Software Authority, Hardware Identity

Brandon Ramsay
Irrefutable Labs Inc.

July 2026

Abstract

A Deterministic State Machine, or DSM, advances state by local deterministic acceptance rather than by global consensus. Ordinary DSM operation does not require a blockchain, validator set, sequencer, wall clock, or online settlement step on the common path.

This paper replaces the previous offline bearer specification. The previous specification treated a hardware monotonic counter as the transfer uniqueness authority and built a receiver witnessed counter positioned commit around it. Adversarial review showed that a scalar counter read never binds the transition it brackets, and further review showed that this binding was never needed. Transfer uniqueness is already a software property of DSM: the device state is one resource, it is consumed as a whole, and one parent state admits exactly one accepted and executed successor at an honest acceptance point. Hardware cannot add to that property. Hardware can only get in its way.

This paper states the separation directly. DSM software enforces correctness and transfer uniqueness by whole state consumption. Hardware enforces physical device identity. The hardware never becomes a transaction authority, never orders transactions, and never decides uniqueness. The receiver acceptance predicate reads no live chip state, no live host state, no raw hardware counter, no relay session, no verifier slot, no boot ticket, and no MACANDD output.

The remaining, irreducible job of hardware is device identity. A software clone holds every byte of host readable state. Software alone cannot distinguish such a clone from the original. This paper therefore assigns hardware exactly one role: making device identity physically unclonable enough for offline bearer use and bounding offline exposure while doing it. This does not mean offline releases work without hardware witnesses. Offline identity acceptance still requires the enrolled chip and host signatures. The separation is that those signatures prove identity, not transaction uniqueness.

The design has two permanent identity domains on one device. The online domain derives its identity from a BIP39 seed alone. It requires no hardware, costs nothing beyond a phone, and supports all online DSM operation. The offline domain derives its identity from a fusion of three factors: the BIP39 seed held on the phone, a PUF rooted non exportable key inside a TROPIC01 secure element, and a measurement gated partition sealed key inside the RP2350 secure partition. Every offline bearer release must be witnessed by all three factors over the same root advance message.

The RP2350 host witness key is generated at birth under a partition seal. The unsealing gate is the enrolled firmware measurement. Only firmware whose measurement matches the enrolled policy may invoke the host signing authority. Firmware replacement therefore does not give an attacker a programmable signing oracle: rogue firmware receives an unsealing failure and cannot produce the host witness. The chip and host witnesses are minted at commit, after the one way counter decrement, so no valid release witness for an origin exists while that origin is still spendable. Under enrolled firmware, the appliance is single threaded, maintains one active state, permits at most one prepared record per frontier, exports no release before commit, and recovery only re emits the same committed release. Therefore a single enrolled appliance cannot

originate two distinct valid offline releases from the same origin, including to disconnected first contact receivers.

The design adds one structural element to the DSM device state: a monotone counter committed as a leaf of the device sparse Merkle tree. The SMT counter makes state position explicit inside the consumed state, allows offline acceptance to decide locally whether a proposed transition consumes the receiver's current frontier and advances it exactly once, and is the software register the TROPIC01 monotonic counter must track one to one, one physical decrement per offline commit. The physical counter is not an acceptance authority. It is a tracker of the SMT counter: a non rewind floor, a stale image tripwire, and an offline exposure cap.

The receiver witnessed counter positioned commit, the MACANDD transfer witness, the fused anchor head, the boot ticket chain, and the first transfer disclosure round trip of the previous specification are removed. The July 8, 2026 silicon validation on Raspberry Pi Pico 2 W with TROPIC01 is retained and reinterpreted under the new claims: it validates counter initialization, single commit discipline, refusal of a second commit, counter stability, and distinct per die chip identity across two physical chips. The July 9, 2026 ECC gate validation adds the resident witness key result: a fresh TROPIC01 slot read fails cleanly, in slot Ed25519 key generation succeeds, on die signatures verify off chip, erase is reversible, and the public key persists byte identically across power cycle.

1 Purpose and Scope

This document specifies the DSM device identity and offline bearer authority. It replaces the specification titled *Boot Fenced Fused Anchor Authority for DSM Offline Bearer State* in full. Section 25 enumerates what was removed and why.

The central claim of this document is a separation:

> Software (DSM) enforces correctness and uniqueness of state transitions.
>
> Hardware (anchor) enforces uniqueness of the physical device instance.

These two jobs are not mixed. The hardware never authorizes a transaction. The software never attempts to prove physical possession.

The authority provides:

1. one online device identity derived from a BIP39 seed, requiring no hardware;
2. one offline device identity derived from a three factor fusion of seed, PUF rooted chip key, and measurement gated partition sealed host key;
3. a permanent dual domain structure in which both identities coexist on one device;
4. an upgrade ceremony by which an online only device gains the offline domain;
5. transfer uniqueness enforced by whole state consumption at DSM acceptance points;
6. strict first contact offline uniqueness for a single enrolled appliance under the measurement gate;
7. a monotone state counter committed in the device SMT, making position explicit, allowing local offline validation of state consumption, and making hardware synchronization definable;

8. a three factor release witness on every offline bearer transfer;
9. a physical counter synchronized one to one to the SMT counter, providing a non rewind floor and an offline exposure cap;
10. recovery by re emitting the same committed release;
11. Tripwire exposure and propagation of incompatible proposals, non mergeability of divergent histories, and bricking of split identities on first honest contact.

The target hardware for the offline domain is unchanged.

Layer	Part	Role
Controller	Raspberry Pi Pico 2 W	host and transport board
MCU	RP2350	secure partition, host key, appliance policy
Secure element board	MIKROE 6559 Secure Tropic Click	TROPIC01 over SPI
Secure element	TROPIC01	PUF rooted chip key, monotonic counter
Interface	SPI at 3.3 V	secure element command transport

The online domain requires none of it.

2 Trust Decomposition

Every component is assigned exactly one job. No component is trusted for a job assigned to another.

Component	Provides
DSM / device SMT	transition validity; transfer uniqueness by whole state consumption; one parent, one accepted and executed successor at an honest acceptance point
BIP39 seed (phone)	online identity; factor one of the offline identity
PUF chip key (TROPIC01)	factor two; per release hardware witness; per die identity root
Partition key (RP2350)	factor three; measurement gated host witness; unsealed only under enrolled firmware
Monotonic counter	non rewind floor; stale image bricking; offline exposure cap
Tripwire	exposure and propagation of incompatible proposals; non mergeability of divergent histories; identity split bricking

Explicit non goals for hardware: ordering, transaction uniqueness, transaction authority, double spend prevention. The previous specification assigned all four to hardware. All four are removed.

The RP2350 partition key is not merely a key stored on the host. It is a key generated at birth under a seal whose unsealing predicate includes the enrolled firmware measurement. Only firmware matching the enrolled policy hash may obtain the host signing service. Firmware that fails the measurement gate receives `UNSEALING_FAILED` or an equivalent hard failure. There is no fallback signing path.

3 Naming Discipline

The protocol uses the following names. These names are not interchangeable.

Name	Meaning
R_i	sender device SMT root before transfer
R_{i+1}	sender device SMT root after transfer
h_i	offline frontier root advanced from
h_{i+1}	offline frontier root advanced to
ℓ_i	relationship leaf or chain head inside the sender SMT
κ_{res}	concrete resource occurrence consumed by the transfer
u_i	anchor counter coordinate committed in R_i
H_0	enrolled TROPIC01 physical counter value
H	live raw TROPIC01 physical counter value
B	immutable enrollment bundle for the offline domain
I_{on}	online device identity
I_{off}	offline device identity
D_{i+1}	transition digest
M_{i+1}	root advance message
r_R	receiver challenge
μ	firmware measurement
μ_{enrolled}	enrolled firmware measurement committed by policy

The device root is the per device SMT root. The offline frontier root h_i is a dedicated forward only lineage advanced exactly once per offline bearer transfer; it is the object a counterparty tracks. The relationship leaf is the bilateral chain head inside the device SMT. Each bilateral relationship is its own independent straight hash chain with exactly one receiver.

The anchor counter is not the raw TROPIC01 counter. The anchor counter increases:

$$u = H_0 - H.$$

The raw TROPIC01 counter counts down:

$$H \leftarrow H - 1.$$

All subtraction is checked; $H > H_0$ is rejected as counter mismatch.

4 Terminology Discipline

The protocol distinguishes proposed, accepted, executed, realized, consumed, rejected, and exposed objects.

- A *proposal* is a candidate transition or release package that can be constructed or transmitted as bytes.
- An *accepted* proposal is one that passes the receiver acceptance predicate.

- An *executed* or *realized* transition is the accepted successor that advances state.
- A *consumed* state is a parent state that has been used by an accepted and executed successor.
- A *rejected* proposal fails the acceptance predicate and does not advance state.
- An *exposed* proposal is an incompatible proposal whose conflict with another proposal or accepted lineage becomes known to a party.

A double spend is not the construction of two proposals. A double spend would require two incompatible executions of the same consumed state inside one coherent state history. DSM excludes that at acceptance by whole state consumption. Tripwire exposes attempted divergence and prevents incompatible histories from merging; it is not the source of execution uniqueness.

5 Cryptographic Preliminaries

Let H denote BLAKE3 256, modeled as collision resistant and second preimage resistant. Let HKDF denote a domain separated key derivation function. All structured objects use canonical byte encoding; verifiers reject non canonical encodings. If X is structured, $\mathrm{enc}(X)$ is its canonical encoding.

Three signature schemes appear.

1. $(\mathrm{DsmSign}, \mathrm{DsmVerify})$: the DSM device signature scheme. Device keys are derived from the seed. Every DSM transition is signed under this scheme; this is factor one.
2. $(\mathrm{ChipSign}, \mathrm{ChipVerify})$: Ed25519 executed inside TROPIC01 under a resident key pair generated in slot at birth. The private half is not exported through the TROPIC01 API, and key storage inside TROPIC01 is protected by the die's physically unclonable function. This is factor two.
3. $(\mathrm{HostSign}, \mathrm{HostVerify})$: a signature scheme under a key pair generated inside the RP2350 secure partition at birth and sealed to it. The private half is not exported. The signing service is measurement gated: it is available only when the running firmware measurement matches the enrolled policy. This is factor three.

