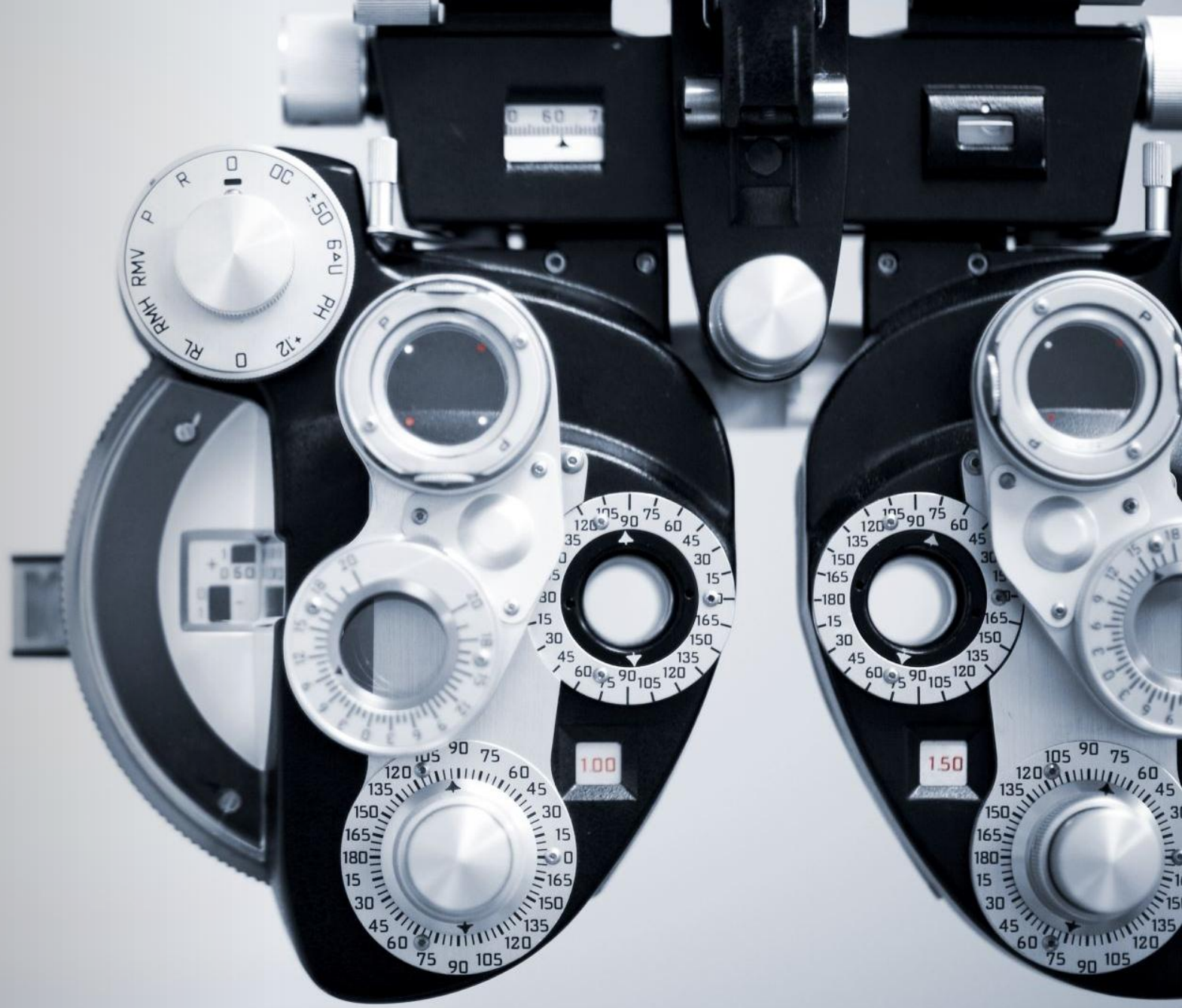
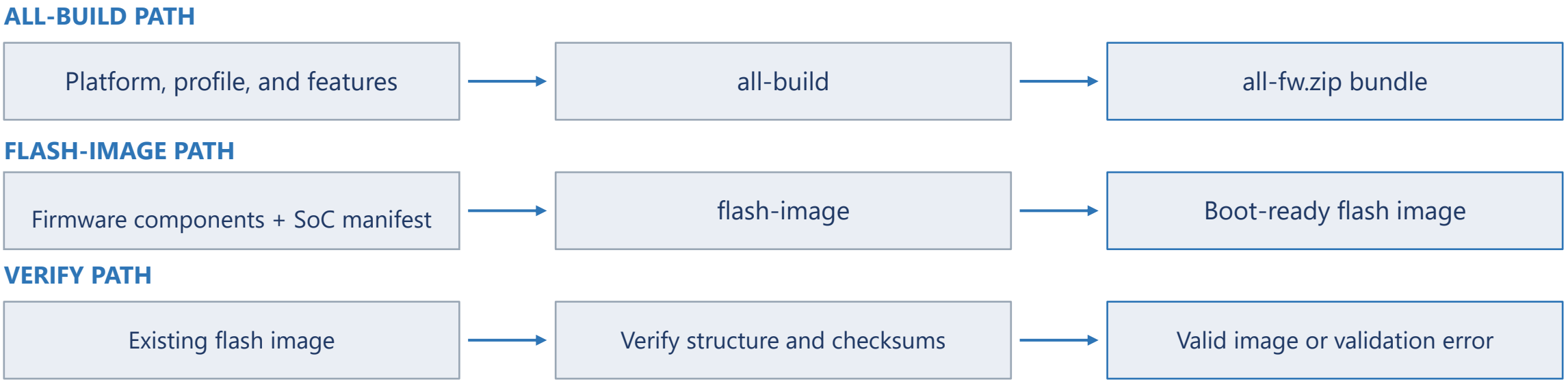
Caliptra Tools & Utilities

