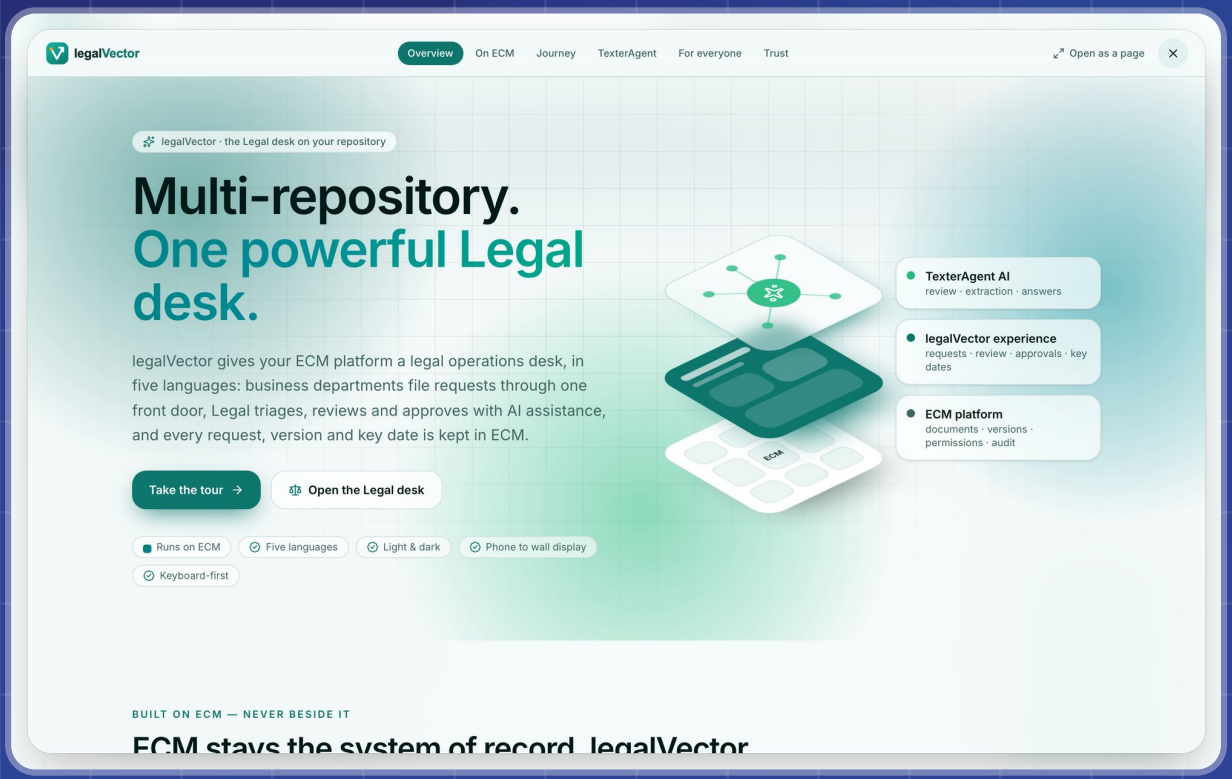
TEXTERBLUE · WHITE PAPER

legalVector

Multi-repository. One powerful Legal desk.

The AI-assisted Legal operations desk for the content repository you already run — in five languages.



Edition 2026 · v1 **Published by** TexterBlue **Classification** Public — Product White Paper **Contact** www.texter.ai

CONTENTS

Inside this paper

Multi-repository. One powerful Legal desk. 4

The repository stays the system of record 5

One Legal desk, on the repository you already run 6

Common work is scattered across too many tools 7

One Legal desk, on the repository you already run 8

PART 1 · THE LEGAL DESK 9

From “can Legal look at this?” to an approved contract — in six visible steps 10

A request form people actually use 12

Triage in seconds, with the whole team in view 14

Everything about a matter, on one page 15

An AI that reads contracts the way your lawyers do 17

Real Word redlines, generated with care 20

Jurisdiction-aware, multilingual by design 22

Notice deadlines that never slip 23

Figures you can trust — every number opens its list 24

A playbook that evolves — under Legal’s control 27

Two layers of protection — enforced on the server 29

Set up in minutes, on your repository 30

PART 2 · WORKFLOW, PLATFORM AND TRUST 31

One Legal process, on the engine that fits 32

Every capability, through the repository’s own interfaces 33

The same calm experience on every screen, in every language 34

A good citizen of your content landscape 36

Fits your organisation — without code 37

User story catalogue — 77 stories 38

Functional map 43

Glossary 44

EXECUTIVE SUMMARY

Multi-repository. One powerful Legal desk.

Every organisation's Legal team does the same work: it receives requests, triages them, reviews and negotiates contracts, approves, and watches the dates that follow. Most of that work still happens in e-mail, attachments and spreadsheets — beside the content repository where the contracts already live. **legalVector** brings that work onto the repository: a complete, AI-assisted Legal operations desk that runs on the content repository the organisation already has.

legalVector is a modern web application from TexterBlue. It adds no second repository and no parallel permission model: every request, version, field, permission and key date stays in the organisation's repository, and every call runs as the signed-in user with that user's own rights. The Legal desk — requests, triage, AI review, Word redlines, the approval gate, key dates, analytics and the rulebook — works the same on every supported repository, and each passes the same 124 end-to-end checks.

It brings three things to a Legal team:

- ✓ **A front door to Legal for every department.** Requests in a minute, prefilled from the Word document, NDAs generated from Legal's template, status visible to the requester — in their own language.
- ✓ **An AI-assisted review desk.** TexterAgent reviews every contract against the organisation's own playbook, proposes wording and anchors each finding to the exact words; lawyers turn accepted fixes into genuine Word tracked changes, and an approval gate makes every accepted risk explicit.
- ✓ **Legal intelligence and governance.** Dashboards where every number opens its list, key dates on the contract, Counterparty 360, portfolio exposure, a board pack, and a versioned rulebook that only the Head of Legal approves.

7	77	6	5
request types — from contract review and NDAs to a legal opinion	user stories, listed by persona in Appendix A	visible steps from a business question to an approved contract	languages — English, Spanish, Portuguese, French and Arabic

legalVector is supported today on Doxis 4 and Alfresco Content Services. Workflows run on an embedded BPMN engine, or on the repository's own BPM.

Our position. legalVector is built on the repository, through its official APIs, beside its own clients — never in front of them. It exists to give Legal one powerful desk on the platform the organisation already trusts.

All screenshots in this paper are taken from the legalVector application (November 2026 build) with fictional demonstration data. People, organisations and contracts shown are invented.

BUILT ON YOUR REPOSITORY — NEVER BESIDE IT

The repository stays the system of record

Every capability in legalVector is designed around one principle: **the repository stays the system of record.** legalVector builds on it.

Organisations choose their content repository for its security model, retention and compliance capabilities. legalVector duplicates none of them. It talks to the repository through its official interfaces, always under the identity of the person using it.

That has three consequences that matter to every customer:

- **Your user, your rights.** What the repository refuses, legalVector never shows. There is no super-user behind the scenes.
- **Your documents stay in your repository.** Files, versions, fields and permissions are protected, retained and audited by the repository.
- **A considerate guest.** A cap on parallel calls keeps legalVector gentle on busy systems.

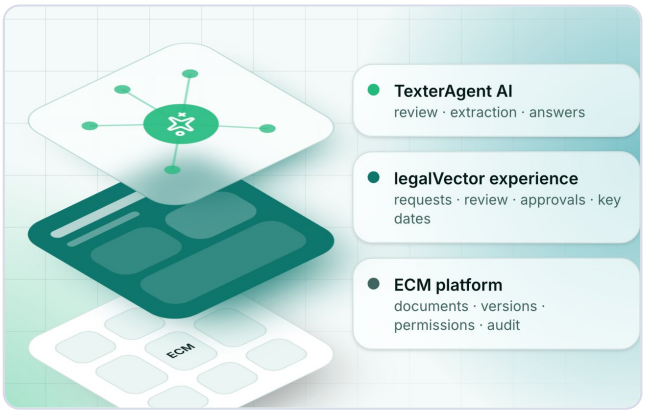


Figure 1: The repository at the base, legalVector's Legal desk on top, TexterAgent as the AI.

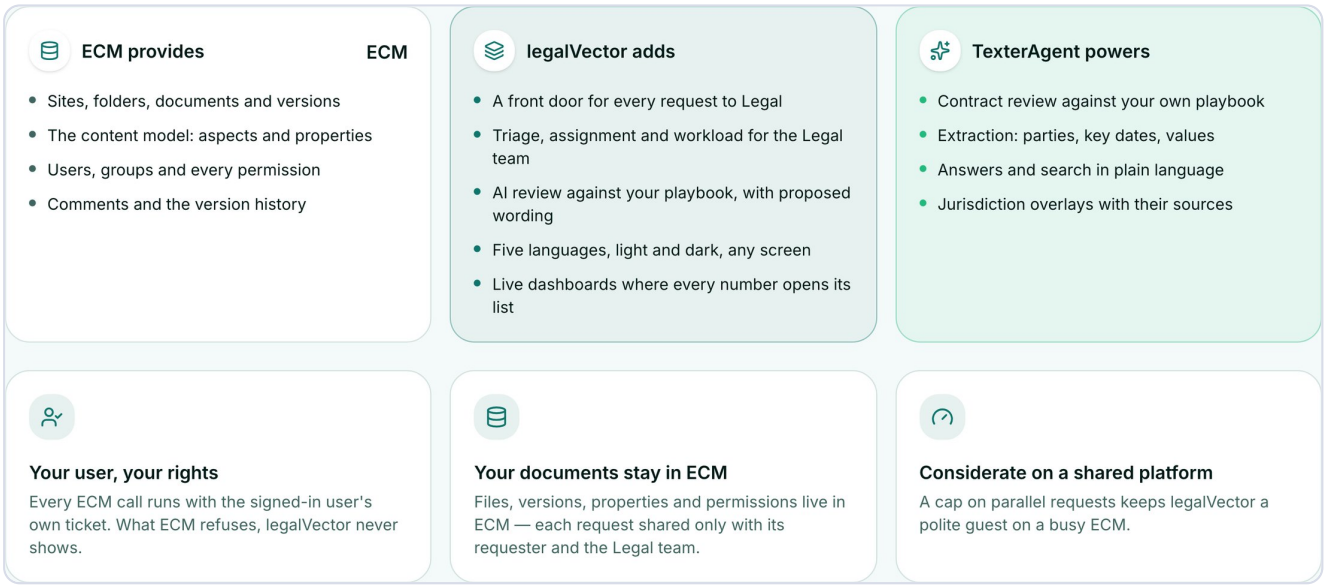


Figure 2: What each layer contributes — the repository provides, legalVector adds, TexterAgent powers.

YOUR REPOSITORY

One Legal desk, on the repository you already run

legalVector talks to its repository through **one interface**. The Legal desk — every screen, rule, report and AI capability — sits above it; the repository keeps every request, version and fact under its own permissions, and its own clients keep working on the same documents.

	In the repository
A request is	a document in the repository, in Legal's own area
Its facts	the repository's own metadata (status, assignee, due date…)
Legal's library	documents of its own kind, in the repository
Access	per request: the requester and the Legal team only
Versions	the repository's own versions
Timeline in the repository	a comment per event
Set up by	a previewed plan that creates only what is missing
Administrators	the repository's own administrators

Supported today: Doxis 4 and Alfresco Content Services.

✓ **The same desk, verified:** the end-to-end suite — roles, the workflow, redaction, access, AI review, dashboards, calendar, e-mails, compare, Ask Legal, forward-to-Legal — passes 124 of 124 checks on each supported repository.

THE CHALLENGE

Common work is scattered across too many tools

In most organisations, the contracts live in the repository — but the legal work around them does not.

A business manager who needs a contract reviewed writes an e-mail. The draft travels as an attachment. Status lives in a spreadsheet maintained by hand; deadlines in personal calendars. Redlines are exchanged as Word files with names like final_v3_LEGAL_comments.docx. Questions such as “where is my contract?” arrive by chat and phone.

Each of these steps is reasonable in isolation. Together they create cost nobody sees in one place: duplicated effort, lost versions, missed notice deadlines, and an audit trail that stops at the edge of the repository.



Figure 3: legalVector brings the legal work around the documents onto the repository, while the content stays in the repository.

legalVector answers this with one simple idea: **bring the legal work to where the contracts already are.** Requests, reviews, negotiation and approval happen in role-based screens on top of the repository, and every result is written to it — where it is secured, versioned, retained and audited.