The previous specification derived a one time witness key from a MACANDD output. That construction is removed. The reason is load bearing for the design: the MACANDD slot state is a pure function of the call input and the slot index. The vendor's own reference PIN flow restores consumed slots by replaying a known input. Slot state is therefore host restorable and carries no forward only lineage. A resident non exportable signing key is the correct possession witness; its signatures are portable and verifiable offline by any receiver with no relay session to the chip.

TROPIC01 Ed25519 signatures are not assumed to be deterministic. The July 9, 2026 ECC gate probe observed different signatures for the same key and message, both valid. The protocol never hashes a chip signature into D_{i+1}, h_{i+1}, R_{i+1}, or M_{i+1}. Receivers verify signatures; they do not compare signature bytes for stability.

6 Identity Domains

The device carries two identities in two hash separated domains. Both are permanent. Relationships are tagged by domain at establishment and never migrate.

6.1 Online Identity

Let seed be the BIP39 derived master secret held on the phone. The online root secret is

$$k_{\text{on}} = \text{HKDF}(\text{seed}, \texttt{"DSM/identity/online/v1"}),$$

from which the DSM device key pair $(sk_{\text{on}}, pk_{\text{on}})$ is derived. The online identity is

$$I_{\text{on}} = H(\texttt{"DSM/identity/online-id/v1"} \parallel pk_{\text{on}}).$$

The online domain requires no hardware anchor. All online DSM operation, including relationship establishment, bilateral transfer, and reconciliation, proceeds under I_{on} with DSM signatures alone.

6.2 Offline Identity

The offline identity exists only after appliance birth and is the hash of the anchor bundle:

$$I_{\text{off}} = H(\texttt{"DSM/identity/offline-id/v1"} \parallel B).$$

The bundle B binds all three factors. Producing a valid offline bearer release requires live cooperation of all three: the DSM transition is signed under seed derived keys, and the release certificate carries both the chip signature and the host signature over the same root advance message.

6.3 Upgrade Ceremony

A device operating online only may add the offline domain at any time. The ceremony is: perform appliance birth, obtain B, and publish an upgrade certificate

$$U = (B, pk_{\text{on}}, \sigma_{\text{on}}),$$

where

$$\sigma_{\text{on}} = \text{DsmSign}_{sk_{\text{on}}}\big(H(\texttt{"DSM/identity/upgrade/v1"} \parallel B)\big).$$

The certificate binds I_{off} to I_{on} under the online identity's own signature. Counterparties that have themselves upgraded may then establish offline domain relationships with the device. Existing online relationships continue unchanged. Neither domain is legacy; the device uses each domain with counterparties who operate in it.

Implementation staging note. The final protocol requires U for genesis adoption of an offline relationship. A staging implementation may keep first transfer TOFU admission while SDK wiring is being proven. Real upgrade certificate persistence and enforcement belong to the final acceptance rule and must not be confused with staging admission.

6.4 Compromise Matrix

Attacker holds	Online domain	Offline domain
phone / seed only	takeover	safe; lacks chip and host
chip only	safe	safe; lacks seed and host
host partition only	safe	safe; lacks seed and chip
phone + chip	takeover	safe; lacks host
phone + host	takeover	safe; lacks chip
chip + host	safe	safe; lacks seed
phone + chip + host, enrolled firmware intact	takeover	one live appliance; strict single appliance uniqueness, no fork
phone + chip + host + measurement breach or perfect emulation	takeover	outside offline distinguishability

The online domain is deliberately one factor: it is the zero hardware entry path. Phone compromise is therefore online domain takeover. Because the device root lineage is shared by both domains while relationships are domain tagged, phone compromise can still damage the online side of the shared device lineage and can disrupt availability. Significant value should be held in offline domain relationships or under policy that requires offline authority for high value movement.

Value held in offline domain relationships is protected by the full fusion. A user may keep the phone, the appliance, and the seed backup physically separate; the offline domain then survives the loss or compromise of any one of them, and of any two. Offline compromise of all three factors while the enrolled firmware boundary remains intact gives the attacker the ability to operate the single appliance, not to fork it. Forking requires breaking the measurement gated boundary or perfectly emulating the live appliance state.

7 One Way Birth Fuse

Definition 1 (One Way Birth Fuse)**.** *The one way birth fuse s_{birth} is a secret enrollment preimage formed from RP2350 partition entropy, TROPIC01 birth witness material, host entropy, the online identity commitment, device context, firmware measurement policy, and authority policy. Public enrollment objects commit only to $H(s_{\text{birth}})$. The preimage is destroyed immediately after deriving the initial private state.*

At birth, the appliance derives

$$s_{\text{birth}} = H\big(\texttt{"DSM/anchor/birth-secret/v2"} \parallel \text{trng}_P \parallel \text{wit}_T \parallel \text{nonce}_{\text{host}} \parallel H(pk_{\text{on}}) \\ \parallel \mathsf{device\_id} \parallel \mu_{\text{enrolled}} \parallel \mathsf{policy\_hash}\big),$$

publishes

$$S_{\text{birth}} = H(s_{\text{birth}}),$$

derives the initial partition seal

$$p_0 = \text{HKDF}\big(s_{\text{birth}}, \texttt{"DSM/partition-seal/v2"} \parallel B \parallel \mu_{\text{enrolled}}\big),$$

and destroys

$$s_{\text{birth}} \leftarrow \perp.$$

The partition key pair is generated under the seal. The seal is not a naked storage key. It is a measurement gated authority boundary. Unsealing succeeds only when the executing firmware measurement equals the enrolled measurement committed by policy. If $\mu \neq \mu_{\text{enrolled}}$, the partition returns `UNSEALING_FAILED` or an equivalent hard failure and refuses HostSign.

The chip witness key pair is generated in slot inside TROPIC01; the private half is not exported through the chip API and its at rest protection is the die PUF.

Remark 1. *The enrolled counter value H_0 is not destroyed. Counterparties and auditors need H_0 to evaluate $u = H_0 - H$. The destroyed value is the birth preimage, not H_0.*

8 Anchor Bundle and Counter Birth

Definition 2 (Anchor Bundle)**.** *The anchor bundle B is the immutable enrollment digest of the offline domain:*

$$B = H\big(\texttt{"DSM/anchor-bundle/v2"} \parallel H(pk_{\text{on}}) \parallel st_{\text{pub}} \parallel pk_{\text{chip}} \parallel pk_{\text{host}} \parallel \text{le64}(H_0) \\ \parallel \textsf{device\_id} \parallel \mu_{\text{enrolled}} \parallel \textsf{policy\_hash} \parallel S_{\text{birth}}\big).$$

Here st_{pub} is the chip's static per die public identity, unique to the physical die and rooted in its PUF, and pk_{chip} is the resident witness verification key. Offline bearer releases under a different bundle are not valid successors of state committed to B.

Counter birth is part of enrollment. On a production fresh chip the monotonic counter is initialized to the maximum value; that value is the enrolled H_0 and is bound into B. Because B is immutable and committed in the device SMT from genesis of the offline domain, H_0 cannot be silently re enrolled: a re initialized counter under a new H_0 is a new bundle, a new I_{off}, and a fresh identity with no claim to the old lineage. Implementations must additionally lock the counter initialization capability at provisioning where the hardware supports it, so that re initialization requires visible rebirth rather than a host call.

9 The SMT State Counter

The offline domain adds exactly one structural element to the base DSM device state: a monotone counter committed as a leaf of the device sparse Merkle tree. The addition is required, not decorative. Without a committed counter, the position of a state is implicit in its root history, staleness is decidable only by chain comparison, and hardware synchronization is not even definable. With it, position is a scalar inside the root, and the root consumes it.

The SMT counter is not merely replay metadata. It is the explicit software position of the offline origin inside the consumed device state. A proposed offline transfer must prove that R_i commits the current anchor state leaf (B, h_i, u_i) and that R_{i+1} commits the successor anchor state leaf $(B, h_{i+1}, u_i + 1)$. The receiver therefore checks state consumption and position advancement before accepting the proposal. No live hardware query is needed for that decision.

9.1 Placement and Birth Embedding

The birth is embedded in the tree. The counter leaf is keyed by the anchor bundle B, which commits S_{birth}, H_0, and the enrolled measurement policy. The counter starts at that leaf, at zero, when the offline domain is born, and advances by exactly one per offline bearer commit, atomically with the relationship advance, under the same root replacement.

Definition 3 (Anchor State Leaf)**.** *The anchor state leaf is the device SMT leaf keyed by the anchor bundle whose value commits the offline domain position:*

$$L_i = H(\texttt{"DSM/anchor-state/v2"} \parallel B \parallel h_i \parallel \mathrm{le64}(u_i)).$$

Definition 4 (Offline Frontier Root)**.** *The offline frontier root is a dedicated forward only hash lineage advanced exactly once per offline bearer transfer:*

$$h_{i+1} = H(\texttt{"DSM/anchor-root-advance/v2"} \parallel h_i \parallel D_{i+1}),$$

seeded at birth by h_0. It is the object a counterparty durably tracks for the holder.

9.2 What the Committed Counter Provides

1. **Explicit position.** u_i is the coordinate of the sender state, proven to any receiver by an SMT inclusion proof against R_i. Position is not inferred from history. It is read from the state.
2. **Consumption with the whole state.** The counter lives inside the root. Advancing it is not a side effect of a transfer; it is part of the single root replacement that is the transfer. A parent root therefore admits exactly one indexed successor at position $u_i + 1$: the index cannot advance without the state, and the state cannot advance without the index.
3. **Up front offline validation.** Since the counter position is committed inside the sender root and consumed with the whole state, the receiver can locally reject a stale or incompatible proposal at acceptance time. This is not after the fact double spend detection. It is validation before execution.
4. **Staleness on sight.** Against any known later frontier, an older state is exposed by a scalar comparison of committed counters. No chain walk and no reconciliation round is required to reject a stale proposal.
5. **Proposal comparability.** Two proposals from one origin necessarily collide at position $u_i + 1$ if they claim distinct successors of the same consumed state. The conflict is a decidable leaf collision, not a blockchain style fork requiring global ordering.
6. **The hardware synchronization target.** The physical counter is meaningful only because there is a committed software register for it to track. The invariant $u = H_0 - H$ is a statement about this leaf.

