**Story C — “License Typing & Operationalization: From Codes to Controls”**

**Intent (one line)**

Use material classifications + corridors to **derive license types**, select the **best available license**, and **reserve counters**—with explainable rules and zero quota overruns.

**Actors**

* **Rules & Corridor Engine** (from earlier)
* **License Engine** & **License Vault**
* **Gatekeeper Service**

**Preconditions**

* Materials classified (Story B) with latest approved versions.
* Ruleset ACTIVE (which ECCN/HS + corridor ⇒ license type/exception).
* License Vault populated with licenses (scope, validity, counters).

**Happy Path**

1. **License Type Derivation**
   * On order/delivery line, Rules Engine evaluates:
     + Exporter → Destination (corridor)
     + **ECCN/HS** from the **approved material classification version**
     + End-use/end-user flags, embargo overlays
   * Emits **license\_type\_required** (e.g., “NLR”, “GL X”, “SIEL”, “EAR99 NLR”, etc.) with **rule\_id** and explanation.
2. **License Selection & Reservation**
   * License Engine queries Vault for licenses matching license\_type\_required, corridor, product scope.
   * **Smart Picker** chooses best fit (expiry soonest, adequate counters, tenant policy).
   * **Atomic reservation** of counters (qty/value) recorded with reservation\_id.
3. **Gate Decision**
   * With valid assignment + reservation → **ALLOW** (with lineage badge).
   * If no license available → **HOLD**; Fix: “Apply for License X” or “Reduce qty to Y”.
4. **Lifecycle Updates**
   * On shipment/invoice post: **commit** reservation (decrement counters).
   * On cancel/return: **release** reservation back to counters.
5. **Change Propagation**
   * If material classification version changes or ruleset updates, **re-evaluate** open lines; flip gates if needed (delta only).

**Failure / Alternate Flows**

* **Multiple Candidate License Types**: Rules Engine returns **ranked options**; License Engine picks by policy (or asks reviewer if tie).
* **Counters Race**: If counters depleted mid-flow, try next license; else **HOLD** with clear message.
* **Exception Eligible**: If rule allows exception (e.g., de minimis), mark ALLOW under exception; badge shows rule reference.

**Acceptance Criteria**

* **Deterministic**: Same inputs → same license type derivation.
* **Safe counters**: No overrun under concurrency (ACID reservations).
* **Explainable**: Decisions show code lineage (material classification version + snapshot), rule id, license id, counters before/after.
* **Auto-refresh**: Any relevant change (classification/rule/license) re-evaluates affected lines within SLA.

**Innovation**

* **Explainable Corridor Graph** in the decision (tiny visual path)
* **License Smart Picker** tuned for SME behavior (avoid expiry waste)
* **What-If Simulation** per order to preview license/counter impact

**Where everything “sits” in your system (clear mental map)**

* **Cold Room (storage):** raw provider XML + headers.
* **Snapshot Vault:** signed classification snapshots (immutable).
* **Classification Registry:** catalog of datasets → latest snapshot, counts, status.
* **Classification Index:** fast search (code lookups, hierarchy, correlations).
* **Material Master:** golden materials with **Classification Versions** (approved only).
* **Rules & Corridor Engine:** derives **license types** from **classification + corridor**.
* **License Vault:** licenses, scopes, counters, validity; reservation/commit logs.
* **Gatekeeper:** answers **ALLOW/HOLD/DENY** in ≤400ms, with lineage badges.
* **Audit Vault:** everything—snapshots, approvals, reservations, flips—tamper-evident.

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

* **XML Drift Mapper:** Suggests XSD mapping updates when provider changes fields.
* **Classifier Assistant:** Ranks ECCN/HS candidates + highlights matching phrases.
* **Ambiguity Coach:** Tells users which attributes to add to improve confidence.
* **License Recommender:** Chooses best license with rationale; forecasts counter burn-out.

**Quick confirmations (so I can lock this and format into your blueprint PDF later)**

1. Which **classification datasets** will your provider deliver first (ECCN only? HS global? Country HTS too)?
2. For **material approval**, do you want **dual control** (Reviewer + Approver) or single reviewer is enough in v1?
3. Default behavior if **classification missing** on a line: **HOLD with “Classify material”** (recommended)?

If you confirm these, I’ll keep going with **UAT checklists** and **Ops Console views** for these three stories (so your team can build screens and tests directly).