LEGALVECTOR AT A GLANCE

One Legal desk, on the repository you already run

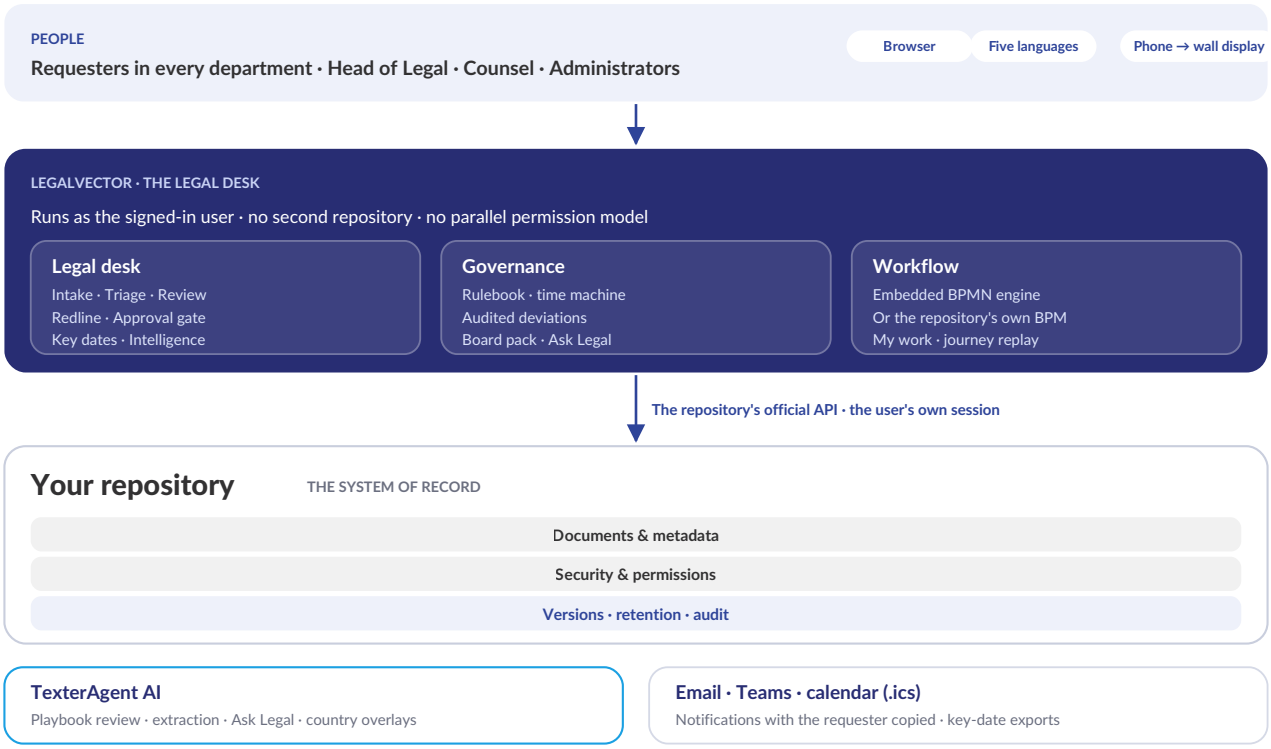


Figure 4: Architecture: people use legalVector; legalVector works through the repository with each user's own session; TexterAgent provides the AI.

Legal desk

Intake, triage, AI review, redline, approval gate, key dates, dashboards, Counterparty 360 and the command centre.

Your repository

The desk runs on the content repository you already have — requests, versions and key dates stay in it.

TexterAgent

TexterBlue's AI engine: playbook review, extraction of parties, dates and values, Ask Legal, country overlays — with lawyers always deciding.

Built on official APIs

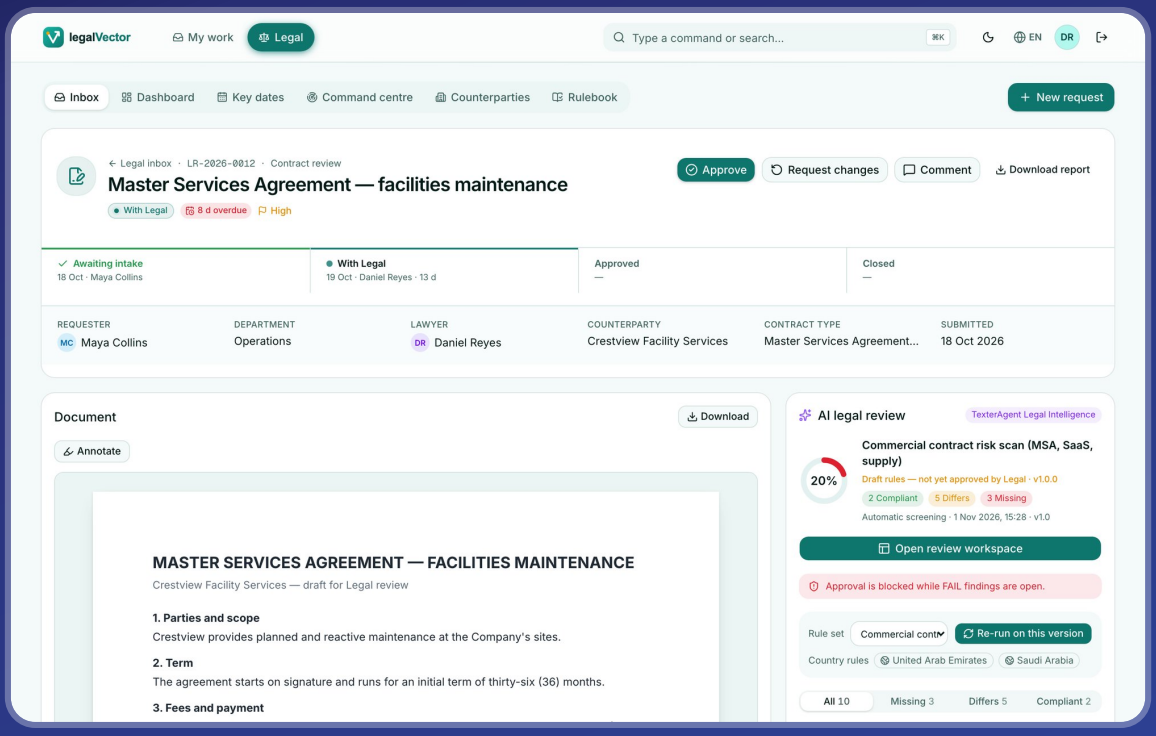
Every call goes through the repository's official interface, as the signed-in user, with that user's own rights.

01

PART 1 · THE LEGAL DESK

Give every department a front door to Legal

A complete, AI-assisted Legal operations desk on the repository you already run: from the first business question to an approved contract, with every step recorded in the repository.



A REQUEST'S JOURNEY

From "can Legal look at this?" to an approved contract — in six visible steps

Everyone sees where a request stands. Requesters see their own requests; Legal sees the whole picture. Each step is recorded in the repository.

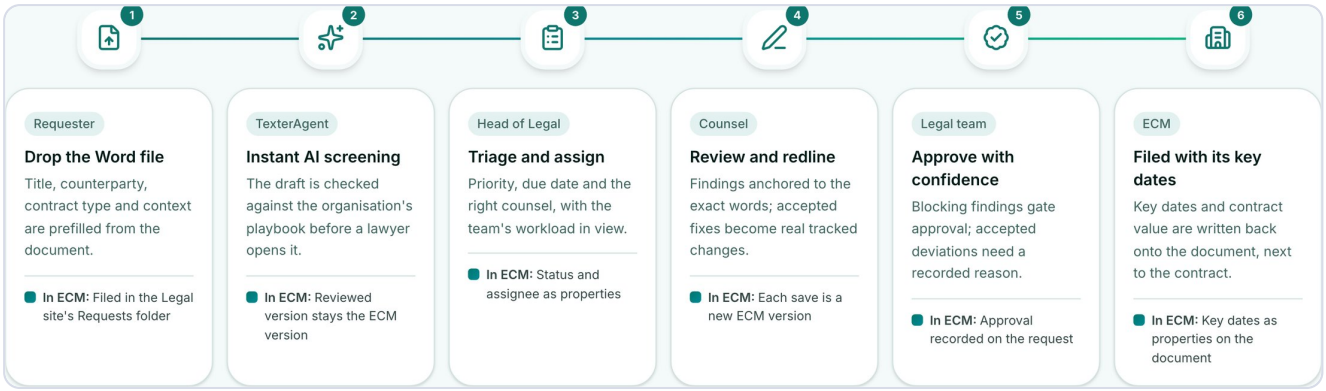


Figure 5: The journey as people experience it — and what is written to the repository at each step.

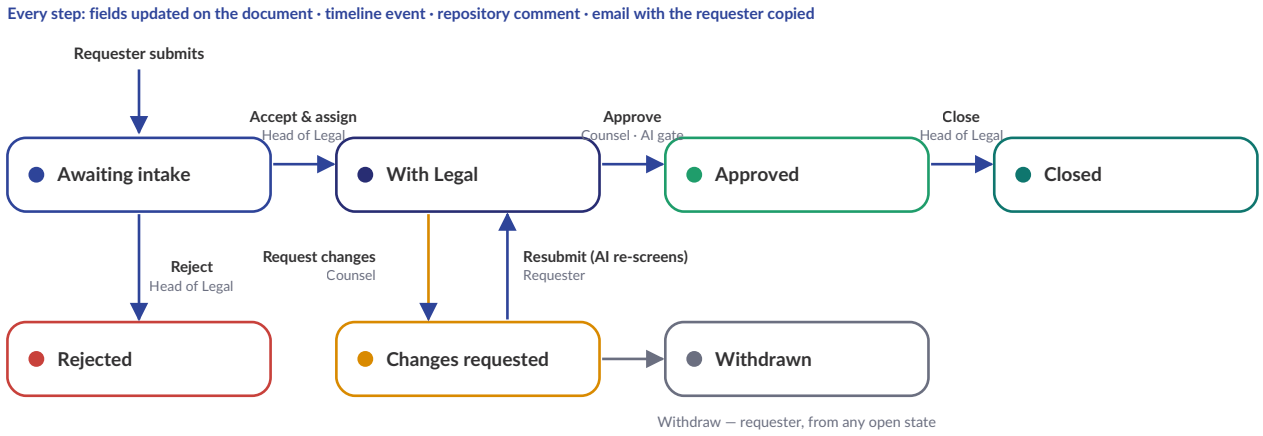


Figure 6: The Legal workflow: states, transitions and who may take them. The server re-checks every rule; the screen only offers allowed actions.

Role	What they do
Requester (any employee)	Raises requests, answers Legal's questions, uploads revised versions, resubmits or withdraws; sees only their own requests.
Head of Legal (intake)	Accepts and assigns, sets priority and due date, rejects with a reason, reassigns, closes; sees all requests and the whole team's workload.
Counsel	Reviews, annotates, requests changes, redlines with AI assistance, approves through the AI gate; sees every request in full.
Administrator	Sets up the Legal model in the repository, configures the platform and integrations; does not approve rules or requests.

INTAKE

A request form people actually use

The first moment decides whether a Legal desk is adopted. legalVector makes it fast: pick what you need, drop the document, confirm.

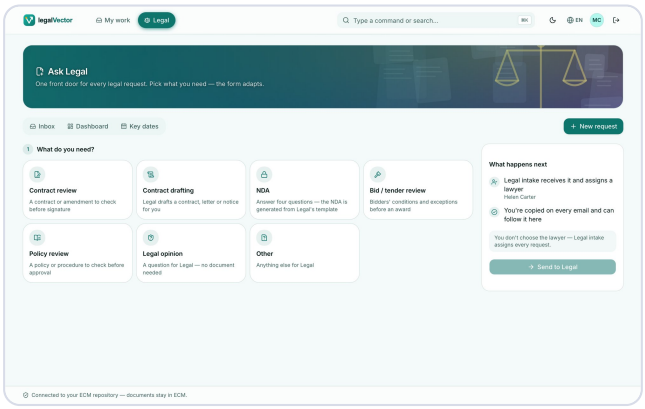


Figure 7: Ask Legal: the form adapts to the kind of request.

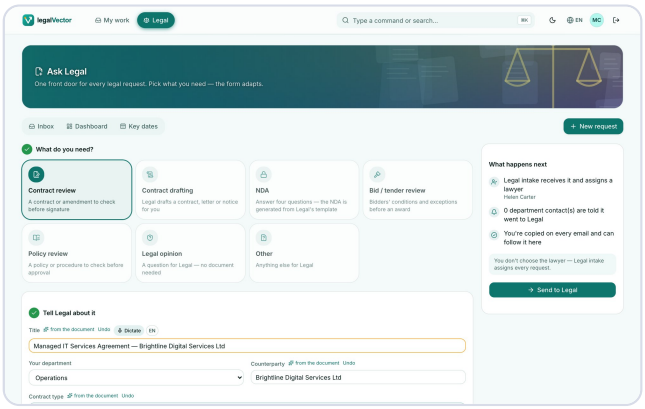


Figure 8: A dropped Word contract fills the title and the other empty fields — each value marked “from the document”, with Undo.

- ✓ **Seven request types** — contract review, contract drafting, NDA, bid/tender review, policy review, legal opinion and other — each asking only what matters for it.
- ✓ **Prefill from Word:** title, counterparty, contract type and “what Legal should know” are read from the document by rules and by TexterAgent; only empty fields are filled.
- ✓ **NDA in four answers:** counterparty, mutual or one-way, purpose and term generate the NDA from Legal’s approved Word template, with its own document number.
- ✓ **Department review attestation:** the requester confirms which departments reviewed the document, and each department’s contact is notified.
- ✓ **Forward to Legal:** an e-mail forwarded to the Legal mailbox becomes a prefilled draft that the sender confirms under their own login.
- ✓ **Voice dictation** for the title and description, in any of the five interface languages.

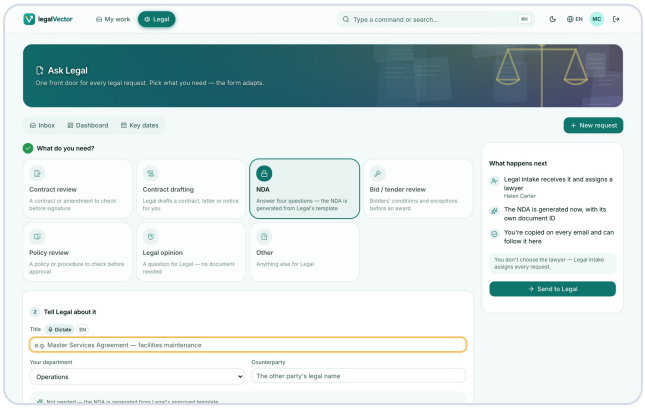


Figure 9: NDA requests generate the document from Legal's template.

