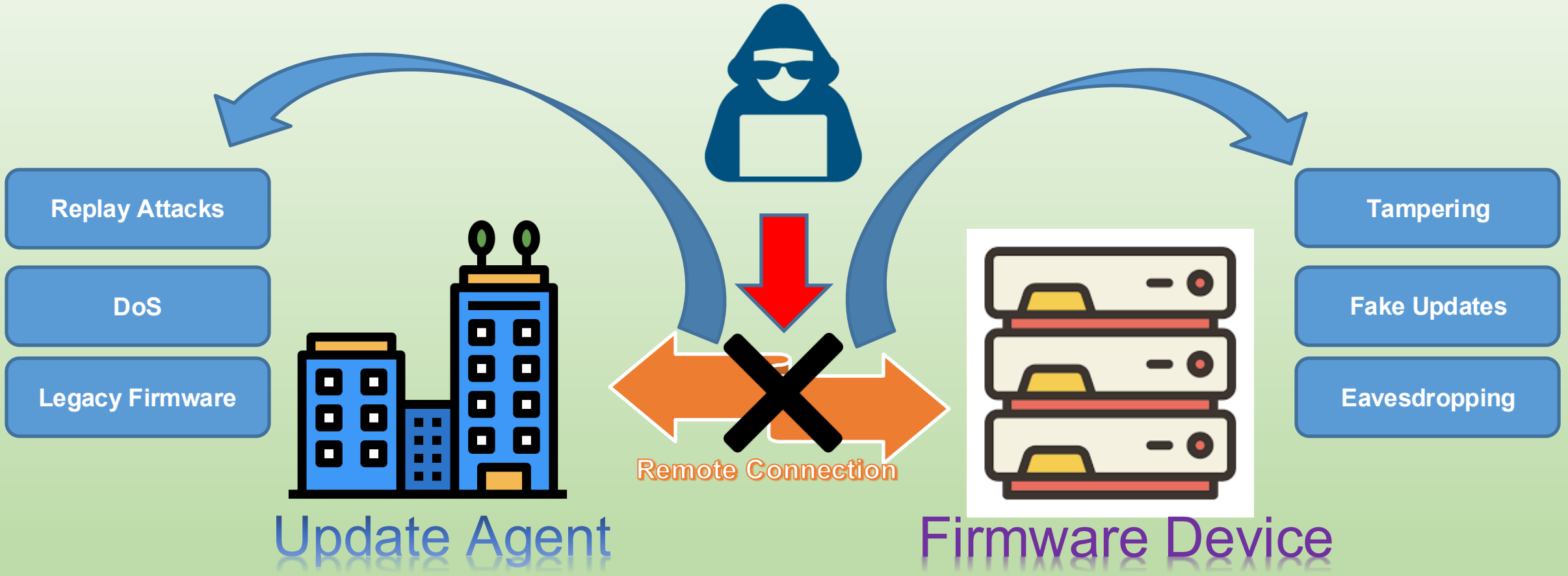
Over-The-Air (OTA)

Update Agent

Remote Connection

Firmware Device

Motivation - BIOS / Firmware Update



Ensuring secure remote firmware updates is pivotal to safeguarding devices from malicious intrusions, maintaining device functionality, and protecting sensitive data.

Problem Statement

- Insecure firmware updates risk vulnerabilities and device malfunctions.
- Traditional OTA update mechanisms can be insecure.
- Diverse update protocols can create compatibility issues.

Design secure firmware update protocol that is device independent and is standardizable

Benefits and Relevance

- Protecting device ecosystems from evolving cybersecurity threats and ensuring uninterrupted device performance
- As industries increasingly rely on interconnected devices, ensuring secure firmware updates becomes paramount to maintaining operational integrity, protecting intellectual property, and complying with regulatory standards
- Modern devices are an amalgamation of components from various suppliers. Using standardized and secure update mechanisms ensures consistency and security across the board

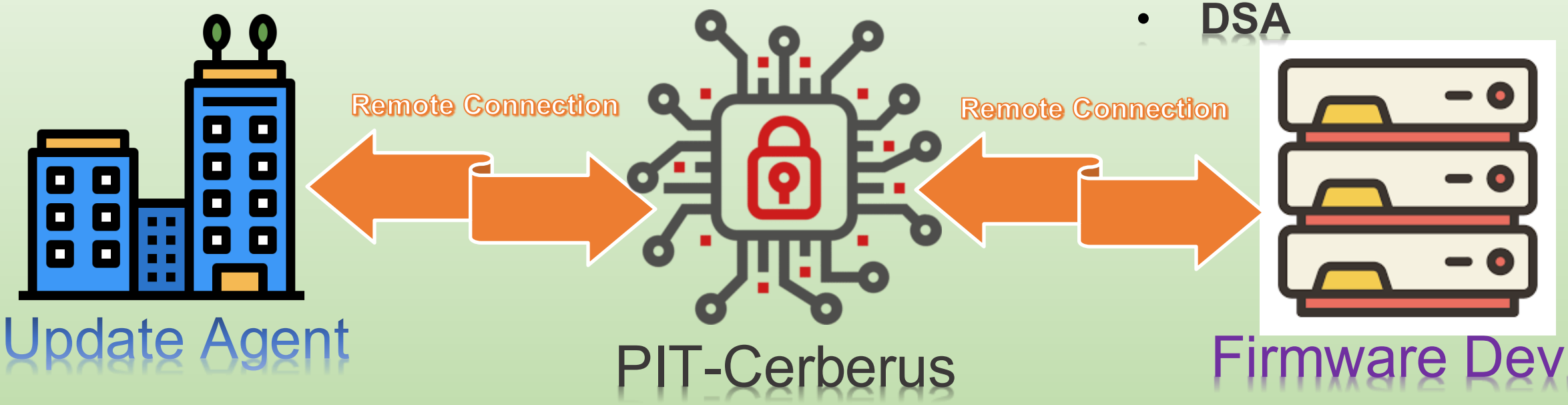
Solution – S-RFUP

Standardization

- PLDM
- MCTP

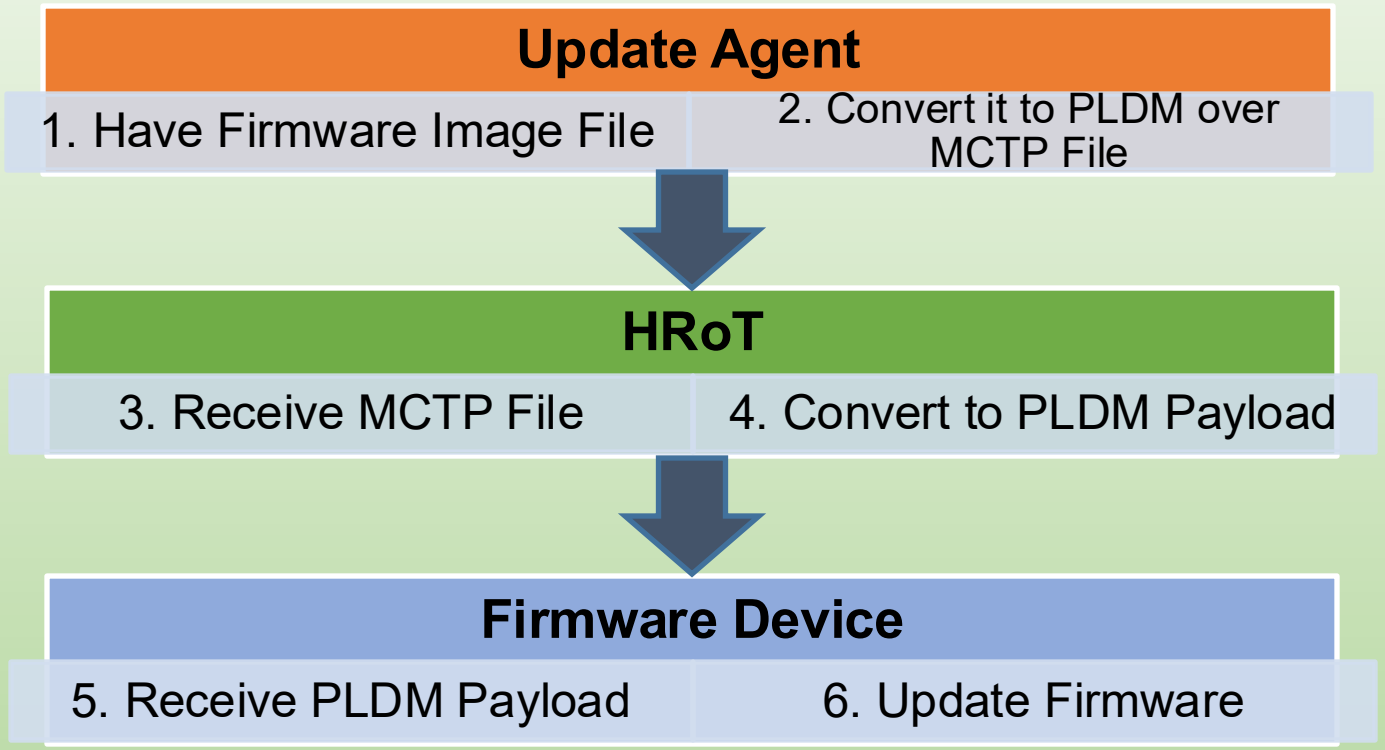
Security

- AES-256
- ECDH
- DSA



Establish an end-to-end secure pipeline for remote firmware updates

Hi-Level Specification



Platform Level Data Model (PLDM) for Firmware Update Specification

Platform Level Data Model (PLDM) over MCTP for Remote Update

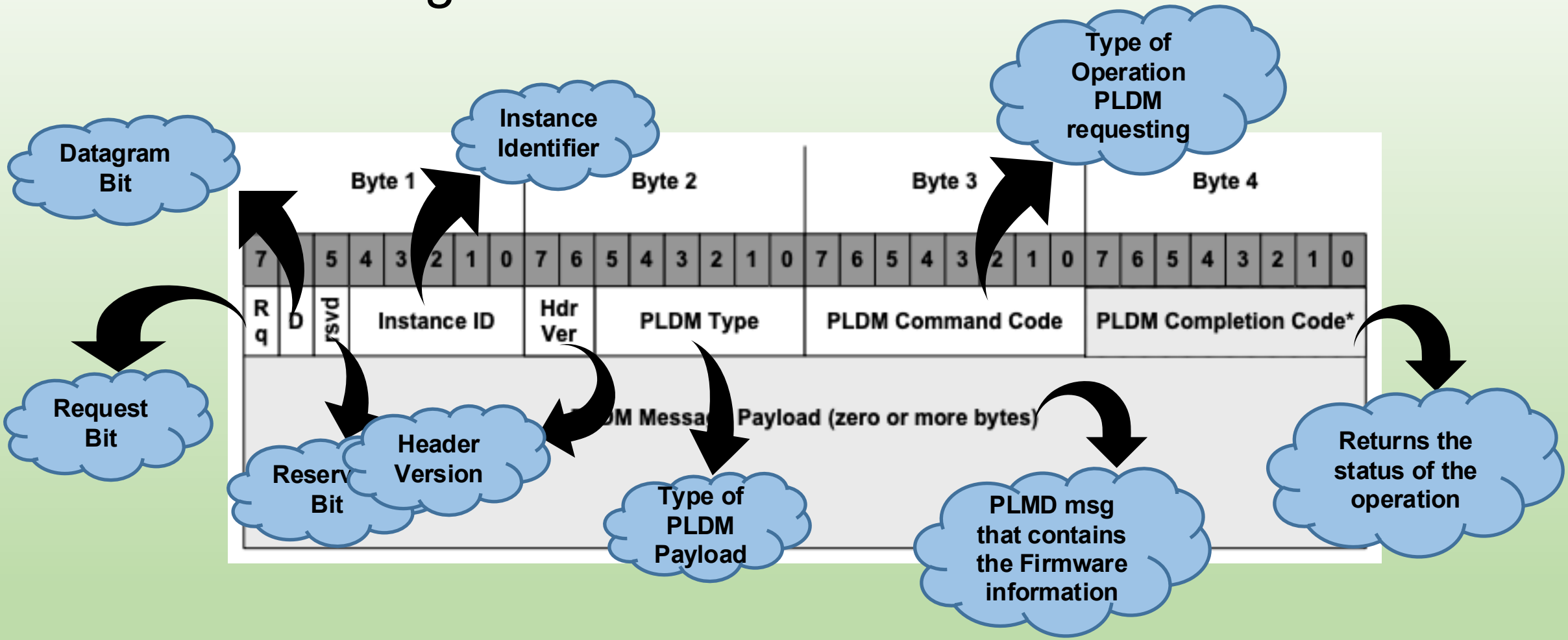
Project Cerberus

- Hardware root of trust for server platforms
- Provides functionality to enforce secure boot for firmware
- Provides mechanism to securely attest to the state of the device firmware