9.3 Counter Synchronized State

Definition 5 (Counter Synchronized State)**.** *The appliance state is counter synchronized when*

$$u_i = H_0 - H$$

holds for the committed u_i and the live raw counter H, outside of an in flight commit. Every offline commit advances both by exactly one, atomically with the relationship advance and the anchor leaf update.

Online transfers do not touch the counter leaf, the frontier root, or the hardware counter. The SMT counter counts offline bearer commits only. Base DSM online operation needs no counter for uniqueness.

10 Software Exclusion and Measurement Gated First Contact

This section states the property the previous specification tried to buy from hardware and separates it into two claims.

First, DSM itself excludes double execution of one consumed state at honest acceptance points. This is software: roots, leaves, chains, counter positions, and signatures are state objects verified by the receiver.

Second, the measurement gated appliance excludes the disconnected first contact fork shape for a single enrolled appliance. This is an identity and emission claim, not a transaction authority claim. A valid first contact release requires σ^{DSM}, σ^{chip}, and σ^{host} over the same message. The host signature cannot be obtained from rogue firmware, it is minted only after the one way counter step that consumes the origin, and enrolled firmware enforces one active state and one prepared record per frontier.

Definition 6 (Whole State Consumption)**.** *A DSM device state is a single resource. An accepted transfer consumes the parent device root R_i entirely and yields exactly one successor R_{i+1}. There is no partial advance. The relationship leaf, the object leaves, and, for offline transfers, the anchor state leaf move together under one root replacement.*

Definition 7 (Offline Origin)**.** *The offline origin of a transfer is the tuple*

$$(R_i, h_i, u_i),$$

the sender device root, the offline frontier root, and the anchor counter coordinate committed by the anchor state leaf of R_i.

Definition 8 (Double Spend)**.** *A double spend exists only if two incompatible proposals become accepted executions of the same consumed origin and the same resource occurrence κ_{res} inside one coherent state history.*

Across relationships, distinct relationships consume distinct occurrences. Committing the same κ_{res} into two relationship advances requires two executed successors of one device root. DSM excludes that by whole state consumption. Each DSM transition names a single recipient, so a single transition core cannot deliver one occurrence to two receivers.

Lemma 1 (Root Singularity)**.** *At any honest receiver, at most one successor of a given adopted parent frontier is accepted and executed.*

Proof. Acceptance persists the successor frontier h_{i+1} for the holder before value is treated as received. A later proposal claiming the previous frontier h_i no longer matches the adopted frontier, and cannot repair the mismatch with a Branch Proof: a verifying hop chain from h_{i+1} back to h_i would close a cycle

$$h_i \to h_{i+1} \to \cdots \to h_i$$

in a forward only hash lineage, which yields a collision in H. □

Lemma 2 (Straight Bilateral Chains)**.** *Within a bilateral relationship, same parent, two receivers is unconstructible as an execution: the relationship is an independent straight hash chain with exactly one receiver and one tip, and the tip is consumed by the advance.*

Proof. The relationship leaf commits the chain tip. An advance replaces that tip under the same leaf key inside the root replacement that is the transfer. Two advances of one tip are two proposed successors of one parent root, both addressed to the chain's single receiver, of which that receiver accepts and executes at most one by Root Singularity. □

Lemma 3 (Position Collision)**.** *Any two distinct proposals from one origin (R_i, h_i, u_i) that claim successor execution must commit the same counter position u_i+1 at the same leaf key under different roots. Identical counter position under an identical bundle is a competing claim to one consumed state position. Any party holding both proposals can decide the conflict by direct leaf comparison, with no reconciliation round.*

Proof. Acceptance requires the successor anchor leaf to commit exactly u_i+1 by checked arithmetic. Two distinct proposed successors present (B, h', u_i+1) and (B, h'', u_i+1) with $h' \neq h''$: two claims to one position under one bundle, decided by direct leaf comparison. □

Theorem 1 (Adopted Frontier Software Exclusion)**.** *For every adversary, no honest receiver accepts and executes two successors of one adopted origin. A second proposal from a consumed origin is rejected at the acceptance predicate by every receiver holding a frontier at or later than that origin's accepted successor. This rejection is up front: it happens before execution, requires no comparison round, requires no reconciliation round, and reads no hardware state.*

Proof. A receiver whose adopted frontier for the holder is at or later than h_{i+1} rejects a second proposal claiming previous frontier h_i at the frontier chaining check. The only repair would be a Branch Proof from a later frontier back to h_i, which would close a cycle in the forward only lineage and yield a collision in H by Root Singularity. Within a bilateral relationship, Straight Bilateral Chains removes the same parent, two receivers shape as an execution. The proof examines adopted frontiers, hash lineage, SMT roots, and DSM validity only. It reads no hardware state. Lemma 1 relies on the receiver persisting the successor frontier before releasing value; this is the acceptance obligation stated in Definition 12 and requirement (2) of the acceptance predicate. □

Theorem 2 (Single Appliance First Contact Exclusion Under Measurement Gate)**.** *Assume:*

1. *the TROPIC01 resident private key is not exported;*
2. *the RP2350 host private key is not exported;*
3. *HostSign is available only when the firmware measurement equals the enrolled measurement;*
4. *the host and chip release witnesses are minted only during Commit, after the single monotonic counter decrement that consumes the origin;*
5. *enrolled firmware is single threaded over offline bearer state, maintains one active frontier, permits at most one prepared record per frontier, exports no release certificate before commit, and recovery only re emits the same committed release;*
6. *no perfect live state clone exists.*

Then a single enrolled appliance cannot emit two distinct valid offline release certificates from the same offline origin (R_i, h_i, u_i), even to disconnected first contact receivers with no prior frontier for the holder.

Proof. A valid offline release requires three witnesses over the same root advance message M_{i+1}: the DSM signature, the chip signature, and the host signature. A disconnected first contact receiver has no adopted frontier against which to reject a first package, but it still verifies the three factor release witness and the SMT update.

A second distinct release from the same origin would require a second distinct message M'_{i+1} binding the same R_i, h_i, and u_i but a different successor root or recipient. That second message requires a valid HostSign(M'_{i+1}) and ChipSign(M'_{i+1}).

By assumption (4), no host or chip witness for the origin exists until Commit, and Commit performs the one monotonic decrement that moves the live counter from u_i to u_i+1. After that decrement, re pinning $H_0 - H = u_i$ fails, so a second Commit at the same origin is refused, and no second witness at position u_i can be minted. Preparing and aborting produces no witness, because Prepare mints nothing. Therefore at most one host witness ever exists for the origin, and it is the one bound into the single committed release.

If the attacker runs enrolled firmware, the firmware exposes only the enrolled offline bearer state machine: one active frontier, one prepared record per frontier, minting confined to Commit, no certificate export before commit, and recovery by re emission of the same committed release. Once the prepared transition for (R_i, h_i, u_i) is committed, the active state advances to $(R_{i+1}, h_{i+1}, u_i+1)$; the old origin is no longer available to prepare or commit a second distinct release. If power fails, recovery re emits the same committed certificate, or, if the decrement completed but the witnesses were not yet minted, mints the witnesses for the single message determined by the committed record and emits that same successor. It never signs a different successor.

If the attacker runs rogue firmware, the measurement gate refuses to unseal the partition signing authority. The attacker cannot obtain σ^{host} for either release, and the package fails receiver acceptance. Rogue firmware may still invoke ChipSign freely, since the chip witness alone is not measurement gated, but a package lacking σ^{host} is not accepted.

The remaining possibilities are export or forgery of a private signing key, a break of H or the signature schemes, or perfect live emulation of all required non exportable state. These are outside the stated assumptions. Therefore two distinct valid releases from one enrolled appliance and one origin are unconstructible. □

Corollary 1 (Hardware Independence of Adopted Frontier Exclusion)**.** *The adopted frontier uniqueness result has no live hardware dependency. Removing live chip reads, host partition reads, physical counter reads, boot tickets, fused heads, MACANDD witnesses, and verifier relay sessions does not affect the receiver's ability to reject a stale or incompatible proposal against an adopted frontier.*

This does not remove the chip and host witness requirements for offline identity acceptance. Hardware serves device identity, strict single appliance emission under the measurement gate, and offline exposure bounding. It contributes nothing to the receiver's transfer validity computation beyond verifiable signatures over the same root advance message.

11 Three Factor Release Witness

Let Δ°_{i+1} be the canonical transition core: action, recipient identity, object identifier, payload, old state proofs from R_i, the counter pair (u_i, u_{i+1}), and the receiver challenge r_R. The core deliberately excludes the successor root. The transition digest is

$$D_{i+1} = H(\texttt{"DSM/transition-digest/v2"} \parallel \mathrm{enc}(\Delta^{\circ}_{i+1})).$$

The dependency order is a clean directed acyclic graph:

$$D_{i+1} \to h_{i+1} \to L_{i+1} \to R_{i+1} \to M_{i+1} \to \sigma^{\mathrm{chip}}_{i+1}, \sigma^{\mathrm{host}}_{i+1}.$$

The root advance message binds both roots on each side:

$$\begin{aligned} M_{i+1} = H\big(&\texttt{"DSM/root-advance/v2"} \parallel B \parallel R_i \parallel R_{i+1} \parallel h_i \parallel h_{i+1} \\ &\parallel \mathrm{le64}(u_i) \parallel \mathrm{le64}(u_{i+1}) \parallel D_{i+1} \parallel \mathrm{recipient} \parallel r_R\big). \end{aligned}$$

The hardware identity witnesses are

$$\sigma^{\mathrm{chip}}_{i+1} = \mathrm{ChipSign}(M_{i+1}), \qquad \sigma^{\mathrm{host}}_{i+1} = \mathrm{HostSign}(M_{i+1}).$$

These witnesses are minted during Commit, after the counter decrement, not during Prepare. Section 13 specifies the timing.

