ANCHORED · JUNE 2, 2026

A real authority decision. Cryptographically anchored. Independently reconstructable.

On **June 2, 2026** at 00:28:40 UTC, H33 produced its first real anchored authority decision for a real customer. The bundle is retrievable. The receipt is anchored. The chain that produced it is open to inspection.

Run the Proof →

View the Bundle ↗

Download PDF Report

ONE CLICK

Step through the exact chain that produced this bundle — **identity** → **authority** → **replay** → **receipt** → **anchor** → **bundle** — with real values displayed at every stage.

Open Demo →

WHAT WAS PROVEN · 10-SECOND READ

Three claims, all reconstructable from public artifacts.

01

Identity can be mapped to authority.

02

Authority can be replayed

03

Evidence can be

from canonical history.

independently reconstructed.

READING ANY H33 PROOF · THE SIX QUESTIONS

Same six answers. Different scope. The reader recognizes the machine.

1 WHAT HAPPENED?

A real customer (`princ_customer_9`) signed in, presented an Auth1 Bearer, requested an `export_content_bundle` decision through the V101 endpoint, and received an H33-74-anchored content bundle.

2 WHO HAD AUTHORITY?

`princ_customer_9` (Eric Beans, customer_id=9), via `auth_44962d9b-…_v101_export` against `pol_v101_exporter_v1` , rooted at `princ_root_v101_44962d9b-…` .

3 HOW WAS AUTHORITY RECONSTRUCTED?

`replay_until(events, T, tenant=tenant_v101_44962d9b-…, root=princ_root_v101_44962d9b-…)` against the canonical event log, signed with production three-PQ keys.

4 WHAT STATE WAS PRODUCED?

Authority `state_id = 96a29047…be4a` , verdict `Valid` , one active grant authorizing `export_content_bundle` , zero excluded authorities.

5 WHAT ARTIFACT WAS RETURNED?

6 HOW CAN A THIRD PARTY VERIFY IT?

Bundle `d9adcfb0-e0bc-426b-8725-fc12d555692b` — embedded 74-byte H33-74 receipt over the IssuedReceipt commitment, anchored on `h33-substrate-v1`. Publicly fetchable.

`GET app.v101.ai/v101/bundle/d9adcfb0-…` · `auth.h33.ai/.well-known/jwks.json` for the Bearer-validating keyset · scif-backend @ `99756176c` for the reproducible chain.

01 What was proven

A customer signed in. They asked for an export. H33 looked up their authority, reconstructed it from the canonical event log, confirmed a real policy backed it, issued a decision receipt, anchored that receipt cryptographically, and returned the result. **Every arrow above is real.** No synthetic identities. No test credentials. No mocked authority. No simulated replay.

DETERMINATION

H33-CANONICAL-AUTH-v1: proven in operation for one customer, one bundle. The next customer earns the same yardstick. The next surface earns it again. That discipline is the proof.

02 Direct evidence

Every value below is checkable. The bundle URL returns the full JSON. The anchor field encodes a 74-byte H33-74 receipt that anyone with the public keys can verify. The chain of identifiers ties back to the original signed events in the canonical log.

BUNDLE

d9adcfb0-e0bc-426b-8725-fc12d555692b ↗

RECEIPT STATUS

anchored

CUSTOMER (PRINCIPAL)

princ_customer_9

CREATOR UUID

44962d9b-25f5-5622-bd9a-98d5580bb8a2

AUTHORITY ID

auth_44962d9b-25f5-5622-bd9a-98d5580bb8a2_v101_export

POLICY

pol_v101_exporter_v1

ANCHOR CHAIN

h33-substrate-v1

COMMITMENT (SHA3-256)

ff770fc838fde707d91f35248946d6928b0a3a999dbd28a2906ce4f0274745e7

Full anchor reference (148-hex: commitment ‖ 42-byte CompactReceipt) +

```
ff770fc838fde707d91f35248946d6928b0a3a999dbd28a2906ce4f0274745e70
16fb294cb7ddf8073700a2cd13531a352da28068b4921c05839b82b8633547fd
d0000019e85bb875107
```

First 32 bytes (64 hex) = the receipt's SHA3-256 commitment. Next 42 bytes = the H33-74 CompactReceipt — a fixed-width binding of the commitment to three post-quantum signatures (ML-DSA-65, FALCON-512, SLH-DSA-SHA2-128f) plus a verified-at timestamp and algorithm flag. Anyone holding the three public keys can recompute the verification hash and confirm authenticity without consulting any external chain.

03 The chain that produced it

Six stages. Every stage runs against deployed production infrastructure. No stage can be skipped, faked, or replayed in isolation — the receipt's `source_jti` ties it to the exact Bearer, the Bearer's claims tie back to Auth1's signing key, the authority grant traces back to a signed canonical event, and the anchor encodes the receipt's commitment in a way only the production PQ keys can sign.