Platform Level Data Model (PLDM)

- Standardized framework for system management
- Enables structured control of hardware components
- Used in data centers and servers
- Binary encoding for efficient data transmission
- Supports hardware queries, configuration, and diagnostics
- Extensible for custom management tasks
- Integrates with management tools

PLDM Message Format



Reference: Platform Level Data Model (PLDM) Base Specification DMTF

MCTP (Management Component Transport Protocol)

- Industry-standard transport protocol for platform management
- Designed for communication between management controllers and components
- Enables remote management and monitoring of platform hardware
- Supports various transport layers (e.g., SMBus, PCIe, etc.)
- Provides a framework for secure communication
- Enables efficient and standardized hardware management

Platform Level Data Model (PLDM) over MCTP Binding Specification

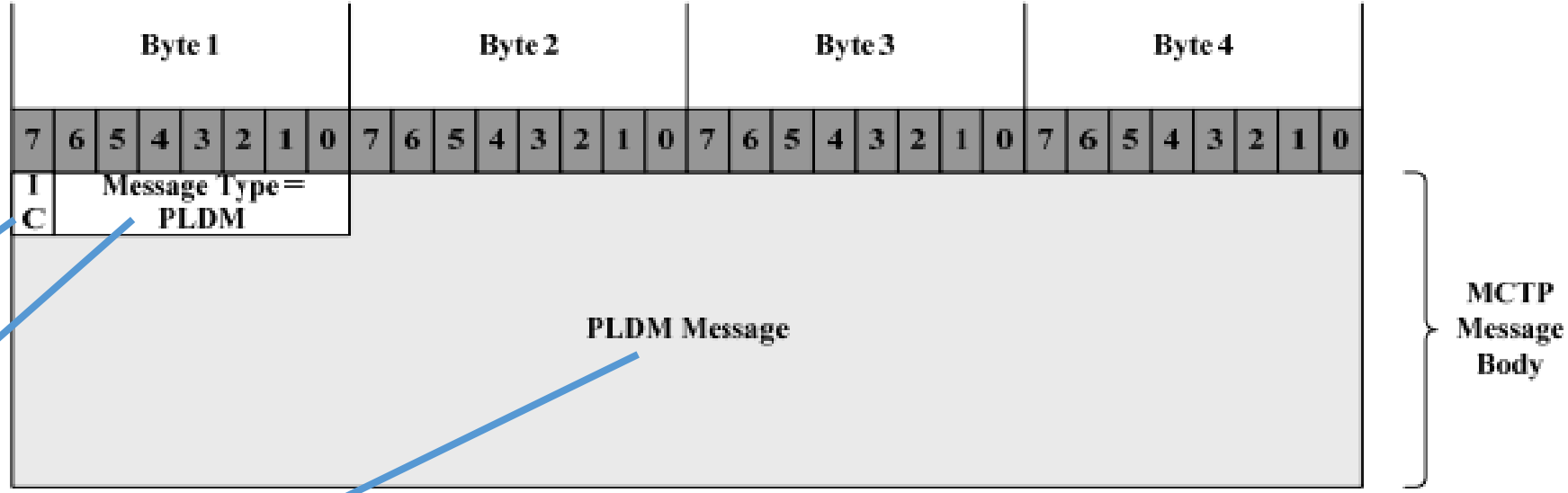


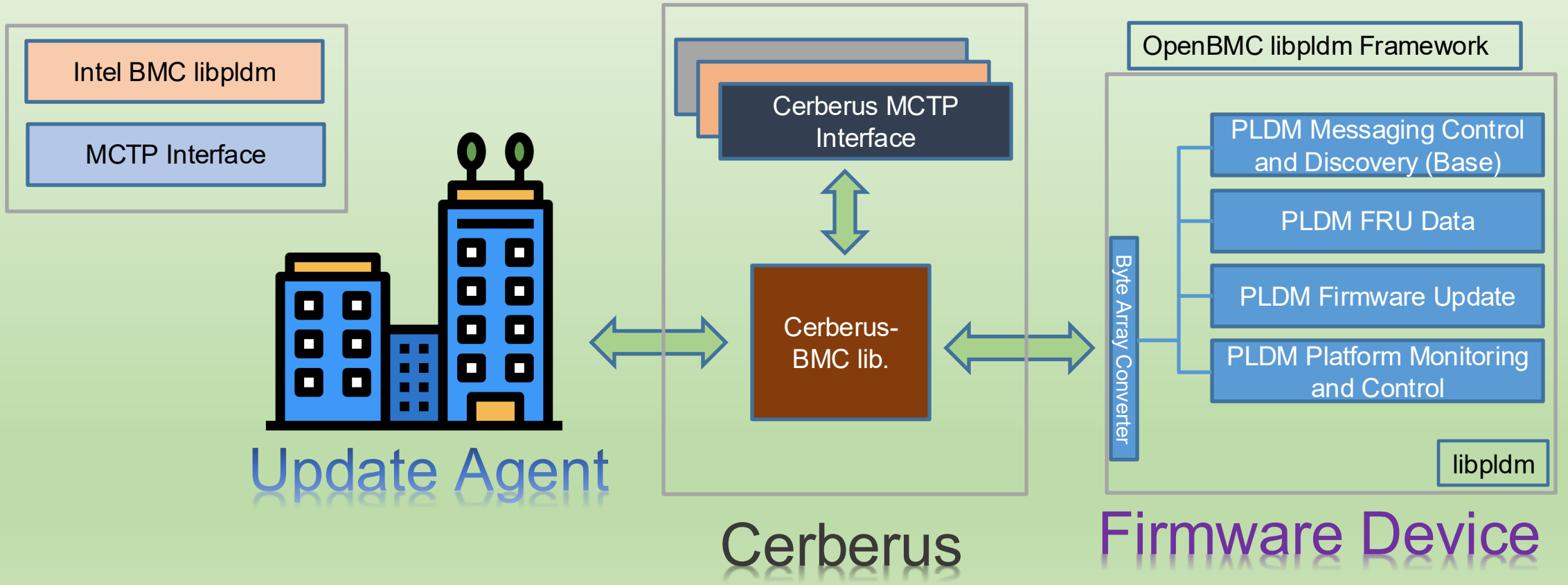
Figure 1 – PLDM over MCTP Message Fields

Table 1 – PLDM over MCTP Message Field Descriptions

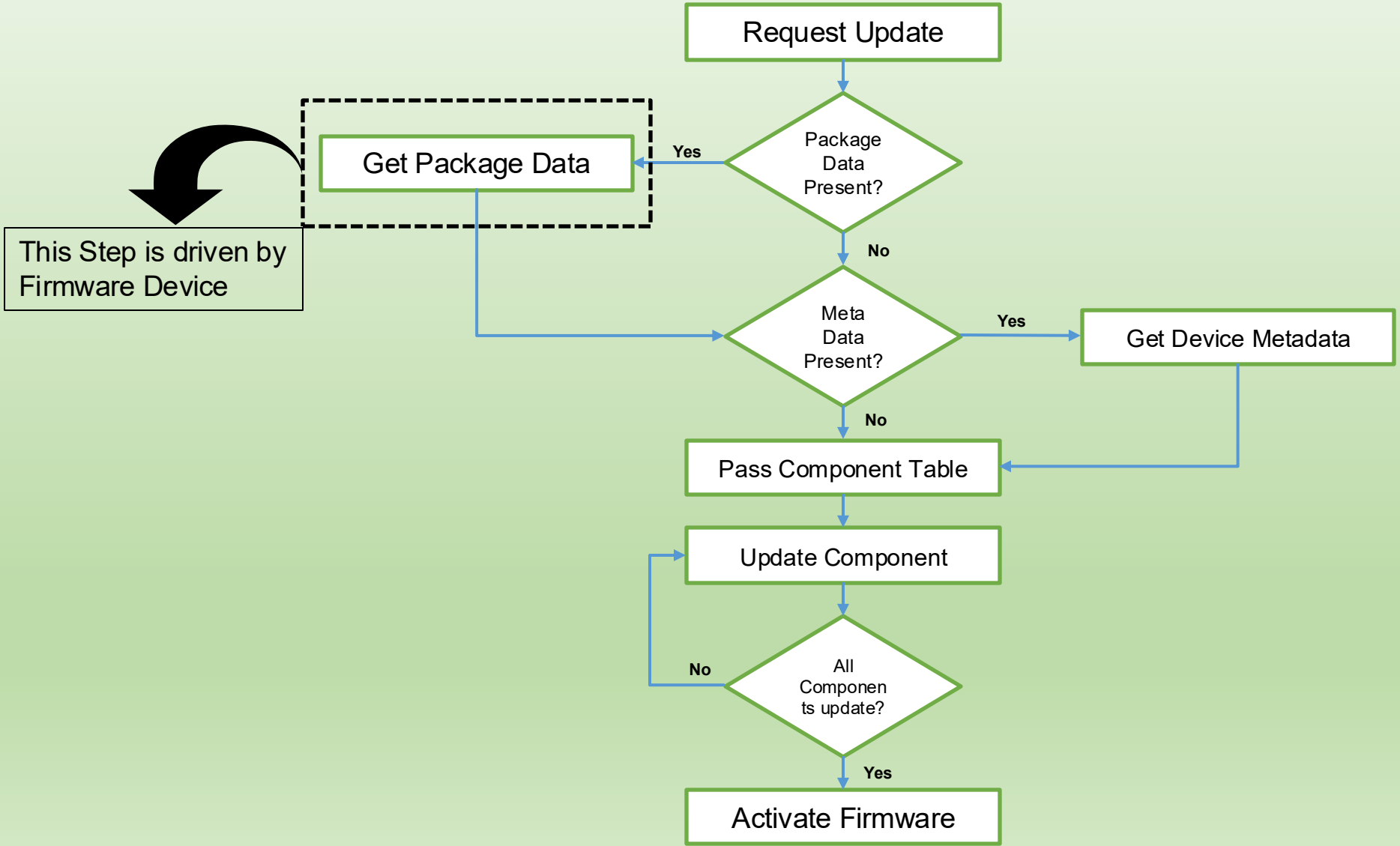
Field Name	Field Size	Description
IC	1 bit	Message Integrity Check bit = 0b PLDM over MCTP messages do not include an overall Message Integrity check field.
Message Type	7 bits	PLDM = 0x01 (000_0001b) This field identifies the MCTP message as carrying a PLDM message.
PLDM Message	Variable	The base PLDM message fields are defined in DSP0240.