Imaging Utilities



Build & Flash Image Utilities

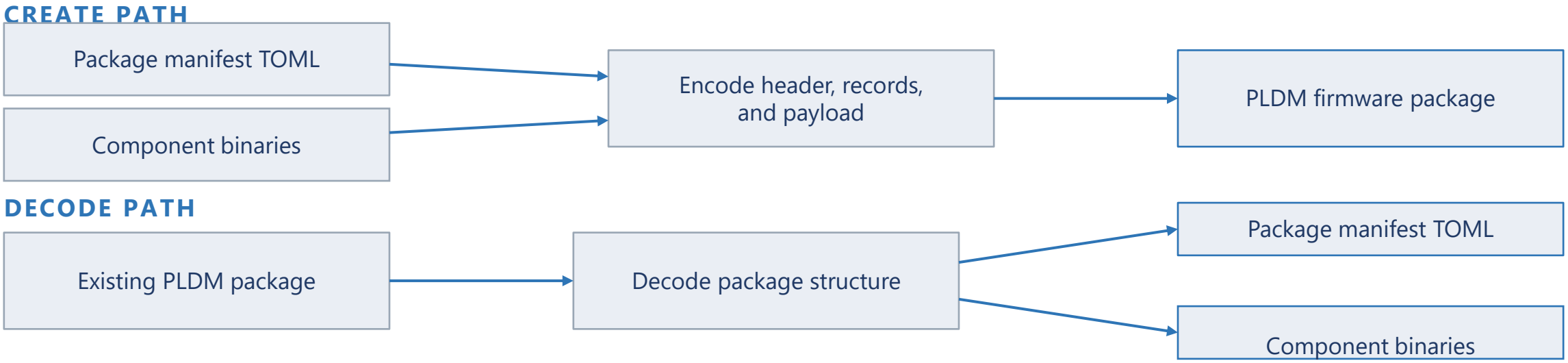
- **What it does:** all-build creates complete firmware bundles (all-fw.zip); flash-image creates or validates bootable flash images. all-fw.zip contains separate build artifacts (Caliptra FMC+RT,MCU RT, auth manifest, etc), whereas a flash image is a boot-formatted binary representing the contents of device flash.
- **Inputs:** Platform, profile, features, optional MCU/SoC images, firmware components, and SoC manifest; existing image to verify.
- **Outputs:** all-fw.zip and build artifacts; bootable flash image or validation result.