1 **Identity**

Customer signed in to `auth.h33.ai` via the production OTP flow. Auth1 minted a fresh EdDSA Bearer with `sub=princ_customer_9`, `iss=https://auth.h33.ai`, `aud=substrate-receipts`, signed by `kid-eddsa-prod-active-2026-06-01-d31134fbc177`.

2 **Authority lookup**

V101 forwarded the Bearer to `api.h33.ai/api/v1/h33-auth/v101-bundle-issue`. JwksValidator verified the signature against the deployed JWKS. The subject was canonicalized to `princ_customer_9`.

3 **Replay**

The canonical event log was replayed forward to *now*. The grant `auth_44962d9b-…_v101_export` — signed by the tenant root, granting `export_content_bundle` to `princ_customer_9` — resolved as active.

4 **Policy evaluation**

The grant's `policy_basis = pol_v101_exporter_v1` was looked up. The policy explicitly grants the requested capability. The decision: allowed.

5 **Anchor**

The receipt was canonicalized, SHA3-256 hashed, and anchored via H33SubstrateAnchorSink — producing a 74-byte H33-74 receipt over the commitment using the production three-PQ signer (ML-DSA-65 + FALCON-512 + SLH-DSA-SHA2-128f). Chain identifier: `h33-substrate-v1`.

6 **Bundle**

V101 embedded the anchored receipt into the customer's content bundle, persisted it, and returned `d9adcfb0-e0bc-426b-8725-fc12d555692b` — publicly retrievable right now.

04 Why this is different

Most platforms record an authority decision. H33 reconstructs it. The receipt does not say "trust us, we logged this" — the receipt is a fixed-width binding of the decision's content (the commitment), the production identity (the principal), the underlying authority grant (the authority_id), and the policy that justified it (the policy_basis). The anchor is not a

trust marker. The anchor IS the proof, signed by three independent post-quantum families that would each have to break simultaneously for the receipt to be forged.

Anyone with the public keys — published in the JWKS at `auth.h33.ai/.well-known/jwks.json` for the Bearer side, and the three substrate public keys (whose fingerprints are recorded in the operator's secret store and reproducible from the keypairs themselves) for the anchor side — can verify the chain without contacting H33. That's the meaning of **independently reconstructable**.

05 Trust statement

> STRICT WORDING
>
> This is the **first operational proof** of the H33-CANONICAL-AUTH-v1 chain. It is **not** "shipped." It is **not** "production-ready at scale." It is **not** "deployed for all customers." Every next customer earns the same proof. Every next surface earns it again. Every claim of production readiness gets measured against this exact yardstick: real identity, real authority, real replay, real receipt, real anchor, real artifact. Same chain. Every time.

06 Evidence appendix

FIELD	VALUE
Bundle ID	d9adcfb0-e0bc-426b-8725-fc12d555692b
Creator UUID	44962d9b-25f5-5622-bd9a-98d5580bb8a2
Authority ID	auth_44962d9b-25f5-5622-bd9a-98d5580bb8a2_v101_export

FIELD	VALUE
Policy ID	pol_v101_exporter_v1
Commitment (SHA3-256)	ff770fc838fde707d91f35248946d6928b0a3a999dbd28a2906ce4f0274745e7
tx_reference (148 hex)	ff770fc838fde707d91f35248946d6928b0a3a999dbd28a2906ce4f0274745e7016fb294cb7ddf8073700a2cd13531a352da28068b4921c05839b82b8633547fdd0000019e85bb875107
Anchor chain	h33-substrate-v1
Authority principal	princ_customer_9
Source JWT jti	jti-1780359511-cf79e5f189cb41fd
Issued at (ms)	1780359626000
Submitted at (ms)	1780360120145
Receipt status	anchored

07 Deployment commit SHAs

Auth1 — auth.h33.ai

SHA	SUBJECT
2f49d0a	Merge MR !3: Auth1 Phase 2 — POST /api/auth/canonical/token issuance endpoint **(deployed)**
489e8a8	Merge MR !2: Auth1 Phase 1 — asymmetric (EdDSA) signing + JWKS endpoint

Deployed via systemd `cachee-auth` on `i-0f64d17ee49b88a6f`. JWKS live at `auth.h33.ai/.well-known/jwks.json`; active kid `kid-eddsa-prod-active-2026-06-01-d31134fbc177`.

