**Story — “Classify → Bind → License: Proof-Token Flow”**

**Intent (one line)**

Turn provider XML catalogs into **provable classifications** on materials and **live license coverage** on transactions, using tiny, signed tokens that make every decision **self-verifying, explainable, and fail-safe**.

**What makes this unique**

* **Proof Tokens, not flags:** each line carries a **Classification Binding Token (CBT)** and a **License Reservation Token (LRT)** — cryptographic “receipts” that force re-validation when anything important changes.
* **Drift-proof & snapshot-true:** every step cites the **exact provider snapshot** and ruleset used; any deviation auto-surfaces.
* **Impact-first safety:** changes (new snapshot, reclassification, expiring license) only re-touch **impacted** items, with delta pushes and no noise.
* **Explainable by design:** decisions include a one-line **Lineage Badge** plus human-readable **WHY + FIX** that your SME apps can show directly.

**Actors & Artefacts**

* **Provider Feed Daemon** – pulls XML classification datasets (ECCN, HS/HTS, other taxonomies).
* **Classification Foundry** – parses XML losslessly, mints **Trust-Stamped Snapshots**.
* **Material Master** – holds **Material Classification Versions (MCVs)**.
* **CBT** – Classification Binding Token: signed digest tying a material to a specific MCV & snapshot.
* **Rules & Corridor Engine** – derives license need/type from corridor + ECCN/HS.
* **License Vault** – licenses, scopes, validity, counters.
* **LRT** – License Reservation Token: signed digest proving live license coverage & reserved counters.
* **Gatekeeper** – answers ALLOW / HOLD / DENY in ≤400ms with tokens + lineage.
* **Audit Vault** – immutable trail for snapshots, versions, tokens, flips, counter moves.

**Preconditions**

* Provider credentials & dataset registry configured (ECCN XML, HS/HTS XML, etc.).
* KMS keys available for signing snapshots & tokens.
* Material sync from SME systems is live (IDs, descriptions, specs, origin).
* Ruleset version active; License Vault populated.

**End-to-End Flow (happy path)**

**1) XML In → Trust-Stamped Classification Snapshot**

1. Provider release detected → raw XML fetched and stored verbatim (headers included).
2. Foundry validates (XSD), parses **without loss**, normalizes to canonical fields, and mints a **Trust-Stamped Snapshot** (snapshot\_id + signed content hash).
3. Classification Index updates atomically (code lookup, hierarchy, correlations).
4. Event: “ClassificationSnapshotActivated” with counts & summary.

**Outcome:** a tamper-evident catalog your UI and engines can cite by ID forever.

**2) Material Alignment → Classification Binding Token (CBT)**

1. New/updated material arrives; de-duped into a **golden material**.
2. **Classifier Assistant** proposes ECCN/HS candidates from the active snapshot (with confidence and highlighted text matches).
3. Reviewer **approves** the right code(s) → system writes a **Material Classification Version (MCV)** capturing: ECCN, HS/HTS, authority, snapshot\_id, reviewer, approved\_at, notes.
4. System mints a **CBT** (tiny, signed digest of material\_id + mcv\_id + snapshot\_id + approved\_at).
5. Event: “MaterialClassified” → all **open lines** using this material are queued to refresh.

**Outcome:** every material has a current, auditable classification; **every line will carry a CBT** proving which version it used.

**3) Line Decision → License Need, Selection, and LRT**

1. At order/delivery line time, the **Rules & Corridor Engine** evaluates: exporter→destination, **ECCN/HS from CBT**, end-use/user overlays.
2. It returns: **license\_required**, **license\_type\_required**, rule\_id, and the **Explainable Corridor Path** (“US → CN with ECCN 3A001” → License needed).
3. **License Engine** queries the **License Vault**, selects the **best fit** (scope, validity, counters; prefer imminent-expiry to minimize waste), and performs an **atomic reservation** of counters.
4. System mints an **LRT** (license\_id + reservation\_id + material\_id + line\_id + qty/value + validity + counter before/after, signed).
5. Gatekeeper evaluates CBT + LRT:
   * Both valid → **ALLOW**
   * License required but no viable license → **HOLD** with clear **FIX** (apply license X; reduce qty; alternative route)
   * Partner BLOCK / license expired / counters overrun → **DENY**

The gate response includes: **Lineage Badge**, **WHY + FIX**, **CBT**, **LRT**.

