necodoc

ARCHITECTURE REVIEW PACK

Self-hosted document generation for insurers

How necodoc is built and deployed, where data goes, which controls it provides and where each one stops. For architecture and security review boards.

VERSION	0.1.0
COMMIT	`addc7c0fda`
GENERATED	2026-10-12
BENCHMARK RUN	2026-10-12
LICENCE	MIT

This PDF was rendered by necodoc's own renderer from the repository at the commit above, as PDF/A-2u and tagged PDF/UA-1.

Contents

01 About this document 3

02 Product summary 3

03 Reference architecture 4

04 Components and state 5

05 Deployment options 5

06 Integration patterns 6

07 Data flows 6

08 Identity and access 7

09 Security controls 9

10 Audit and observability 10

11 Data protection 11

12 Compliance: what helps and what stays yours 11

13 Performance 12

14 Known limits 14

15 Support and licence 14

Each section ends with links to the specifications it summarises, at commit `addc7c0fda`. The specs carry intent and the full list of known limitations.

01

About this document

This pack describes necodoc as of the release on the cover, for an architecture or security review. It states what the software does, where data goes, which controls exist and where each one stops.

- **Built from the repository.** The controls table and the performance figures are read from the same sources as the public site, and this PDF was rendered by necodoc's own renderer.
- **Every section links to its specification** at the commit on the cover. The specs record intent and known limitations in more detail than fits here.
- **No attestations.** necodoc is software you run. There is no hosted service, so there is no SOC 2 report or ISO certificate to attach. It runs inside your boundary and inherits your controls.

02

Product summary

necodoc is self-hosted document generation. Your systems post data and get a PDF back; staff fill a guided form and get the same PDF. It produces documents three ways:

- **Designed templates and HTML**, rendered by headless Chromium, with headers, footers, page numbers, conditional sections and repeating groups.
- **Overlays on existing PDFs**, such as an obligee's bond form or a regulator's form: fields are placed on the original pages, so the output looks exactly like it. When AI is enabled, it proposes field positions and an editor accepts each one.
- **Packets** that combine generated and uploaded PDFs into one file.

Templates have typed fields that drive the form, API validation and a JSON Schema per published version. Published versions are immutable; review and publishing are separate permissions. Output can be PDF/A-2b or 2u, tagged PDF/UA-1, and certified with your certificate. One template produces English and fr-CA documents.

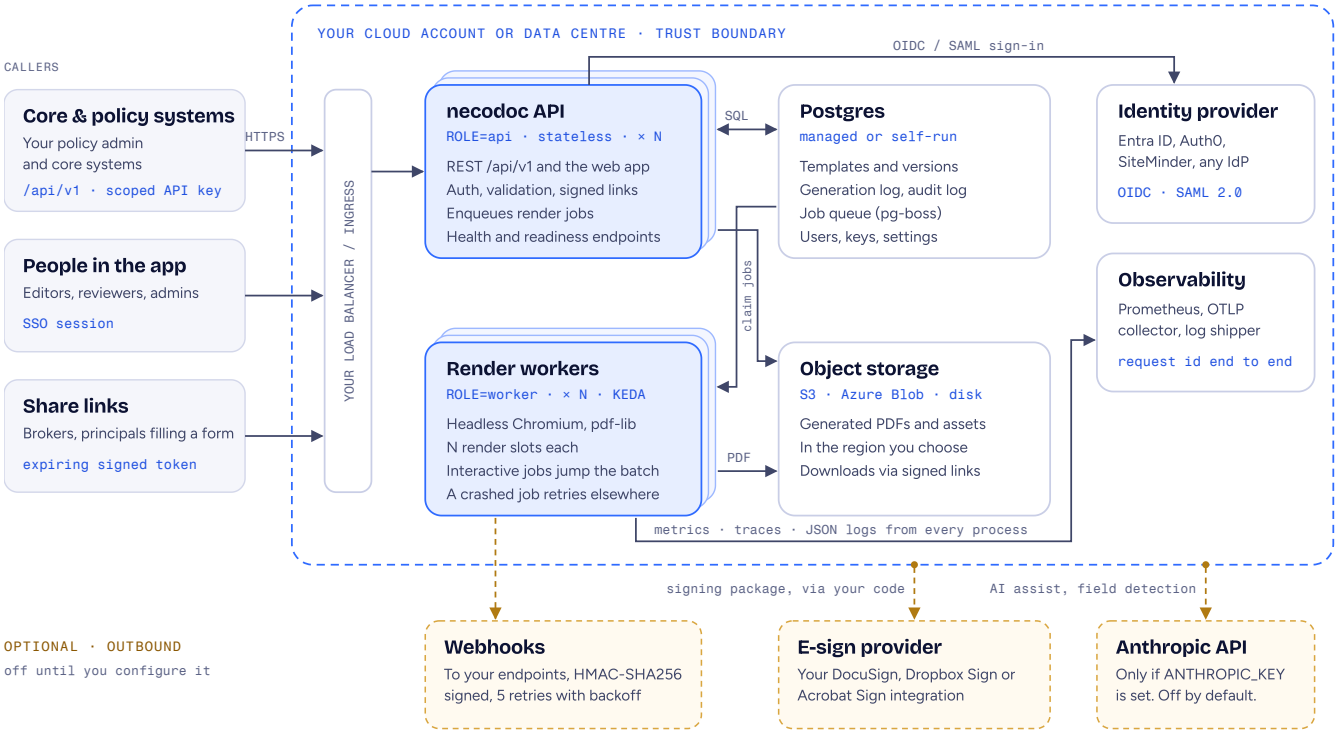
It runs as containers next to one Postgres database, in your cloud account or data centre.