Factor one is the DSM transition itself: Δ°_{i+1} is signed under seed derived DSM device keys per ordinary DSM rules. The release certificate carries the two on device signatures. The DSM signature rides on the transition proof. A complete release therefore proves live cooperation of phone, chip, and host over this exact transition, this exact recipient, and this exact challenge.

The receiver does not verify "the PUF." The receiver verifies a TROPIC resident key signature whose public key is pinned in B. The PUF is the at rest, per die substrate for the resident chip identity. The signature is the portable offline witness.

Definition 9 (Release Certificate and Package)**.**

$$\mathrm{Cert}_{i+1} = (B, R_i, R_{i+1}, h_i, h_{i+1}, u_i, u_{i+1}, D_{i+1}, M_{i+1}, \sigma^{\mathrm{chip}}_{i+1}, \sigma^{\mathrm{host}}_{i+1}, r_R, \mathrm{recipient}),$$

$$\mathrm{Pkg}_{i+1} = (\Delta^{\circ}_{i+1}, \Pi_i, \Pi_{i+1}, \mathrm{Cert}_{i+1}, \mathrm{BranchProof}\ \textit{optional}, U\ \textit{genesis only}),$$

where Π_i, Π_{i+1} are the SMT inclusion and update proofs for the anchor state leaf and for the relationship advance from R_i to R_{i+1}.

Definition 10 (Branch Proof)**.** *When the receiver's stored frontier for the holder is h_k with $k < i$, the release carries the ordered hop certificates for*

$$h_k \to \cdots \to h_i,$$

each hop being a prior certificate whose σ^{chip} and σ^{host} verify and whose roots chain. The receiver advances its frontier along the sender's own witnessed lineage. Interleaved counterparties therefore do not require online readmission.

12 Synchronized Anchor Counter

The authoritative register is the SMT counter: committed position, consumed with the state. This section specifies the physical tracker bound to it. The physical counter has three jobs. None of them is acceptance authority.

1. **Non rewind floor.** The hardware counter is monotonic and non resettable under the enrolled H_0. A device image restored to an older state carries an anchor leaf with $u < H_0 - H$. The appliance refuses offline operation on it under the recovery rule, and any authenticated diagnostic counter read exposes it.

2. **Stale image and split bricking.** Of two divergent lineages claiming the same identity, at most one is counter consistent with the live chip. Reconciliation or any authenticated diagnostic read distinguishes them; the inconsistent lineage is bricked.

3. **Offline exposure cap.** Policy bounds

$$u - u_{\mathrm{rec}} \leq W,$$

   where u_{rec} is the counter coordinate at the holder's last reconciliation and W is the policy window bound into policy_hash. Lifetime capacity is bounded by H_0, the exposure cap, and flash wear.

The hardware counter does not determine validity. It mirrors the committed software position and provides an independent non rewind witness.

The exposure cap W is meaningful only together with the measurement gate. Enrolled firmware is what ensures one physical decrement corresponds to one committed release, that witnesses are minted only after the decrement, and that releases are not exported before commit. If the measurement boundary is broken, the system has left the single appliance model and entered the perfect live clone boundary.

Commit discipline: prepare moves nothing and mints nothing; commit re pins $H_0 - H = u_i$ with checked arithmetic, performs exactly one counter update, then mints the chip and host witnesses over M_{i+1}; $H = 0$ returns `EXHAUSTED_ONLINE_ONLY`; a second commit against the same prepared transition is refused. The counter decrement is exactly what the July 8, 2026 silicon run executed.

Authenticated counter audit is outside the offline acceptance predicate. It is diagnostic or post acceptance policy evidence only. It cannot make an invalid release valid, cannot repair a frontier mismatch, cannot replace any of the three signatures, and cannot upgrade the validity of a package. The default receiver path performs no live counter read.

13 Offline Transfer Protocol

The appliance has three transfer states: Ready, Prepared, Committed. Offline bearer mode is enabled only when the firmware measurement matches the enrolled policy. This gate is load bearing for the host witness. It is not an optional implementation check.

13.1 Prepare

The receiver checks the proposed transfer at the human and DSM level and supplies a fresh challenge

$$r_R \xleftarrow{\$} \{0,1\}^{256}.$$

The DSM SDK constructs or simulates the successor transition first. It computes Δ°_{i+1}, D_{i+1}, h_{i+1}, L_i, L_{i+1}, the parent root R_i, the successor root R_{i+1}, and the SMT proof material Π_i, Π_{i+1}.

The appliance is not the SMT engine. It checks:

1. firmware measurement equals enrolled policy;
2. status is Ready;
3. the supplied parent root and frontier match the active appliance state;
4. counter synchronization $u_i = H_0 - H$;
5. the supplied fields are internally consistent with the current frontier and policy.

It recomputes M_{i+1} from the supplied roots and transition digest and writes a durable prepared record containing M_{i+1} and the staged transition fields. **Prepare mints no signatures.** No counter has moved. No release exists. Exactly one prepared record may exist per frontier.

This is the load bearing change from the previous revision. The chip and host witnesses are not produced at Prepare and are therefore never present in flash while the origin is still spendable. A malicious host cannot prepare, abort, and re prepare to harvest multiple signed certificates for one origin, because Prepare produces nothing to harvest. Single use of the host witness is a timing property of the one way counter, not a confidentiality property of at rest storage.

13.2 Commit

The appliance stores the committed candidate durably with

$$\texttt{counter\_committed} = \text{false},$$

re pins $H_0 - H = u_i$, and if the check holds and $H > 0$, issues the single counter update

$$H \leftarrow H - 1,$$

marks

$$\texttt{counter\_committed} = \text{true},$$

then mints $\sigma^{\text{chip}}_{i+1} = \text{ChipSign}(M_{i+1})$ and $\sigma^{\text{host}}_{i+1} = \text{HostSign}(M_{i+1})$ over the message fixed by the committed record, and stores them durably. The witnesses come into existence only after the irreversible counter step. If the re pin fails, the operation downgrades to online recovery without moving the counter and without minting.

13.3 Emit and Finalize

The appliance exports the committed certificate only after the counter commit and witness minting. The SDK assembles Pkg_{i+1} by attaching the transition core, DSM signature material, SMT proofs, the certificate, and any Branch Proof. Finalization requires

$$\mathrm{Active}.u + 1 = H_0 - H$$

and writes the successor active state. If power fails before finalize, recovery re emits the same committed release and finalizes the same successor; if the decrement completed but minting did not, recovery mints the single message determined by the committed record and emits that same successor.

13.4 First Transfer Offline

A first transfer to a new counterparty requires no counter disclosure round trip. Under the final protocol, the receiver supplies r_R and enrollment material, receives Pkg_{i+1} together with the upgrade certificate U and bundle disclosure, and verifies the package self contained.

Genesis adoption persists the release's next frontier root only after all predicate checks pass and the receiver's own canonical commit succeeds. Policy may require online first contact for high value relationships; that is a policy knob, not a protocol change. The prepared and proceed disclosure round trip of the previous specification is removed; it existed only to serve receiver witnessed counter reads, which are themselves removed.

Under the measurement gate, a single enrolled appliance cannot produce two distinct first contact releases at the same origin. Enrolled firmware mints witnesses only at commit, permits one prepared record per frontier, and exports only after commit. The one host witness that can exist for an origin is minted after the one way counter step. Rogue firmware cannot unseal HostSign.

Implementation staging note. A staging implementation may keep first transfer TOFU admission while producer, receiver, proof, and signature paths are wired to v2. That is an implementation staging choice only. The final protocol owns the real dual identity upgrade certificate and genesis upgrade proof.

14 Receiver Acceptance Predicate

Definition 11 (Accepted Frontier)**.** *For each enrolled holder, the receiver maintains a durable adopted offline frontier root. A proposal must chain from it: either the proposal previous frontier equals it exactly, or the proposal carries a verifying Branch Proof from it to the proposal previous frontier. If no frontier exists, the relationship is at genesis and genesis adoption rules apply.*

Definition 12 (Offline Acceptance)**.** $\mathrm{Accept}_{\mathrm{off}}(\mathrm{Pkg}_{i+1}) = 1$ *iff all of the following hold:*

1. *all encodings are canonical;*
2. *the proposal chains from the receiver's adopted frontier for the holder, directly or by Branch Proof, and the receiver persists the successor frontier before treating value as received;*
3. Π_i *proves* R_i *commits* (B, h_i, u_i) *through the anchor state leaf;*

4. *D_{i+1} recomputes from Δ°_{i+1}, and $h_{i+1} = H($"DSM/anchor-root-advance/v2"$\,\|\, h_i \,\|\, D_{i+1})$;*

5. *the claimed next anchor counter is $u_i + 1$, using checked arithmetic;*

6. *r_R is the challenge supplied by this receiver, and the recipient field names this receiver;*

7. *M_{i+1} recomputes from the bound fields;*

8. $\mathrm{ChipVerify}(pk_{\mathrm{chip}}, M_{i+1}, \sigma^{\mathrm{chip}}_{i+1}) = 1$ *with pk_{chip} pinned in B;*

9. $\mathrm{HostVerify}(pk_{\mathrm{host}}, M_{i+1}, \sigma^{\mathrm{host}}_{i+1}) = 1$ *with pk_{host} pinned in B;*

10. *the DSM transition proof verifies: Δ°_{i+1} is validly signed under the holder's DSM device keys, and $R_i \to R_{i+1}$ is the correct SMT update including the relationship advance and the anchor leaf update;*

11. *Π_{i+1} proves R_{i+1} commits $(B, h_{i+1}, u_i + 1)$ through the anchor state leaf;*

12. *the transfer gives the claimed object or value to the receiver, and the authority policy hash matches;*

13. *if this is final protocol genesis adoption, the upgrade certificate U verifies and binds I_{off} to I_{on};*

14. *no known compromise or policy event invalidates the anchor.*

No live chip read, live host read, raw counter read, relay session, verifier slot read, passthrough, boot ticket, MACANDD output, or fused anchor head appears in $\mathrm{Accept}_{\mathrm{off}}$. The receiver trusts public DSM verification, its own challenge, and the two pinned hardware identity witnesses. The receiver does not trust host reported state, copied wallet files, or any unauthenticated field carried inside the release.

