**Story A — “XML Taxonomy In → Trust-Stamped Classification Snapshot”**

**Intent (one line)**

Turn the provider’s **XML classification catalogs** (ECCN, HS/HTS, any extras) into **immutable, signed snapshots** your engines can trust, search, and cite—without ever losing provider fidelity.

**Actors**

* **Provider Feed Daemon** (pulls XML)
* **Classification Foundry** (parses, normalizes, stamps)
* **Classification Registry** (catalog of datasets & versions)
* **Audit Vault** (stores proofs)
* **Ops Pilot** (observes / promotes)

**Preconditions**

* Provider credentials set; datasets registered:
  + ECCN\_XML, HS\_GLOBAL\_XML (and optionally HTS\_US\_XML, HTS\_EU\_XML, etc.)
* KMS signing keys live.
* Storage zones created:
  + **Cold Room** (raw XML), **Snapshot Vault** (signed classification snapshots).

**Happy Path**

1. **Fetch XML**
   * Daemon detects a new provider **release** (webhook/schedule).
   * Saves **raw XML** verbatim to Cold Room with headers, release id, retrieved\_at.
2. **Validate & Schema Guard**
   * XSD/schema validation (versioned).
   * If schema drifts, move to **Drift Hold** (see failure flows).
3. **Normalize (Non-Lossy)**
   * Parse to a **Canonical Classification Record** while **preserving original XML** fields (verbatim).
   * Canonical core fields (examples):
     + code (e.g., “3A001”), title, description
     + authority / jurisdiction (US BIS, WCO, country HTS)
     + effective\_from/to, notes, footnotes, license\_hints (if provider supplies)
     + parents[] / children[] (hierarchy), correlations[] (ECCN↔HS where given)
   * Country tags normalized to ISO2; dates to UTC ISO8601.
4. **Mint Snapshot (Trust Stamp)**
   * Compute content\_sha256 over **provider-canonical** JSON representation (not our prettified form).
   * Create snapshot\_id = {provider}:{dataset\_id}:{release\_id}:{retrieved\_at}.
   * **Sign** with KMS → store in Snapshot Vault (immutable).
5. **Indexing**
   * Build **Classification Index** per dataset:
     + Code lookup, keyword search (title/desc), hierarchy traversal,
     + Correlation graph (ECCN↔HS/HTS) if provided.
   * **Atomic swap** to make the snapshot available for lookups.
6. **Registry Update & Event**
   * Classification Registry records dataset → latest snapshot\_id, counts.
   * Publish ClassificationSnapshotActivated event with summary.

**Failure / Alternate Flows**

* **Schema Drift Firewall:** XML fails XSD → mark dataset “Needs Mapping,” keep last snapshot ACTIVE; alert Ops with a **Field Drift Map** suggestion.
* **Suspicious Delta:** Massive adds/removes in codes → **Quarantine**; require **Dual Attestation** to promote.
* **Provider Stale:** Last good snapshot remains; console shows **“classification stale”** banner; downstream keeps working with safe defaults.

**Acceptance Criteria**

* Every release yields a **signed** snapshot with reproducible content\_sha256.
* Activation ≤ **5 min** from retrieval in steady state.
* Index swaps are **atomic**; no partial code tables.
* Audit Proof includes: raw XML headers, snapshot id, signer key id, counts.

**Innovation**

* **Trust Stamp Receipts** (QR-friendly)
* **Drift Firewall** with suggested mappings
* **Correlation Graph** (ECCN↔HS) ready for later license logic