On-flash format: https://github.com/chipsalliance/caliptra-mcu-sw/blob/main/docs/src/flash_layout.md

PLDM Type 5 Packaging Utilities

- **What it does:** Creates PLDM firmware update packages and decodes packages for inspection or reuse.
- **Inputs:** Create: TOML manifest and component binaries. Decode: PLDM package and destination directory.
- **Outputs:** Standards-compatible PLDM package; decoded manifest and component binaries.

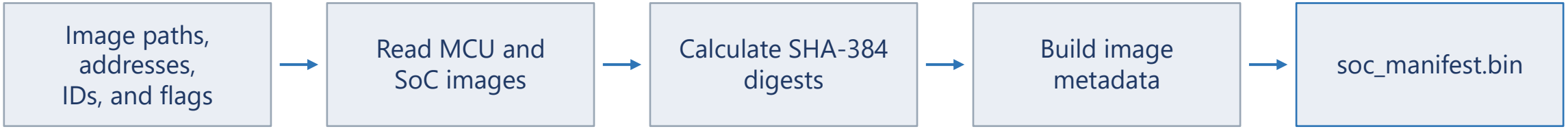


Package layout: https://github.com/chipsalliance/caliptra-mcu-sw/blob/main/docs/src/pldm_package.md

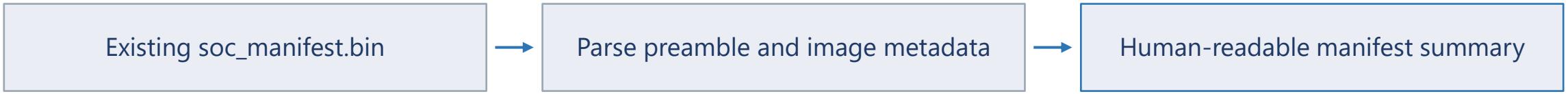
Auth / SoC Manifest Utilities

- **What it does:** Creates Caliptra authorization manifests and parses existing manifests with SHA-384 metadata.
- **Inputs:** Create: MCU/SoC images, addresses, IDs, and flags. Parse: existing manifest binary.
- **Outputs:** Binary manifest for secure boot; readable identity, version, image, and digest report.