15 Power Loss and Recovery

Power may fail between any two operations. The recovery rule is: if a committed release exists, re emit that same release and finalize that same successor. Recovery never signs a new release for the same counter step. Because witnesses are minted only at or after the counter decrement, a prepared but not committed state has no witnesses to lose and no counter step to reverse; a committed state either already holds its witnesses or has performed the decrement that fixes the single message those witnesses cover.

```
recover(H0, H, Active):
    live = checked_sub(H0, H)
    if live == ERROR:
        return COUNTER_MISMATCH
    if firmware_measurement_invalid():
        return DOWNGRADE_ONLINE

    if Active.status == COMMITTED:
        rec = Active.record
        if rec.counter_committed == TRUE:
            if rec.next_u != live:
                return DOWNGRADE_ONLINE
```

```
        if not signatures_present(rec):
            # decrement done, witnesses not yet minted (power loss in the window);
            # M is fully determined by rec, so exactly one message is signable
            mint_release_witnesses(rec)    # ChipSign, HostSign over rec.M
        return REEMIT_COMMITTED(rec.next_root)
    if rec.counter_committed == FALSE:
        # decrement not yet applied; origin is still live at position 'live'
        if live == rec.prev_u:
            if rec.prev_root != Active.root:
                return DOWNGRADE_ONLINE
            if rec.bundle != Active.bundle:
                return DOWNGRADE_ONLINE
            counter_update()               # single decrement
            mark_counter_committed(rec)
            mint_release_witnesses(rec)     # mint only after the decrement
            return REEMIT_COMMITTED(rec.next_root)
        return DOWNGRADE_ONLINE
    return DOWNGRADE_ONLINE

if Active.status == PREPARED:
    # Prepare minted nothing and consumed no counter step
    if Active.u != live:
        return DOWNGRADE_ONLINE
    return CANCEL_TO_READY                  # safe: no witness minted, no step burned

if Active.status == READY:
    if Active.u < live:
        return FAIL_CLOSED_STALE_IMAGE
    if Active.u > live:
        return COUNTER_MISMATCH_OR_DOWNGRADE_ONLINE
    if H == 0:
        return EXHAUSTED_ONLINE_ONLY
    return ACCEPT(Active.root)

return DOWNGRADE_ONLINE
```

The `Active.u < live` branch identifies a restored stale host image: the chip has already consumed more offline counter steps than the restored image records. Offline operation fails closed. The `Active.u > live` branch is impossible for the honest appliance unless the host state is ahead of the live chip, the wrong chip is attached, or a counter invariant has been violated; it does not resume offline bearer operation and must route to mismatch handling or online recovery.

The PREPARED branch is a clean cancel back to Ready: because no witness was minted and no counter step was consumed at Prepare, there is nothing to harvest and nothing to reverse. The re emission window inside COMMITTED covers only the single message that the committed record already fixes, so recovery can mint the missing witnesses without any freedom to sign a different successor.

If firmware measurement fails, recovery does not expose signing authority and does not attempt offline continuation. It downgrades to online recovery or authority rotation.

16 Online Domain Operation

Online transfers proceed under I_{on} with DSM signatures alone: no chip, no host key, no counter, no anchor leaf. Value held by the device is domain agnostic at the base DSM layer because the device state is one resource; the relationship, not the value itself, carries the domain tag.

A production wallet may add an explicit device bound offline allocation leaf as a value earmark for cash like UX: online available balance can be debited into a device bound offline allocation and later reconciled back online. This is a compatible extension to the base device SMT discipline. It does not change receiver acceptance, the three factor witness, the SMT counter, or the hardware role. It simply makes the "cash withdrawn from account" model explicit and limits device loss exposure to the loaded slice.

A holder with both domains spends into an offline relationship through the offline protocol and into an online relationship through ordinary DSM operation, from the same device root lineage. A holder with only the online domain participates fully in online DSM. This is the zero hardware entry path: a phone and a seed. The offline domain is added later, if ever, by the upgrade ceremony, without disturbing any existing relationship.

17 Threat Model

Definition 13 (Software Clone)**.** *A software clone holds all host readable state: seed, keys, chain history, local database, cached proofs, and application state. It does not hold the TROPIC01 resident private key or counter state, and does not hold the RP2350 partition sealed host private key.*

Definition 14 (Partial Hardware Compromise)**.** *A partial hardware compromise yields the non exportable state of the chip, or of the host partition, but not both, and not the seed.*

Definition 15 (Full Factor Adversary Under Enrolled Firmware)**.** *A full factor adversary under enrolled firmware controls the phone and seed side and has live access to the enrolled appliance, including the chip and host signing services, but cannot replace firmware without failing measurement, cannot extract the host private key, cannot extract the chip private key, and cannot make the enrolled firmware violate its single active state discipline.*

Definition 16 (Measurement Boundary Breach)**.** *A measurement boundary breach permits firmware not matching the enrolled policy to invoke HostSign, export host signing material, extract a minted host witness before the committing counter step, or bypass the single active state appliance discipline while still producing host signatures accepted under pk_{host}. This is outside the normal full factor adversary model and is grouped with perfect live state cloning.*

Definition 17 (Perfect Live State Clone)**.** *A perfect live state clone extracts and perfectly emulates the exact current non exportable state of all three factors, or otherwise bypasses the measurement gated signing boundary so that multiple independent devices can produce accepted offline releases as if they were the same enrolled appliance. No offline only protocol can distinguish such an emulation from the original. This is a stated limit, not a defended case.*

Behavior of an online domain software clone with the same seed on two hosts: both instances hold I_{on}. Neither can double spend into the same honest receiver; whole state consumption admits one accepted and executed successor per parent at an honest acceptance point. The two instances may originate incompatible proposals and diverge at their first independent attempts. The identity splits. Tripwire exposes the split at first cross contact or reconciliation. Counterparties reject both lineages pending readmission. The clone buys chaos and self bricking, not value creation. Confidentiality of the copied state is lost; integrity of the network is not.

PUF fault injection, for example laser assisted extraction, has been demonstrated against PUF constructions in laboratory settings. The design does not rest the offline identity on the PUF alone for exactly this reason: extracting the chip factor still leaves the attacker without seed and host. Policy should treat confirmed physical compromise of the chip as grounds for authority rotation.

18 Tripwire Composition

Tripwire supplies exposure of incompatible proposals on reconciliation. Each device commits relationship tips into its device SMT; a valid receipt proves adjacency from an old root to a new root and binds the relevant tip update. Tripwire does not make disconnected receivers instantly aware of each other. It guarantees that conflicting accepted tips cannot survive comparison as one coherent history.

Assumption 1 (Tripwire Security)**.** *Assume H is collision resistant and DSM signatures are secure against chosen message forgery. Then two distinct accepted successors of the same predecessor cannot survive reconciliation without a hash collision, a signature forgery, a violated DSM predicate, or a measurement boundary breach that exits the single appliance model.*

Tripwire is not the source of double execution prevention. Whole state consumption is the source of execution uniqueness. Tripwire makes divergence non mergeable, propagates evidence of incompatible proposals, and bricks split identities once the incompatibility is known.

No non local fork relay is part of this protocol. Fork proofs may be carried where protocol flows already require them, such as Branch Proofs, reconciliation, tombstone handling, or recovery exposure processing. The base offline release does not carry a fork list gossip layer, recent observation relay, or social graph propagation mechanism.

19 Security Claims

Theorem 3 (Birth Non Recreation)**.** *Public enrollment data is insufficient to recreate the offline identity on new hardware.*

Proof. B commits $S_{\text{birth}} = H(s_{\text{birth}})$, st_{pub}, pk_{chip}, pk_{host}, H_0, and μ_{enrolled}. The birth preimage is destroyed; the chip private key is not exported through the chip API and is PUF rooted per die; the host key is partition sealed and measurement gated. A party with only public data must invert H, extract non exportable state, or bypass the measurement gate. □

Transfer uniqueness at adopted frontiers is the software theorem of Section 10. Strict first contact offline uniqueness for one enrolled appliance is the measurement gated emission theorem of Section 10. The claims below are the identity and durability claims that hardware does carry.

Theorem 4 (Clone Exclusion, Offline Domain)**.** *A party lacking any one of the three factors cannot produce an accepted offline bearer release for an enrolled identity, except by forging a signature, breaking H, or perfectly emulating the missing factor in the perfect live state clone boundary.*

Proof. Acceptance requires a valid DSM transition under seed derived keys, σ^{chip} under the resident chip key pinned in B, and σ^{host} under the partition sealed key pinned in B, all over the same transition package. Absent a factor, the corresponding signature cannot be produced. In particular, a software clone holding every host readable byte produces no offline release at all. □

Theorem 5 (Online Domain Scope)**.** *Compromise of the phone alone yields the online domain only.*

Proof. The seed suffices for I_{on} operation by construction. Offline acceptance additionally requires the chip and host signatures. □

Theorem 6 (Stale Image Rejection)**.** *An appliance image restored to an earlier state cannot resume offline bearer operation.*

Proof. The restored image commits $u_{\text{old}} < H_0 - H$ since the physical counter cannot rewind and has already consumed more offline steps than the restored host image records. Recovery's Ready branch refuses offline operation on Active.$u < H_0 - H$ in the stale direction. Any counterparty holding a later frontier also rejects the image's proposals on the frontier check; any authenticated diagnostic counter read exposes the desynchronization. □

Theorem 7 (Identity Split Bricking)**.** *No receiver accepts and executes proposals from both lineages of a split identity. A receiver whose adopted frontier lies on one branch rejects the other branch at the acceptance predicate, with no comparison round and without knowing a split exists. Any party holding proposals from both branches decides the split on sight by leaf comparison, and honest counterparties thereafter refuse both lineages pending readmission.*

Proof. The two lineages diverge at some parent and commit the same position at the same leaf key under one bundle. A receiver whose adopted frontier lies on one branch rejects any proposal from the other at the frontier chaining check: a Branch Proof connecting the two branches would have to pass back through the divergence parent and close a cycle in a forward only lineage, yielding a collision in H. This rejection is a property of acceptance and requires no knowledge that a split exists. A party holding proposals from both branches holds two claims to one position under one bundle, decided by direct leaf comparison with no reconciliation round.

