Live webinar · Wed 3 June 2026

Under the Hood: How agentic AI actually does contract work (reliably)

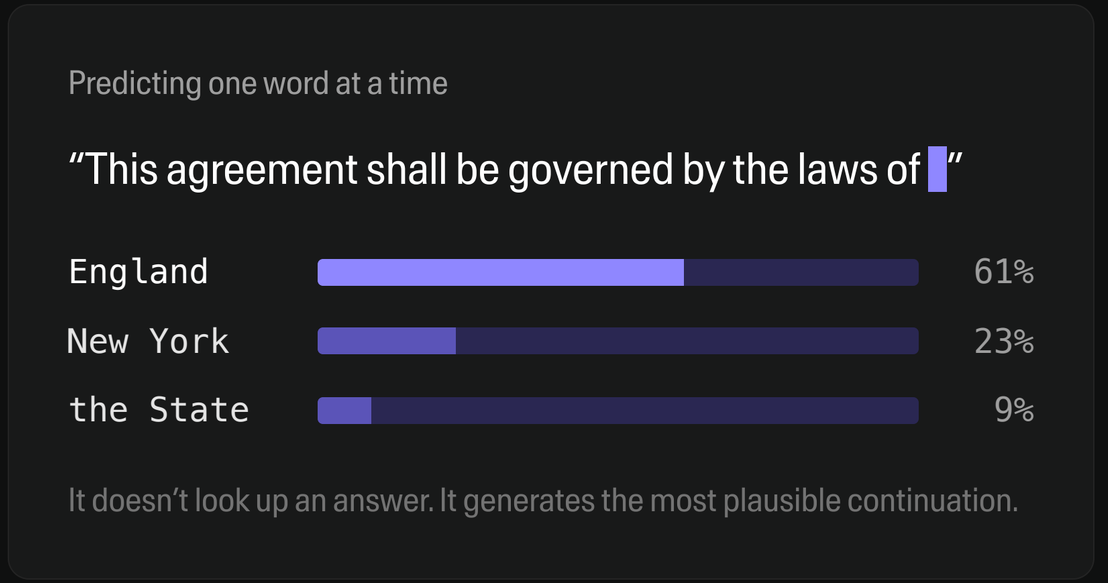
Martin Lukac

CTO, Flank

Flank*

What an LLM actually is.

- A statistical model of language — nothing more.
- At runtime it predicts the next word. One at a time, in a recursive loop.
- Trained on an internet-scale pile of text — books, code, contracts, the web.
- That is the entire core trick. Everything else is engineering around it.



How it learned — two phases.

Internet-scale text — books, code, contracts, the web

→

01 Pre-training

Read everything. Adjust **billions of weights** until its predictions match what humans actually wrote.

Not learning facts — learning word probabilities.

base model

02 Post-training

Teach it to be **helpful** — follow instructions, decline harm. Humans rate; the model is nudged.

Most recent capability jumps come from here.

The leaps you've seen lately come from the **second** phase — not from ever-bigger models.

Three consequences you can’t ignore.

No real-time knowledge

It was trained on a snapshot. Ask about yesterday and it guesses — or confesses.

No database lookup

It doesn’t retrieve facts — it reconstructs them from patterns. Lossy compression, not lossless.

No guarantee of truth

It optimises for plausibility, not correctness. That is the root cause of hallucination.

Truth is a problem you solve at the system layer — not the model layer.

How you make it reliable — four wrappers.

Context

Feed it your documents as it works — playbooks, precedents, clause library. The industry calls it RAG.

Tools

Let it search, query systems, send mail, update records. This is when a chatbot becomes an agent.

Guardrails

Hard rules. Off-limits clauses, escalation thresholds, what may ship without a human.

Human in the loop

Review the right outputs — exceptions, flagged risk — not every single one.

90% of whether a legal AI product is any good is these four wrappers — not the underlying model.

The capability ladder.