CREATE PATH



PARSE PATH

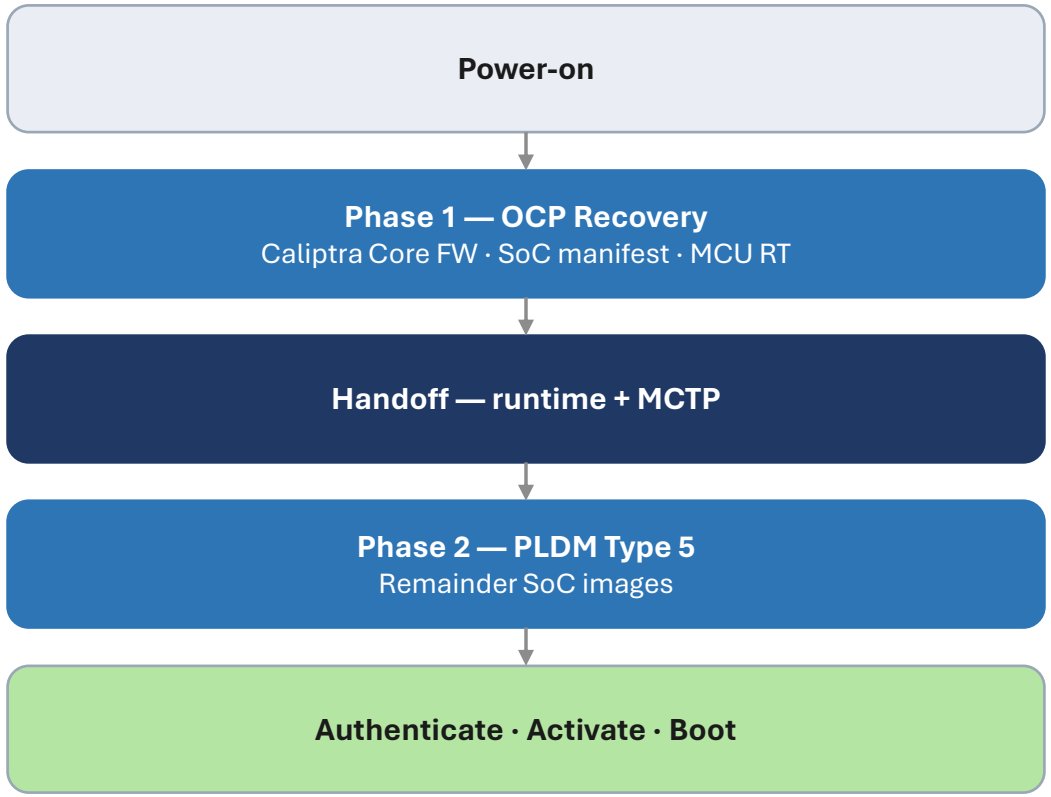



What Does the Streaming Boot Agent Do?

Streaming Boot: A Two-Phase Secure Boot

The BMC establishes the trusted runtime, then the live MCU pulls the rest.

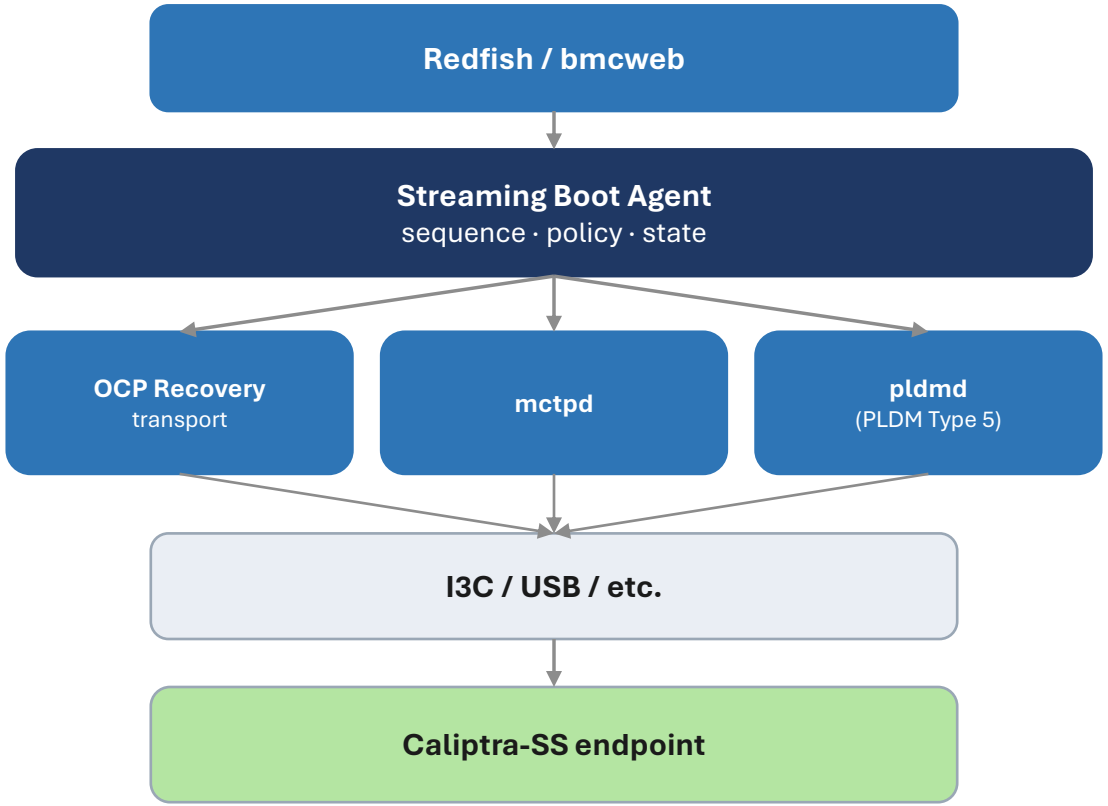
- **Phase 1 — OCP Recovery:** BMC pushes an ordered image set (Caliptra FW, SoC manifest, MCU runtime) over the register interface. No MCTP/PLDM/runtime yet.
- **Handoff:** MCU runtime starts, exposes an MCTP endpoint; bus ownership changes.
- **Phase 2 — DMTF PLDM Type 5:** MCU pulls remaining SoC images via RequestFirmwareData; BMC acts as Update Agent.
- Caliptra authenticates and gates every stage — streaming boot is the normal boot path.