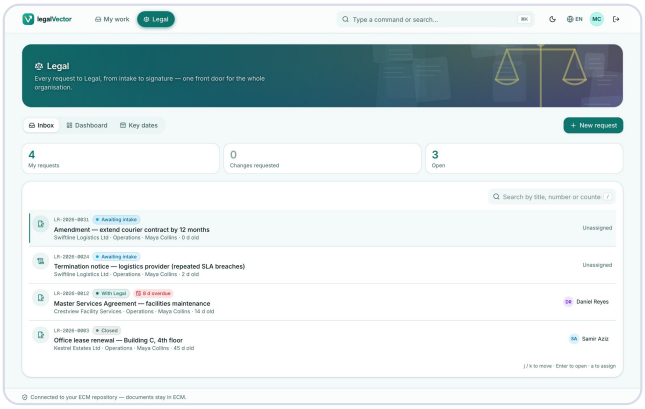


Figure 10: Requesters follow their own requests — status, lawyer and due date — without chasing Legal.

TRIAGE

Triage in seconds, with the whole team in view

The Head of Legal sees an “Awaiting intake” queue in arrival order, with a suggested lawyer for each request and every lawyer’s live workload and availability.

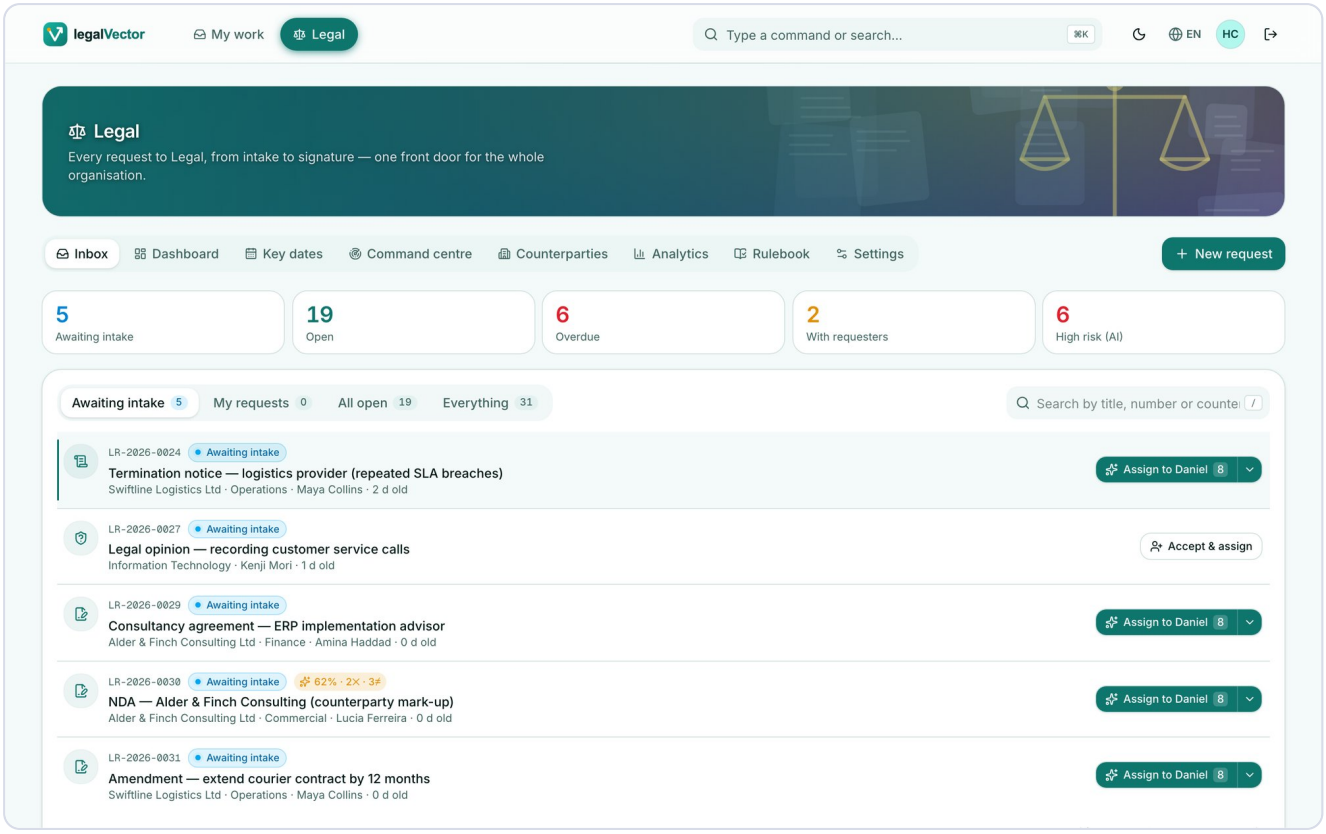


Figure 11: The Legal inbox for the Head of Legal: queues, KPI tiles where every number opens its list, and one-click “Assign to …”.

- ✓ **One-click assignment** to the suggested lawyer — routing by contract type, then request type — or choose another with open/overdue counts and out-of-office shown.
- ✓ **Priority and due date** on acceptance, with a default turnaround.
- ✓ **Reject with a reason** the requester receives; reassign at any time.
- ✓ **Keyboard-first:** j/k to move, Enter to open, a to assign.



Figure 12: Today’s brief: built from exact figures, every line opens its list.

ONE PAGE PER REQUEST

Everything about a matter, on one page

The request page tells each person what is theirs to do next, and shows the document, the AI review, the people, the timeline and the live process — in one place.

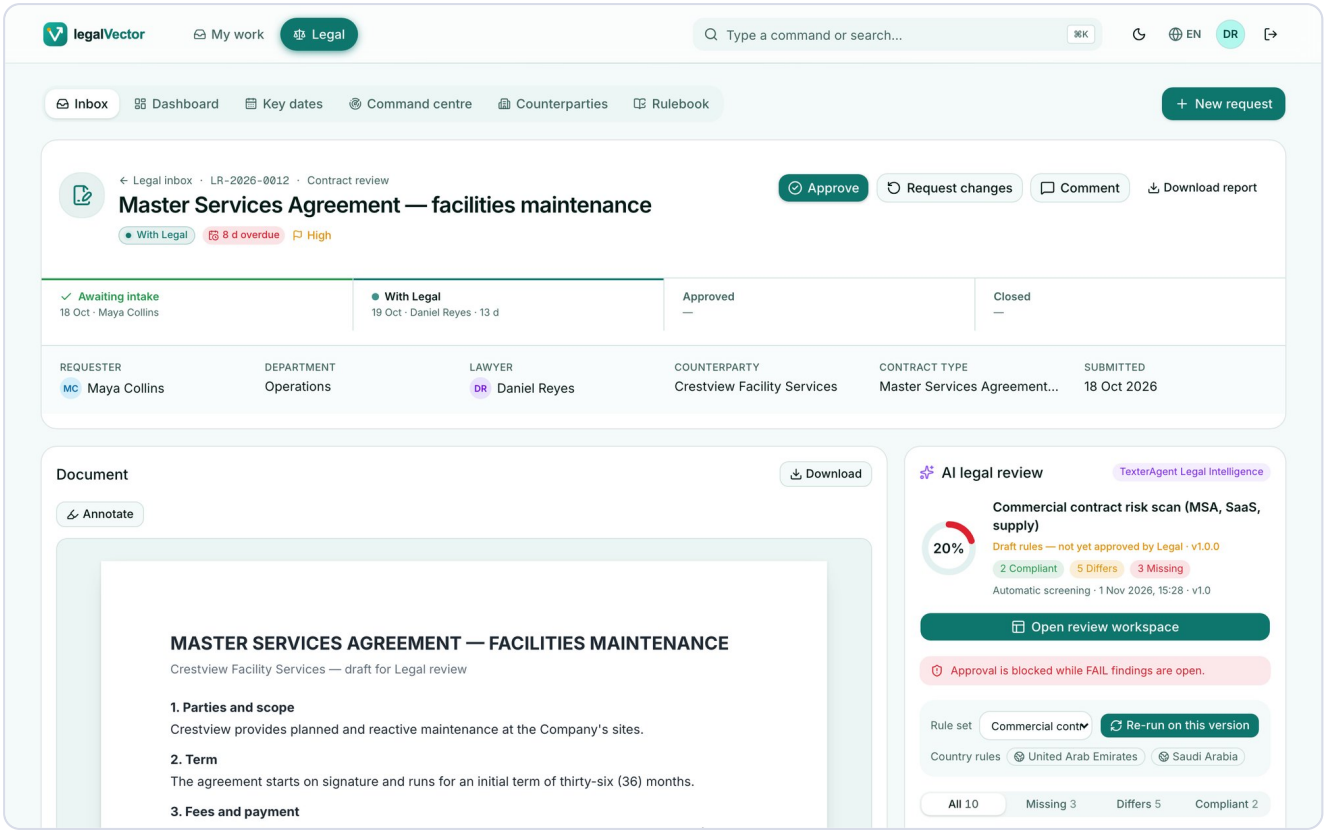


Figure 13: A contract review with Legal: the journey stepper, people, the Word document rendered in place and the action bar for the assigned lawyer.

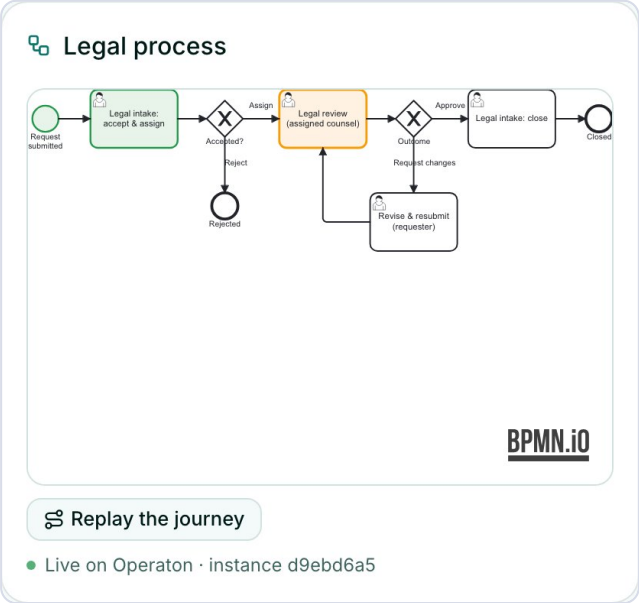


Figure 14: The timeline records every event — and every notified person with the real delivery result.

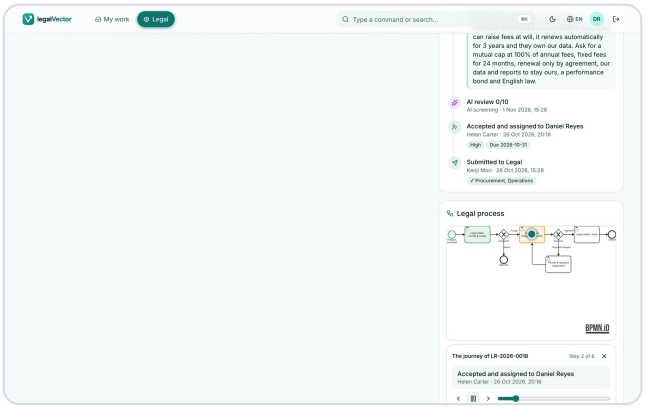


Figure 15: Journey replay: the request's real history travels the BPMN diagram, step by step.

- ✓ **Annotations anchored to the exact words** of the Word document, sent to the requester with "Request changes".
- ✓ **Every event mirrored in the repository** as a comment on the document, with its fields (status, assignee, due date, priority) updated on the document.
- ✓ **An audit report PDF** per request, respecting the reader's visibility.

TEXTERAGENT · THE AI INSIDE THE LEGAL DESK

An AI that reads contracts the way your lawyers do

TexterAgent is TexterBlue's AI engine. In legalVector it reviews every contract against the organisation's own playbook, explains each finding, proposes wording — and leaves every decision to a lawyer.

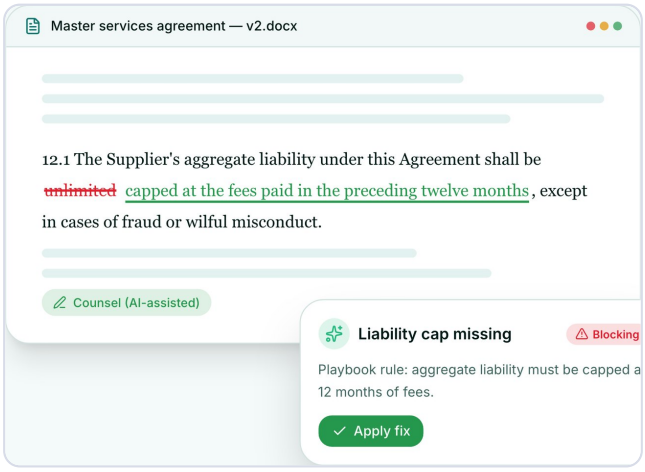


Figure 16: A finding anchored to the exact words becomes a genuine tracked change.