01

Chatbot

You drive every turn. Ask, it answers. ChatGPT in a browser.

15% autonomous

02

Copilot

Embedded where lawyers already work. It suggests, you execute. Every contract still passes through a lawyer.

50% autonomous

03

Agent

Given an outcome, it executes multi-step work. A lawyer supervises exceptions — the work leaves the queue.

90% autonomous · human on exceptions

Part 2 · Why models keep improving

Four drivers — none close to exhausted.

Scale

More data, parameters, compute. Still pays off — returns flattening.

Reasoning

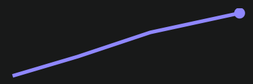
More invisible work before answering. Most of last year's gains.

Post-training

Verify its own reasoning, follow complex rules, refuse cleanly.

Scaffolding

Retrieval, tools, evaluation, multi-agent. Improving fastest — and what matters most for legal.



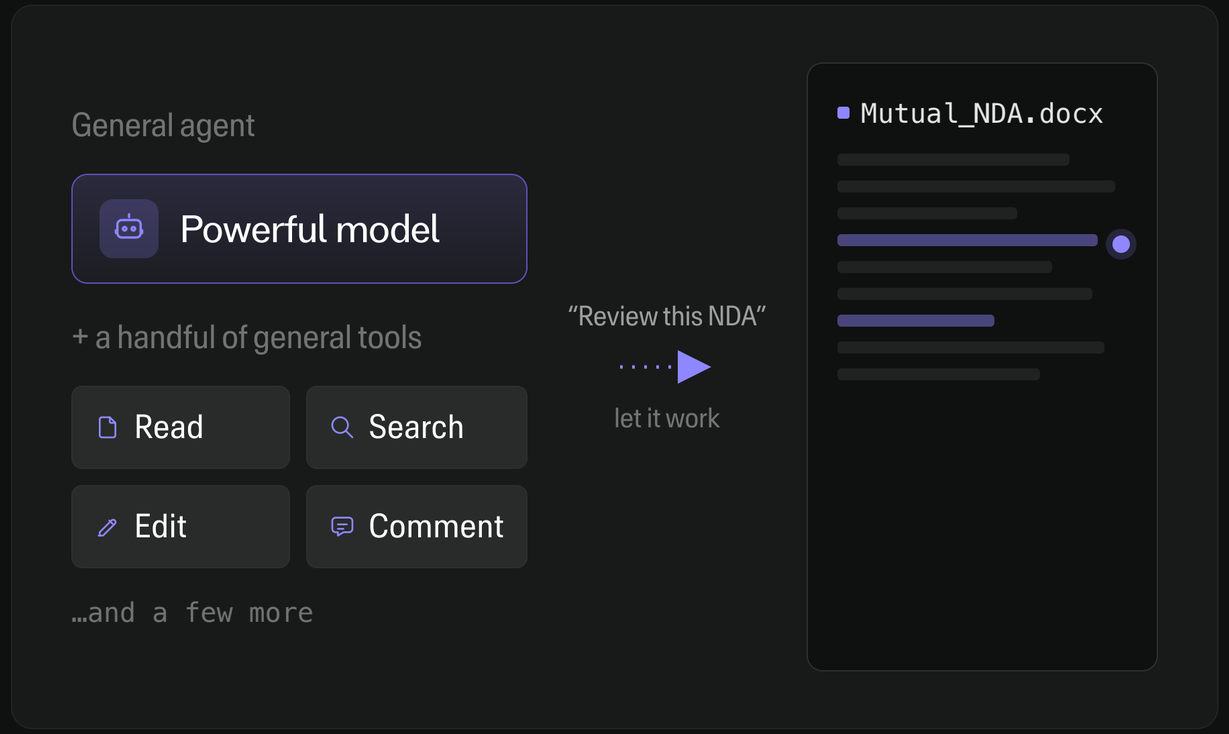
Everyone has the same engines. The question is no longer how good the model is — it's what you do with it.