Streaming Boot Agent: OpenBMC Orchestration

The agent owns the sequence — existing daemons own the wire protocols.

- **Streaming Boot Agent (SBA):** per-endpoint state machines — policy, ordering, handoff, persistence, reporting.
- **OCP Recovery transport:** Phase 1 register-level transfers.
- **mctpd:** runtime endpoint discovery and EID policy.
- **pldmd:** DMTF PLDM Type 5 Update Agent and package handling.
- **Redfish / bmcweb:** operator-facing status, per-endpoint and platform aggregate.



Streaming Boot Agent: Reference Implementation

Reference implementation — not yet a production OpenBMC distribution.

Validated today (pre-silicon, QEMU)

- Real SBA daemon + real OCP Recovery transport on OpenBMC.
- Three-image Phase 1 over virtual I3C into the Caliptra MCU emulator.
- Boot through ROM into the Tock runtime.
- Live progress + classified authentication/authorization failures; developer CLI + tests.

Architecture extension points

- Runtime → MCTP endpoint handoff · full PLDM Phase 2.
- Multi-endpoint dependency supervision · restart/resume.
- Production hardware transport · Redfish rendering.

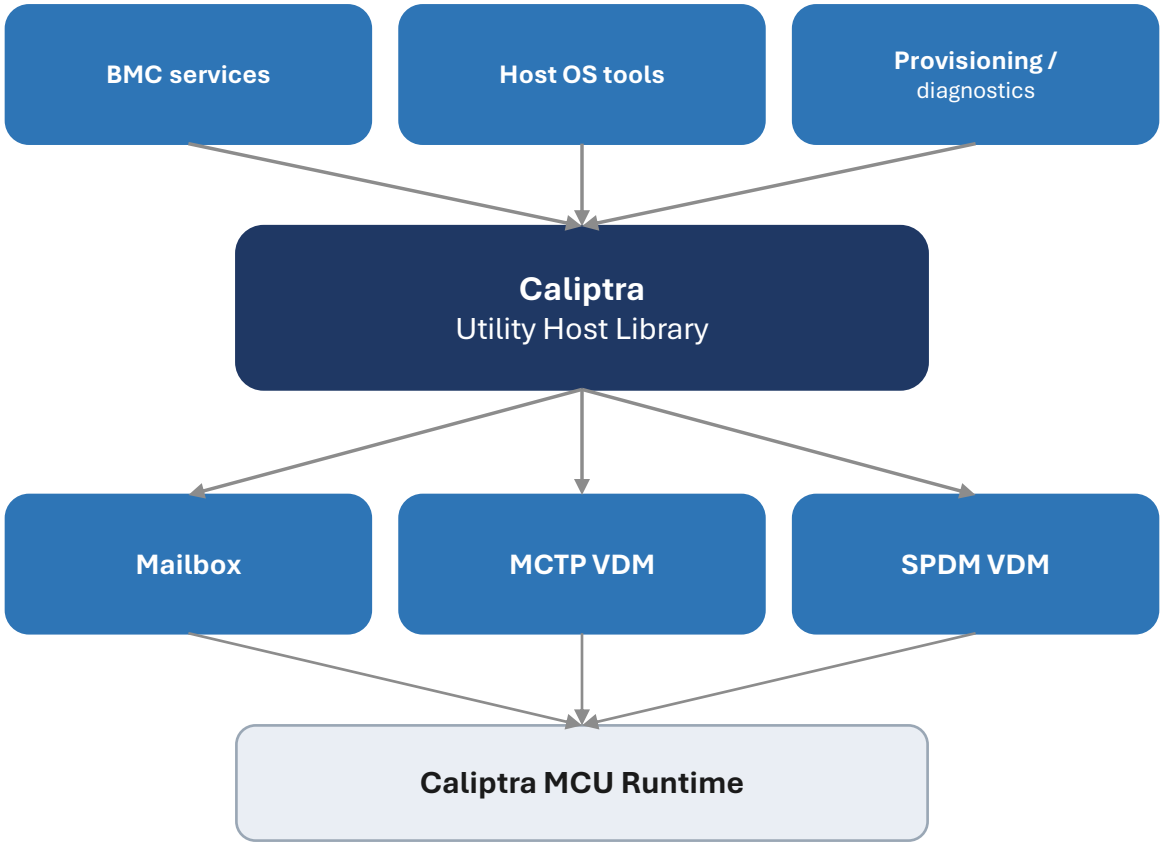
🔗 *Repo: github.com/helloxiling/streaming-boot-agent-ref/tree/main-dev*

