



SECURITY

Provisioning Flows for Caliptra Subsystem


Tim Trippel, Google
Darpana Munjal, Microsoft
Willy Zhang, Google
Emre Karabulut, Microsoft
Mojtaba Bisheh-Niasar, Microsoft

Objectives

1. Provide guidance to SoC integrators and manufactures on how to securely provision Caliptra Subsystem fuses and identity certificates.
2. Be vendor and integration agnostic.

What assets are provisioned at manufacturing?

Two categories of Caliptra Subsystem assets to be provisioned:

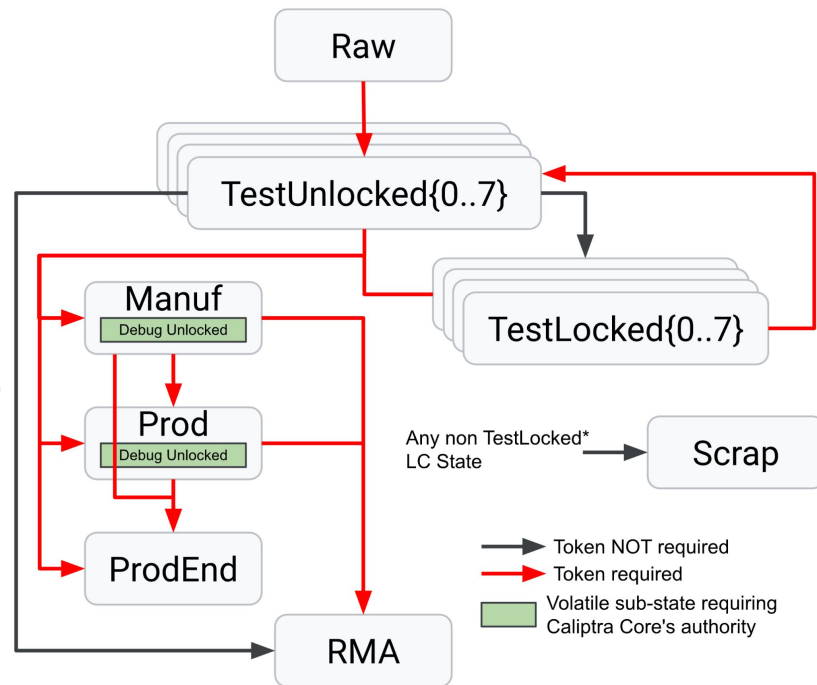
1. **Fuses**, defined in the [fuse map](#). 
 - a. Found in caliptra-ss GitHub repo and linked above.
 - b. Link above breaks down point during provisioning process each fuse should be written.
2. **IDeVID certificate**, for which there are two key/signature types:

| Certificate Types | Signature Size (Bytes) |
|-------------------|------------------------|
| ECDSA secp384r1 | 96 |
| ML-DSA-87 | 4627 |

