legal-rights-extractor - PDD - v1.2

Project Definition Document (PDD)

Legal Rights Extraction System (Portfolio Term Sheets / SHA)

Version 1.2 • Updated • Author: Corentin Marion

1. Executive Summary

This document defines the scope, constraints, and implementation of a **deterministic system for extracting investor rights from legal documents**.

The system processes portfolio legal instruments (term sheets, shareholders' agreements, amendments) and produces **structured, auditable representations of explicitly stated investor rights**, each backed by direct textual evidence.

The system is not a legal analysis tool.
It does not infer, interpret, or normalize rights beyond what is explicitly written in the source documents.

The MVP targets four fixed clause families:

- veto
- liquidity
- exit
- anti_dilution

Outputs are designed for **portfolio review, audit, and triage**, and are exportable to CSV/XLSX.

2. Context and Goals

2.1 Business Context

Private equity portfolios accumulate legal complexity over time:

- multiple legal instruments per company,

- heterogeneous drafting styles,
- amendments overriding earlier provisions,
- rights scattered across pages, articles, and annexes.

Manual extraction of investor rights is:

- time-consuming,
- error-prone,
- difficult to audit,
- hard to reproduce when documents evolve.

The objective is not summarization, but **traceable extraction under legal constraints**.

2.2 Objectives

Primary objective

- Enable fast, repeatable extraction of key investor rights with explicit evidence and provenance.

Secondary objectives

- Support portfolio-level visibility and comparison.
- Enable deterministic re-runs as documents change.
- Surface ambiguity and conflicts explicitly rather than resolving them implicitly.

Output objective

- Produce a structured, exportable dataset suitable for review workflows (Excel / CSV).

2.3 Non-Goals

The system does **not**:

- provide legal advice or replace counsel,
- assess enforceability or market standards,
- infer missing clauses,
- reconcile conflicting provisions automatically,

- draft, negotiate, or modify legal language,
- perform full contract summarization.

---

3. Users and Stakeholders

- **Primary users**: investment and portfolio teams performing reviews, reporting, or exit preparation.
- **Secondary users**: legal counsel using outputs as a review accelerator.
- **Stakeholders**: fund management requiring auditability and portfolio-level visibility.

---

4. Scope and Perimeter

4.1 In Scope (MVP)

- **Documents**: term sheets, shareholders' agreements, investment agreements, amendments.
- **Languages**: French and English (document-by-document).
- **Clause families** (fixed enum):
  - veto
  - liquidity
  - exit
  - anti_dilution
- **Processing mode**: batch execution across multiple companies.
- **Outputs**:
  - structured portfolio table,
  - per-company detailed view with evidence and flags.

4.2 Out of Scope (MVP)

- Automatic document retrieval from data rooms.
- Jurisdiction-specific legal interpretation.
- Market benchmarking.

- OCR beyond basic fallback handling.
- Any form of automated inference or decision-making.

---

5. Definitions and Clause Taxonomy

The system relies on a **fixed taxonomy** to avoid semantic drift.

5.1 Veto

Any provision requiring investor consent or granting blocking rights over specific actions.

5.2 Liquidity

Provisions affecting share transfer or liquidity, including ROFR/ROFO, lock-ups, tag-along, drag-along, and IPO restrictions.

5.3 Exit

Provisions governing exit mechanics, including drag/tag triggers, liquidation preferences, and change-of-control definitions.

5.4 Anti-Dilution

Mechanisms adjusting economic ownership or conversion terms following down-rounds or defined triggers.

---

6. Inputs

6.1 Document Set per Company

Each company may include:

- term sheet(s),
- shareholders’ agreement,
- investment agreement,
- amendments or side letters.

6.2 Supported Formats

- PDF (text-based preferred),

- DOCX,
- TXT.

6.3 Assumptions

- The user provides the authoritative document set.
- If multiple versions conflict, conflicts are surfaced explicitly.
- Every extracted item retains provenance (file, page, article when available).

---

7. Outputs

7.1 Portfolio Table

Primary output: one row per extracted right.

Each row corresponds to a `RightItem` with the following fixed fields:

- company_id
- clause_family
- clause_type
- present
- beneficiary
- trigger
- effect
- thresholds
- evidence
  - quote (≤ 25 words)
  - source_file
  - page
  - article
- confidence
- flags

7.2 Per-Company Detail View

- Structured list of extracted rights.
- Direct evidence excerpts.
- Explicit flags:
  - ambiguous
  - conflicting
  - missing_evidence
- “Not Found” items listed without implying existence.

---

8. Requirements

8.1 Functional Requirements

- Deterministic ingestion and chunking with page mapping.
- **Lexical retrieval using BM25** over chunked text.
- Schema-locked extraction using LLMs.
- Deterministic validation and conflict detection.
- Batch execution across companies.
- Export to CSV and XLSX.

8.2 Non-Functional Requirements

- Accuracy prioritized over coverage.
- Full auditability of every output field.
- Reproducibility (model, prompt version, timestamp).
- Local-first processing by default.
- Best-effort latency target < 2 minutes per company.

---

9. System Architecture

Fixed pipeline stages (order enforced):

1. Ingestion
2. Chunking

3. Retrieval (**BM25**)
4. Extraction (LLM, schema-locked)
5. Validation
6. Conflict Detection
7. Aggregation
8. Export / UI

9.1 Core Design Choices

- Deterministic scaffolding around the LLM.
- Lexical retrieval favored over semantic embeddings to:
  - preserve exact legal language,
  - avoid semantic drift,
  - ensure transparent scoring.
- LLM restricted to extraction and classification only.
- Conflicts are surfaced, never resolved implicitly.

---

10. Prompting Strategy and Schemas

10.1 Extraction Contract

Each clause family returns a list of `RightItem` objects.

The LLM must:

- output valid JSON only,
- rely exclusively on provided context,
- set `present=false` when evidence is insufficient,
- flag ambiguity instead of guessing.

10.2 Validation Rules

- Reject items without evidence.
- Enforce quote length and enum validity.
- Conflicting definitions across documents are flagged, not merged.

11. Evaluation and QA

- Gold set of manually verified companies.
- Metrics: precision, evidence accuracy, conflict detection rate.
- Regression runs after any prompt or code change.
- Human review prioritized by flags and confidence.

12. Security, Privacy, and Compliance

- Confidential by default.
- Minimal retention of raw documents.
- No logging of full text.
- Local execution preferred.
- External model usage subject to policy review.

13. Delivery Status

The MVP pipeline is implemented end-to-end:

- ingestion → BM25 retrieval → extraction → validation → export,
- deterministic stages enforced,
- schema-locked outputs,
- reproducible runs.

Future work is incremental, not architectural.

14. Risks and Mitigations

Risk	Mitigation
Hallucinated rights	Evidence required; otherwise `present=false`
Drafting variability	Fixed taxonomy + BM25 lexical matching
Conflicting versions	Explicit conflict flags
Scope creep	Frozen clause families

Risk	Mitigation
Confidentiality	Local-first, minimal retention