Hand a contract to a general-purpose agent.

Claude Cowork · the Claude plug-in for Word · even a tool tuned for legal — all the same shape underneath.

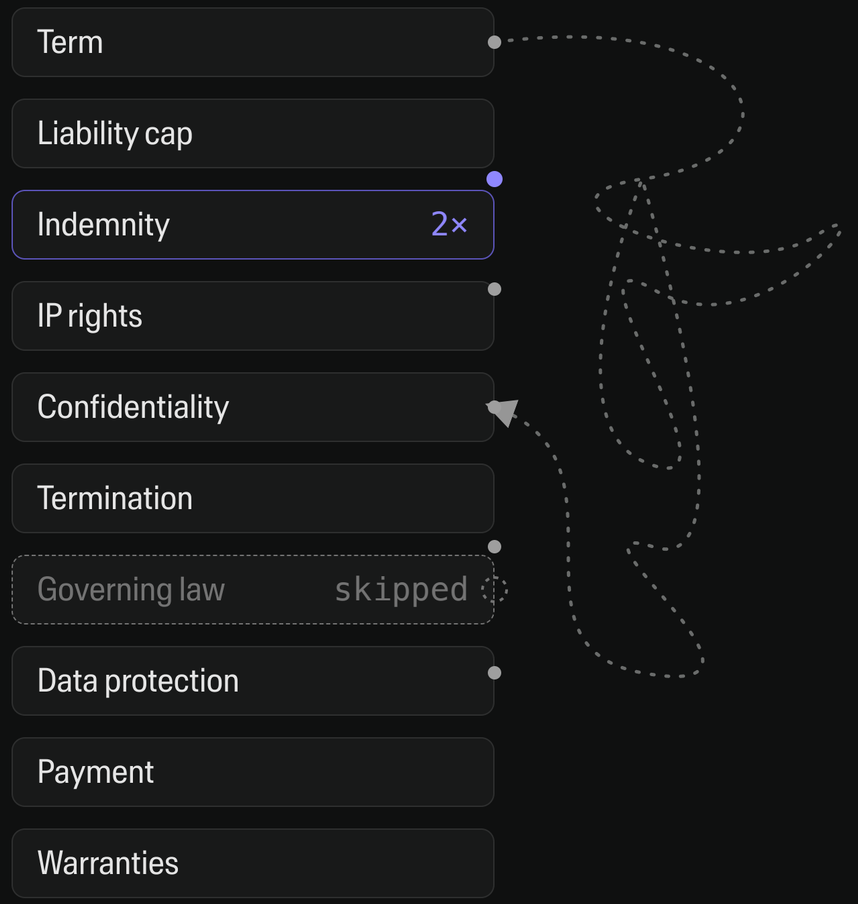
- A powerful model.
- A handful of general-purpose tools — read, search, edit, comment, a few more.
- Point it at a document, give an instruction, let it work.

And it works. Genuinely.



...until you watch how it spends its effort.

The agent decides where to look



- Burns effort orienting, not reviewing.
- Reviews non-systematically — jumps around, no fixed order.
- Forgets provisions as its context fills.
- Switches — a different answer on the same clause across runs.
- Breaks formatting and tracked changes on the way through.