Caliptra Utility

Host Utility Library: One Runtime Interface

One API to interact with Caliptra devices — regardless of where the tool runs or which transport reaches the device.

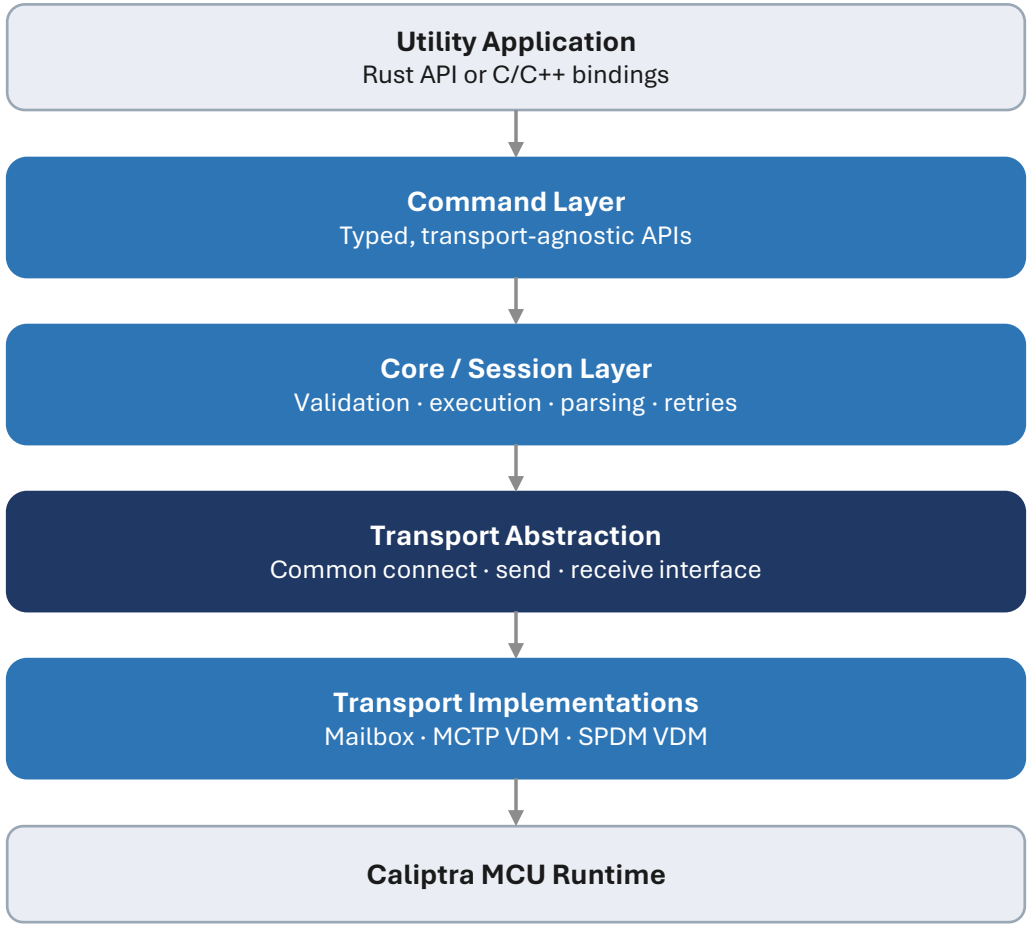
- Runs on BMC, host OS, provisioning stations, and service tools.
- Supports in-band and out-of-band device management.
- A common typed API hides the transport details.
- **Transports today:** Mailbox, MCTP VDM, and SPDM VDM
- Rust implementation with C/C++ bindings for existing platform software.