Reference: Platform Level Data Model (PLDM) over MCTP Binding Specification DMTF

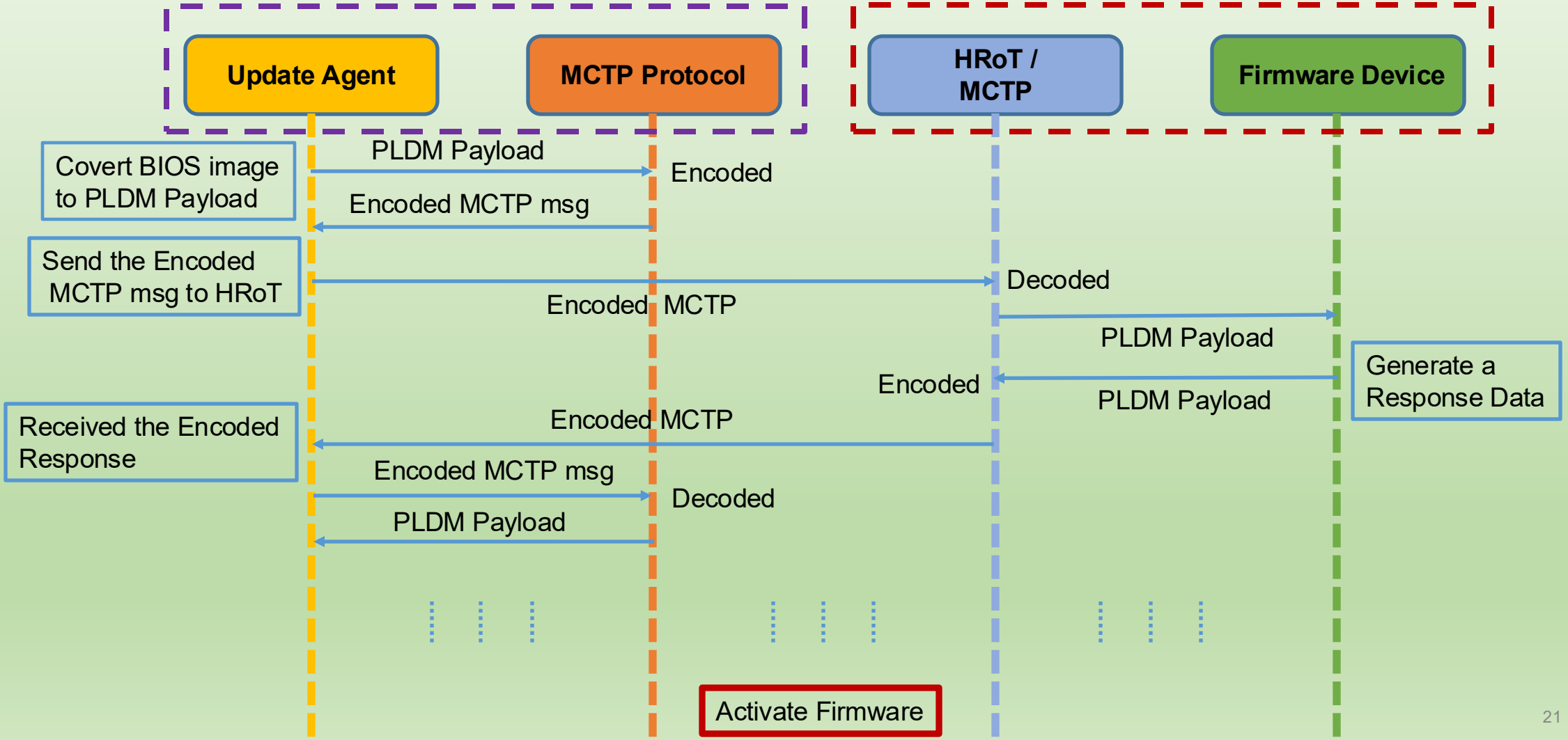
Libraries and Framework



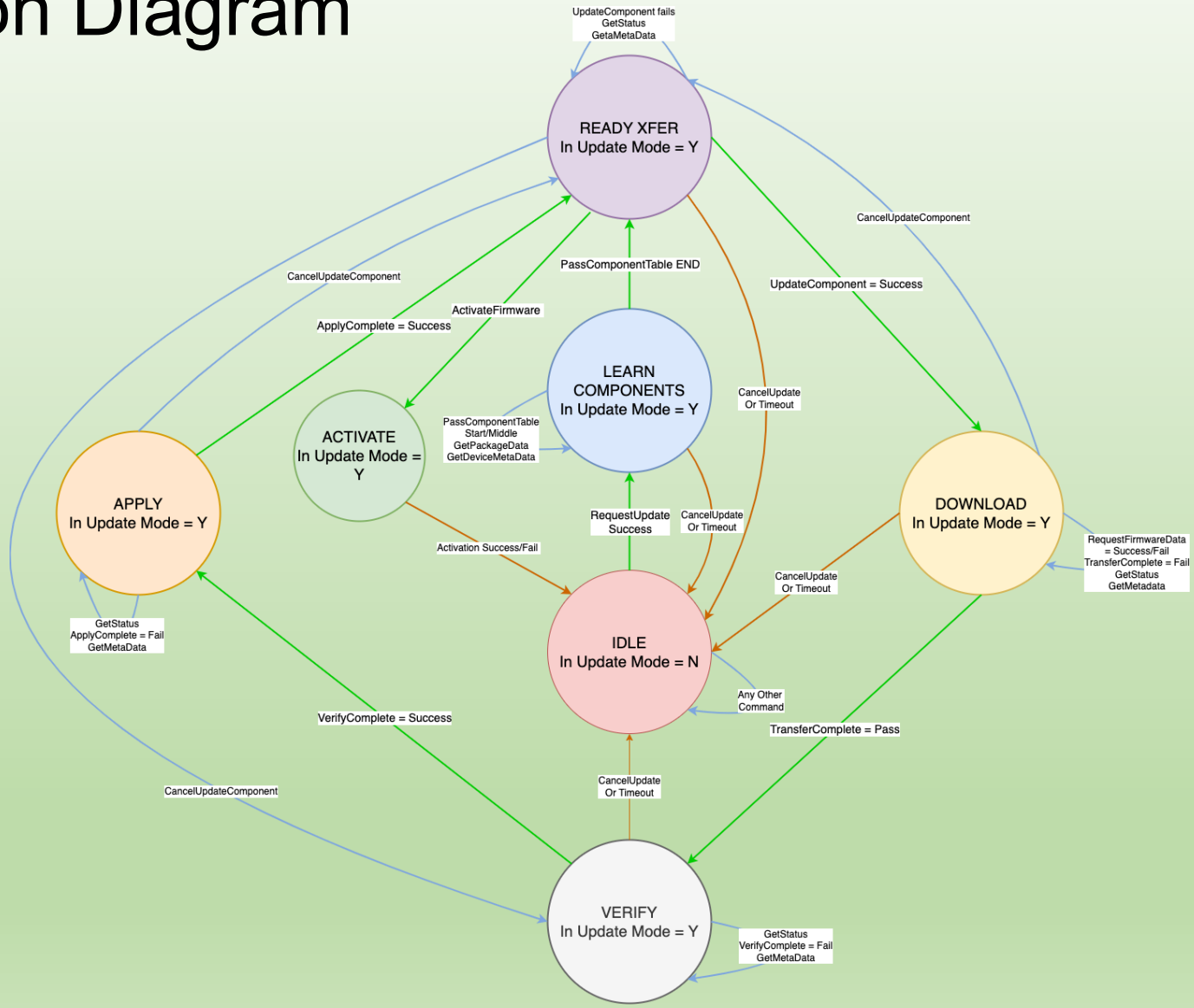
High Level Firmware Update Flow



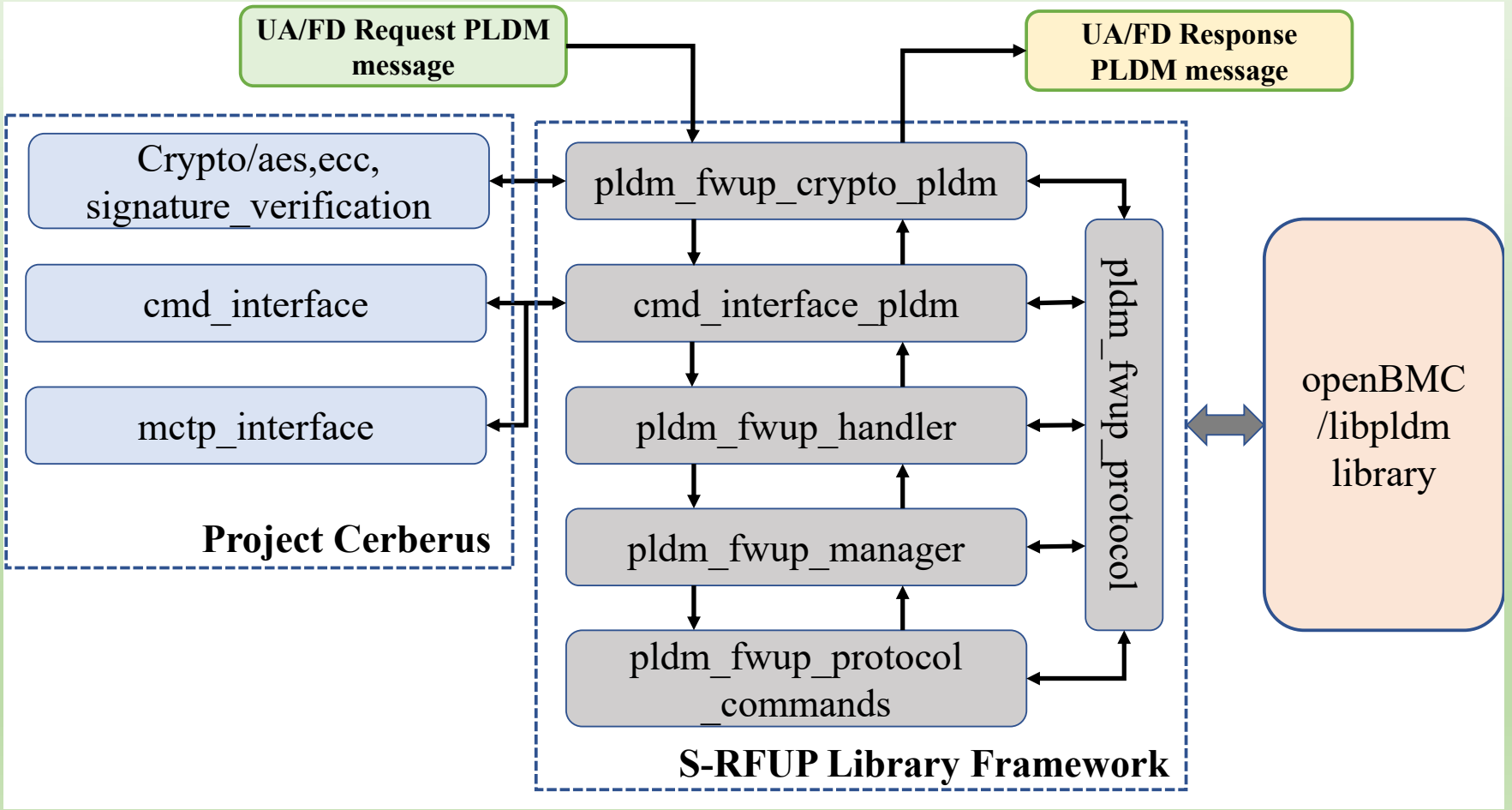
S-RFUP Protocol



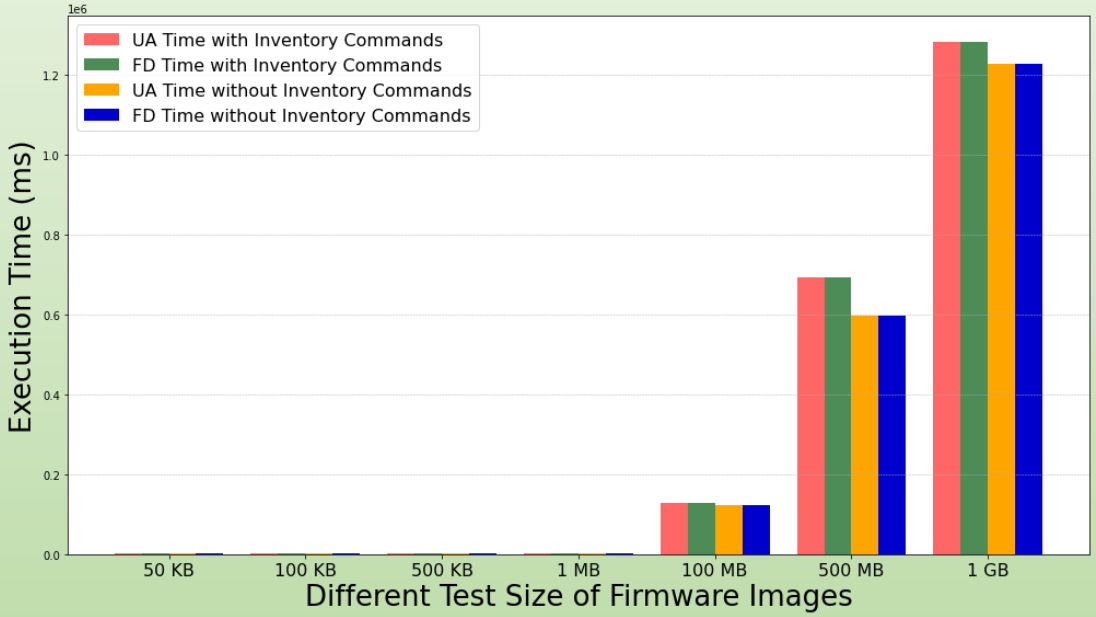
S-RFUP State Transition Diagram



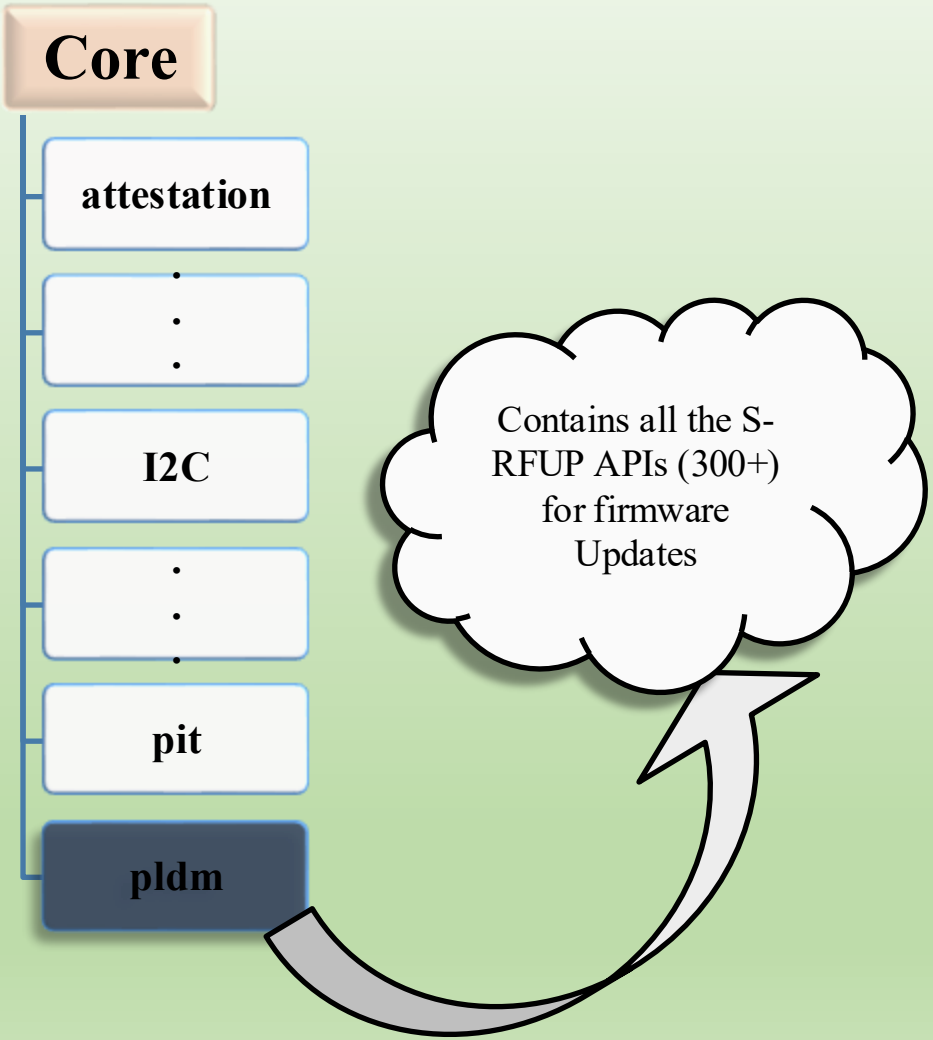
Libraries and Framework



Evaluation



UA Time vs FD Time for Firmware Update Tests (with and without Inventory Commands).



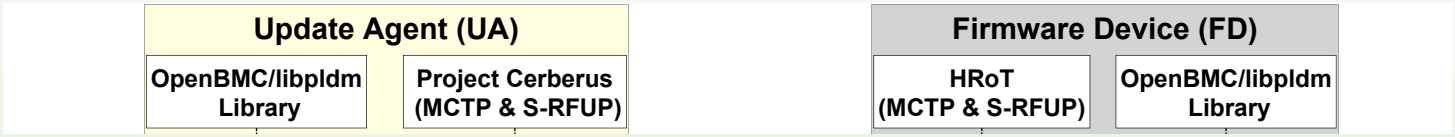
Conclusion & Future Work

- Have a complete and secure Firmware Update Protocol that adheres to industry standards.
- Incorporated protocol into Microsoft Cerberus Framework.
- Ongoing testing and optimizing the proposed protocol.
- Formal Verification of S-RFUP protocol

Thank You

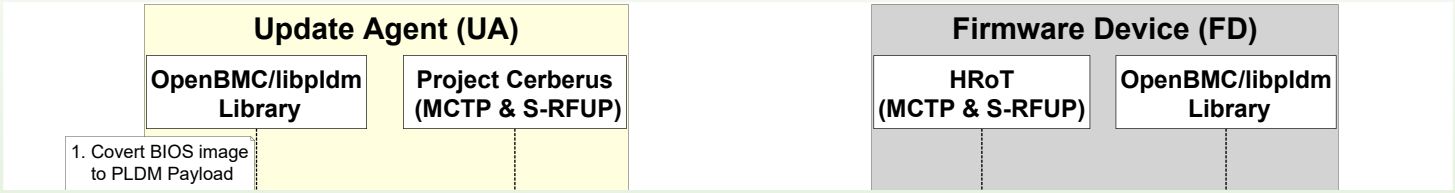
Questions?