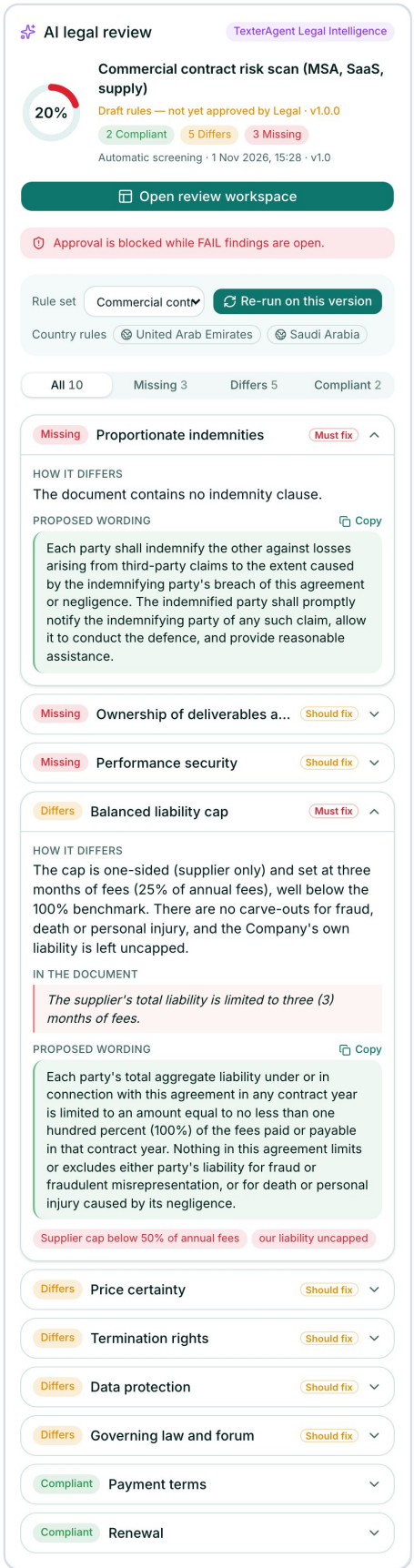


Figure 17: The AI review on a request: score, compliant / differs / missing, proposed wording, rule set version and country overlays.

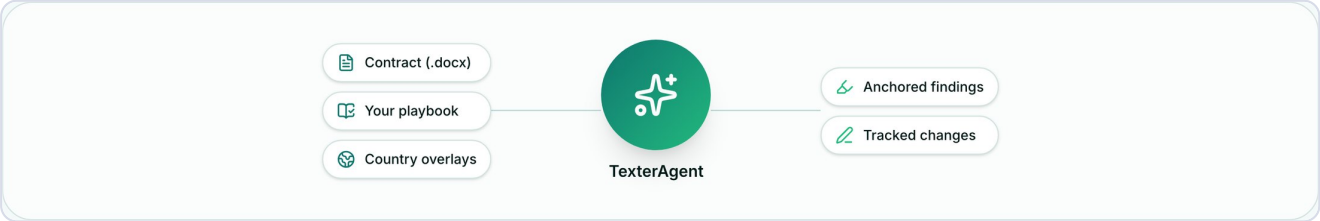


Figure 18: Contract, playbook and country overlays in; anchored findings and tracked changes out.

Playbook review

Each clause is checked against the organisation's rules, with country overlays where they apply.

Exact anchoring

Every finding points at the exact characters of the judged version — no guessing where.

Proposed wording

Fixes are suggested in the organisation's standard language and applied as Word tracked changes.

Delta re-analysis

A new version is re-checked automatically: resolved · new · still open.

Live progress

Findings stream in clause by clause while the review runs — real progress, not an animation.

Smart intake & dates

Word documents prefill the request form; key dates and values are extracted onto the document.

AI that proposes. Lawyers who decide. Redlines are authored "<lawyer> (AI-assisted)". Accepted deviations require a written reason and stay visible. Requesters never see internal reviews. Shipped rule sets are labelled drafts until Legal approves them, and the AI key never leaves the server.

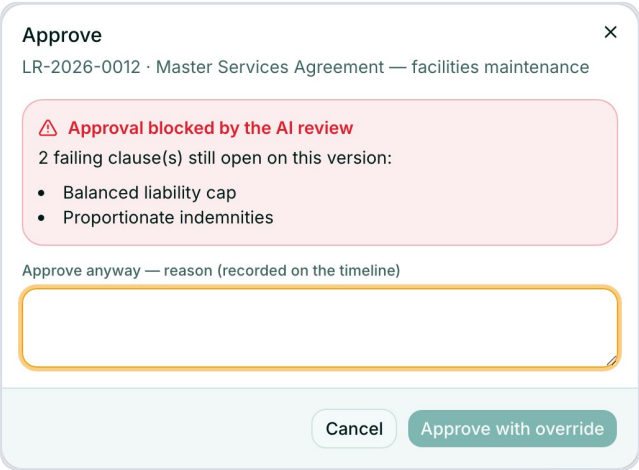


Figure 19: The approval gate: approval waits while must-fix findings are open — an override needs a reason, recorded on the timeline.

The approval gate

When enabled, a lawyer cannot approve while the latest review of the current version still has open must-fix findings. Accepting a deviation with a reason, applying the fix or overriding with a recorded justification are the only ways forward — so risk acceptance is always explicit and auditable.

REVIEW WORKSPACE

Real Word redlines, generated with care

The review workspace puts the Word document beside a Smart Review panel. Every staged change appears live as a tracked change; saving writes a genuine .docx as a new version in the repository.

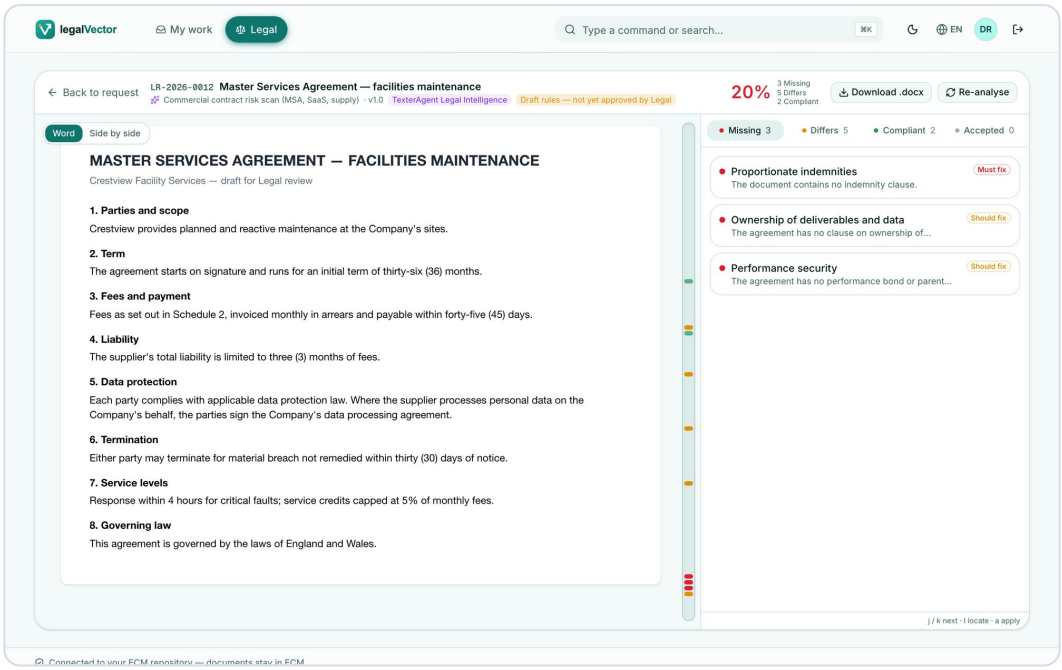


Figure 20: Review workspace: the contract as Word renders it, the clause map and findings grouped as Missing, Differs, Compliant and Accepted.

- ✓ **Locate** highlights the exact words; **Apply change** and **Insert clause** stage edits — with the position chosen by clicking the clause it should follow.
- ✓ **The Redline Engine** writes genuine w:ins/w:del tracked changes, authored by the lawyer (AI-assisted), with minimal word-level differences. Numbering, styles, tables and signatures stay untouched.
- ✓ **Server-side integrity:** changes are rebuilt from the stored review, never from browser offsets; a save is refused if the review judged an older version.
- ✓ **Automatic re-analysis** after every save shows what was resolved, what is new and what is still open.

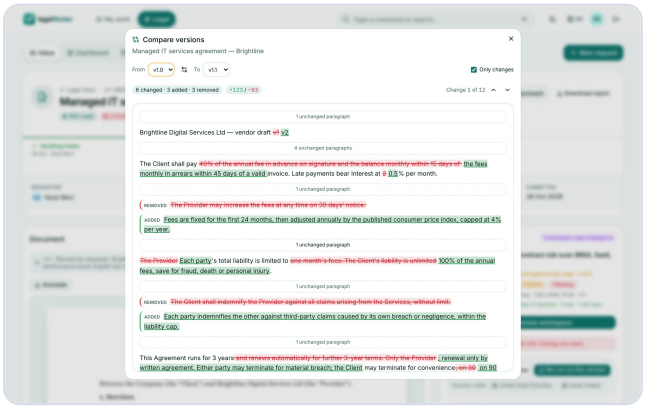


Figure 21: Compare any two versions: paragraphs aligned, changes word by word.

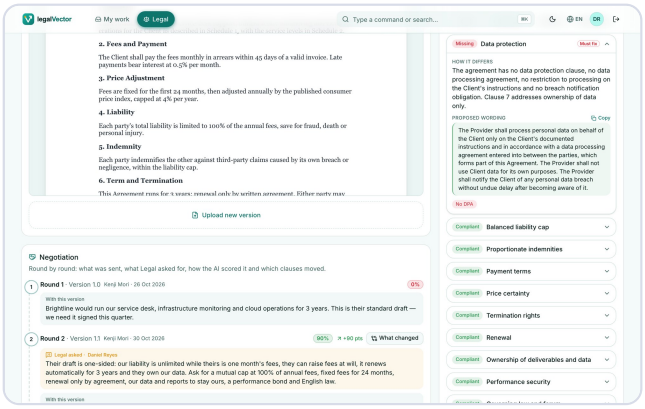


Figure 22: Negotiation timeline: round by round — what Legal asked, the score delta, clauses that moved.

LOCAL LAW AND LANGUAGES

Jurisdiction-aware, multilingual by design

Contracts are checked against curated jurisdiction packs stored in Legal's library in the repository — each fact numbered, sourced and dated — never against a model's general memory.

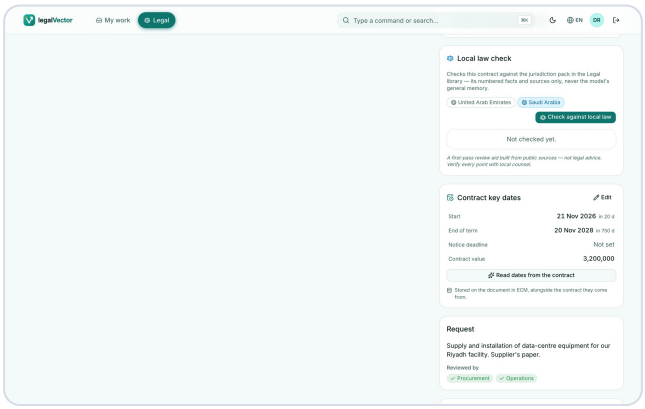


Figure 23: Local law check and contract key dates on the request page.

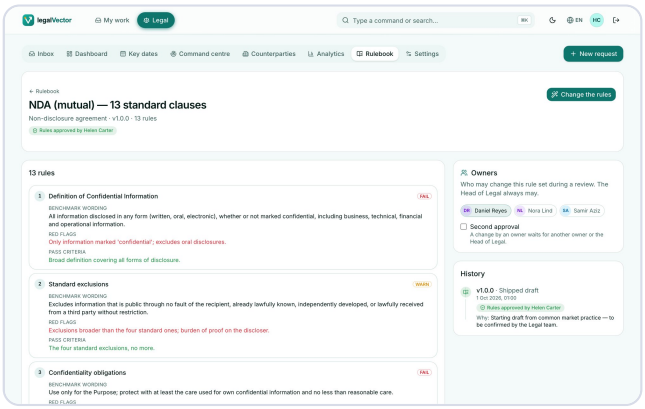


Figure 24: A rule set: owners, versions, approval state and every rule with benchmark wording, red flags and pass criteria.

- ✓ **Country overlays** are added automatically when a request mentions the jurisdiction; Legal can add further country overlays in the Rulebook editor, without code.
- ✓ **Cited facts** with an as-of date and "verify with local counsel" — a first-pass aid, clearly not legal advice.
- ✓ **A side-by-side view** places the contract beside a clause-by-clause reading translation into any of the five interface languages, each clause keeping its review colour — labelled "for reading, not the signed text".

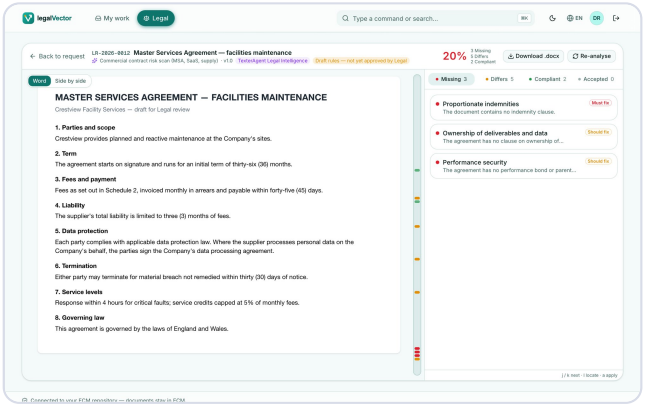


Figure 25: The review workspace: the Word document beside Smart Review, in the reviewer's own language.

KEY DATES ON THE CONTRACT

Notice deadlines that never slip

Start, end of term, notice deadline, renewal and contract value are read from the contract — by rules first, then by TexterAgent — and saved **on the request document in the repository**: as the repository's own metadata, where the repository's own tools can act on them.