SPECS Self-hosting · Competitive scope

03

Reference architecture

Everything inside the dashed boundary runs in your environment. The items outside it are optional and outbound only.



- **Callers:** core and policy systems over `/api/v1` with API keys; staff in the browser app through your identity provider; recipients of public share links.
- **API and render workers** are stateless and scale by replica count. Each worker runs headless Chromium with JavaScript off and no network access.
- **State** lives in your Postgres and your object storage only.
- **Outbound, optional:** webhooks to endpoints you configure, your own e-signature integration fed by signing packages, and Anthropic's API only if an admin enables AI.

SPECS Render queue, batch jobs and scaling · ADR 0012

04

Components and state

Only Postgres and object storage hold state. Every necodoc process can be killed and replaced.

COMPONENT	DOES	SCALES BY	STATE
API (`ROLE=api`)	Serves `/api/v1` and the web app's API; renders live previews; authenticates keys and sessions; validates data against the template's fields; queues renders; issues signed download links	Replicas behind your load balancer	None
Render worker (`ROLE=worker`)	Claims render jobs; runs Chromium for designed and HTML documents; stamps overlays; applies PDF/A, PDF/UA and certification; writes the PDF	Replicas, each with `RENDER_CONCURRENCY` slots; KEDA on queue depth	None; a job on a lost worker retries
Postgres	Users, templates and versions, themes, assets and uploaded PDFs, the generation log, the audit log, settings and the job queue (pg-boss)	Your database tier; this is the throughput ceiling	All durable records
Object storage	Generated PDFs: an S3-compatible bucket, an Azure Blob container, or a shared disk	Your storage service	PDF bytes
Identity provider	Sign-in for people over OIDC, SAML 2.0 or a trusted header; groups map to roles	Yours	Yours
Single process (`ROLE=all`)	API and worker in one process, for evaluation and small installs	Not meant to	None

Back up Postgres, then storage, plus three secrets: `BETTER_AUTH_SECRET`, `DOWNLOAD_SIGNING_SECRET` and the storage credentials.

SPECS ADR 0014 · ADR 0020 · ADR 0023

05

Deployment options

Every option uses the same images and the same environment variables. Each process validates its configuration at start. A bad value stops the process with an error that names the variable. necodoc doesn't call home, so it runs without outbound internet access.

OPTION	FOR	WHAT YOU GET
Docker images	Any container platform	An API and worker image (Node, with Chromium) and nginx images for the app and docs, built from the repository's Dockerfile
Docker Compose, one host	Evaluation and small installs	Postgres, migrations, API, workers and the app from one `.env`; `--scale worker=N`; an optional monitoring profile with Prometheus and Jaeger

OPTION	FOR	WHAT YOU GET
Helm on Kubernetes	Production	Separate API and worker Deployments behind one Ingress, migrations as a pre-install hook, disruption budgets, graceful drain on scale-down, Prometheus annotations or a `PodMonitor`; bring your own Postgres
KEDA autoscaling	Peaks such as renewal runs	Worker replicas follow queue depth read from Postgres; without KEDA, a CPU autoscaler works

Every process has health and readiness endpoints. Readiness fails if the database, queue or storage is unavailable. To upgrade, deploy the new image tag, run the migrations, then roll the pods.

> LIMIT The Helm chart passes `helm lint`, `helm template` and kubeconform but has not yet been run on a live cluster, and OpenShift isn't tested. There is no published image registry: you build the images from source.