S-RFUP FIRMWARE UPDATE PROCESS

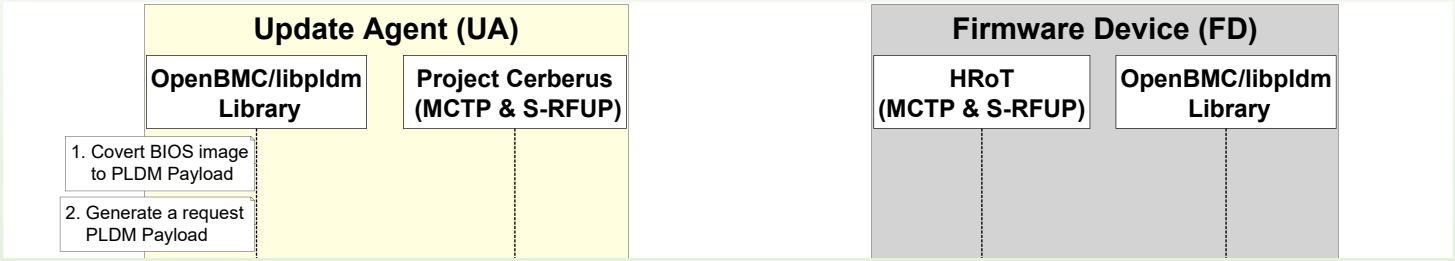


S-RFUP FIRMWARE UPDATE PROCESS

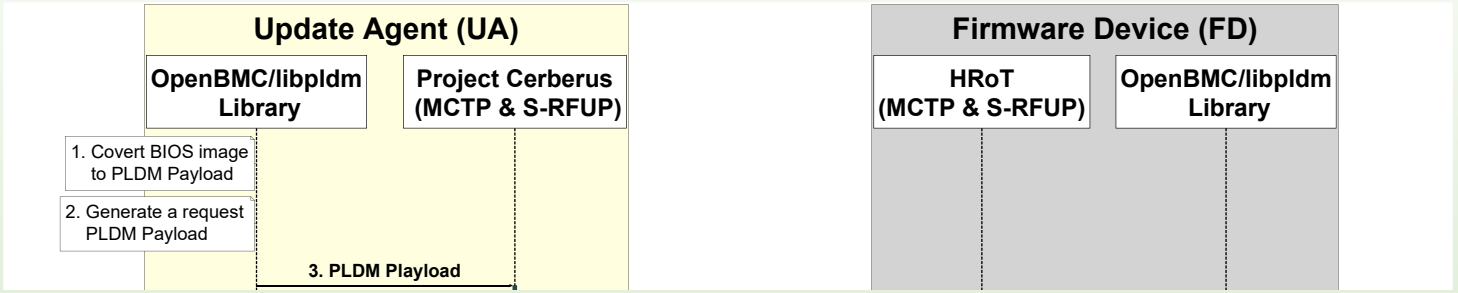
S-RFUP FIRMWARE UPDATE PROCESS



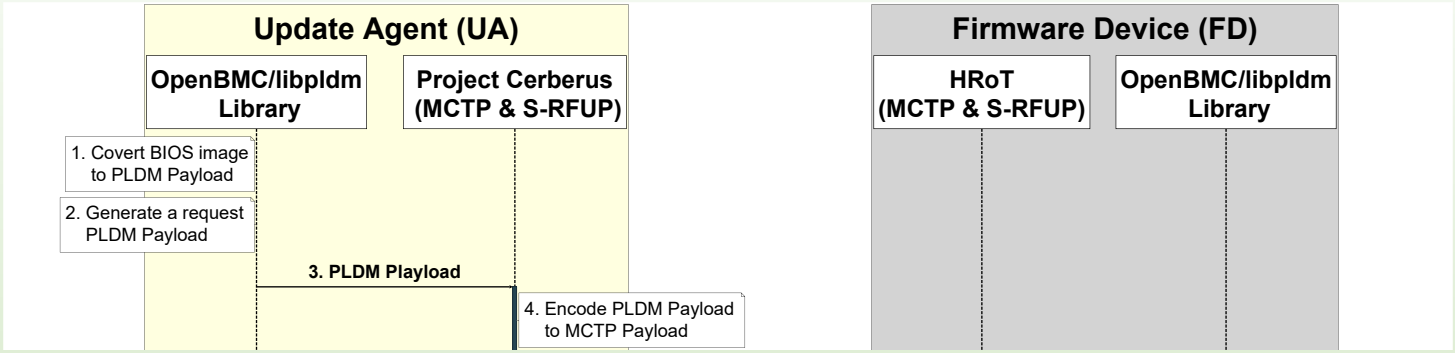
S-RFUP FIRMWARE UPDATE PROCESS



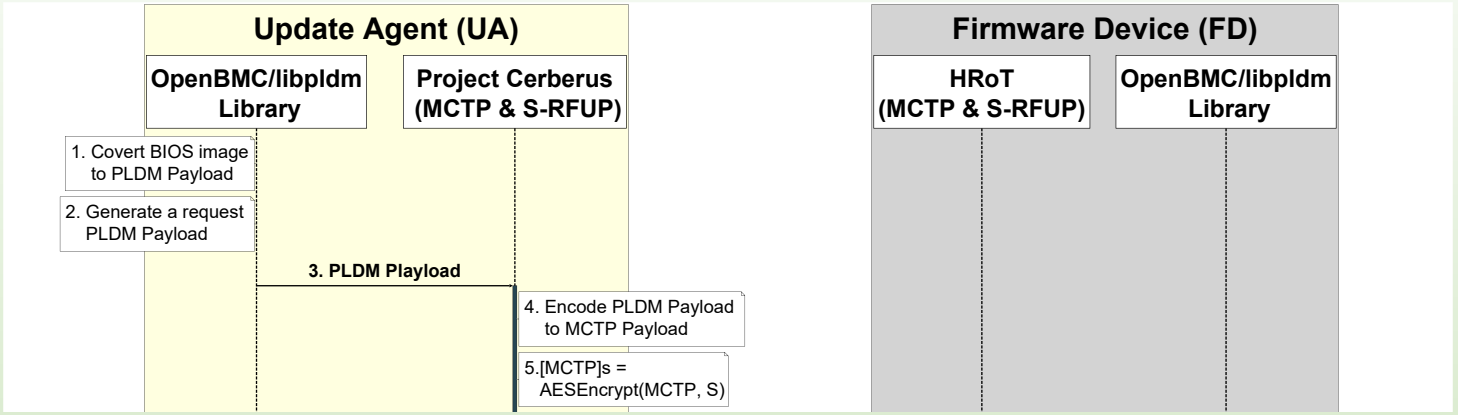