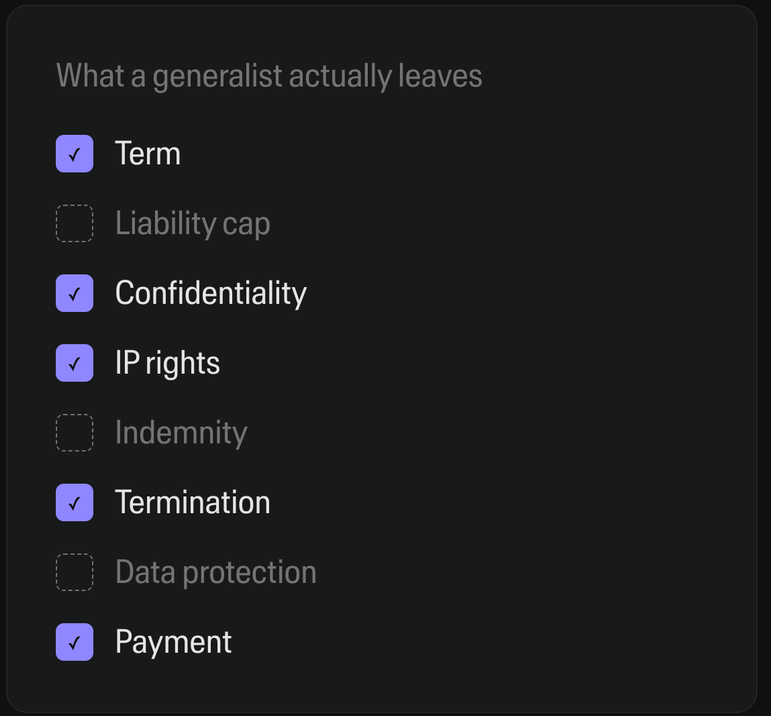
None of this is the model being weak. It's the model being unconstrained.

Contract review has a hard requirement.

Exhaustive, consistent coverage.

Every provision. Every time. The same way.

A self-directing generalist can't guarantee that. It can only try — and "usually" isn't the bar when the clause it skipped ends up in front of the regulator.



A tool is not an outcome.

What a tool gives you

Capability

A capable model and a handful of general actions. Even one tuned for legal. Necessary — and not the thing you need.

the gap

the actual work

What you actually need

An outcome

Every contract reviewed — completely, consistently — inside your workflow, in a form you can stand behind.

Closing that gap takes four things — none of them the model:

01 Orchestration

02 Legal engineering

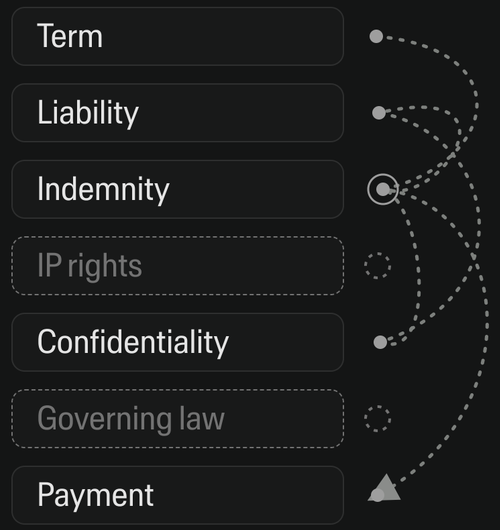
03 Integration

04 Supervision

Part 4 · Closing the gap

Take the wheel back from the model.

Self-directed — the model decides



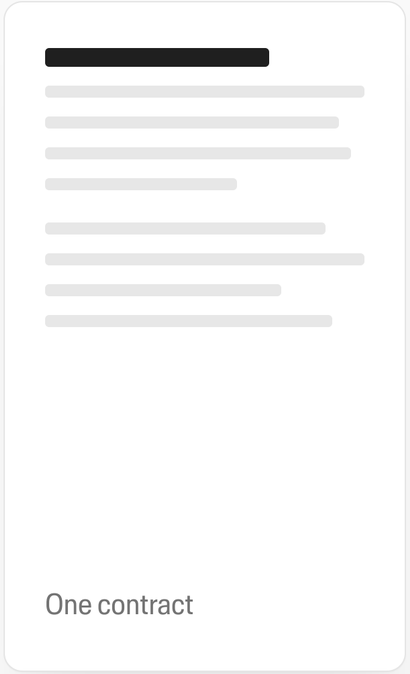
○ skipped ◎ double-reviewed wandering order

System-directed — the system decides

Every provision · in order · nothing skipped

The model still does the thinking. The system decides what gets thought about, and how much.

