ConsentLens

“Explain This Before I Agree”

AI-Driven Consent Interception, Power Asymmetry Analysis, and Worst-Case Interpretation

---

1. Refined Foundational Concept

ConsentLens is not a legal summarizer.
It is a **consent intelligence system** designed to expose *real-world consequences* of agreement that are:

- Legally valid but cognitively hidden
- Structurally asymmetric
- Triggered only under stress, dispute, or exit

Core Thesis

Consent is invalid when consequences are invisible, delayed, or asymmetric.

ConsentLens operationalizes this by treating contracts as **behavioral traps**, not documents.

It directly inherits **scamcheck.py’s logic primitives**:

- Trap detection
- Withdrawal friction
- Escalation pathways
- Asymmetric control

—but applies them to **legal, contractual, and platform consent environments**.

---

2. Expanded System Logic Flow

```
Agreement Capture (Static or Live)
   ↓
Document Normalization & OCR
   ↓
Clause Segmentation & Structural Parsing
   ↓
Clause Semantics & Legal Intent Classification
```

```
   ↓
Trap Pattern Detection
   ↓
Power Asymmetry Modeling
   ↓
Risk Pattern Matching (Corpus + Heuristics)
   ↓
Worst-Case Scenario Inference
   ↓
Human-Readable Consent Explanation
```

Each stage emits **explainable artifacts**, not black-box scores.

---

3. Agreement Intake & Parsing Layer

Supported Inputs (MVP → v2)

Input Type	Method
PDFs	`PyPDF2`, `pdfplumber`
Scanned docs	`pytesseract`
Web checkouts	`selenium`
Inline text	direct ingestion
Clickwrap flows	DOM interception

Core Challenges Solved

- Multi-column legal layouts
- Footnotes and references
- Embedded tables
- Obfuscated formatting

Output: Canonical Text Blocks

```
{
  "doc_id": "hash",
  "source": "pdf | web | scan",
  "clause_blocks": [
    {
      "block_id": "...",
      "text": "...",
      "location": "page 7, para 3"
    }
  ]
}
```

---

4. Clause Segmentation & Structural Detection

Objective

Identify **legally meaningful units**, not paragraphs.

Techniques

- Sentence boundary detection (legal-aware)
- Heading inference
- Numbered clause reconstruction
- Cross-reference resolution

Libraries

- `spaCy` (custom pipeline)
- `lexnlp`
- Rule-based fallback logic

Clause Types Detected

- Arbitration
- Class action waiver
- Indemnification
- Auto-renewal
- Termination & exit
- Unilateral modification
- Data ownership & resale
- Jurisdiction / venue lock-in

---

5. Clause Classification (ML Layer)

Purpose

Determine **what power is being asserted**, not just the topic.

Libraries

- `transformers`
- `torch`

Model

- `nlpaueb/legal-bert-base-uncased`
- Fine-tuned on labeled clause-intent dataset

Example Labels

Label	Meaning
`USER_OBLIGATION`	You must act or pay
`PROVIDER_DISCRETION`	They may act unilaterally
`RIGHTS_WAIVER`	You give up legal rights
`EXIT_PENALTY`	Leaving has consequences
`DATA_TRANSFER`	Your data becomes an asset

Training Objective

- `CrossEntropyLoss`
- Multi-label classification (clauses often serve multiple functions)

---

6. Trap & Risk Logic Engine (Core Differentiator)

This is **explicitly non-ML-first** and is what makes ConsentLens defensible.

Inherited Scamcheck Patterns

Trap Pattern	Contractual Analog
Withdrawal friction	Complex cancellation
Silent escalation	Auto-renew + price change
Asymmetric risk	Indemnification
Control isolation	Arbitration clauses
Delayed harm	Data resale, liability shift

Libraries

- `fuzzywuzzy` (pattern variants)
- `sqlite3` (rule corpus)
- Deterministic logic engine

Example Rule

```
IF
  unilateral_modification = true
AND
```

```
  auto_renew = true
AND
  notice_period > 30 days
THEN
  trap = "Silent Obligation Escalation"
```

---

7. Power Asymmetry Analysis

Core Question

Who can act freely, and who cannot?

Signals Modeled

- Who can change terms
- Who bears liability
- Who controls dispute venue
- Who pays costs
- Who loses rights

Output Vector

```
{
  "user_power": 0.21,
  "provider_power": 0.79,
  "asymmetry_type": ["exit_control", "legal_shielding"]
}
```

This enables **comparative consent intelligence** across platforms.

---

8. Risk Matching & Corpus Intelligence

Purpose

Anchor interpretation in **real-world outcomes**, not hypotheticals.

Risk Corpus

- Prior contracts
- Known disputes
- Regulatory actions (future)
- User-submitted harm cases

Matching Techniques

- Clause embedding similarity
- Pattern overlap scoring
- Historical consequence mapping

Result

“This clause is structurally similar to X contracts that resulted in Y disputes.”

---

9. Worst-Case Scenario Inference Engine

This is the philosophical core of ConsentLens.

Legal documents describe **best-case legality**.
ConsentLens explains **worst-case lived reality**.

Logic

- If clause grants discretion → assume unfavorable use
- If exit is restricted → assume conflict scenario
- If liability is shifted → assume failure occurs

Example Output

“If a billing dispute occurs, you cannot sue, cannot join a class action, must arbitrate in Utah, and may still be required to pay disputed fees.”

This is **interpretation**, not advice.

---

10. User-Facing Explanation Layer

Outputs (Structured)

- **What you give up**
- **What they gain**
- **How to exit**
- **What happens if things go wrong**
- **Risk score (0–100)**

Design Principle

One screen. No legalese. No hedging.

---

11. MVP Timeline (90 Days – Expanded)

Phase 1 (Days 1–25): Intake & Parsing

- PDF + OCR reliability
- Clause segmentation accuracy
- Live DOM capture proof-of-concept

Phase 2 (Days 26–50): Clause Intelligence

- Legal-BERT fine-tuning
- Clause taxonomy validation
- Annotation tooling

Phase 3 (Days 51–75): Trap & Risk Engine

- Scam-pattern inheritance
- Rule authoring system
- Power asymmetry scoring

Phase 4 (Days 76–90): Delivery

- Browser extension
- Explainability UX
- Logging & feedback loop

---

12. Defensibility & Moat

ConsentLens is defensible because:

- **Interpretation > summarization**
- Rule corpus grows with use
- Worst-case framing is rare and hard to automate
- Legal ML alone cannot replicate trap logic
- Network effects via shared consent intelligence

---

13. Strategic Positioning

ConsentLens is **not**:

- Legal advice
- A chatbot
- A ToS summary tool

ConsentLens **is**:

- A consent truth engine
- A power imbalance detector
- A pre-commitment defense system