Both colliding certificates replicate through every Branch Proof crossing the divergence hop, so the set of deciding parties grows with every transfer on either branch. Tripwire is the contact channel by which parties holding one branch or neither come to hold both; it performs no rejection. Once the split is decided, honest counterparties refuse both lineages pending readmission under rotated authority. For the offline domain, at most one branch is counter consistent with the live chip at any time, so an authenticated diagnostic read exposes at least one branch as desynchronized; it does not certify the survivor as honest, since the holder chooses which branch consumes decrements. □

Theorem 8 (Measurement Gated Full Stack Bound)**.** *Against a full factor adversary confined to enrolled firmware, offline first contact forks are unconstructible. The adversary may operate the enrolled appliance and choose the next valid successor, but cannot obtain two distinct valid releases from one origin.*

Proof. This is the Single Appliance First Contact Exclusion theorem applied to a full factor adversary whose firmware boundary remains intact. The adversary has access to the live appliance as an oracle, but the oracle is the enrolled state machine: one active frontier, one prepared record per frontier, witness minting confined to Commit after the counter decrement, no release export before commit, one counter decrement per commit, and recovery by re emission of the same committed release. A second distinct release at the same origin would require a second host witness over a distinct M_{i+1} at position u_i; but no witness is minted before the decrement, and after the decrement the re pin at u_i fails. The remaining routes are export of private signing material, extraction of a minted witness before its committing step, or rogue firmware with access to HostSign. Each violates the model. □

Theorem 9 (Measurement Breach Residual Bound)**.** *If the measurement boundary is broken or a perfect live state clone exists, the protocol does not claim strict offline first contact uniqueness. In that boundary case, conflicting proposals cannot merge into one coherent DSM history; they are exposed and bricked on comparison, and their count between reconciliations is bounded by policy window W only if the counter discipline remains enforced.*

Proof. A measurement breach or perfect live state clone exits the single appliance model. If such an attacker can produce signatures for distinct successors of one origin, the proposals collide at the same anchor counter position under the same bundle and cannot be jointly represented as one linear DSM history. Any receiver pinned to one branch rejects the other by frontier chaining. Any party holding both proposals decides the split by direct leaf comparison. If the physical counter discipline remains enforced, each committed release consumes one counter step and policy bounds steps between reconciliations by W. If the attacker also bypasses that discipline, the system is outside the hardware identity model and must rotate authority. □

Theorem 10 (Replay Idempotence)**.** *Re emitting the same release package does not create a second spend.*

Proof. The package binds the same roots, digest, challenge, recipient, and witnesses. A receiver that accepted it recognizes the identical transition; re emission is duplicate delivery, not a distinct successor. □

Theorem 11 (Recoverable Commit)**.** *If the counter moves for a release whose committed candidate is durable, recovery either re emits the same release or downgrades to online recovery; it never signs a different release for the same counter step.*

Proof. By inspection of `recover`: every branch under COMMITTED returns re emission of the stored record, mints only the single message the committed record already fixes, or downgrades; no branch derives a different successor for the same counter step. □

20 TLA+ Sketch

The reduced model captures the software uniqueness invariant, the counter synchronization invariant, and the single appliance emission invariant. Sessions, pre evidence, and proceed gates of the previous model are gone. `NoTwoAcceptedSameOrigin` is the adopted frontier software theorem in mechanical form. `CounterSync` is the counter synchronized state definition. `NoTwoEmittedSameOrigin` is the measurement gated single appliance theorem in mechanical form. Emission is modeled at Commit, guarded by the counter decrement, which mirrors witness minting occurring only after the one way counter step.

```
VARIABLES H, Active, Frontier, Delivered, Emitted

LiveU == H0 - H

Init ==
    /\ H = H0
    /\ Active = [status |-> "ready",
                 u |-> 0,
                 root |-> R0,
                 h |-> h0,
                 record |-> NULL]
    /\ Frontier = [r \in Receivers |-> NULL]
    /\ Delivered = {}
    /\ Emitted = {}

MeasurementOK == FirmwareMeasurement = EnrolledMeasurement

Prepare(pkg) ==
    /\ MeasurementOK
    /\ Active.status = "ready"
    /\ pkg.prev_root = Active.root
    /\ pkg.prev_h = Active.h
    /\ pkg.u = Active.u
    /\ pkg.next_u = Active.u + 1
    /\ Active' = [Active EXCEPT
                  !.status = "prepared",
                  !.record = pkg]
    /\ UNCHANGED <<H, Frontier, Delivered, Emitted>>

Commit ==
    /\ MeasurementOK
    /\ Active.status = "prepared"
    /\ LiveU = Active.u
    /\ H > 0
    /\ H' = H - 1
    /\ Emitted' = Emitted \cup {Active.record}    \* witness minted post-decrement
    /\ Active' = [Active EXCEPT !.status = "committed"]
    /\ UNCHANGED <<Frontier, Delivered>>

Accept(r) ==
    /\ Active.status = "committed"
    /\ LET pkg == Active.record IN
       /\ Frontier[r] \in {NULL, pkg.prev_h}
       /\ Delivered' = Delivered \cup {pkg}
       /\ Frontier' = [Frontier EXCEPT ![r] = pkg.next_h]
       /\ Active' = [status |-> "ready",
                     u |-> pkg.next_u,
                     root |-> pkg.next_root,
                     h |-> pkg.next_h,
                     record |-> NULL]
    /\ UNCHANGED <<H, Emitted>>

NoTwoAcceptedSameOrigin ==
    \A p, q \in Delivered :
        (p # q) =>
            ~(p.prev_root = q.prev_root /\
              p.prev_h = q.prev_h /\
              p.u = q.u)
```

```
NoTwoEmittedSameOrigin ==
    \A p, q \in Emitted :
        (p # q) =>
            ~(p.prev_root = q.prev_root /\
              p.prev_h = q.prev_h /\
              p.u = q.u)

CounterSync ==
    (Active.status = "ready") => (Active.u = H0 - H)
```

This sketch is an abstraction over an honest single threaded appliance. It corroborates the emission and counter invariants at the state machine level; it assumes the signature and hash primitives rather than modeling an unforgeability adversary. A model that mechanizes Theorem 1 against arbitrary proposals requires an explicit adversary action that injects well formed packages under an unforgeability assumption and a receiver side invariant of frontier chaining rejection.

21 Wire Protocol

The wire protocol carries the transition core, proofs, and the release certificate. The release certificate carries the two hardware identity signatures. The DSM signature is part of the transition proof. The counter evidence messages and the first transfer disclosure messages of the previous specification are deleted.

```
syntax = "proto3";
package dsm.anchor.v2;

message UpgradeCertificate {
    bytes anchor_bundle    = 1;
    bytes online_pubkey    = 2;
    bytes online_signature = 3;
}

message ReleaseCertificate {
    bytes  anchor_bundle              = 1;
    bytes  sender_device_root_before = 2;
    bytes  sender_device_root_after  = 3;
    bytes  frontier_root_before      = 4;
    bytes  frontier_root_after       = 5;
    uint64 anchor_counter            = 6;
    uint64 next_anchor_counter       = 7;
    bytes  transition_digest         = 8;
    bytes  root_advance_message      = 9;
    bytes  chip_signature            = 10;
    bytes  host_signature            = 11;
    bytes  receiver_challenge        = 12;
    bytes  recipient                 = 13;
}

message BranchHop {
    ReleaseCertificate cert = 1;
}

message OfflineRelease {
    bytes              canonical_transition_core = 1;
```

```
    bytes               smt_proofs                = 2;
    ReleaseCertificate certificate               = 3;
    repeated BranchHop branch_proof              = 4;
    UpgradeCertificate upgrade                   = 5;
}
```

No CounterAudit message is part of OfflineRelease. Diagnostic counter reads, if implemented, live outside acceptance.

Removed relative to `dsm.anchor.v1`: CounterEvidencePre, CounterEvidencePost, CounterAdvanceEvidence, AnchorDisclosure-as-gate, BilateralBearerPrepared, BilateralBearerProceed, BootTicket, fused anchor head, MACANDD witness frames, BLE frames 16 and 17.

22 Reference Implementation Requirements

1. `accept.rs` keeps the anchor state proofs from R_i and R_{i+1} and the checked $u_i + 1$ rule.
2. `accept.rs` replaces all FROM to TO counter verification with the two hardware identity signature verifications over M_{i+1} plus the ordinary DSM transition signature verification. The counter appears in acceptance only as the committed pair (u_i, u_{i+1}) inside signed data.
3. No verifier read API, passthrough, live chip session, raw counter read, or audit transcript gates default acceptance. Any counter audit code must live outside the protocol authority path and must never make an invalid release valid.
4. Chip provisioning at birth: generate the Ed25519 witness key pair in slot inside TROPIC01; export pk_{chip} and st_{pub}; bind both into B. Clean production birth uses the default key slot. A used chip whose default slot is unavailable requires an explicit override and must produce a different bundle.
5. Counter birth: initialize the monotonic counter to H_0 at enrollment; lock the initialization capability where hardware supports it; bind H_0 into B.
6. Host provisioning at birth: generate the partition key pair under the seal p_0; never export the private half.
7. The partition seal must be measurement gated. Offline bearer mode is enabled only when the firmware measurement matches enrolled policy. This gate must be enforced before any call to HostSign. Failure to unseal returns `UNSEALING_FAILED` or equivalent. There is no fallback to online signing, no debug bypass, and no host provided override.
8. The chip and host witnesses must be minted only during Commit, after the monotonic counter decrement, never during Prepare. Prepare stores only the root advance message and the staged transition fields, with no signatures. No valid host witness for an origin may exist before the counter step that consumes that origin. This makes single use a timing property of the one way counter rather than a confidentiality property of at rest storage, and it closes any prepare, abort, and re prepare harvesting shape. A release certificate is exported only after Commit.
9. MACANDD carries no protocol role. Implementations must not treat MACANDD slot state as lineage: slot state is a pure function of call input and slot index and is host restorable,

as demonstrated by the vendor reference PIN flow, which reinitializes consumed slots by replaying a derived input. MACANDD may be used for local unlock policy per the vendor application note; that use is outside this specification.