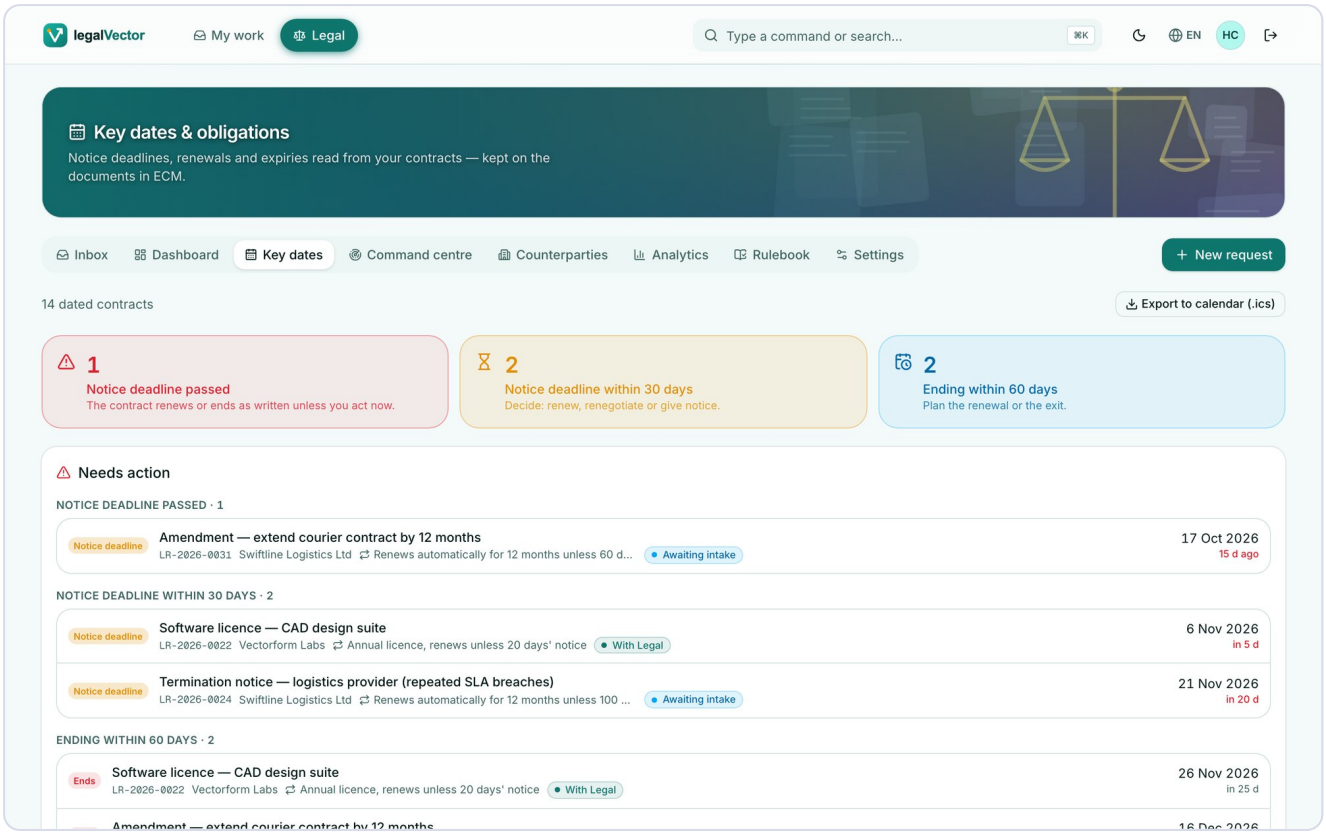


Figure 26: Key dates & obligations: needs-action buckets, the month view, the next 90 days and an .ics export.

- ✓ Each value keeps its **source clause and method** — read by AI, by rules or entered by hand.
- ✓ **Needs action**: notice deadline passed while the contract runs, notice within 30 days, ending within 60 days.
- ✓ **Calendar export** so deadlines reach the team's own calendars.

LEGAL INTELLIGENCE

Figures you can trust — every number opens its list

Dashboards in legalVector are doors, not posters: every count opens exactly the list behind it, built from the same filter definition.

The **board pack** turns the week into one PDF for the Head of Legal — figures, attention points, exposure, counterparties, key dates and team load — with every figure linking back to its list. Dashboards export to CSV, trimmed to what the reader may see.

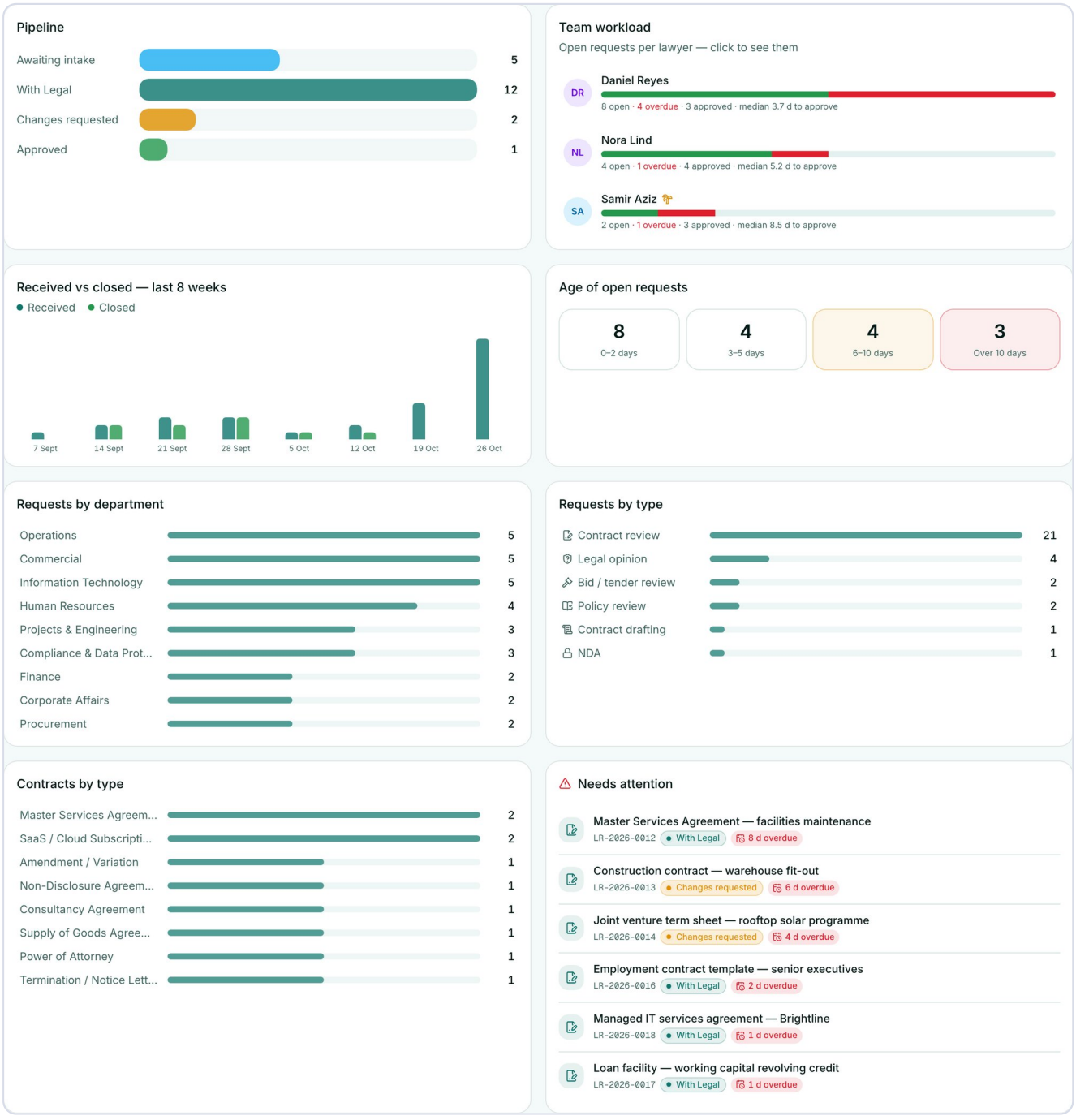


Figure 27: Pipeline and team workload — the Head of Legal sees every lawyer; lawyers see themselves.

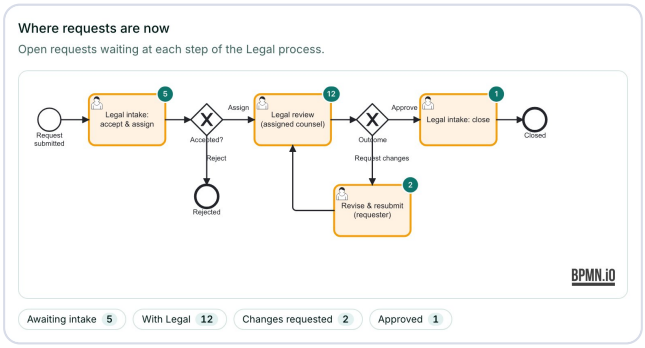


Figure 28: Analytics: open requests at each step of the live BPMN, and the median days each party holds a request.

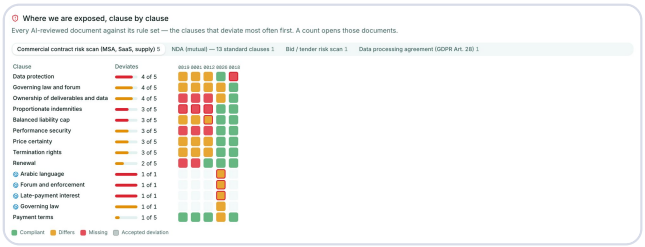


Figure 29: Portfolio exposure: which clauses deviate most across reviewed contracts — counts open the documents.

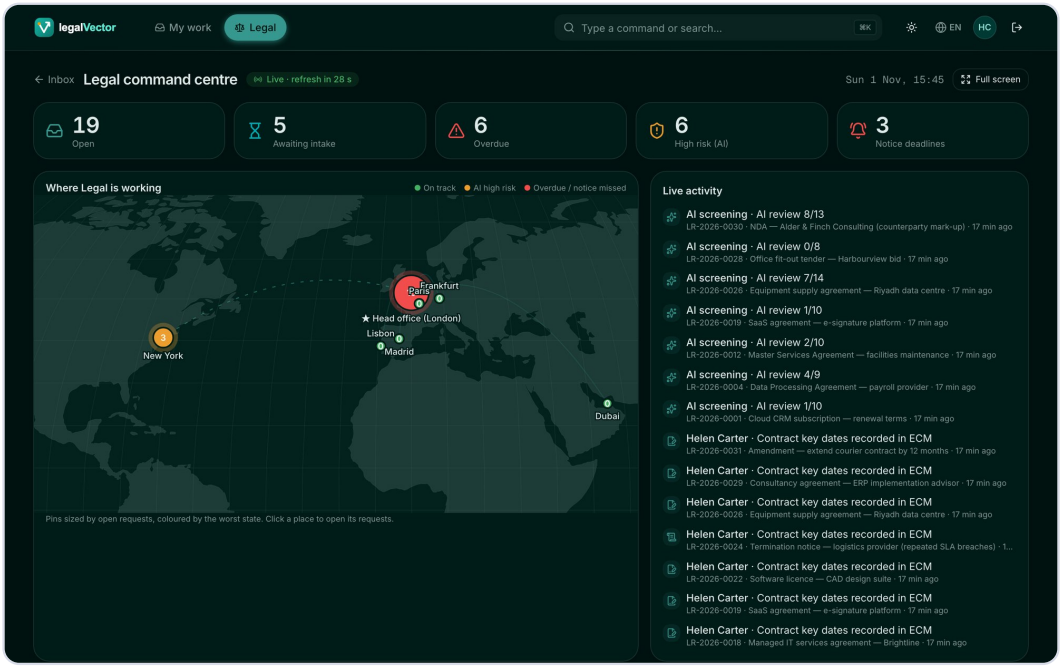


Figure 30: The command centre on a wall display: site map, KPIs, live activity, SLA watch and dates ahead (dark theme).

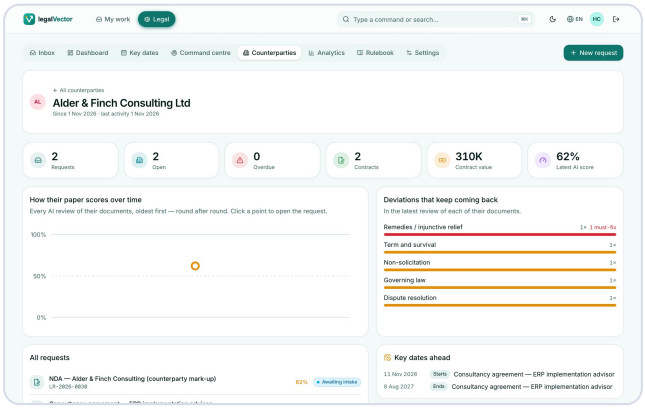


Figure 31: Counterparty 360: requests, value, AI score over time and deviations that keep coming back.

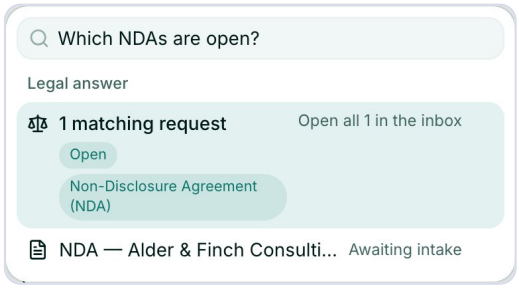


Figure 32: Ask Legal in ⌘K: questions in any of the five languages become the inbox's own filters — "Open all" shows exactly that list.

RULEBOOK & GOVERNANCE

A playbook that evolves — under Legal's control

Rule sets are versioned. Each review records exactly which version judged it and who approved it — so any past review can be explained by the rules in force that day.