| A | B | C | D |
|--------------------------------|--|----------|-----------------|
| Partition | Item | Size (B) | Lifecycle State |
| SW_TEST_UNLOCK_PARTITION | CPTRA_SS_MANUF_DEBUG_UNLOCK_TOKEN | 64 | TEST_UNLOCKED → |
| SECRET_MANUF_PARTITION | CPTRA_CORE_UDS_SEED | 64 | MANUF → |
| SECRET_PROD_PARTITION_0 | CPTRA_CORE_FIELD_ENTROPY_0 | 8 | In Field → |
| SECRET_PROD_PARTITION_1 | CPTRA_CORE_FIELD_ENTROPY_1 | 8 | In Field → |
| SECRET_PROD_PARTITION_2 | CPTRA_CORE_FIELD_ENTROPY_2 | 8 | In Field → |
| SECRET_PROD_PARTITION_3 | CPTRA_CORE_FIELD_ENTROPY_3 | 8 | In Field → |
| SW_MANUF_PARTITION | CPTRA_CORE_ANTI_ROLLBACK_DISABLE | 4 | MANUF → |
| | CPTRA_CORE_IDEVID_CERT_IDEVID_ATTR | 96 | MANUF → |
| | SOC_SPECIFIC_IDEVID_CERTIFICATE | 4 | MANUF → |
| | CPTRA_CORE_IDEVID_MANUF_HSM_IDENTIFIER | 16 | MANUF → |
| | CPTRA_CORE_SOC_STEPPING_ID | 4 | MANUF → |
| | CPTRA_SS_PROD_DEBUG_UNLOCK_PKS_0 | 48 | MANUF → |
| | CPTRA_SS_PROD_DEBUG_UNLOCK_PKS_1 | 48 | MANUF → |
| | CPTRA_SS_PROD_DEBUG_UNLOCK_PKS_2 | 48 | MANUF → |
| | CPTRA_SS_PROD_DEBUG_UNLOCK_PKS_3 | 48 | MANUF → |
| | CPTRA_SS_PROD_DEBUG_UNLOCK_PKS_4 | 48 | MANUF → |
| | CPTRA_SS_PROD_DEBUG_UNLOCK_PKS_5 | 48 | MANUF → |
| | CPTRA_SS_PROD_DEBUG_UNLOCK_PKS_6 | 48 | MANUF → |
| | CPTRA_SS_PROD_DEBUG_UNLOCK_PKS_7 | 48 | MANUF → |
| SECRET_LC_TRANSITION_PARTITION | CPTRA_SS_TEST_UNLOCK_TOKEN_1 | 16 | TEST_UNLOCKED → |
| | CPTRA_SS_TEST_UNLOCK_TOKEN_2 | 16 | TEST_UNLOCKED → |
| | CPTRA_SS_TEST_UNLOCK_TOKEN_3 | 16 | TEST_UNLOCKED → |
| | CPTRA_SS_TEST_UNLOCK_TOKEN_4 | 16 | TEST_UNLOCKED → |
| | CPTRA_SS_TEST_UNLOCK_TOKEN_5 | 16 | TEST_UNLOCKED → |
| | CPTRA_SS_TEST_UNLOCK_TOKEN_6 | 16 | TEST_UNLOCKED → |
| | CPTRA_SS_TEST_UNLOCK_TOKEN_7 | 16 | TEST_UNLOCKED → |
| | CPTRA_SS_TEST_EXIT_TO_MANUF_TOKEN | 16 | TEST_UNLOCKED → |
| | CPTRA_SS_MANUF_TO_PROD_TOKEN | 16 | TEST_UNLOCKED → |
| | CPTRA_SS_PROD_TO_PROD_END_TOKEN | 16 | TEST_UNLOCKED → |
| | CPTRA_SS_RMA_TOKEN | 16 | TEST_UNLOCKED → |
| SVN_PARTITION | CPTRA_CORE_FMC_KEY_MANIFEST_SVN | 4 | In Field → |
| | CPTRA_CORE_RUNTIME_SVN | 16 | In Field → |
| | CPTRA_CORE_SOC_MANIFEST_SVN | 16 | In Field → |

Lifecycle (LC) Architecture

- Provisioning flows are enabled by Caliptra SS LC architecture.
- Caliptra SS implements a hardware FSM in the lifecycle controller (LCC) block.
 - Based on the OpenTitan lc_ctrl block.
- FSM states are fuse-backed.
 - LC states are encoded using a 48-byte monotonically incrementing counter in fuses.
 - Transitions:
 - are actuated via a dedicated JTAG TAP
 - persist across resets (except volatile states)
 - some require password-like tokens or signatures (provisioned in fuses)
- Each life cycle state provides varying granularity of debug access.
- All devices start in Raw state (brick safe).
- Devices end in a functional state (Prod or ProdEnd) with limited debug access.
- TestUnlocked* and Manuf states aid provisioning by providing debug access.