S-RFUP FIRMWARE UPDATE PROCESS



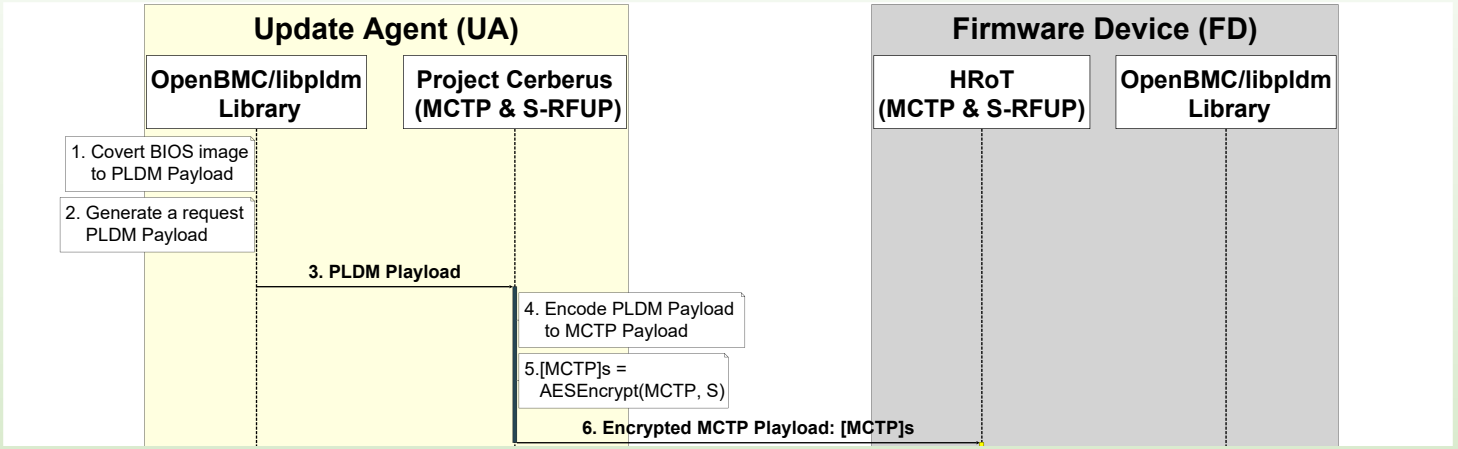
S-RFUP FIRMWARE UPDATE PROCESS



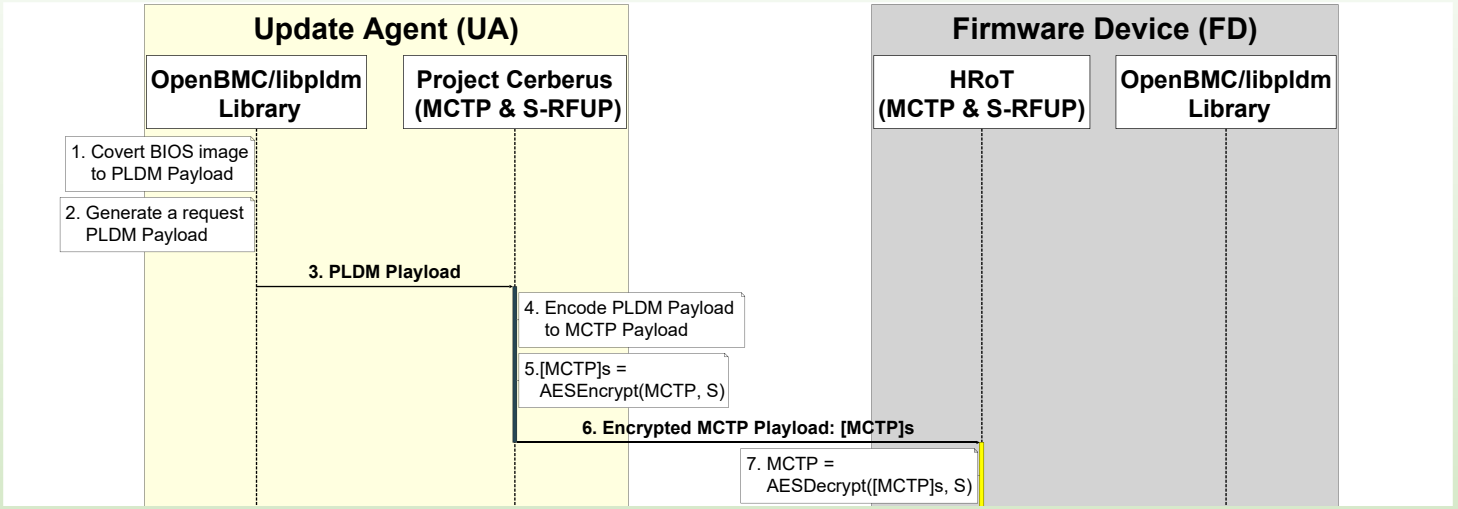
S-RFUP FIRMWARE UPDATE PROCESS



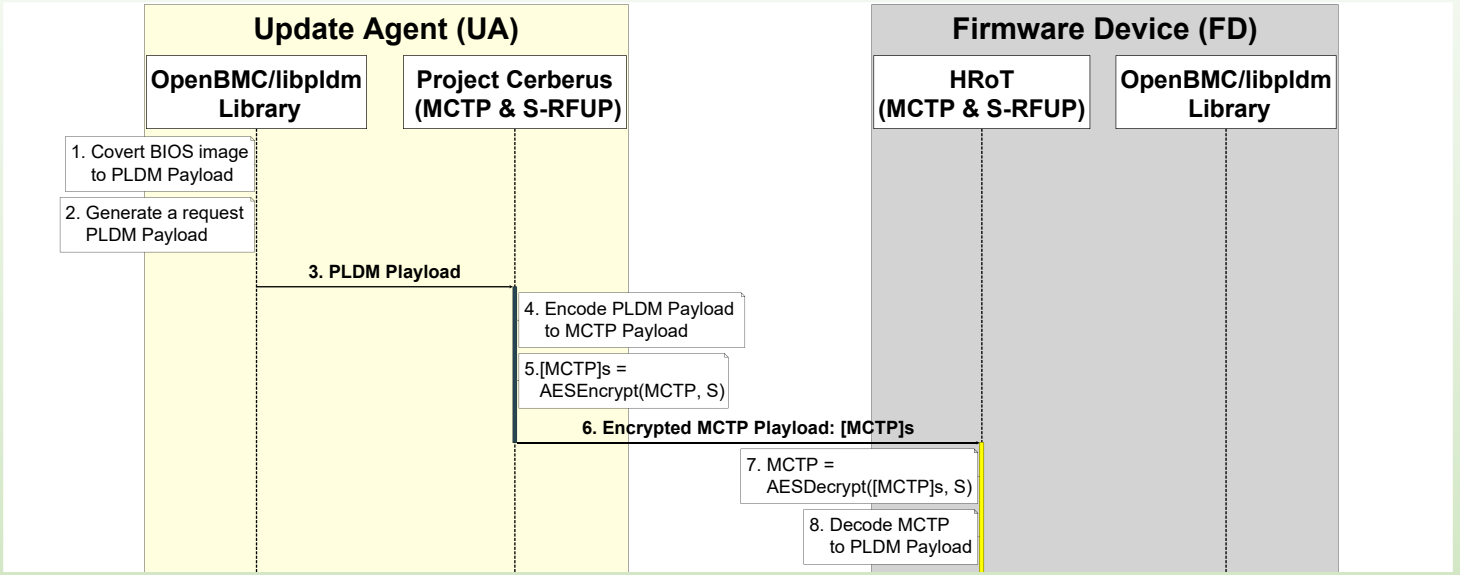
S-RFUP FIRMWARE UPDATE PROCESS



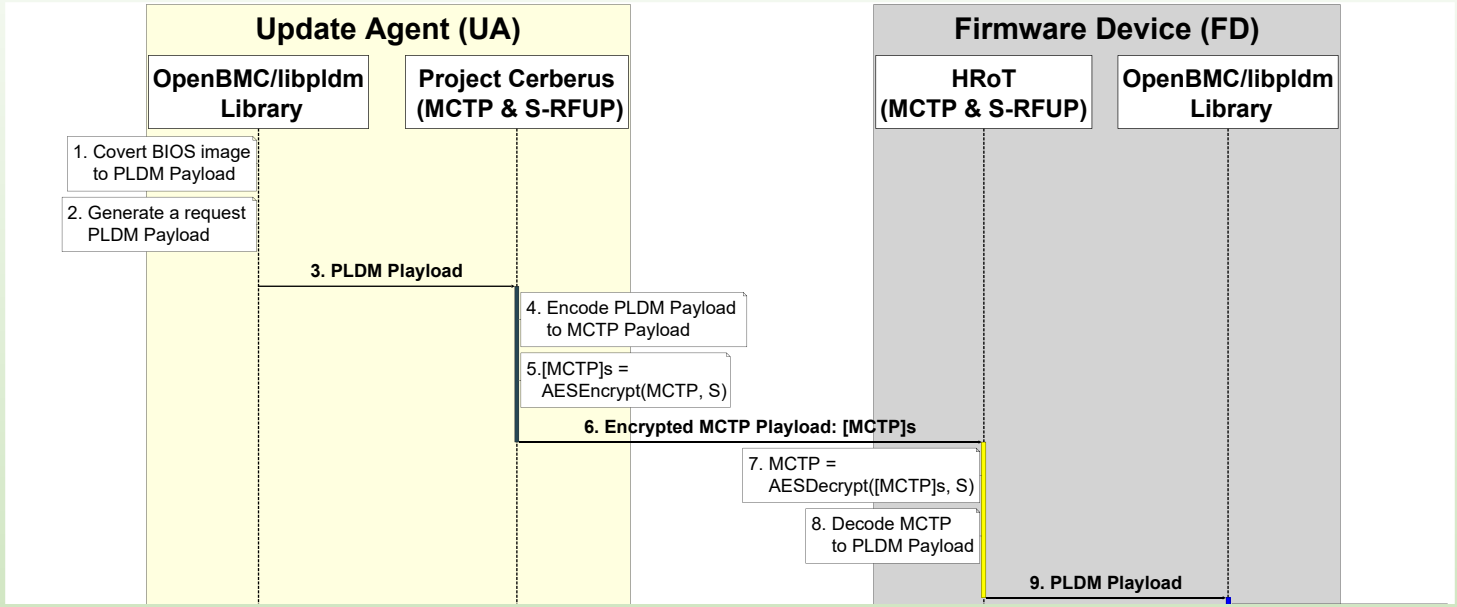
S-RFUP FIRMWARE UPDATE PROCESS