H33 — api.h33.ai (scif-backend)

SHA	SUBJECT
`99756176c`	fix(canonical-auth): background JWKS refresh in build_production_validator **(deployed)**
`ea0f4e9dc`	fix(canonical-auth): block_in_place in PostgresEventLogSource::events_for
`403f511c5`	chore(canonical-auth): log message backend-agnostic
`1334525d3`	refactor(canonical-auth): H33SubstrateAnchorSink — SK accessors removed, env vars renamed
`16f050f42`	feat(canonical-auth): H33SubstrateAnchorSink — chain-agnostic first-proof anchor
`5ef818235`	feat(canonical-auth): wire production V101BundleIssueState in server.rs
`fa90a9271`	feat(canonical-auth): h33-sign-canonical-event CLI + signing module
`76d9fc554`	Merge MR !25 — Postgres canonical_auth_events + EventLogSource + seed CLI
`1f5c27469`	Merge MR !24 — canonical-auth route mount + auth-before-body
`a5d61a696`	Merge MR !22 — canonical-auth public-path exemption
`f5f824484`	Merge MR !21 — production wiring (BearerValidator + PolygonZkEvmAnchorSink)

SHA	SUBJECT
`8280f2e8a`	Merge MR !20 — V101 Content Bundle export endpoint
`ac8273918`	Merge MR !19 — issue_and_anchor_receipt wrapper + AnchorSink trait
`e767d0c9e`	Merge MR !18 — JWKS validator + Postgres ApiKeyStore
`59e55cdfd`	Merge MR !17 — issue_receipt entry point + API-key exchange
`6b40719f8`	Merge MR !16 — Bearer middleware + principal mapping + replay enforcement

Deployed image: `h33-rust:canonical-auth-99756176` on `i-099b8356ab956a480`. Anchor backend `h33-substrate-v1`. All six `H33_SUBSTRATE_*_B64` env vars populated from AWS Secrets Manager `h33/production/canonical-event-signer`.

V101 — app.v101.ai

SHA	SUBJECT
`68034b1`	V101 Content Bundle endpoint with CANONICAL_AUTH_REQUIRED=true, ANCHOR_HOST=https://api.h33.ai **(deployed)**

08 Known limitations

These are the explicit constraints under which v1.0 was earned. They are not flaws — they are the boundary of what this proof claims.

1. **Single customer.** Only `princ_customer_9` (eb@h33.ai, customer_id=9) was used. The chain is not yet proven for any other principal.

2. **Single bundle.** Bundle `d9adcfb0-…` is the only artifact produced through the full chain. Volume behavior is not yet observed.
3. **No scale validation.** No load testing, no concurrency testing, no throughput claims. Single-request demonstration.
4. **No failover validation.** No disaster-recovery drill. No cross-region failover. No database-failover test. No JWKS-source-unreachable test. Recovery characteristics are not yet measured.
5. **`source_jti` replay behavior observed.** The same Bearer (jti `jti-1780359511-cf79e5f189cb41fd`) produced two receipts in this proof cycle — once from the H33-side smoke test, once from the V101-side bundle issuance. The receipts have distinct `commitment_hex`, distinct `submitted_at_ms`, distinct `tx_reference`, but the same `source_jti`. Pending policy decision: declare idempotent OR enforce single-use at the receipt-issuance layer.

09 Reference documents

- Production Readiness Report v1.0 — Markdown source (frozen).
- Production Readiness Report v1.0 — PDF (frozen, downloadable, share with auditors and regulators).
- H33 Open Standards — HATS, HICS, CRV, OIS, Q-Key, and the H33-CANONICAL-AUTH-v1 specification.
- The bundle itself — `GET app.v101.ai/v101/bundle/d9adcfb0-…`, public read.
- Auth1 JWKS — the public keys that minted and signed the Bearer.

10 Readiness determination

H33-CANONICAL-AUTH-v1: PROVEN IN OPERATION for one customer, one bundle.

What this proof unlocks: distribution. Conversations with prospects, auditors, insurers, and regulators can now move from "we are building this" to "we ran this; here is the artifact." The framework is reusable: every next customer milestone follows the same nine-section proof format at `/proofs/<proof-id>/`.

What this proof does **not** unlock: any claim of scale, multi-tenant production readiness, disaster-recovery readiness, or operational proof on any surface beyond V101. Each requires its own published proof, earned by the same yardstick.

11 Version

FIELD	VALUE
Report version	**v1.0 (Final)**
Frozen	2026-06-02
Supersedes	None
Superseded by	None

Future operational proofs are published at `/proofs/<proof-id>/` using the structure defined in /proofs/proof-template/. This v1.0 is the reference implementation of that structure.

Issued by H33, Inc. · Eric Beans, CEO · 2026-06-02

This page itself is reconstructable from public artifacts. Independent reconstruction inputs: `auth.h33.ai/.well-known/jwks.json` · `h33.ai/standards/` · `app.v101.ai/v101/bundle/d9adcfb0-…`