Host Utility Library: Layered Architecture

Applications use the same typed commands while sessions and transports handle platform-specific communication.

- **Command layer:** typed, transport-agnostic APIs for device operations.
- **Core / session layer:** validation, command execution, response parsing, and retries.
- **Transport abstraction:** a common connect, send, and receive interface.
- **Transport implementations:** Mailbox, MCTP VDM, and SPDM VDM.
- **Integration surfaces:** native Rust APIs and C/C++ bindings.



Host Utility Library: OpenBMC & Standalone Integration

- **Utility role today:** Caliptra-specific commands over Mailbox, MCTP VDM, and SPDM VDM.
- **OpenBMC integration:**

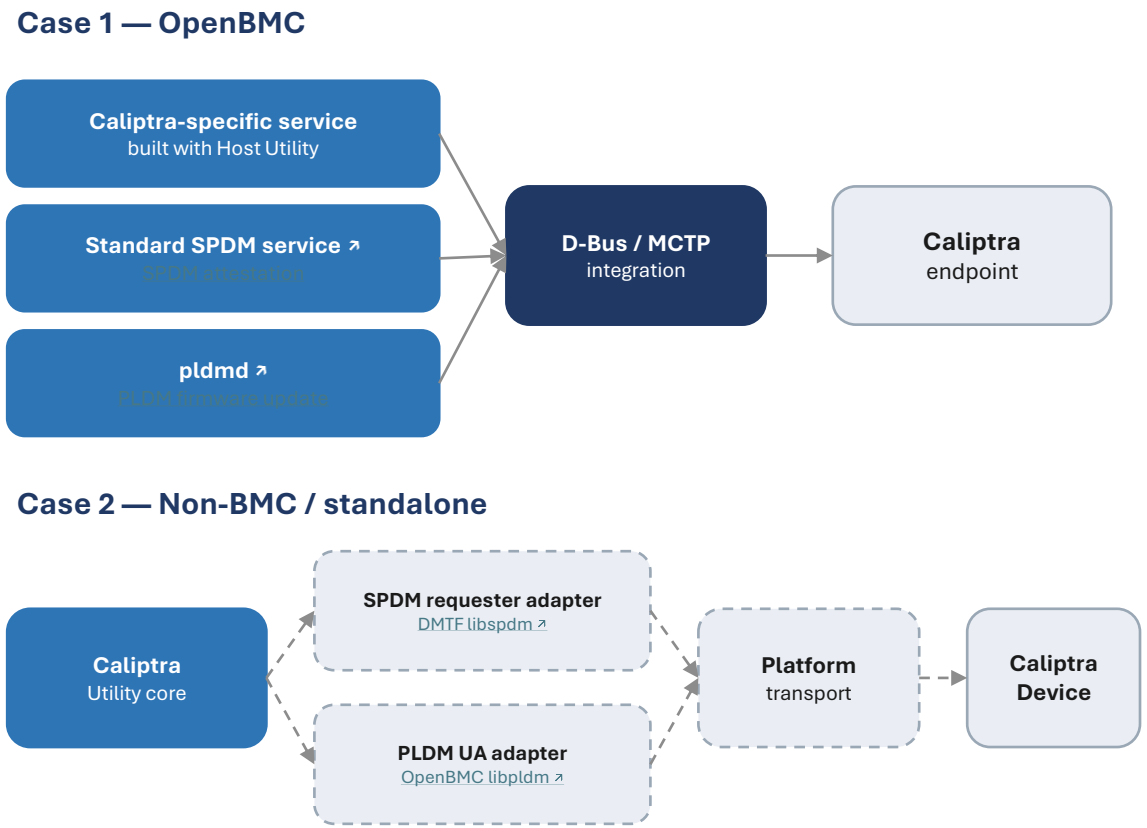
Leverage the host utility library to build Caliptra-specific services integrated with OpenBMC D-Bus and MCTP.

Use standard OpenBMC services for SPDM attestation and PLDM firmware update.