Lifecycle States and Debug Access

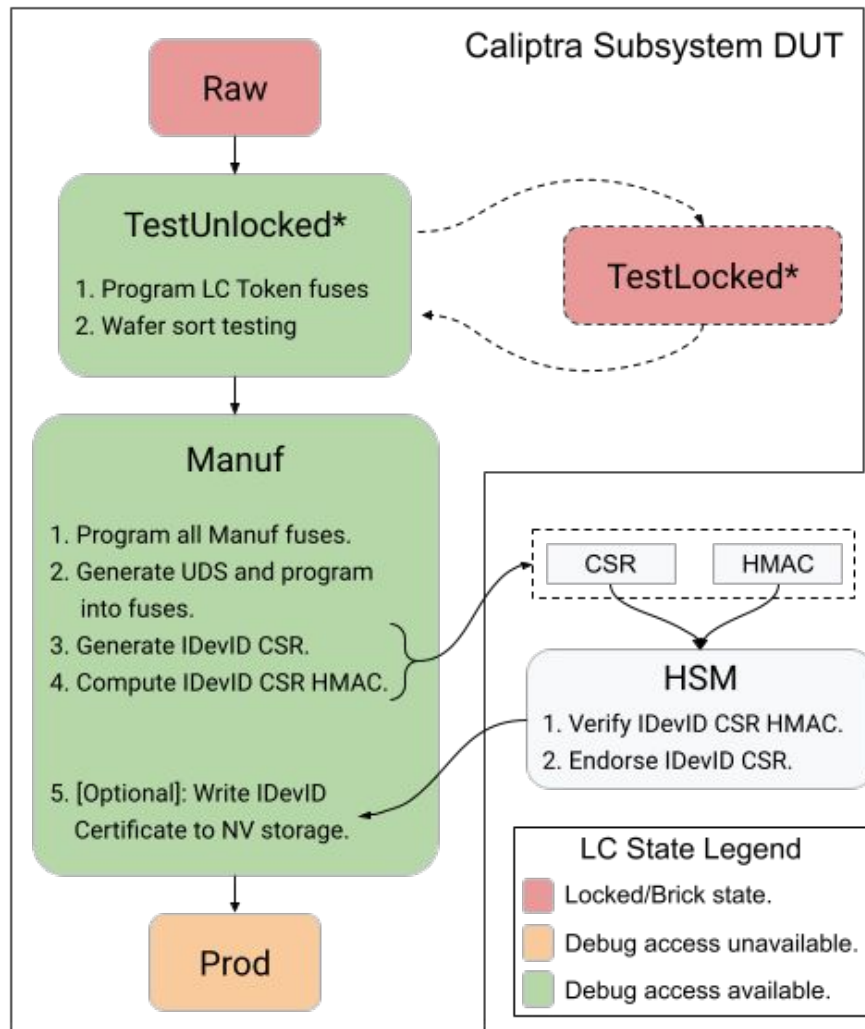
- LC states gate debug access:
 - JTAG TAP access
 - DFT access
- Three Caliptra Subsystem JTAG TAPs
 - Caliptra Core
 - Caliptra MCU
 - LCC (always accessible)
- Guidelines for additional Chip-Level TAP (CLTAP)
- Two DFT paths:
 - Subsystem
 - SoC

| Lifecycle State | JTAG Access | DFT Access |
|--------------------|------------------------|------------|
| Raw | LCC | None |
| TestUnlocked[0–7] | LCC, CLTAP*, MCU, Core | SS, SoC |
| TestLocked[0–7] | LCC | None |
| Manuf | LCC, CLTAP* | None |
| Manuf Debug Unlock | LCC, CLTAP*, MCU, Core | SoC* |
| Prod | LCC | None |
| Prod Debug Unlock | LCC, CLTAP*, MCU, Core | SoC* |
| ProdEnd | LCC | None |
| RMA | LCC, CLTAP*, MCU, Core | SS, SoC |
| Scrap | LCC | None |

* Integration Configurable

Provisioning Flow

- All device start in "Raw" LC state.
- March through LC states, performing provisioning operations along the way.
- End in a "Prod[End]" LC state.
- Each LC state gates chip functionality for security.
 - Brick states
 - Debug access enabled states
 - Debug access disabled states
- **TestUnlocked***
 - Program token fuses
 - Debug access enables wafer testing
- **Manuf**
 - program most fuses
 - generate/endorse IDevID certificate