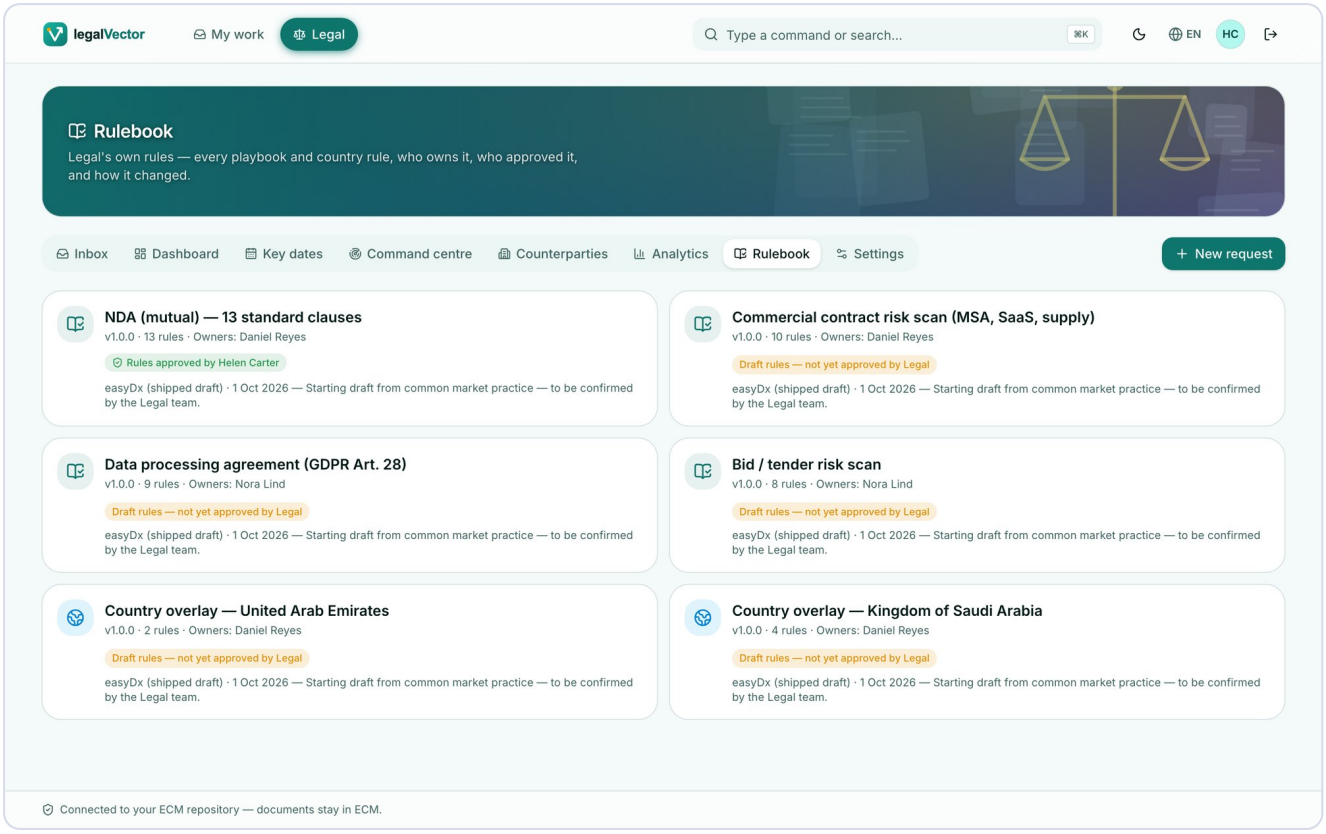


Figure 33: The rulebook: every rule set and country overlay, with version, owners and approval state.

- ✓ **Owners per rule set**, an optional second approval, and only the Head of Legal approves.
- ✓ **Change the rule in plain language**: TexterAgent proposes exact edits, flags conflicts and weakening, and previews the impact on recent documents before publishing.
- ✓ **Template edits** travel with rule changes: on approval, the NDA Word template is written as a new version in the repository.
- ✓ **The time machine** replays every version — who changed what, why, and which reviews each version judged.

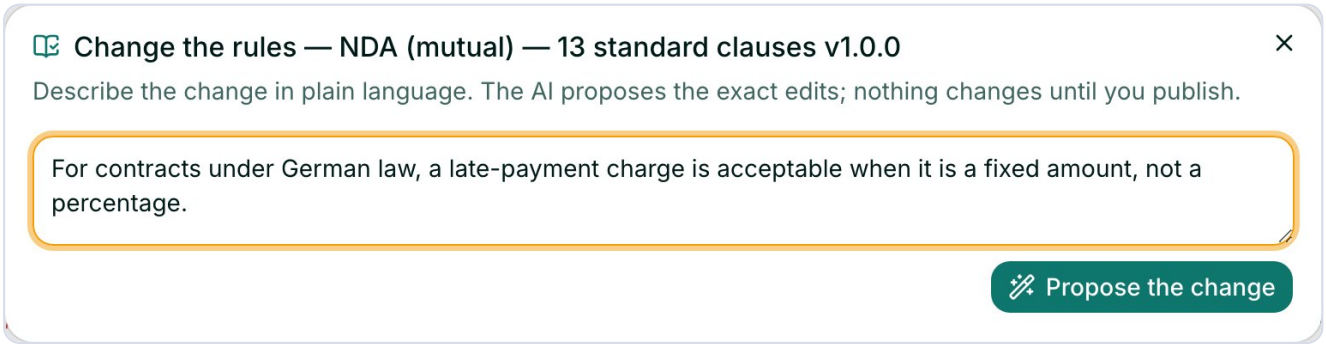


Figure 34: "Change the rules": describe the change in plain language; nothing changes until it is published.

WHO SEES WHAT

Two layers of protection — enforced on the server

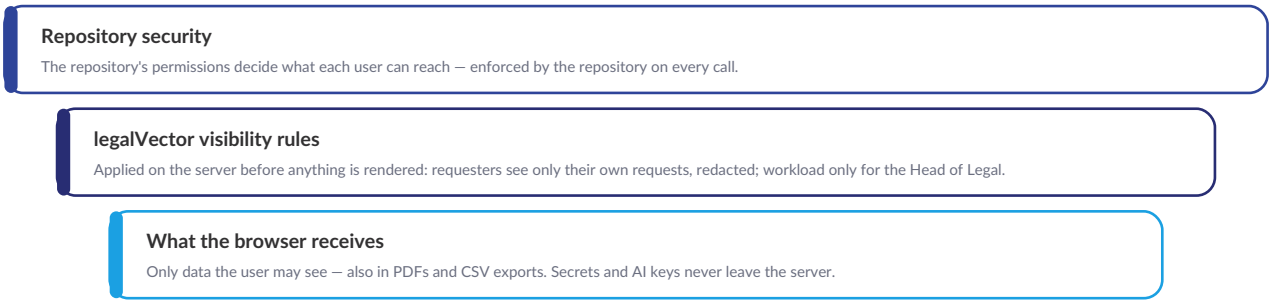


Figure 35: Repository permissions first; legalVector's Legal visibility rules second; the browser receives only what the user may see.

Audience	Sees
Head of Legal	Every request and every lawyer's workload and turnaround
Counsel	Every request in full; workload figures only about themselves
Requester	Only their own requests — without AI reviews, internal notes or priority
Anyone else	"Access denied" — nothing about the request is shown

The same rules apply to the request PDF report and to CSV exports. Role checks run in the server actions and again in the domain layer; the workflow engine completes its task before the repository is written, so a refused step never leaves a half-done change.

SETTING UP LEGAL

Set up in minutes, on your repository

“Set up Legal” creates what is missing of the Legal model, after a preview of every step in plain names and the administrator’s password: where requests and Legal’s library are kept, the fields they carry, their permissions and the Legal group. The Legal process is deployed to the workflow engine.

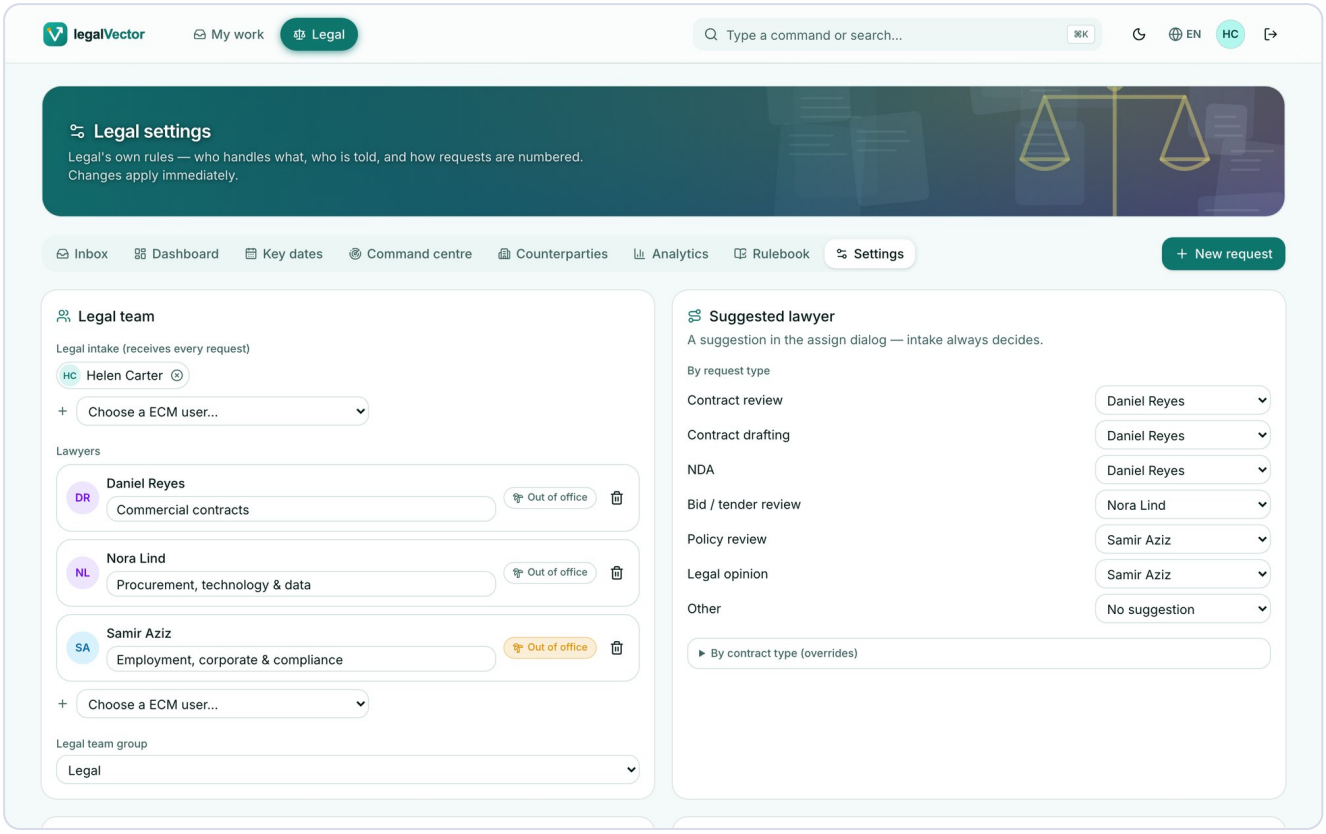


Figure 36: Legal settings: the team, suggested lawyers per request and contract type, departments and contacts, contract types, numbering and AI policy.

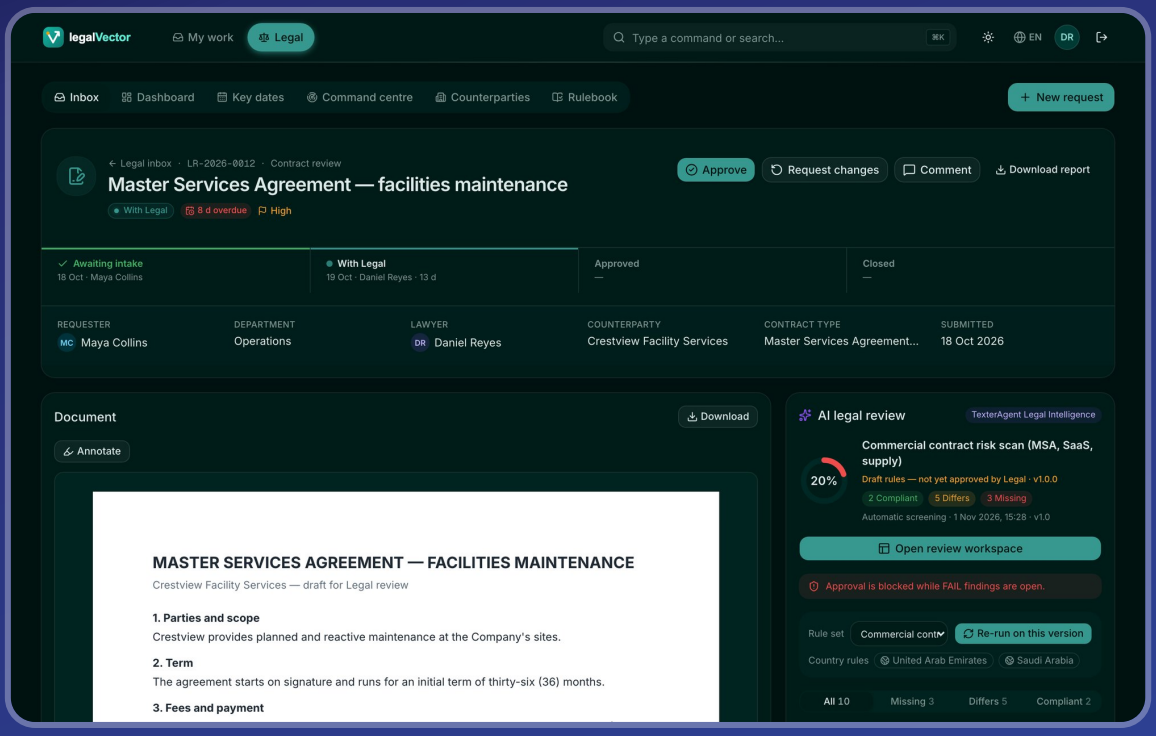
- ✓ **Only what is missing is created** — run it again safely after an upgrade.
- ✓ **Rights included:** every request is closed to all but its requester and the Legal team the moment it is filed.
- ✓ **Configurable without code:** organisation, standard governing law and forum, numbering prefixes, turnaround, routing, departments and contacts, contract types named in every interface language, auto-screening and the approval gate.
- ✓ **Contact health badges** flag disabled accounts, missing e-mails and empty groups before a notification silently fails.

