

Chomper

A universal evidence-ingestion engine with quantitative confidence scoring and extract-and-destroy provenance.

Author: Richard L. Sanders · Utah Bar #15728 **Role:** Sole inventor and implementer
Version: 1.0 — 2026-05-13 **Companion patent:** USPTO provisional `04-chomper-evidence-ingestion.md` (filed 2026-04-26) **Source:** `lib/chomper/index.js`, `lib/chomper/confidence.js`, `lib/chomper/pdf-enhance.js`, `lib/chomper/image-analyze.js`, `lib/chomper/secure-destroy.js`, `tests/chomper-*.test.js` in the public repository at <<https://github.com/Benchslap/Benchslap>>

Abstract

Forensic evidence ingestion in commercial legal tools today is a mess of disconnected one-off extractors: one library for PDFs, another for images, a third for audio, and the resulting metadata is mailed back to the user as opaque categorical labels — "high confidence," "likely match," "consistent with" — that no judge has ever cross-examined. The evidence stays on the vendor's servers in original form, indefinitely, often without the user's informed consent and without a defensible chain of custody.

Chomper is a single ingestion entry point that does three things commercial competitors do not:

1. **One pipe, every file type.** PDF, image, audio, video — same call shape, same output shape, same confidence semantics. Whatever goes in, structured forensic text comes out.
2. **Quantitative confidence by formula, not by label.** Every observation carries a percentage score derived from four measurable factors: signal strength, ambiguity, reliability, and expert calibration. The number is shown to the user, the formula is shown to the user, and the four input factors are shown to the user. There is no "high / medium / low" — there is 73%, with the four factors that produced it.
3. **Extract-and-destroy provenance.** The original file is processed in-memory, its content is extracted as structured text, and the original buffer is zeroed in RAM before the response returns. Originals are never persisted to disk. This is the only architecture that satisfies an attorney's professional-responsibility duty to keep client material out of the vendor's permanent storage.

This paper describes Chomper's confidence formula, its extract-and-destroy enforcement, the four current and two planned modality analyzers, and the architectural reasons this design is not feasible to retrofit onto a conventional legal-tech pipeline.

1 • The problem

A litigator who receives a 200-page PDF from opposing counsel, a 50-image discovery dump, a recorded deposition, or a body-camera video has a clock. They have to read it, classify it, decide what is responsive, what is privileged, what is exculpatory, and what is admissible — and they have to do it before the next hearing. They cannot read every word of every file at every cycle. They need a tool that reads the file once, extracts what is in it with a known degree of reliability, and gives them back a forensic summary they can actually rely on in court.

Commercial e-discovery products do part of this. They are expensive (typically \$5–20 per gigabyte processed, before counsel review), they index originals indefinitely on the vendor's storage (a privilege and confidentiality risk that no attorney has ever litigated to a clean conclusion), and they emit confidence as a category — "high relevance," "likely PII" — that no expert witness can defend on cross. When the judge asks "what does 'high' mean," there is no answer in the math.

I needed something different. I needed:

1. **One ingestion contract** for every common evidentiary file type a litigator actually sees in practice — so my pipeline could fan out to the right analyzer without my having to remember which library handles which extension.
2. **A confidence number, not a label** — so I could put the percentage on the page, name the four factors, and defend it on cross-examination.
3. **Extract-and-destroy** — so the original never touches my storage. The text I extract is mine to retain; the file the client gave me is destroyed in RAM the moment its content is extracted. The vendor (me) cannot accidentally redisclose it because the vendor (me) never had it.

Chomper is that pipeline.

2 • The confidence formula

`lib/chomper/confidence.js` implements one expression:

$$C = S \times (1 - A) \times R \times E$$

Where, in the canonical naming I established:

Symbol	Factor	Range	Interpretation
S	Signal strength	0.00 – 1.00	How loud is the underlying data? Pixel contrast on the OCR target, dB on the audio sample, density of cited authority on the page.
A	Ambiguity	0.00 – 1.00	How many alternative interpretations of the same signal would a reasonable expert advance? Zero means there is exactly one explanation.
R	Reliability	0.00 – 1.00	How stable is the observation across the sample? Does the same OCR result come back across passes? Does the same speaker fingerprint hold for the whole clip?
E	Expert calibration	0.00 – 1.00	Would a domain expert, presented with the same evidence and the same observation, agree it follows? This is the only factor that requires human-judgment encoding — but it encodes it once, into the formula, and never disappears into a black box.