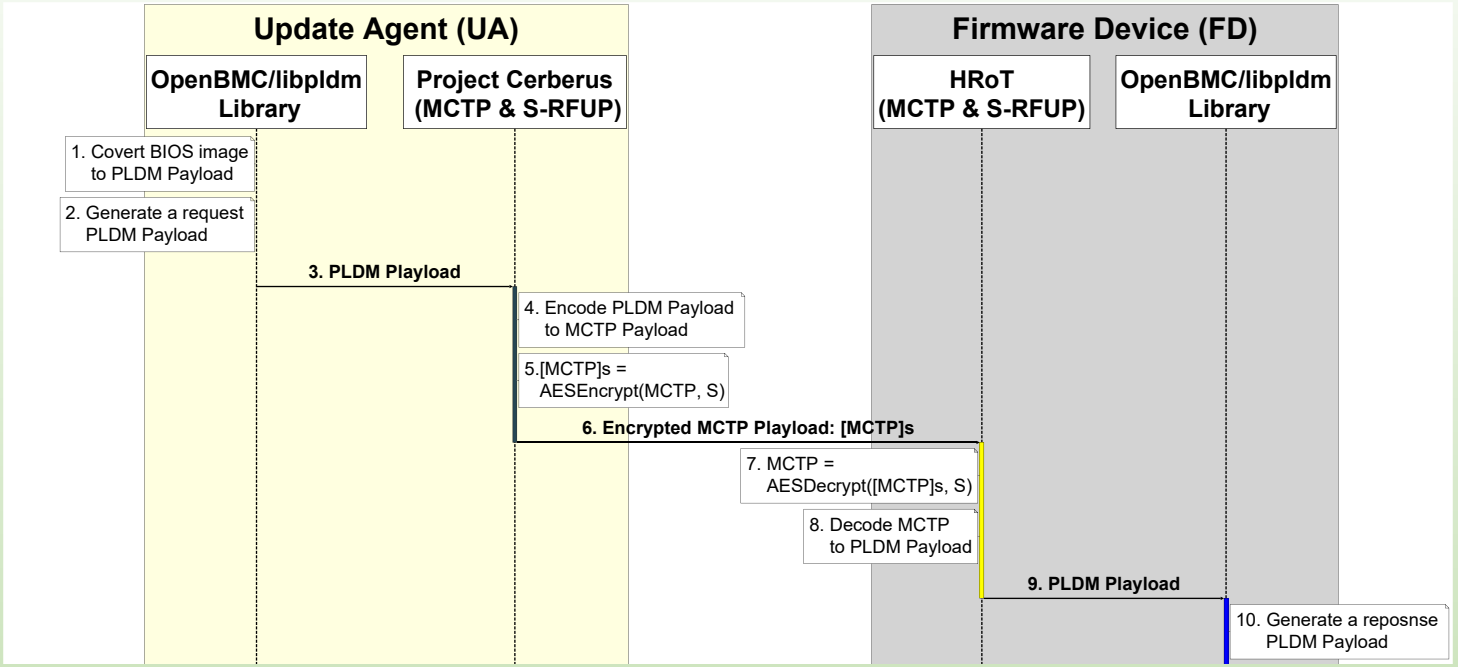
S-RFUP FIRMWARE UPDATE PROCESS



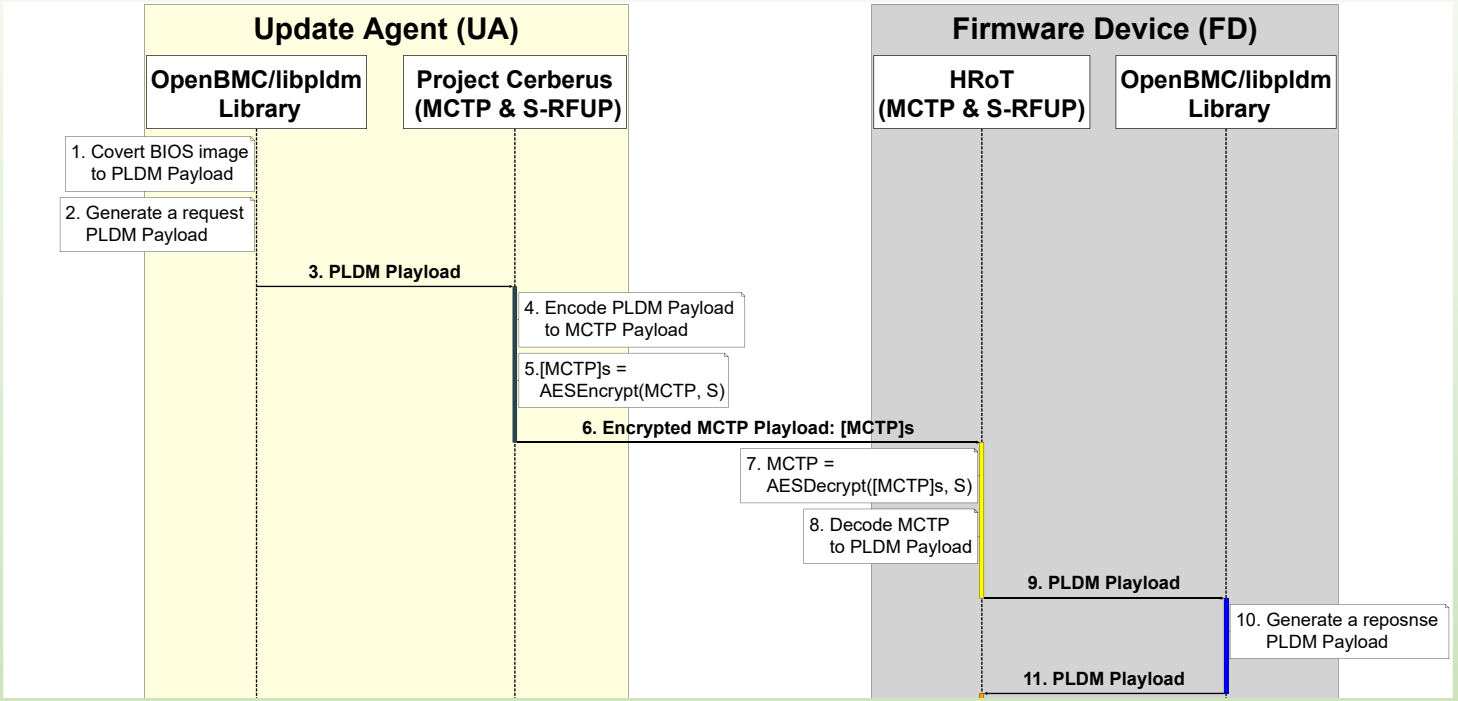
S-RFUP FIRMWARE UPDATE PROCESS



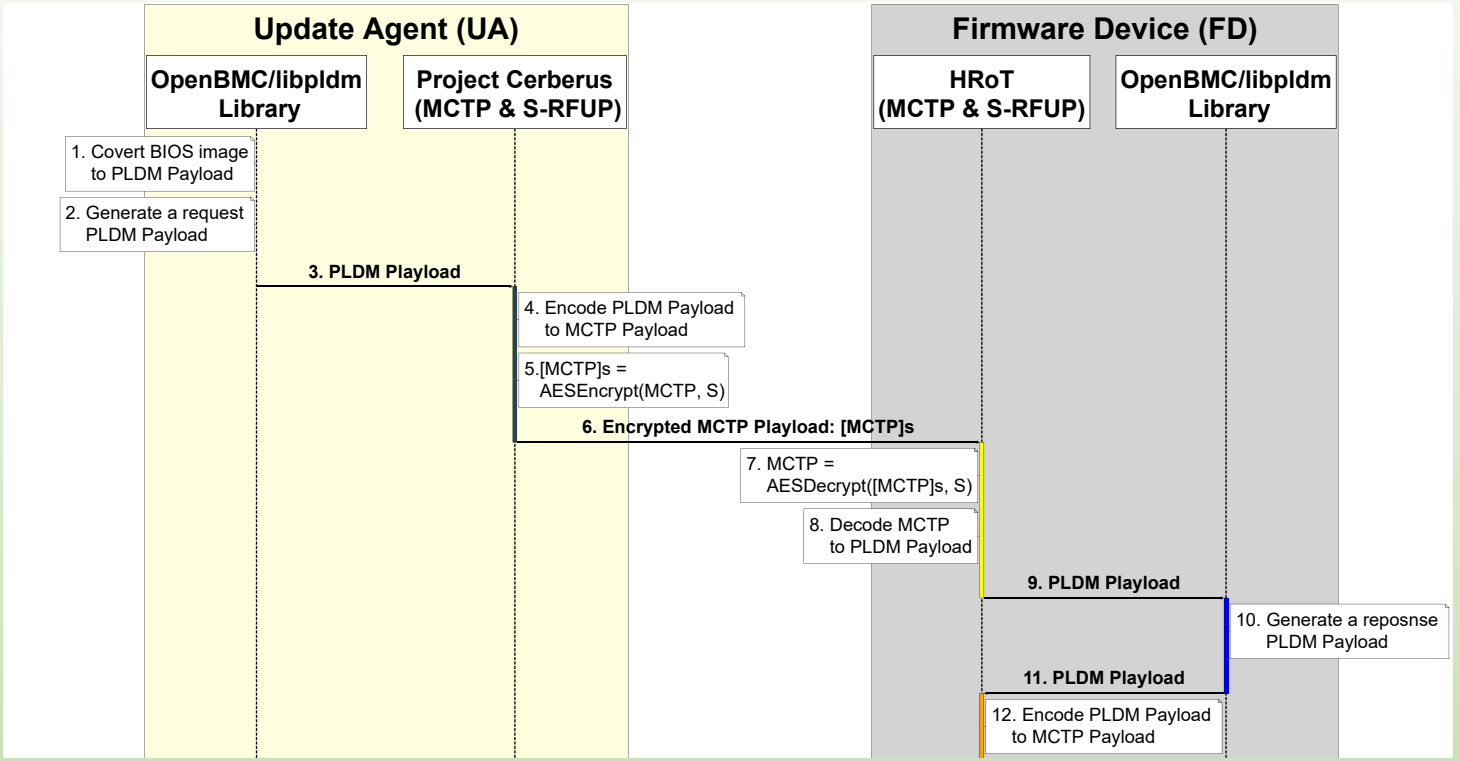
S-RFUP FIRMWARE UPDATE PROCESS



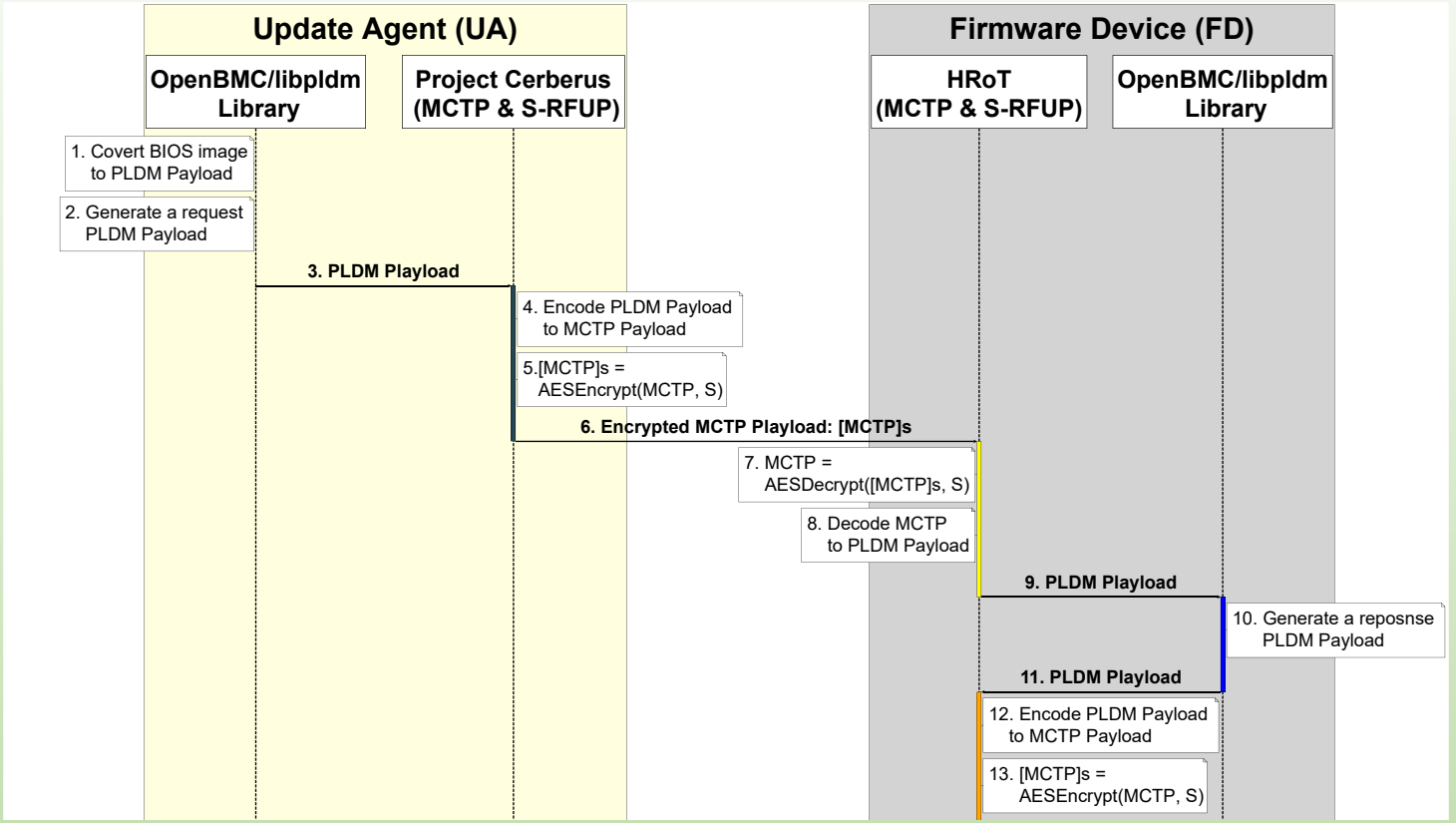
S-RFUP FIRMWARE UPDATE PROCESS