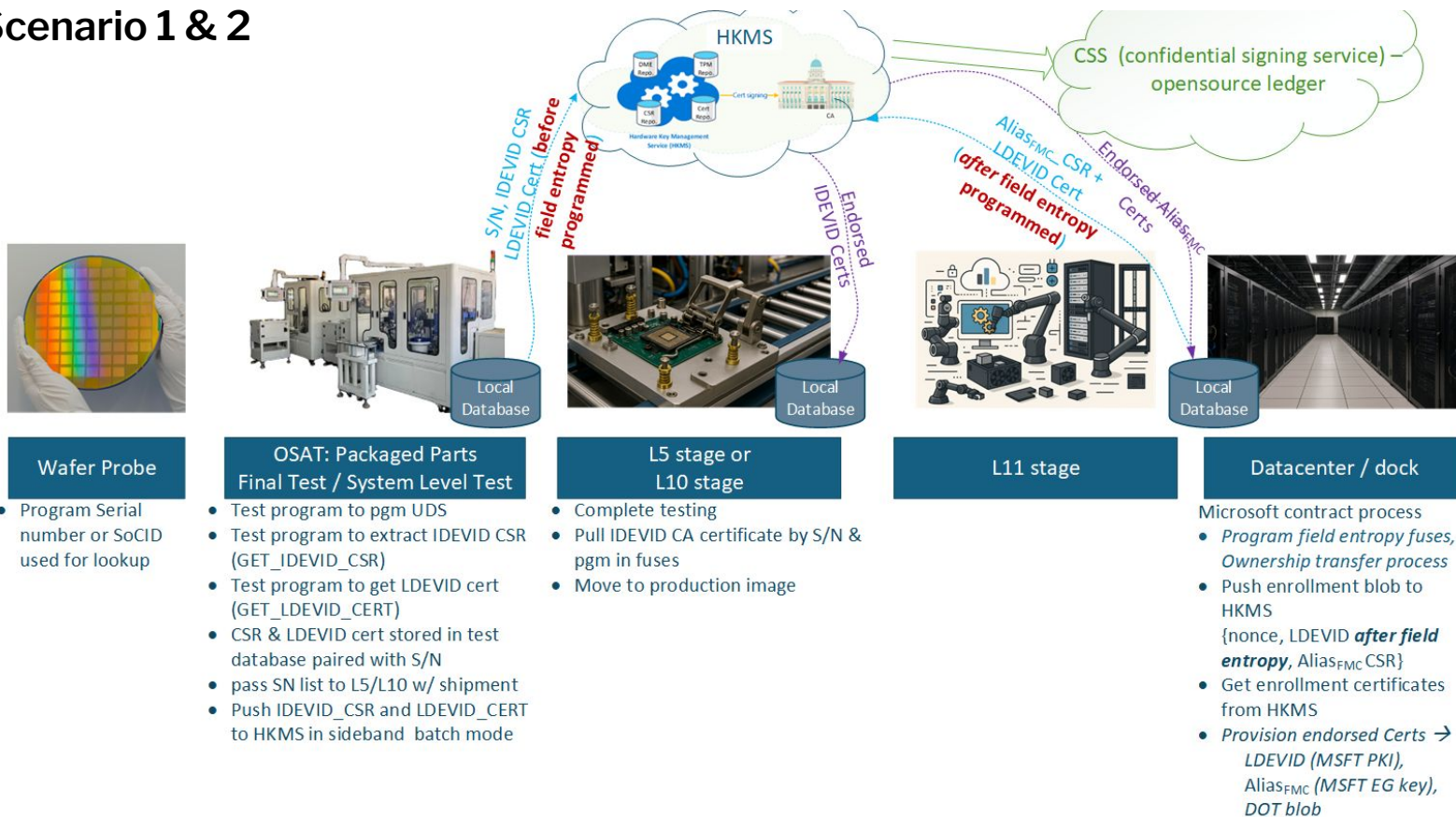


IDevID Provisioning Scenarios

| Scenario | UDS Provision | IDevID CSR Harvesting | IDevID Cert Endorsement | Device Delivery |
|---|---------------|--------------------------------|--|---|
| 1) Vendor and owner are the same | Owner ATE | Owner | <ul style="list-style-type: none">- Endorsed + stored in HSM- Sent to secure database for later endorsement | Owner manages provisioning infra |
| 2) Vendor builds for owner | Vendor ATE | Vendor, securely sent to Owner | Owner endorses CSR at ingestion | Shipped with UDS only (no endorsed IDevID cert) |
| 3) Vendor builds for open market | Vendor ATE | Vendor | Vendor PKI endorses | Shipped fully provisioned (only field entropy programmed later) |