Systematic and complete — nothing omitted.



decompose

- ✓ Term
- ✓ Liability cap
- ✓ Indemnity
- ✓ IP rights
- ✓ Confidentiality
- ✓ Termination
- ✓ Governing law
- ✓ Data protection
- ✓ Payment
- ✓ Warranties

Completeness stops being something you hope for and becomes something you enforce. Omission isn't an option the model is allowed to take.

Deliberate effort — spend thinking where the risk is.

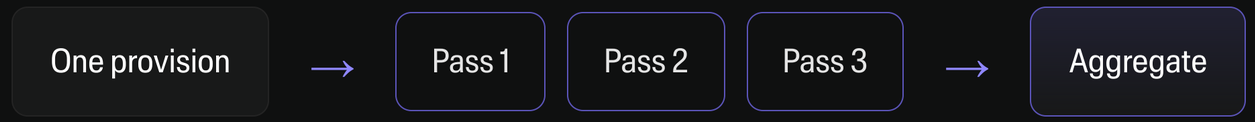
- Boilerplate → a light pass.
- Risky, negotiated, ambiguous → heavy scrutiny.

light — heavy

But everything gets covered. Effort is allocated, not improvised.

Part 4 · Closing the gap

Redundant review — check each provision N times.



Independent passes, then reconciled — like a senior lawyer reading a tricky clause twice.

Hallucinations

A made-up issue shows in one pass, not the others — it **washes out**.

✓ ✓ ☐

Ambiguous wording

Passes **disagree** — that disagreement is the signal. Flag it for a human.

⚑ to human review

Switching

Different answers across runs **collapse** toward the consistent one. Consensus is stable.

✓ ✓ ✓

Part 4 · Closing the gap

Orchestration is one quarter of it.

02

Legal engineering

Your playbooks, your positions, and the implicit knowledge in your senior lawyers' heads — drawn out and made machine-usable. The part no competitor can copy, because it's yours.

03

Implementation & integration

Wired into your DMS, CLM and inbox — the way work actually arrives and leaves. A review that lives in a tool nobody opens is not an outcome.

04

Monitoring & supervision

Every run observed, drift caught early, and the right exceptions — only those — routed to a human. Reliability you can audit over months, not just demo.

The model is the easy quarter — everyone rents the same one. The other three are built, for your team, over time.

Part 5 · The outcome you can trust

Reliability, as a property of the system.

Completeness

Every provision, every time.

Consistency

Same contract in, same review out — 9am or midnight.

Hallucination suppression

Redundancy filters the noise before it reaches a person.

Formatting protected

Deterministic tooling makes the edits — not the model. The hardest one.

“Reliably” isn’t a smarter model — everyone has the same models. It’s the discipline of refusing to let the model decide what to skip.

Part 5 · The outcome you can trust · honest section

Where even this struggles.

Judgment

Coverage and consistency don't manufacture a view where the playbook is silent.

Novel provisions, adversarial drafting, true judgment calls → a human. The system makes sure those are the only things that reach them.

The format itself

A Word file isn't text — under the hood it's a zip of deeply nested XML.

```
<w:p>
  <w:pPr><w:numPr><w:ilvl w:val="0"/>
    <w:numId w:val="7"/></w:numPr></w:pPr>
  <w:r><w:rPr><w:b/></w:rPr>
    <w:t>Indemnification.</w:t>
  </w:r><w:ins w:author="…">…</w:ins>
</w:p>
```

One small malformation and Word breaks. The fix isn't a smarter model — it's not letting the model freely rewrite the document. Nobody has fully solved it.

Any tool — even a great one — gives you capability.

An outcome takes orchestration, legal engineering, integration, supervision.

Same model underneath. The gap between the two is the product.

Questions.

Drop them in the chat. ~15 minutes.