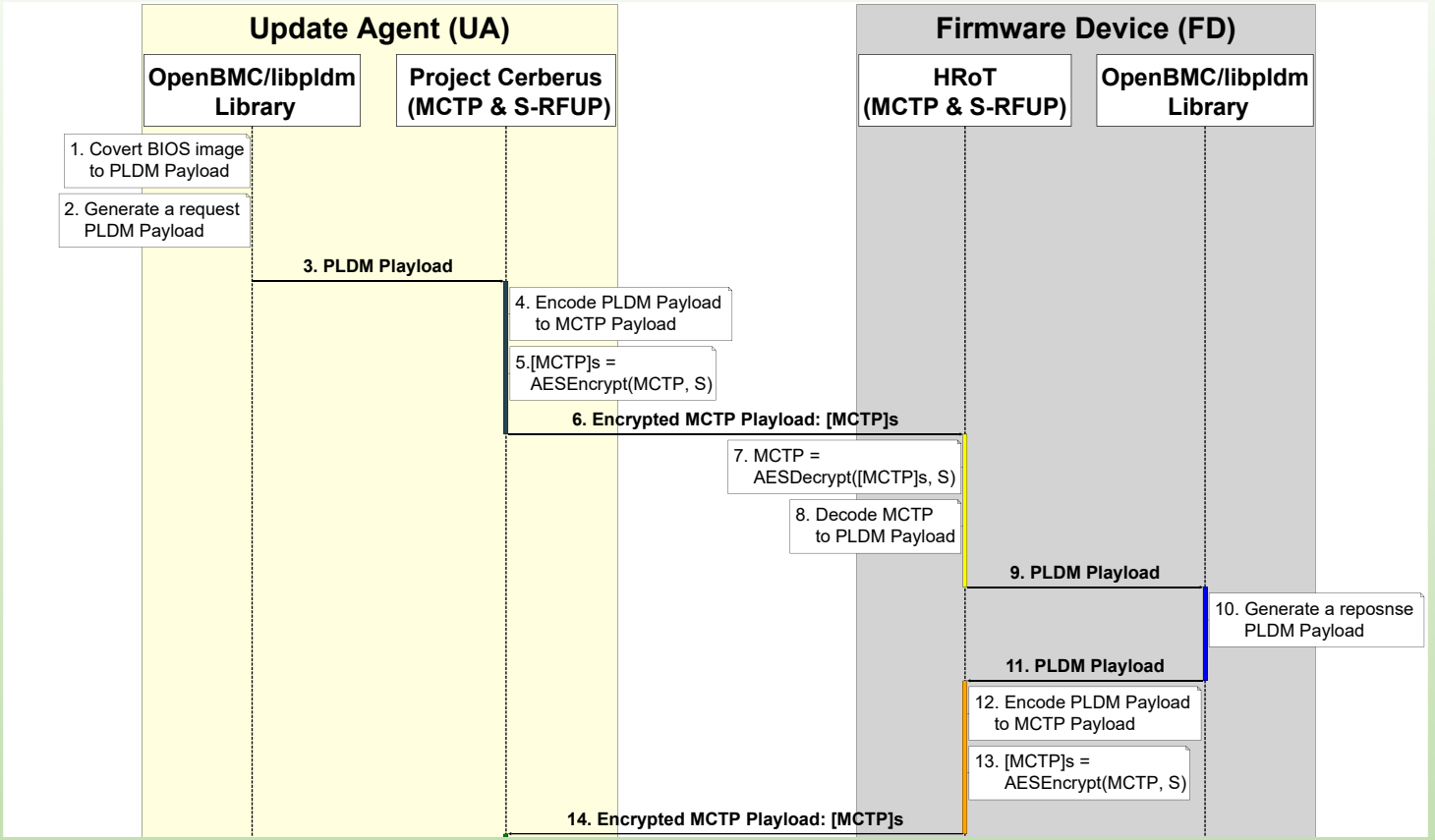
S-RFUP FIRMWARE UPDATE PROCESS



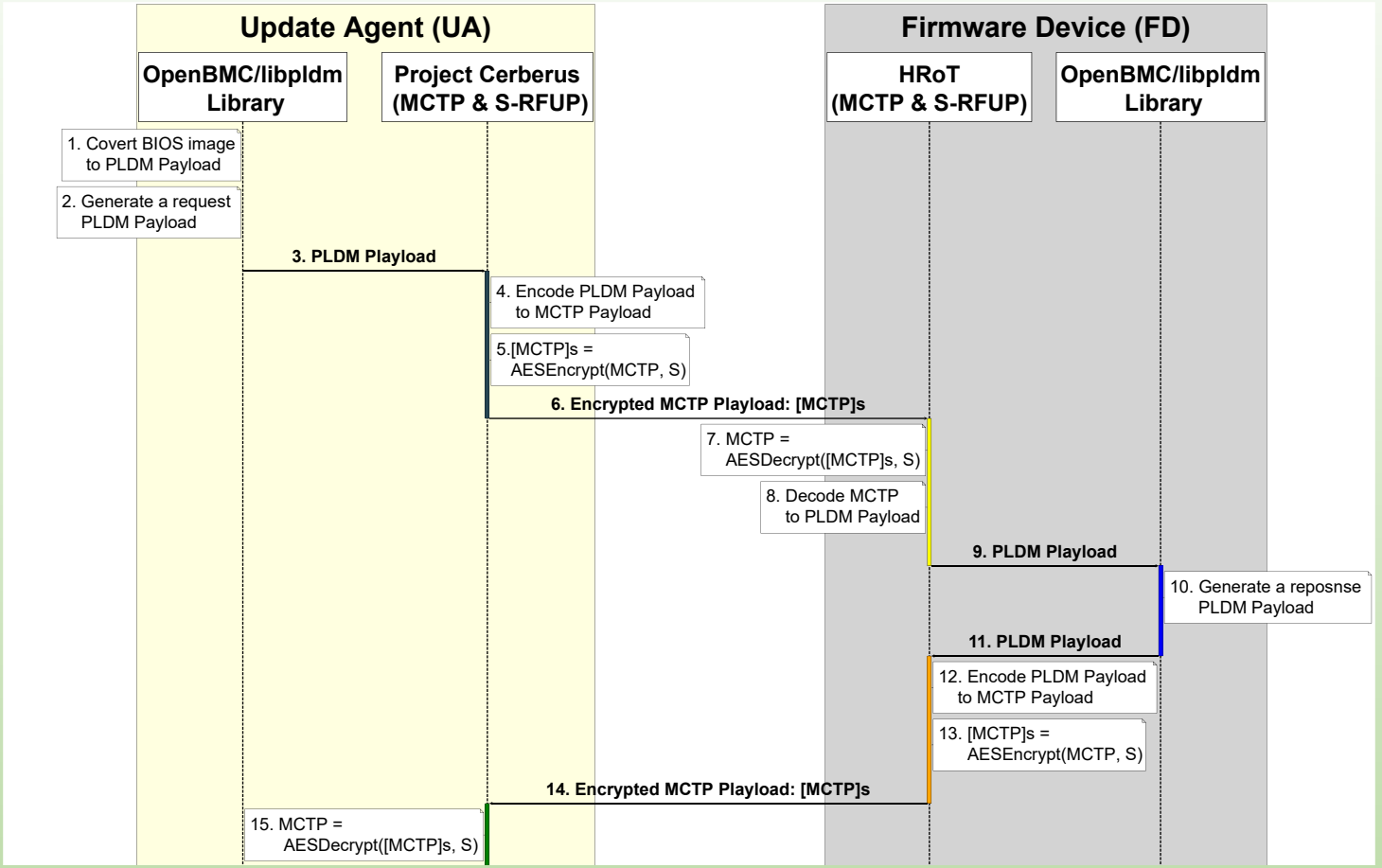
S-RFUP FIRMWARE UPDATE PROCESS



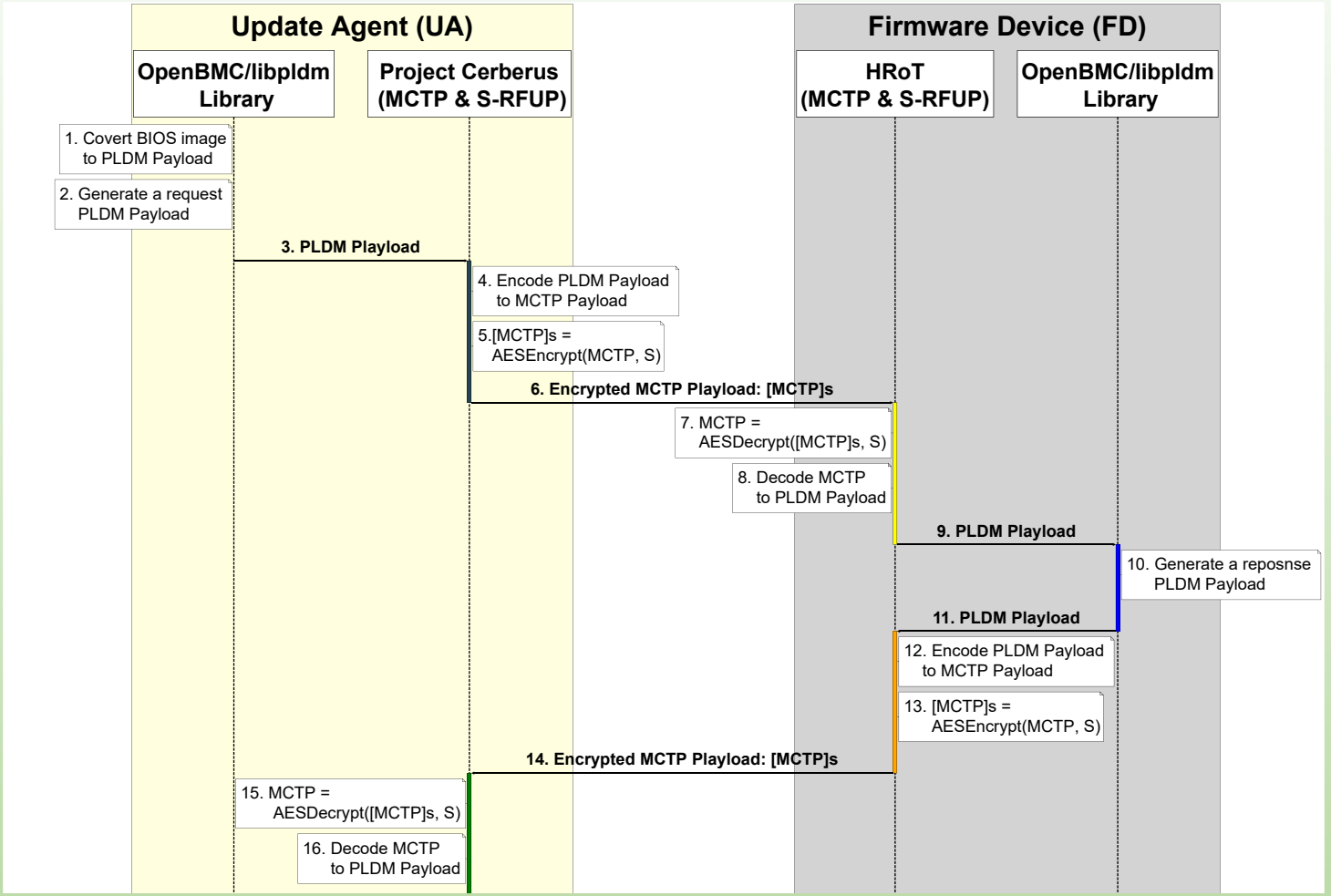
S-RFUP FIRMWARE UPDATE PROCESS



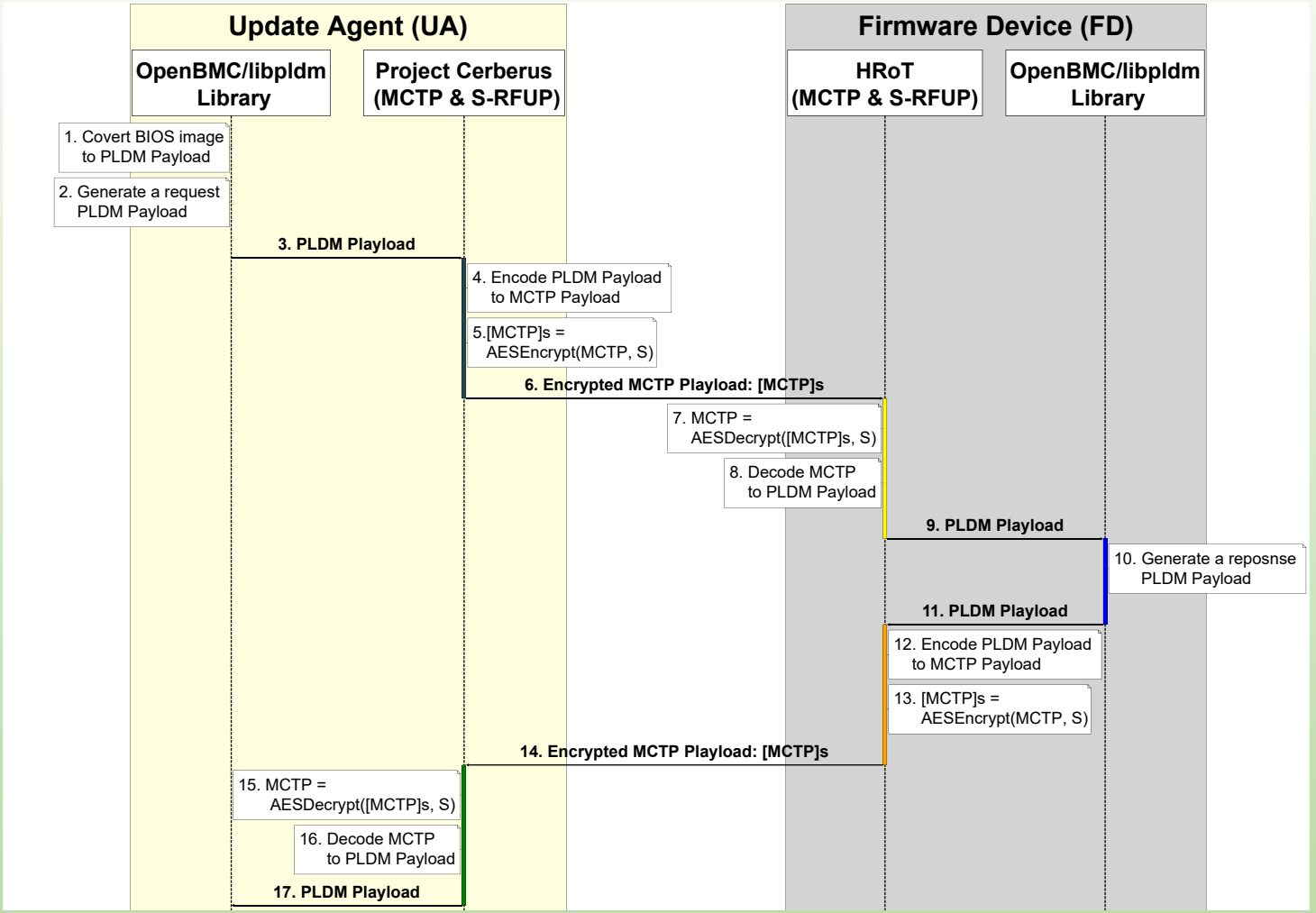
S-RFUP FIRMWARE UPDATE PROCESS