- **Non-BMC / standalone applications:**

Add requester and Update Agent adapters around DMTF libspdm and OpenBMC libpldm.

Reuse standard libraries; standalone SPDM/PLDM adapters are extension points, not implemented today.



[Reference links]

- OpenBMC pldmd: https://github.com/openbmc/pldm/tree/master/fw-update
- libpldm: https://github.com/openbmc/libpldm
- DMTP libspdm: https://github.com/DMTF/libspdm
- OpenBMC spdmd: https://github.com/openbmc/spdm (under development);
- Nvidia/spdmd: https://github.com/NVIDIA/spdm

Host Utility Library: Runtime Management Building Blocks

A unified, typed API turns Caliptra runtime commands into reusable platform-management workflows.

Workflow	Platform capability
Discover	Read firmware versions and device capabilities
Diagnose	Retrieve and clear debug logs; inspect runtime status
Provision Identity	Export attested CSRs and install signed device certificates
Secure Service	Perform production debug unlock and authorization-gated provisioning
Cryptographic Services	Hash, encrypt, sign, verify, and perform key exchange

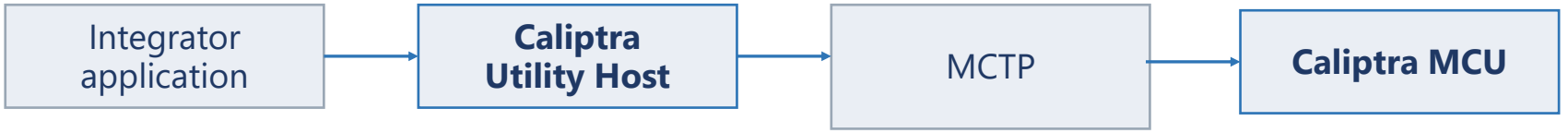
- Design doc: https://github.com/chipsalliance/caliptra-mcu-sw/blob/main/docs/src/caliptra_util_host_library.md
- Code: https://github.com/chipsalliance/caliptra-mcu-sw/tree/main/caliptra-util-host

Attestation Utilities

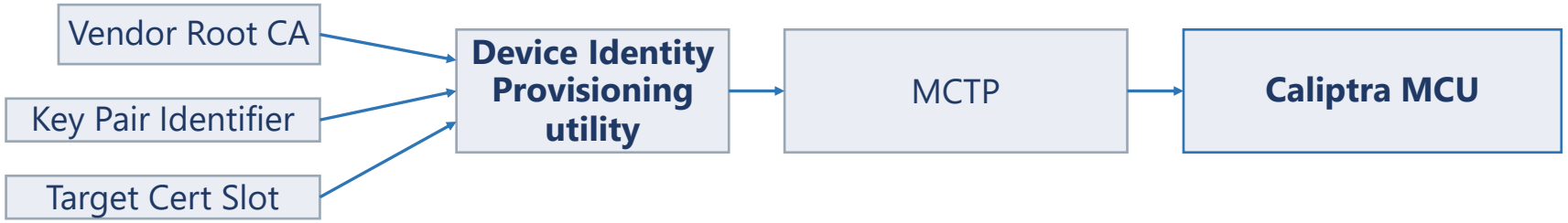
Certificate Provisioning

- **What it does:** Executes SPDM commands out of band through Host Utility Library; Provide device identity provisioning.
- **Inputs:** SPDM Command Arguments **(**see https://github.com/chipsalliance/caliptra-mcu-sw/blob/main/docs/src/caliptra_spdm_vdm_cmds.md**),** Transport implementation**,** target certificate slot, key-pair identifier, and Vendor root certificate
- **Outputs:** SPDM Command Responses (see response definitions in the link provided); A provisioned and verified Owner identity

SPDM COMMAND PATH



OCP DEVICE IDENTITY PROVISIONING



CoRIM Utilities

- **What it does:** Generates CoRIM reference values from an all-fw bundle; discovers Caliptra, MCU, manifest, and SoC measurements.
- **Inputs:** all-fw.zip (firmware bundle); optional JSON config, feature variants, and signing key.
- **Outputs:** Human-readable CoRIM JSON and exchange-ready CBOR; optional test signature.

INPUTS → PROCESSING → OUTPUTS