The formula's behavior is predictable:

- **Zero signal collapses confidence to 0%** regardless of the other factors. You cannot manufacture confidence from clean ambiguity.
- **Full ambiguity collapses confidence to 0%** regardless of signal. A loud signal that means three different things means nothing.
- **Zero reliability collapses confidence to 0%** even on a perfect signal. If the observation does not replicate, the observation does not count.
- **Zero expert calibration collapses confidence to 0%** even with strong signal, no ambiguity, and perfect reliability. If the field would not accept the observation as evidence, the math does not invent acceptance.

The product structure is deliberate: any one factor collapsing the score to zero means the report says "0%" and the underlying factor explains why. No silent rounding. No averaging that hides a fatal weakness behind three strong neighbors.

2.1 · Five canonical presets

`confidence.js` exports five preset confidence shapes for the recurring forensic claim types:

```
metadataFact()      // S=1.00, A=0.00, R=1.00, E=1.00 → 100%
directMeasurement() // S=0.95, A=0.05, R=0.95, E=0.95 → ~80%
patternMatch()      // S=0.80, A=0.15, R=0.80, E=0.80 → ~43%
```

```
inference()           // S=0.60, A=0.30, R=0.70, E=0.65 → ~19%
speculation()         // S=0.25, A=0.70, R=0.30, E=0.40 → ~1%
```

- **metadataFact** — facts pulled directly from the file (EXIF date, MIME type, file size). These never lose confidence because they are not inferences, they are extractions.
- **directMeasurement** — measurements taken on the file (volume in dB, dominant frequency, pixel-level resolution). High but bounded — the measurement instrument has known error bars.
- **patternMatch** — a known pattern is detected (reverb profile, OCR-recognized character, fingerprint match). The pattern could be wrong; the score reflects how often it is.
- **inference** — emotional state, age estimate, document classification. Multiple weak signals support the conclusion; an expert might disagree.
- **speculation** — the report mentions it because it could be material, but the underlying signal is faint. The 1% number tells the reader to take it as a flag, not a finding.

2.2 · The format string

`formatConfidence()` emits the exact text that appears in every Chomper report:

```
[C: 73% | S=0.95 A=0.20 R=0.95 E=0.85 – pattern match against 12-shingle
reverb library]
```

Every observation carries this bracket. The methodology phrase after the dash is the analyst's one-line statement of what the four factors mean for this specific observation. This is the only place in the pipeline where natural language replaces a number, and it is constrained to a single phrase that the court can question directly.

3 · One ingestion contract, every file type

`lib/chomper/index.js` defines the universal entry point:

```
async function process(buffer, filename, options = {}) → {
  type: string,           // 'pdf' | 'image' | 'audio' | 'video'
  report: string,         // markdown forensic report
  metadata: object,       // file-system + format metadata
  observations: Array     // {claim, confidence, factors, methodology}
}
```

The caller does not pick an analyzer. Chomper inspects the filename extension via `isSupported(ext)` and routes to the right analyzer behind the scenes:

Extension family	Analyzer	Status
<code>.pdf</code>	<code>pdf-enhance.js</code> (text extraction + OCR fallback + structure detection)	Production
<code>.jpg .jpeg .png .gif .bmp .tiff .tif .webp .heic .heif</code>	<code>image-analyze.js</code> (forensic description + EXIF + recreation prompt)	Production
<code>.mp3 .wav .aac .m4a .ogg .flac .wma</code>	audio analyzer	Phase 3 (planned)
<code>.mp4 .mov .avi .wmv .mkv .flv .webm .m4v</code>	video analyzer	Phase 4 (planned)

The output shape is identical regardless of input modality. Every observation has a `confidence` object with `{score, percent, factors}`. Every report is markdown. Every metadata object names the format, the file size, the original filename, and the SHA-256 of the input buffer (computed once before destruction).

