Step 5 — Validation & Consolidation (Deterministic)

Objective

Transform *raw LLM extraction outputs* into **validated, conflict-aware, portfolio-ready legal rights**, suitable for human review and reporting.

Step 5 converts *LLM candidates → auditable legal facts*.

This step is **non-probabilistic**: no new information is introduced, only validated, downgraded, or flagged.

Inputs

- `extraction.jsonl` (output of Step 4)
  - One line per `clause_family`
  - Each line contains a list of extracted items with:
    - structured fields (trigger, effect, thresholds…)
    - evidence quote + provenance
    - confidence score
    - initial flags

Outputs

- `validated_items.jsonl`
  - One line per extracted item
  - Deterministically validated and flagged
  - Canonical representation of portfolio rights
- `portfolio_table.csv / .xlsx`
  - One row per legal right

    - Portfolio-level, review-ready format

Each validated item includes:

- `present` (boolean, may be downgraded)
- `clause_family`, `clause_type`
- `beneficiary`
- `trigger`, `effect`, `thresholds`
- **Evidence**:
    - normalized quote (≤ 25 words)
    - source file
    - page / article (when available)
- `confidence`
- `flags` (`ambiguous`, `conflicting`, `missing_evidence`, …)

Core Design Principles

1. Deterministic validation (LLM output is not trusted)

- No LLM calls are made in this step
- All checks are rule-based and reproducible
- The system prefers **downgrading** over keeping uncertain claims

If a rule fails → the item is invalidated or flagged.

---

2. Evidence is mandatory for presence

A right **cannot** be `present = true` unless:

- an evidence quote exists
- quote length ≤ 25 words
- a source file is provided

Violations automatically result in:

- `present = false`
- appropriate flags (`missing_evidence`, `quote_too_long`, …)

3. Explicit handling of uncertainty

- `confidence < 0.6` → `ambiguous`
- Confidence is **triage-only**
- It never upgrades or justifies a claim

Silence is preferred to unsupported certainty.

Conflict Detection & Consolidation

Grouping key

Rights are grouped by:

```
(company_id, clause_family, clause_type)
```

Conflict detection logic

For each group with multiple `present = true` items:

- Normalize `trigger`, `effect`, `thresholds`
- Compare semantic overlap (token similarity)
- If similarity < threshold → conflict detected

Conflict behavior (strict)

- Conflicting rights are **never merged**
- All versions are kept
- All are flagged `conflicting`
- All evidences are surfaced

No silent override, no “latest document wins” logic.

Explicit “Not Found” Semantics

When a clause family or clause type is expected but unsupported by evidence:

- the system emits an explicit item with:
  - `present = false`
  - `flag = missing_evidence`

Absence is **made visible**, not implicit.

Technical Implementation

Validation layer

- Implemented in pure Python
- Schema enforced via **Pydantic**
- Fast-fail on invalid structures
- Fully deterministic and reproducible

Normalization

Applied before comparison:

- lowercase
- whitespace normalization
- symbol normalization (`€`, `%`, etc.)

Prevents false conflicts due to drafting style.

Export & Formatting

Portfolio table

- Single consolidated table
- Stable sorting by:
  - company
  - clause family
  - clause type
  - presence

XLSX formatting

- Frozen header
- Filters enabled
- Conditional formatting:
  - 🔴 `conflicting`
  - 🟠 `ambiguous`
  - ⚪ `present = false`

Designed for direct use by investment teams.

CLI Usage

```
lre validate \
  --extraction path/to/extraction.jsonl \
  --out-dir path/to/step5/
```

Optional tuning:

```
--min-conf 0.6
--max-quote-words 25
--conflict-thr 0.75
```

Observability & Robustness

- Deterministic runtime (no model calls)
- Immediate write after validation
- One corrupted item does not fail the batch
- Outputs are reproducible across runs

Typical runtime:

- < 1 second per document
- Dominated by I/O, not computation

Outcome

At the end of Step 5, the system produces:

- a **clean, auditable portfolio view**
- explicit conflicts and missing rights
- zero hallucinated legal claims

This step is what turns an LLM pipeline into a **professional legal review accelerator**.