02

PART 2 · WORKFLOW, PLATFORM AND TRUST

Built on official repository APIs

One workflow interface, the user's own identity on every call, and a design that works for every person on every screen.



WORKFLOW

One Legal process, on the engine that fits

legalVector talks to workflow through one interface. The Legal process ships as standard BPMN on an **embedded engine** — or runs on the repository's own BPM.

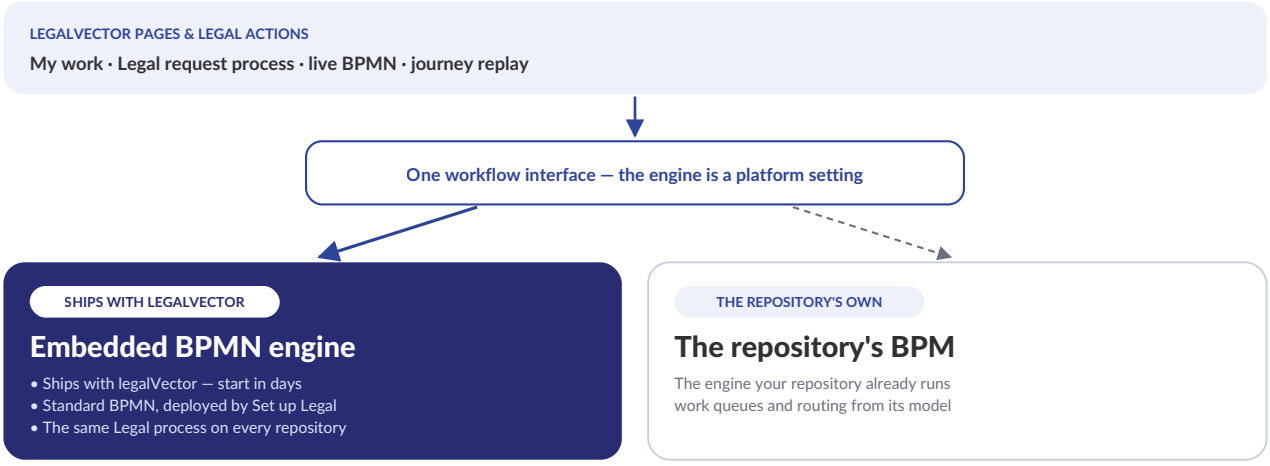


Figure 37: One interface: the embedded BPMN engine, or the repository's own BPM.

The embedded engine lets a team start working in days, and is deployed by "Set up Legal". Because pages and Legal actions only talk to the common interface, moving to the repository's own engine changes no screen and no user habit, and a running request stays on the engine where it started. Identities are the repository's own: assignees are login names, candidate groups are repository groups.

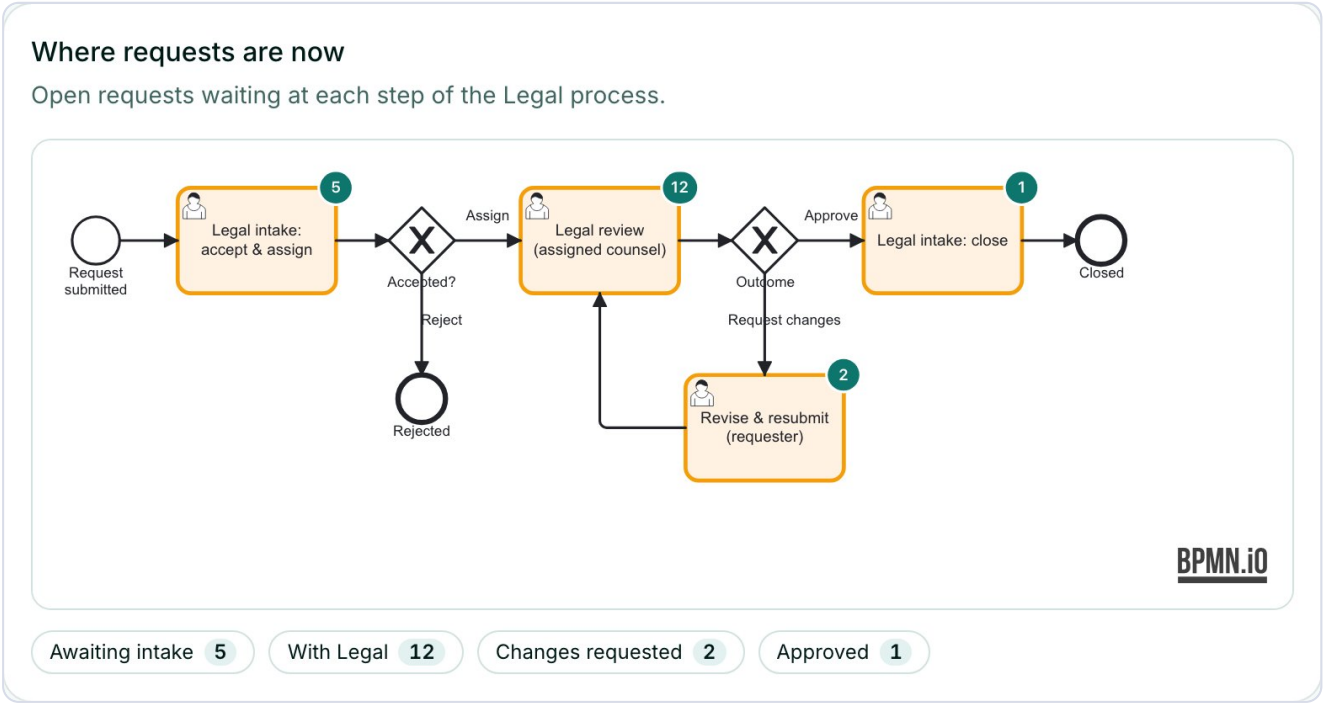


Figure 38: The Legal process as BPMN, with live counts per step — the same view on every engine.

BUILT ON OFFICIAL APIS

Every capability, through the repository's own interfaces

legalVector uses only the repository's official interfaces: its public REST API for all user work, and the platform's own tools for the few schema changes the Legal model needs.

The user's own identity

Every call runs with the signed-in user's own repository session. Sign-out logs the repository session out.

Load discipline

A cap on requests in flight keeps legalVector gentle on a shared system.

Safe writes

Fields re-sent or merged on every new version, so a document never loses its metadata.

Verified behaviour

Built against each supported repository's API, with a faithful simulator of each for the same automated end-to-end tests.

DESIGNED FOR EVERYONE

The same calm experience on every screen, in every language



Figure 39: From a quick approval on a phone to a team dashboard on the wall.

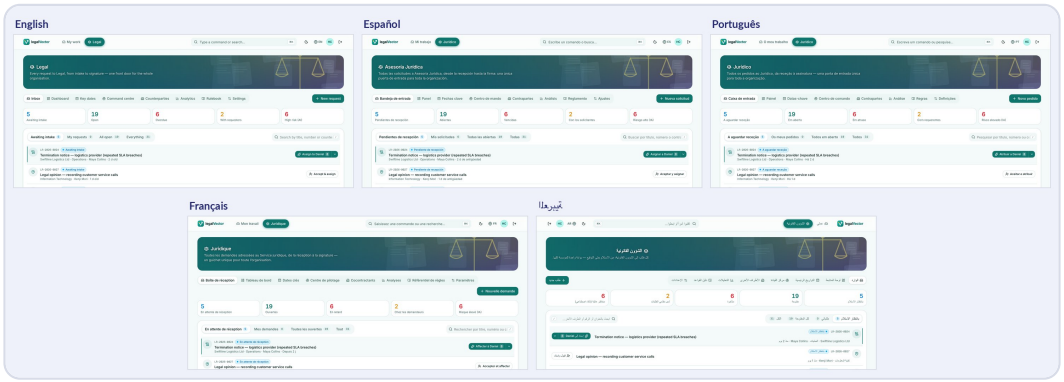


Figure 40: The same Legal inbox in English, Spanish, Portuguese, French and Arabic — every label, date and number follows the user's language.

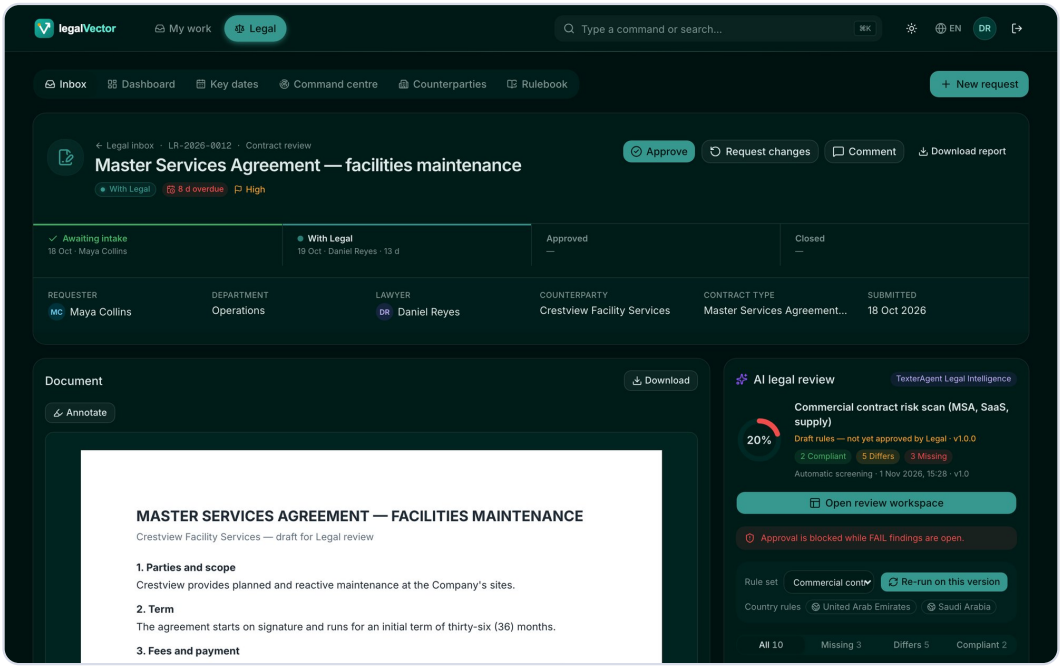
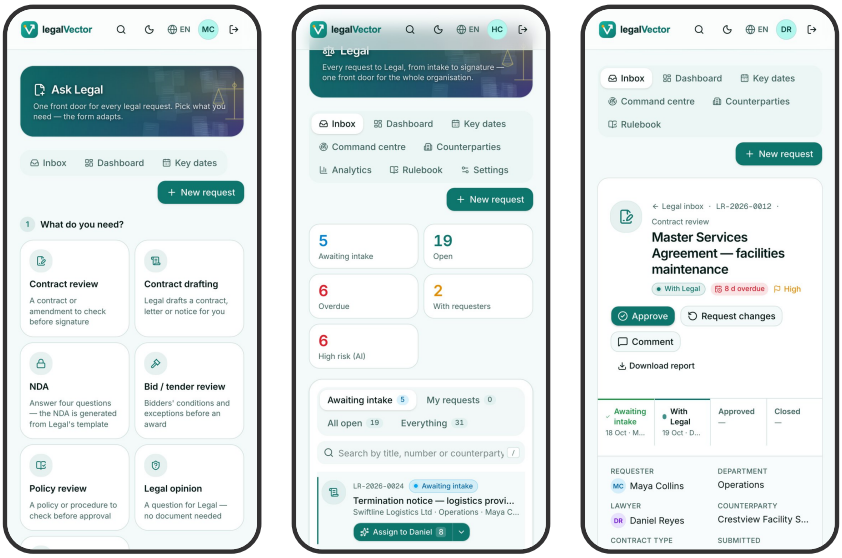


Figure 41: A request in dark mode.



On a phone: Ask Legal, the Legal inbox and a request.

Five languages

English, Spanish, Portuguese, French and Arabic — process labels, settings, dates and numbers included; right-to-left for Arabic.

Light and dark

Comfortable in a bright office or on a night shift.

Keyboard-first

Shortcuts and ⌘K for people who live in their documents.

Your brand

Logo and colours per customer — with the repository clearly running underneath.

TRUST BY DESIGN

A good citizen of your content landscape

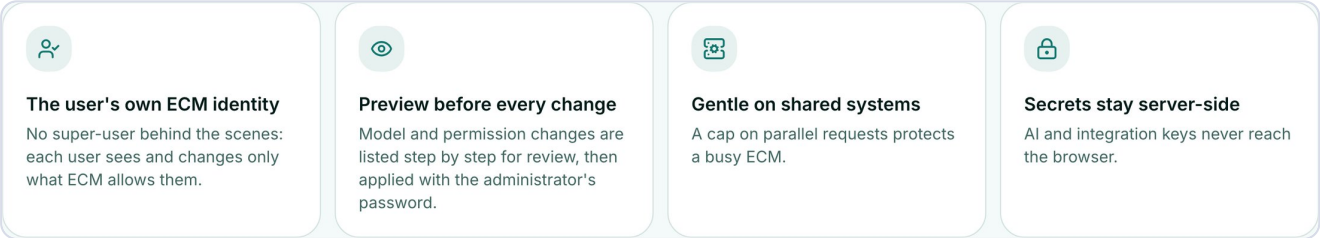


Figure 42: Four commitments that hold everywhere in legalVector.