The classification phase fires after extraction (`classify(text, filename, options)` at `lib/chomper/index.js:38`). It hands the extracted text to the preprocessor pipeline, which builds an `authorityIndex` (every cited case) and a `factFreezer` (every factual assertion frozen with a source pin). The two together are the cross-referenceable index that downstream tools (Drafter, Auditor, Advisor) consume — so a single document upload populates every tool's working memory in one pass.

4 • Extract-and-destroy provenance

`lib/chomper/secure-destroy.js` enforces the rule that the original file never persists outside the request-handling memory window:

1. The buffer comes in on the request body, in-memory.
2. The analyzer reads it, produces structured text, and writes the extracted text to the database.
3. The buffer is zeroed (`buf.fill(0)`) before `process()` returns.
4. The temporary file (if any was written by the multipart middleware) is `fs.unlink` ed.
5. The HTTP response carries only the extracted text + the confidence-scored observations. The original bytes are gone.

The downstream effect is severe and intentional: BenchSlap cannot subpoena-respond with the original of a document the user uploaded. We do not have it. We did not store it. We extracted what was in it and destroyed the container. This forecloses a class of professional-responsibility hazards that vendors using "we'll store your originals indefinitely for indexing" architectures cannot foreclose without lying about their architecture.

It also fails the test the way courts want it to fail: if challenged on what we have, the answer is "we have the extracted text and the SHA-256 of the original; we never had the original after the request completed." That is a defensible record.

5 · Why this isn't a conventional pipeline retrofit

Three architectural choices interlock:

1. **Confidence is computed once, at the analyzer, in the same module that produced the observation.** It is not bolted on later by a classifier downstream of the OCR. A vendor that today emits "high relevance" cannot retrofit a four-factor formula because their analyzers do not surface the factors — they emit a single posterior probability from an opaque model. To produce the four factors, the analyzer's internals must expose them; that requires rewriting the analyzers, not wrapping them.
1. **Extract-and-destroy requires the analyzer to be in-process, not on a separate microservice with its own storage tier.** A vendor whose analyzer is a queued job on an external worker cannot enforce the buffer-zero-on-return contract because the worker does not own the request lifecycle. To enforce it, the analyzer must be a same-process call, which forces an architectural choice the vendor's existing pipeline cannot make without rewriting the pipeline.
1. **The output contract is uniform across modalities.** A vendor whose PDF analyzer returns `{pages, text, tables}` and whose image analyzer returns `{description, tags}` cannot unify the downstream classifier without rewriting the analyzers. Chomper's contract `{type, report, metadata, observations}` is the same for every modality, which is what allows the preprocessor / authority-index / fact-freezer pipeline downstream to be a single classifier rather than a per-modality fan-out.

These are interlocking design choices. A competitor can copy any one in isolation but the value comes from the conjunction.

6 • Novelty claims

1. **Quantitative four-factor confidence formula** applied to every forensic observation produced by a heterogeneous evidence ingestion pipeline, rendered as a percentage with the four factors visible to the user on every report.
2. **Same-process extract-and-destroy provenance** for evidence ingestion, in which the original buffer is zeroed in memory before the response returns, and no original ever persists outside the request-handling window.
3. **Uniform output contract across heterogeneous modality analyzers** (PDF, image, audio, video) such that downstream classification, authority indexing, and fact freezing are a single pipeline rather than a per-modality fan-out.
4. **Confidence presets calibrated by claim type** (metadata fact, direct measurement, pattern match, inference, speculation), each producing a numerical score that is auditable against the four-factor formula and defensible against cross-examination.

USPTO provisional `04-chomper-evidence-ingestion.md` filed 2026-04-26 covers the conjunction of these elements.

Filed alongside USPTO provisional. © Richard L. Sanders 2026. All rights reserved except those granted by the project's open license.