**Outcome:** the line is not just “allowed”; it is **provably covered** right now.

**4) Lifecycle & Settlement**

* **Ship/Invoice:** LRT moves from **reserved** → **consumed** (counters decremented).
* **Cancel/Return:** counters restored with a **reversal entry**; LRT marked released.
* **Split shipments:** one reservation per release; tokens reflect each partial.

**5) Change Handling (auto, safe, and minimal)**

* **New classification snapshot** (provider): we compute **impacted materials**, re-mint CBTs where codes changed, queue **only impacted lines** for re-decision, and push **delta events**.
* **Material reclassified (new MCV):** old CBT becomes **stale**; next gate forces refresh (no silent ALLOW).
* **Ruleset update:** impacted lines recalculated; flips sent as **DecisionDelta** with rule version.
* **License expiry/counter drop:** LRT turns **invalid** at next gate; system re-picks license or **HOLD** with exact reason.

**Outcome:** the system stays current automatically, while tenants only see **changes that matter**.

**6) Evidence & Audit (one-click, watertight)**

* **Each line** has: CBT, LRT, decision, lineage badge, WHY + FIX.
* **Each token** is verifiable (signature + payload hash); any stale token fails fast.
* **Proof Bundle** (download): provider snapshot, MCV text, corridor evaluation, license scope, counter log, timestamps, user approvals.

**Outcome:** auditors can **reproduce the decision bit-for-bit** years later.

**Alternate / Failure Flows**

* **XML schema drift:** Foundry blocks promotion, shows a **Field Drift Map** with suggested mappings; last snapshot stays active; high-risk corridors default to REVIEW (never silent ALLOW).
* **No confident classification:** material flagged REVIEW; gate for lines using it = **HOLD** with “What to add” hints (specs, alloy %, encryption).
* **Counter race condition:** if counters exhaust mid-flow, system retries alternate license; else **HOLD**.
* **SME system offline:** decisions queue with retries; Ops can **re-push** later; tokens are time-stamped so staleness is visible.

**Acceptance Criteria (testable)**

1. **Classification truth on lines**
   * Every new line referencing a classified material carries a **valid CBT** tied to its MCV & snapshot.
   * Changing an MCV **invalidates existing CBTs** on open lines; next gate re-decides.
2. **License coverage integrity**
   * When license\_required = true and a license exists, an **LRT is present**, and **counters never exceed** available (even under concurrency).
   * No license available → **HOLD** with concrete FIX steps.
3. **Determinism & reproducibility**
   * Given CBT, LRT, rule\_version, and snapshot\_id, the same inputs produce the **same outcome**.
4. **Delta discipline**
   * New snapshots/rules trigger **impacted-only** re-evaluation and **delta pushes** within SLA.
5. **Performance**
   * Gate P95 ≤ 400 ms (cached), ≤ 1 s when fresh compute is needed.
   * License determination + reservation ≤ 500 ms median.

**Non-Functional & Security**

* **Signing & keys:** snapshots, CBTs, LRTs signed via KMS; rotation tracked.
* **PII minimization:** classification data contains no PII; license artifacts store only necessary facts.
* **RBAC/ABAC:** only Compliance can approve MCVs or override gates; MFA for overrides.
* **Tenancy isolation:** tokens scoped to tenant; cross-tenant leakage impossible.
* **Storage tiers:** hot (90d) → warm (1y) → archive (7–10y).

**AI (assistive only, never the decider)**

* **Classifier Assistant:** ranks ECCN/HS and highlights supporting phrases; suggests what extra attributes would raise confidence.
* **License Picker Helper:** recommends the least-waste license and explains the choice.
* **Change Narrator:** plain-language “what changed” summaries for snapshot/rule shifts, included in delta events.
* **Forecasts:** predicts counter depletion; suggests pre-emptive license renewals.

*All AI outputs carry confidence + citations, and are fully overrideable.*

**UI moments (so teams “feel” the flow)**

* **Material page:** Classification timeline, current MCV, **CBT status chip**, “Why this code” evidence.
* **Order/Delivery drawer:** “Allowed under License X” with **CBT + LRT chips**, Lineage Badge, WHY + FIX; one-click **Evidence Pack**.
* **License Lens:** quotas by corridor/product, predicted depletion, **rebind alerts** when tokens turn stale.