10. One pending prepared record per frontier; confirm consumes the stored committed release and never rebuilds a different release for the same prepared step.

11. Production hardware paths must be release builds. If debug assertions are enabled, the firmware or harness must fail before touching provisioning, prepare, commit, recover, audit, or signing paths.

12. Firmware measurement gates offline bearer mode. BOOTSEL must not be normally accessible; it must be internal or behind a recessed tamper evident service feature. The USB connector may remain externally accessible as the power and service interface.

13. Domain separation: relationship establishment records the domain tag; final offline domain establishment requires and verifies the upgrade certificate; online paths must be incapable of emitting offline releases.

14. The SDK, not the appliance, owns device SMT root computation and proof attachment. The producer must compute or simulate the successor first, obtain the real R_i, R_{i+1}, Π_i, and Π_{i+1}, and then request the commit that mints signatures over the root advance message. Zero root stamping followed by restamping is invalid.

15. Existing pins are a clean schema boundary. The v2 pin contains pk_{chip} and pk_{host}. Old verifier slot, chip static public key as authority key, boot head, and fused anchor fields must not remain live in the authority path.

23 Verification Plan

The implementation must include tests for:

1. honest offline transfer with DSM transition signature, chip signature, and host signature succeeds;

2. release missing any one of the three required signatures is rejected;

3. chip signature under a key not pinned in B is rejected;

4. host signature under a key not pinned in B is rejected;

5. stale frontier release without Branch Proof is rejected; with verifying Branch Proof it is accepted;

6. second release claiming an already consumed frontier at the same receiver is rejected before execution;

7. same parent cannot execute twice against one adopted frontier;

8. enrolled firmware cannot emit two releases from the same origin, even to first contact receivers;

9. rogue firmware fails to unseal HostSign and therefore cannot produce an accepted release;

10. no host or chip witness exists for an origin before its counter commit; Prepare produces no signatures; prepare, abort, and re prepare at one origin mints nothing;
11. recovery re emits the same committed release, mints only the single message fixed by a committed record when the decrement completed before minting, and never signs a different release for the same counter step;
12. receiver challenge mismatch and recipient mismatch are rejected;
13. post state anchor leaf that does not commit $u_i + 1$ is rejected;
14. restored stale image fails closed in recovery when `Active.u` is behind live counter derived u;
15. online path cannot emit an offline release; final offline establishment without upgrade certificate fails;
16. exposure cap W enforcement at the appliance and its check at reconciliation;
17. replay of an accepted package is recognized as duplicate delivery;
18. chip signatures verify by public key and are not assumed deterministic byte strings;
19. producer generates real R_i, R_{i+1}, Π_i, and Π_{i+1} before appliance commit;
20. receiver acceptance contains no live counter read and no passthrough path.

Silicon status.

- **V1.** In slot Ed25519 key generation with TROPIC01, signing over a 32 byte M, verification against the exported public key, erase, clean empty slot behavior, and power cycle persistence were validated on July 9, 2026. V1 is complete.
- **V2.** End to end three signature release acceptance on the Pico 2 W appliance against the reference verifier remains the next live integration item.
- **V3.** Counter initialization lock behavior after enrollment on a production fresh chip remains to be validated where the hardware exposes that lock.
- **V4.** Measurement gated host signing must be validated: enrolled firmware can call HostSign, rogue firmware receives `UNSEALING_FAILED`, witness minting occurs only after the counter decrement, and release export is impossible before commit.

24 Silicon Validation

24.1 July 8, 2026 Counter Discipline Run

The July 8, 2026 hardware validation was executed on physical RP2350 plus TROPIC01 appliances running release builds with debug assertions disabled, using the one commit Phase 3 harness and authenticated caged verifier slot reads. Two distinct anchors were exercised:

Run	Chip	Birth mode	H_0	Final state
A	used chip A	bench adopted	4294967281	$u = 1$, $H = 4294967280$
B	clean chip	production fresh	4294967294	$u = 1$, $H = 4294967293$

Clean chip anchor identifier:
`8RPYNMX7G26GNTQB3BKFK4QXEKJEW0CVTSARZ8BVB2T9ZADGQ3V0`.

Used chip A anchor identifier:
`1SZ0KC8H6WJ0JYMX2YDD49VM8RX4R6ZGKCK692FDBDZ186HYN15G`.

Under the claims of this specification, the run validates:

1. counter birth: a virgin chip's uninitialized counter was initialized to the maximum, read back, and adopted as H_0;
2. prepare moves nothing: witness and certificate formation left the counter untouched;
3. commit discipline: exactly one counter update per commit; H decremented by exactly one;
4. second commit refusal: the appliance refused a second commit against the same prepared transition, and H remained stable after the refusal;
5. synchronization: $u = H_0 - H$ held at every observation point on both chips;
6. per die identity: the two chips presented distinct static public identities, the empirical basis of the PUF rooted per die uniqueness that the offline identity fuses.

What the run does not validate under the new claims: the receiver witnessed FROM and TO evidence path it was originally built to demonstrate. That path is removed from the protocol. The reads survive only as diagnostic audit outside acceptance.

24.2 July 9, 2026 Resident Ed25519 ECC Gate Probe

The July 9, 2026 ECC gate probe was executed on a fresh board with an untouched TROPIC01 slot 0. The monotonic counter was never touched. The probe deliberately exercised the resident Ed25519 witness key path required by ChipSign.

Gate	Result
Empty slot read	clean error; no enrollment poison from bogus bytes
Ed25519 key generation	`ecc_key_generate(slot 0, Ed25519)` succeeded
Public key read stable	pk_{chip} exported from slot 0
On die sign, off chip verify	Ed25519 `verify_strict` passed
Erase	erase succeeded; slot returned to empty state
Power cycle persistence	public key after reboot matched byte for byte
Signature behavior	signatures over the same key and message differed; both verified
Counter safety	monotonic counter was not read, initialized, decremented, or otherwise touched

The observed resident public key for the provisioned board was:
`FYKMTS7RQTNMQM3173BYW0VP306E2KWDTGSXFGNCVKZFXYSX2Q8G`.

This is a public verification key. It is not a secret.

The probe proves the generate once path required by birth:

1. a virgin slot fails cleanly before key generation;
2. production silicon permits Ed25519 key generation in the chosen slot;
3. the resident key signs the root advance message class;
4. off chip receivers can verify the chip signature against the exported public key;
5. the key persists across power cycle;
6. erase is reversible for this batch;
7. signature bytes are not a stable protocol object.

This retires the former V1 bench item. The remaining live validation is the full end to end three signature release through the reference verifier, any hardware supported counter initialization lock behavior, and the measurement gated host signing boundary.

25 Changes from the Previous Specification

Identified during adversarial review, July 2026.

1. **The counter is demoted from authority to floor.** A scalar counter read proves the chip's coordinate, freshly, over a session. It never proves the chip reached that coordinate because of this transition: the counter primitive does not see M_{i+1}, and binding metadata wrapped around a scalar is receiver asserted, not chip attested. Two receivers could witness the same physical decrement as their own. The apparatus built on those reads, CounterEvidencePre, CounterEvidencePost, CounterAdvanceEvidence, and predicate checks 18 through 23 of the previous acceptance definition, is removed.
2. **Transfer uniqueness moves to software, where it already lived.** The device state consumes itself as a whole; one parent, one accepted and executed successor at an honest acceptance point. The counter argument of the previous specification is replaced by the adopted frontier software theorem and the SMT position collision lemma.
3. **First contact offline uniqueness is now an enrolled appliance emission property.** The prior revision allowed conflicting first contact proposals under full stack compromise and bounded them by exposure, non mergeability, and bricking. This revision tightens the model: while the measurement gated firmware boundary remains intact, the single enrolled appliance cannot emit two distinct valid releases from one origin. Rogue firmware cannot unseal HostSign. Witnesses are minted only after the counter decrement. Enrolled firmware enforces one active state and one prepared record per frontier. The remaining fork case is measurement boundary breach or perfect live state clone, explicitly outside offline distinguishability.

4. **The SMT committed counter becomes the authoritative position register.** The previous specification committed u_i in the anchor state leaf but treated the physical counter as the authority and the leaf as corroboration. This specification inverts the relation: the SMT counter is the authoritative software position, consumed with the whole state, and the physical counter is a tracker bound to it one to one.

5. **MACANDD is removed from the acceptance path.** MACANDD slot state is a pure function of call input and slot index and is host restorable; it carries no forward only lineage. The MACANDD derived one time witness key is replaced by a resident non exportable Ed25519 key, portable and offline verifiable.

6. **The fused anchor head and boot ticket chain are removed as cryptographic objects.** Their job, refusing resumed copied images and new hardware, is done by the three factor identity, the measurement gated host seal, and the counter floor. A copied image lacks chip and host. A stale image fails closed because Active.$u < H_0 - H$. Firmware measurement survives as a load bearing unsealing gate, not as a receiver side protocol object.

7. **The first transfer disclosure round trip is removed.** BilateralBearerPrepared and BilateralBearerProceed existed only to obtain a live FROM read before commit. With the reads removed, first transfer offline is one round trip like any transfer, plus the upgrade certificate at final genesis adoption.

8. **The dual domain structure is added.** Online identity from BIP39 alone, zero hardware; offline identity as a three factor fusion; permanent coexistence; upgrade ceremony. This removes the hardware cost barrier from online participation entirely.

9. **The full stack adversary claim is corrected.** Against a malicious owner or full factor adversary confined to enrolled firmware, first contact offline forks are unconstructible. Against a measurement boundary breach or perfect live state clone, no offline only protocol can distinguish the clone from the original. The corrected claim is strict single appliance uniqueness under the measurement gate, with non mergeability, exposure, bricking, and policy rotation as the residual boundary after the gate is broken.

10. **Verify live is removed from acceptance.** Authenticated counter reads may exist only as diagnostic or post acceptance audit evidence. They are not part of $\text{Accept}_{\text{off}}$, do not gate the common path, and cannot make an invalid release valid.