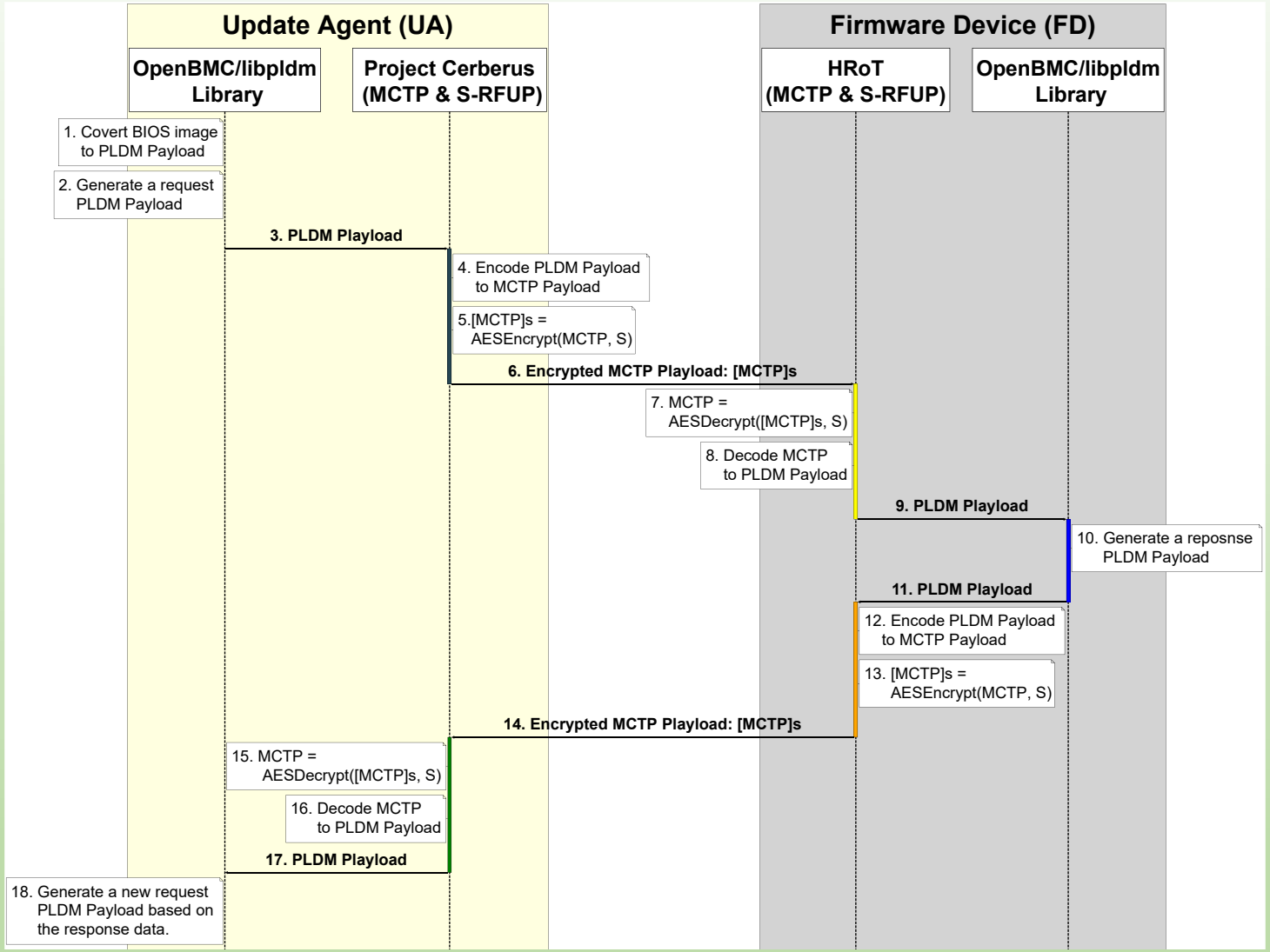
S-RFUP FIRMWARE UPDATE PROCESS



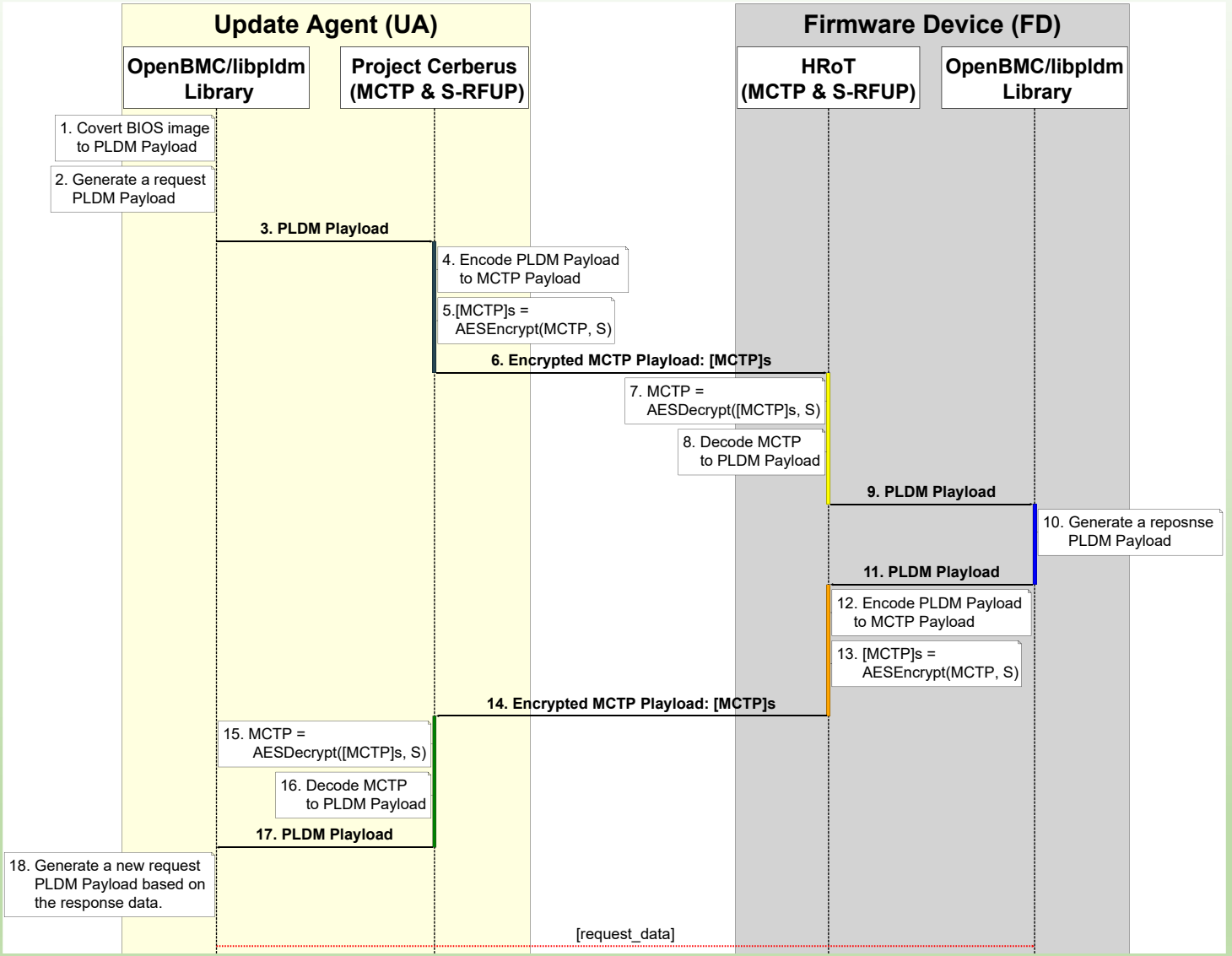
S-RFUP FIRMWARE UPDATE PROCESS



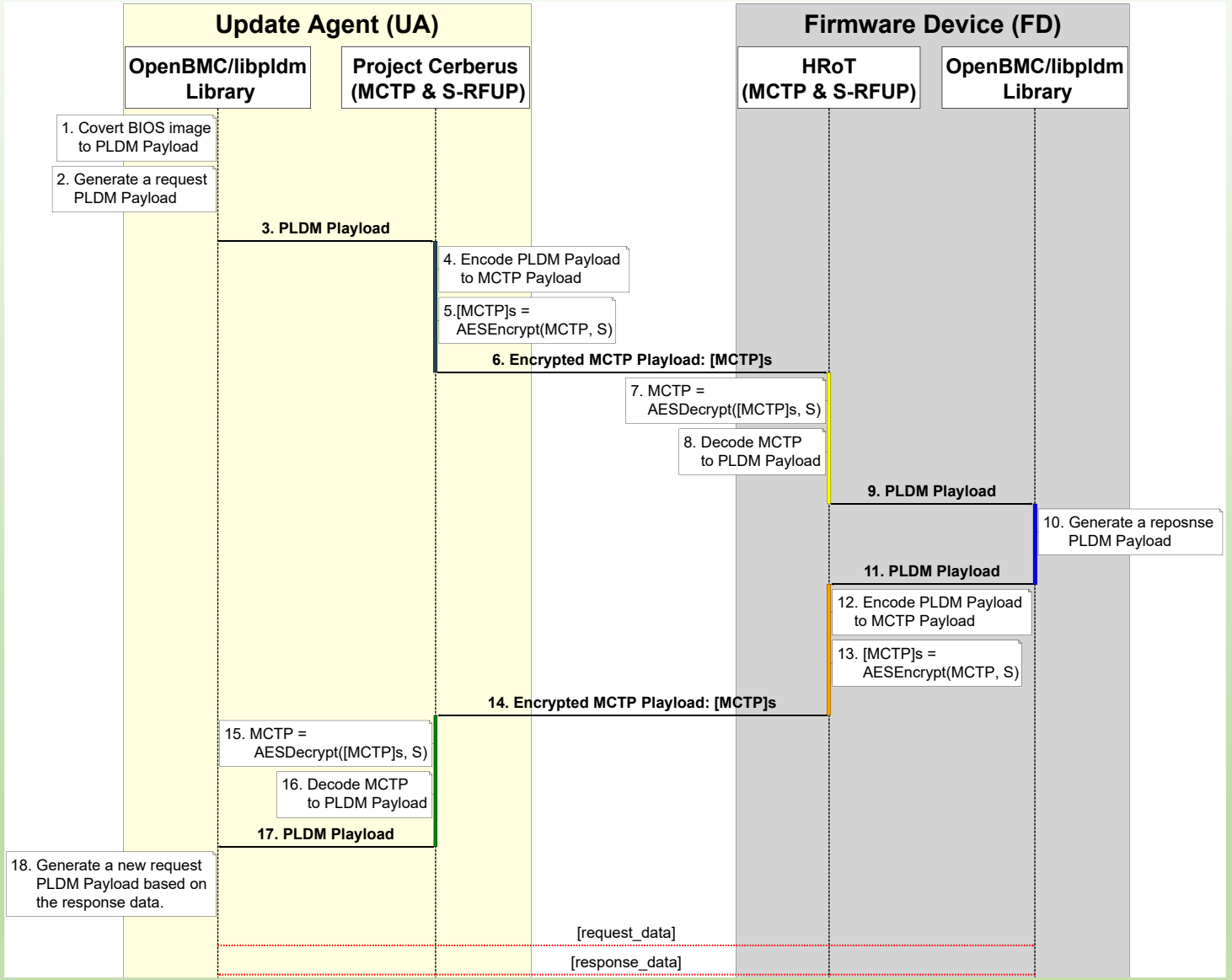
S-RFUP FIRMWARE UPDATE PROCESS



S-RFUP FIRMWARE UPDATE PROCESS



S-RFUP FIRMWARE UPDATE PROCESS



S-RFUP FIRMWARE UPDATE PROCESS