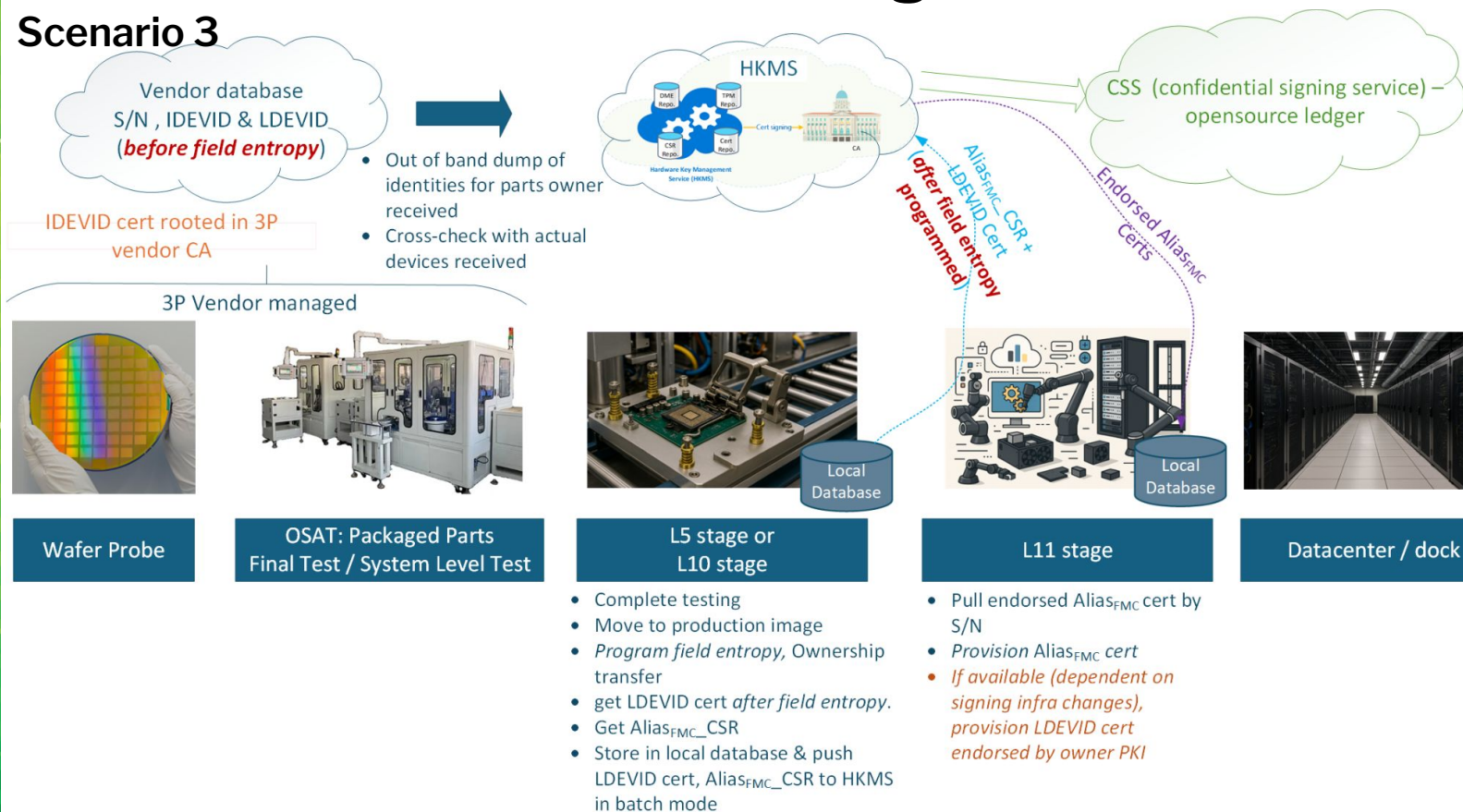
IDEVID Certificate Provisioning and DOT

Scenario 1 & 2



IDevID Certificate Provisioning and DOT

Scenario 3



In Field Fuse Programming

Purpose:

Enable owners to update select fuse partitions after manufacturing for security actions like key rotation or invalidation.

- Enter a production (Prod/ProdEnd) lifecycle state.
- Only designated ***in-field programmable*** partitions can be updated.
- Each fuse word remains one-time programmable. (No erased)
- Partition lock fuses (block all further writes unless zeroization is supported.)

Questions



OPEN
Compute
Project®