11. **The SDK and appliance boundary is made explicit.** The SDK computes the device roots and SMT proofs. The appliance signs the supplied, recomputed root advance message during commit, after local frontier, counter, and firmware measurement checks and after the counter decrement. The appliance is not an SMT verifier oracle and does not stamp zero roots for later replacement.

12. **Witness minting is bound to the counter step.** The chip and host witnesses produced for a transfer come into existence only during commit, after the one way counter decrement. They are not free standing host visible artifacts and they do not exist while the origin is still spendable. This closes the prepare, abort, and re prepare harvesting shape and makes host witness single use a timing property of the counter rather than a confidentiality property of the prepared record.

26 Limits

1. Perfect live emulation of all three factors is outside offline distinguishability.
2. Breaking the RP2350 firmware measurement boundary so rogue firmware can invoke HostSign, or extracting a minted host witness before its committing counter step, is outside the enrolled appliance model and is treated as a perfect live state clone boundary.
3. PUF constructions have been attacked by fault injection in laboratory settings; the design tolerates chip factor extraction by fusion, and policy should rotate authority on confirmed physical compromise.
4. The online domain is single factor by design; phone compromise is online domain takeover. Users holding significant value should hold it in offline domain relationships or under policies requiring offline authority.
5. Counter exhaustion at $H = 0$ forces online only operation until authority rotation; lifetime capacity is bounded by H_0, the exposure cap, and flash wear.
6. Theft of the complete appliance with the phone and seed is full factor compromise. Under enrolled firmware, this gives the thief the ability to operate the single appliance, not to fork it. Local unlock policy outside this specification is the mitigation.
7. If the complete appliance is both stolen and the measurement boundary is broken, the system is outside the strict offline uniqueness claim and must rotate authority.
8. Tripwire exposes incompatible proposals at reconciliation or contact, not instantly across disconnected receivers. It does not provide transaction uniqueness; whole state consumption and enrolled appliance emission discipline do.
9. Chip signatures are validity witnesses, not deterministic identifiers. Implementations must verify them cryptographically and must not compare chip signature bytes for equality across emissions.

27 Related Work and Novelty Boundary

Offline value transfer between mutually distrustful parties has a long history, and this specification should be read against it. Three families of prior work bear on it, and the design here sits against all three.

Detection based offline electronic cash begins with Chaum's blind signatures [9] and the offline extension of Chaum, Fiat, and Naor [10], given an efficient construction by Brands [12]. In this family a coin may be spent offline, and offline double spending is not prevented but detected: a coin spent once remains anonymous, and spending it twice reveals the spender's identity when the coins are later deposited at an issuer. Recent offline central bank digital currency work continues the line [14], for example the offline digital euro of Kempen and Pouwelse [15], which pairs zero knowledge transaction privacy with retroactive double spend detection at deposit. This specification is not anonymity first and does not defer to a deposit step. Transfer uniqueness is enforced at acceptance as a property of the consumed state (Theorem 1), and the reconciliation exposure of Tripwire is a boundary case against full compromise, not the mechanism.

The line this specification most directly inverts is the observer, or wallet with observer, model of Chaum and Pedersen [11], in which a tamper resistant chip inside the user's wallet co signs each payment and refuses to sign the same value twice. Brands [12] also situates his cash there. Fielded smartcard purses, and a large share of recent secure element and offline CBDC designs, follow the same template: the hardware is the transaction authority, and the no double spend rule lives inside tamper resistant silicon. That assignment makes the hardware's attack surface the protocol's attack surface. This specification refuses it. Uniqueness is a software theorem that reads no hardware state (Theorem 1, with the hardware independence corollary), and it survives deletion of the anchor entirely. Hardware is left the one job software cannot do, which the observer literature never separates out: proving that the physical device is the enrolled device. The three factor release witness and the measurement gated host seal deliver that job, and the single appliance emission theorem (Theorem 2) makes it strict, so that under enrolled firmware one origin yields at most one release and rogue firmware cannot invoke the host signer at all.

Blockchain systems [8] prevent double spending by global ordering under consensus. Payment and state channels such as the Lightning Network [13] move the common case off chain but still settle disputes by broadcasting a signed state to the consensus layer within a timeout, so their safety reduces to the global ledger. This design has no global order and no settlement broadcast on the common path. Conflicting successors of one consumed origin are excluded locally, and a stalled or hostile counterparty is an availability failure rather than a safety one.

The boundary, stated plainly. Prior offline cash prevents or handles double spending with tamper resistant hardware acting as the transaction authority, or with spending limits, or with after the fact detection at a deposit [10, 12, 11, 15]. Blockchains and channels use global ordering and consensus [8, 13]. This design moves the exclusion into the state predicate, where a valid transition consumes a committed resource and two incompatible successors of that resource cannot both become one coherent history. Hardware here is never the transaction authority. It proves offline device identity and prevents physical duplication of the bearer appliance, and the measurement gated seal makes single appliance first contact uniqueness strict. What is new is the placement of uniqueness and the demotion of hardware, proved rather than assumed.

28 Security Position

For programmable security between mutually distrustful parties, this design differs from a blockchain and from prior secure element cash. A blockchain prevents double spend by global ordering under consensus assumptions. Prior secure element designs prevented it by making the hardware the transaction authority and inherited the hardware's full attack surface as the protocol's attack surface. This design does neither:

> transfer uniqueness is a software property of whole state consumption;
>
> hardware proves only that the device is the device.

The offline identity boundary is stronger than a single hardware secret. It is a fused release witness: seed derived DSM authority, TROPIC01 resident key signature, and RP2350 measurement gated partition signature over the same root advance. A software clone holding every host readable byte cannot produce an offline release. A chip only or host only extraction does not produce the offline identity. Even a full factor adversary operating the real enrolled appliance is still confined to the single enrolled firmware state machine: one active frontier, one prepared record per frontier, witnesses minted only after the counter step, one commit, one exported release.

The claim is narrow and strong: no honest receiver accepts and executes two successors of one adopted origin; a single enrolled appliance cannot emit two distinct valid first contact releases from one origin while the measurement gated host seal holds; no party short of all three factors produces any offline successor; and if the measurement boundary or perfect live state clone boundary is broken, incompatible histories remain non mergeable and require authority rotation. The hardware trust surface has been reduced to two non exportable signing factors, a measurement gated seal, and one monotonic counter. The protocol's receiver side correctness argument for transfer validity does not reference live hardware.

29 Conclusion

The previous specification asked hardware to do what software already did, and the machinery required to make the hardware appear to do it, receiver witnessed counter reads, MACANDD derived witnesses, fused heads, boot tickets, disclosure round trips, was the bulk of the document. Removing the misassignment removes the machinery.

What remains is small. DSM consumes state whole; one parent, one executed successor at an honest acceptance point; that is transfer uniqueness, and it is software. Identity is two domains: a seed for the online world, at zero hardware cost, and a three factor fusion of seed, PUF rooted chip key, and measurement gated partition sealed host key for the offline world, in which every release is witnessed by all three.

The SMT counter makes offline position explicit inside the consumed state. The physical counter is a floor, a tripwire input, and a budget. Tripwire exposes incompatible proposals and bricks whatever splits, but the safety property is already enforced at acceptance by whole state consumption. The measurement gate adds the missing appliance boundary: enrolled firmware can emit one committed successor per frontier; rogue firmware cannot call HostSign; witnesses are minted only after the one way counter step, so no signed release for an origin exists while that origin is still spendable. Therefore disconnected first contact forks are not merely bounded under the intact appliance model. They are unconstructible.

The July 8, 2026 silicon run stands as validation of the counter discipline and the per die identity on which the offline domain rests. The July 9, 2026 ECC gate probe stands as validation of the resident Ed25519 witness key path: empty slot behavior is clean, key generation succeeds, on die signing verifies off chip, erase is reversible, and the public key persists across power cycle. The remaining live bench work is full end to end three signature release acceptance against the reference verifier, any hardware supported counter initialization lock behavior, and the measurement gated host signing boundary.

The offline bearer design is no longer a hardware authority with a software veneer. It is a software authority with hardware identity and a measurement gated appliance boundary, which is what a Deterministic State Machine required all along.

References

[1] J. O'Connor, J.-P. Aumasson, S. Neves, Z. Wilcox-O'Hearn. *BLAKE3: One Function, Fast Everywhere.*

[2] L. Lamport. *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers.*

[3] R. Merkle. *A Certified Digital Signature.* Advances in Cryptology, CRYPTO 1989.

[4] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, B.-Y. Yang. *High-speed high-security signatures.*

[5] M. Palatinus, P. Rusnak, A. Voisine, S. Bowe. *BIP-0039: Mnemonic code for generating deterministic keys.*

[6] Tropic Square. *TROPIC01 Datasheet and Application Note ODN_TR01_app_002, PIN Verification.*

[7] Tropic Square. *TROPIC01 Physical Unclonable Function (PUF), ODR_TR01_PSR_001_PUF.* Version 1.0, June 3, 2026.

[8] S. Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System.* 2008.

[9] D. Chaum. *Blind Signatures for Untraceable Payments.* Advances in Cryptology, CRYPTO 1982, Springer, 1983.

[10] D. Chaum, A. Fiat, M. Naor. *Untraceable Electronic Cash.* Advances in Cryptology, CRYPTO 1988, LNCS 403, Springer, 1990.

[11] D. Chaum, T. P. Pedersen. *Wallet Databases with Observers.* Advances in Cryptology, CRYPTO 1992, LNCS 740, Springer, 1993.

[12] S. Brands. *Untraceable Off-Line Cash in Wallet with Observers.* Advances in Cryptology, CRYPTO 1993, LNCS 773, Springer, 1994.

[13] J. Poon, T. Dryja. *The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments.* 2016.

[14] R. Auer, R. Böhme. *The Technology of Retail Central Bank Digital Currency.* BIS Working Paper, Bank for International Settlements, 2020.

[15] L. Kempen, J. Pouwelse. *Offline Digital Euro: a Minimum Viable CBDC using Groth-Sahai proofs.* arXiv:2407.13776, 2024.

[16] B. Ramsay. *Deterministic State Machines as Guarded Linear Constraint Systems: double-spend prevention as a state property.* Irrefutable Labs Inc. 2026.