Area	How legalVector handles it
Identity	No super-user: each user sees and changes only what the repository allows.
Server-side rules	Legal visibility and redaction applied before anything reaches the browser.
Secrets	AI and integration keys stored server-side, write-only in the UI.
Passwords	Never stored by legalVector: each sign-in goes to the repository; administrative tools refuse empty passwords.
Audit	Every Legal event mirrored as a repository comment; e-mail delivery recorded per recipient.
Shared systems	Request cap, bounded technical sessions; demo data disabled on real systems unless explicitly enabled.

CONFIGURATION & ADOPTION

Fits your organisation — without code

Area	What can be configured
Repository	One setting; Legal settings, timelines and numbering kept per repository system
Branding	Product name, logo, colours (light and dark palette derived automatically), sign-in panel, section imagery, “runs on” the repository
Workflow engine	The embedded engine, or the repository’s own BPM
TexterAgent	Connection, AI search, Ask, language model
Value lists	Choices for text descriptors, enforced on every form
Legal	Organisation, governing law and forum, numbering, team, routing, departments and contacts, contract types (named in every language), AI auto-screening and approval gate, command-centre sites
AI routing & rulebook	Which rule set reviews which request or contract type; owners, second approval, versions
Notifications	SMTP e-mail with the requester copied; Microsoft Teams channel; calendar export; Forward-to-Legal mailbox

A practical adoption path

1 · Connect

Point legalVector at your repository; users sign in with their own accounts.

2 · Set up Legal

Preview and apply the Legal model; configure team, routing and contract types.

3 · Pilot

Run the Legal desk with real requests from one or two departments, on the embedded engine.

4 · Production

Approve the rule sets and enable AI screening — optionally move to the repository’s own engine.

5 · Roll out

Open the front door to every department, each in its own language.

Start small, grow on the repository you own. Every step builds on the content landscape the organisation already has — no data migration, no second repository, nothing to unwind.

APPENDIX A

User story catalogue — 77 stories

Every functionality of legalVector expressed as a user story, grouped by persona. Each story is implemented in the November 2026 build.

#	Story
BUSINESS REQUESTER (ANY EMPLOYEE) · 15	
1	As a requester, I want to pick the kind of legal request first (contract review, drafting, NDA, tender, policy, opinion, other), so that the form only asks what matters for it.
2	As a requester, I want to drop my contract on the form and have the title, counterparty, contract type and "what Legal should know" filled in from it, with each value marked "from the document" and an Undo, so that I don't retype what the document already says.
3	As a requester, I want to dictate the title or description in my own language, so that I can raise a request from my phone.
4	As a requester asking for an NDA, I want to answer four questions (counterparty, mutual or one-way, purpose, term), so that Legal's approved template generates the NDA with its own document ID at once.
5	As a requester, I want to confirm which departments reviewed the document, so that each department's contact is told it went to Legal.
6	As a requester without a document (opinion, drafting), I want legalVector to create a request brief, so that Legal still gets a proper Word document.
7	As a requester, I want to see "what happens next" before I send, so that I know intake assigns a lawyer and I'll be copied on every email.
8	As a requester, I want to see only my own requests, their status, the lawyer and the due date, so that I can follow them without chasing Legal.
9	As a requester, I want to hover a status to see who held my request and for how long, so that I know where it is waiting.
10	As a requester, I want Legal's notes anchored to the exact words in my document, so that I know precisely what to change.
11	As a requester, I want to upload a revised version and resubmit, so that Legal reviews the corrected document (and the AI screens it again).
12	As a requester, I want to withdraw a request with a reason, so that Legal stops work I no longer need.
13	As a requester, I want to comment on my request, so that everyone on it is emailed.
14	As a requester, I want to forward an email with the contract to the Legal mailbox and then confirm the prefilled draft under my own login, so that raising a request takes one forward and one click.
15	As a requester, I want to download a PDF report of my request, so that I have the record (without Legal's internal work).

#	Story
HEAD OF LEGAL (LEGAL INTAKE) · 19	
16	As Head of Legal, I want an "Awaiting intake" queue in first-in-first-out order, so that nothing waits unfairly.
17	As Head of Legal, I want one-click "Assign to ⟨suggested lawyer⟩", with each lawyer's live workload and out-of-office status, so that triage takes seconds.
18	As Head of Legal, I want to set priority and due date when I accept, with a default turnaround, so that every request has a deadline.
19	As Head of Legal, I want to reject a request with a reason the requester receives, so that out-of-scope work is closed politely.
20	As Head of Legal, I want to reassign a request, so that the work moves when a lawyer is overloaded or away.
21	As Head of Legal, I want to close approved requests, so that the requester is told the approved document is in the repository.
22	As Head of Legal, I want a daily brief (overdue, waiting, busiest lawyer, lawyer with room, notice deadlines) where each line opens its list, so that I know the day in seconds.
23	As Head of Legal, I want a dashboard with open, overdue, median days to close, median hours to assign and first-pass approval, where every figure opens its list, so that I can trust the numbers.
24	As Head of Legal, I want to see every lawyer's workload and turnaround, so that I can balance the team.
25	As Head of Legal, I want a board pack PDF where every figure links to its list, downloadable or emailed to me, so that I can report to the board.
26	As Head of Legal, I want analytics (department × type heatmap, open requests at each BPMN step, time each party holds a request), so that I can see where requests come from and where they wait.
27	As Head of Legal, I want a portfolio exposure heatmap (clauses × contracts) whose counts open the documents, so that I know which clauses we concede most.
28	As Head of Legal, I want to approve or send back rule changes, so that only rules Legal stands behind judge contracts.
29	As Head of Legal, I want to set owners and an optional second approval per rule set, so that rule governance is explicit.
30	As Head of Legal, I want to configure the team, routing suggestions, departments and contacts, contract types, numbering, standard governing law and forum, and the AI policy, so that the module fits my organisation without code.
31	As Head of Legal, I want contact health badges (disabled account, no email, empty group), so that notifications don't silently fail.
32	As Head of Legal, I want a command centre on a wall display (site map, live feed, SLA watch, team load, kiosk tour), so that the team sees the function's state at a glance.
33	As Head of Legal, I want to ask in ⌘K "NDAs that end this quarter" (in any of the five interface languages) and open exactly that list, so that I get answers without building filters.
34	As Head of Legal, I want CSV exports of the requests I can see, so that I can work in Excel.

#	Story
COUNSEL / LAWYER · 23	
35	As a lawyer, I want an "Assigned to me" queue ordered by due date, with keyboard navigation (j/k/Enter), so that I work in deadline order.
36	As a lawyer, I want the Word document rendered in place with its versions, so that I don't download files to read them.
37	As a lawyer, I want an AI review of every clause against Legal's rule set and country overlays, showing findings as they arrive, so that I start from a structured first pass.
38	As a lawyer, I want each finding to show the current text, how it differs, red flags, the proposed wording and sources, so that I can decide quickly.
39	As a lawyer, I want "Locate" to highlight the exact words in the document, so that I see the clause in context.
40	As a lawyer, I want to stage "Apply change" and "Insert clause" (editing the wording and choosing the position), and see them live as tracked changes, so that I control the redline.
41	As a lawyer, I want "Save as new version" to write a real .docx with genuine Word tracked changes in my name (AI-assisted), so that the counterparty receives a normal redline.
42	As a lawyer, I want an automatic re-analysis after saving ("resolved · new · still open"), so that I see the effect of my changes.
43	As a lawyer, I want to accept a deviation with a recorded reason, carried over to later versions, so that a business-accepted deviation stops blocking approval.
44	As a lawyer, I want approval blocked while must-fix clauses are open, with an override that requires a written reason, so that risk acceptance is explicit and audited.
45	As a lawyer, I want to annotate words in the document and send them back with "Request changes", so that the requester gets precise instructions.
46	As a lawyer, I want to compare any two versions word by word (only changes, n/p navigation), so that I see what the counterparty changed.
47	As a lawyer, I want a negotiation timeline (round by round: what Legal asked, score, clauses moved, accepted deviations), so that I can explain the negotiation.
48	As a lawyer, I want a local law check against the jurisdiction pack in the Legal library, citing its numbered facts and as-of date, so that I see local-law points without relying on the model's memory.
49	As a lawyer, I want a side-by-side view of the contract with a reading translation into any interface language, clause by clause with review colours, so that stakeholders can read it in their language (not the signed text).
50	As a lawyer, I want contract key dates (start, end, notice deadline, renewal, value) read from the contract and saved on the document in the repository, so that they stay with the contract and the repository's own tools can act on them.
51	As a lawyer, I want a key-dates calendar with "needs action" buckets and an .ics export, so that no notice deadline is missed.
52	As a lawyer, I want Counterparty 360 (all requests, AI score over time, recurring deviations, dates, people), so that I know the other party's track record.

#	Story
53	As a lawyer who owns a rule set, I want to describe a rule change in plain language, see the exact edits, warnings and impact on recent documents, and publish it (or submit it for second approval), so that the playbook evolves from real reviews.
54	As a rule owner, I want a rule change to also edit the NDA Word template, written as a new version in the repository on approval, so that every new NDA uses the agreed wording.
55	As a lawyer, I want the rulebook time machine, so that I can explain any past review by the rules in force that day.
56	As a lawyer, I want my own brief and my own workload figures, so that I can plan my day.
57	As a lawyer, I want to replay a request's journey on the BPMN diagram, so that I can show how it progressed.
LEGAL OPERATIONS / LEGAL ADMIN · 7	
58	As a Legal administrator, I want "Set up Legal" to preview every change to the repository (in names) and then apply it with my password, creating only what is missing, so that setting up is safe and repeatable.
59	As a Legal administrator, I want to load a realistic demo caseload replayed through the real workflow by demo personas, so that I can demonstrate the module.
60	As a Legal administrator, I want to route each request or contract type to a rule set, so that the right playbook reviews each document.
61	As a Legal administrator, I want to create or edit rule sets (including country overlays with keywords, sources and as-of date) as new versions with a reason, so that rules are maintained. Admin edits stay drafts until the Head of Legal approves.
62	As a Legal administrator, I want to import clauses from a Word template as proposed rules, so that a playbook starts from our own standard.
63	As a Legal administrator, I want to see recent AI reviews, median score and the engine connection, so that I can monitor the AI service.
64	As a Legal administrator, I want to set the command centre's sites and keywords, so that the map reflects our footprint.
IT / SECURITY · 11	
65	As IT security, I want every repository call to run with the signed-in user's own repository session (no super-user), so that the repository's permissions always apply.
66	As IT security, I want Legal's visibility rules enforced on the server before data reaches the browser (requesters see only their own requests, redacted), so that hiding is not cosmetic.
67	As IT, I want legalVector to cap its parallel repository requests, so that it cannot overload a shared system.
68	As IT, I want sign-out to log the repository session out, so that repository sessions are released.
69	As IT security, I want schema changes made only by admins, after a password confirmation, through a Java tool that refuses empty passwords and passes the password only through the process environment, so that failed logins don't lock accounts and passwords aren't written to disk.

#	Story
70	As IT security, I want API keys (TexterAgent) stored server-side only and never sent to the browser, so that secrets don't leak.
71	As IT, I want new passwords RSA-encrypted with the CSB's public key, so that they are never sent in plain text.
72	As IT, I want the workflow engine bound to localhost with authorization and basic auth on, so that it is not exposed.
73	As IT, I want every Legal event mirrored as a comment in the repository and the timeline to record each email's delivery result, so that the audit lives in the repository.
74	As IT, I want demo data blocked on real systems unless explicitly enabled per environment, so that test users and documents are never created by accident.
75	As a developer or tester, I want a simulator of each supported repository that reproduces its behaviour, so that features can be built and tested without a shared system.
PLATFORM OWNER · 2	
76	As a platform owner, I want legalVector to run on the repository we already run, chosen by one setting, so that Legal gets its desk where the contracts already are.
77	As a platform owner, I want the same end-to-end tests to pass on both repositories, so that equivalent functionality is proven rather than assumed.

APPENDIX B

Functional map

Screen	Purpose
/inbox	My work
/about	About legalVector (public; overlay from the logo)
/legal	Legal inbox
/legal/new	Ask Legal — new request
/legal/requests/…	Request page
/legal/requests/…/review	Review workspace
/legal/dashboard	Dashboard, daily brief, contracts view, board pack
/legal/calendar	Key dates & obligations
/legal/command-centre	Command centre (wall display)
/legal/counterparties	Counterparty 360
/legal/analytics	Analytics and portfolio exposure
/legal/rulebook	Rulebook, change the rule, time machine
/legal/settings	Legal settings and set-up
/admin/workflows	Workflow engine and processes
/admin/integrations	TexterAgent
/admin/document-intelligence	AI review engine, rule sets, routing

APPENDIX C

Glossary

Term	Meaning
Content repository	The system that stores, secures, versions and retains an organisation's documents; legalVector runs on it.
TexterAgent	TexterBlue's AI engine for review, extraction and answers.
Rule set (skill)	A versioned playbook of clauses that TexterAgent reviews against.
Country overlay	Jurisdiction-specific rules with cited facts, added when a request mentions a country.
Approval gate	The rule that approval waits while must-fix findings are open.
Redline Engine	legalVector's writer of genuine Word tracked changes.

Multi-repository.
One powerful Legal desk.

Give Legal one desk on the platform the organisation already trusts.

TEXTER BLUE

TexterBlue

www.texter.ai

legalVector · the Legal desk on your repository · TexterAgent