SPECS Self-hosting · ADR 0025

06

Integration patterns

There is no packaged connector for any core system. Systems integrate over a versioned REST API described by OpenAPI 3.1, with a JSON Schema per published template version and generated SDKs for TypeScript, Python, .NET and Java.

PATTERN	USE	HOW
Synchronous	A certificate issued while someone waits	`POST …/generate` returns the PDF, or JSON with a signed `downloadUrl`; `X-Generation-Id` names the log entry
Asynchronous	Bind and issue events	`async: true` returns 202 with an id; an HMAC-signed webhook brings the result
Batch	Renewal runs	Up to 10,000 rows of JSON or CSV per batch; progress, cancel, then one ZIP or one combined packet

Interactive renders and synchronous calls go ahead of batch rows. Errors are JSON, with a code and per-field errors: 401 bad key, 403 scope or IP, 422 invalid data, 429 rate limit with `Retry-After`. Every response carries `X-Request-Id`.

SPECS HTML to PDF API · Webhooks · ADR 0030 · ADR 0041

07

Data flows

DATA	STORED IN	LEAVES YOUR BOUNDARY
Templates, versions, snippets, themes, assets	Your Postgres	No

DATA	STORED IN	LEAVES YOUR BOUNDARY
Input data for each PDF	Your Postgres (generation log), for a retention period you set	No
Generated PDFs	Your bucket, blob container or disk, in your region	Only through signed, expiring links you issue
Logs, metrics, traces	Your log shipper, Prometheus and OTLP collector; field values are never logged	No
Webhook payloads	Sent to endpoints you configure	Yes, to your URLs, HMAC-SHA256 signed
Signing packages	Your Postgres; fetched by your e-signature integration	Only if your code sends them on
Draft templates, form page images	Sent to Anthropic's API for the template assistant and AI field detection	Only if `ANTHROPIC_KEY` is set

> **AI is the one thing that can leave your environment, and it's off by default.** If an admin sets `ANTHROPIC_KEY`, the template assistant sends the draft template (fields, content, sample data) and AI field detection sends page images of an uploaded form to Anthropic's API. Generation data, the policyholder data that fills real documents, is not sent. The audit log records each AI request, but necodoc doesn't store prompts or replies. Without the key, the app shows no AI controls and the AI routes return 404.

SPECS AI template assistant · PDF overlay templates

08

Identity and access

People sign in through your identity provider, with email and password, or both.

- OIDC and SAML 2.0 providers such as Entra ID, Auth0 or SiteMinder, plus a trusted-header mode for SiteMinder-style web agents that accepts the header only from proxy addresses you list.
- necodoc creates the account at first sign-in. IdP groups map to roles (admin, publisher, reviewer, editor, viewer), and necodoc syncs them again at every sign-in. A user in no mapped group gets no access until an admin grants it.
- An IdP can't take over an existing account with the same email unless the operator allows linking. Password sign-in can be turned off.
- Admins ban users, revoke sessions and reset passwords. Impersonation is admin-only and audited.

Systems call `/api/v1` with API keys.

- A key belongs to a person or to a service account. Service accounts can never sign in.
- Each key has scopes no wider than its owner's roles, an optional template allowlist, an optional IP allowlist, an optional expiry date and a rate limit, and shows its usage. To change a key, revoke or rotate it. Keys can't be edited.

> LIMIT No SAML single logout, no IdP-initiated sign-in, no signed AuthnRequests or encrypted assertions. Providers are configured by environment variables and a restart. The IP allowlist sees the reverse proxy's address unless `TRUST_PROXY` is set.

SPECS Single sign-on · User management and roles · ADR 0029

09

Security controls

The same list as the controls table on the site's security page, which is also offered there as a CSV for questionnaires. Each row links to the specification that defines the control and its known limits.

CONTROL	HOW NECODOC DOES IT	PROOF
Identity and access		
Single sign-on	Entra ID, Auth0, SiteMinder or any OIDC or SAML 2.0 IdP, or a trusted SiteMinder-style header. necodoc creates the account at first sign-in.	Single sign-on
Role mapping	IdP groups map to roles: admin, publisher, reviewer, editor, viewer. A user can hold several roles, and each role grants one kind of work.	Users and roles
Session control	Admins ban users, revoke sessions and reset passwords. Impersonation is admin-only and audited from start to stop.	Users and roles
System access		
Service accounts	Keys belong to a person or to a service account that can never sign in.	ADR 0029
Scoped keys	Each key has scopes no wider than its owner's roles, an optional template allowlist, IP allowlist and expiry date, and a rate limit. Admins can revoke a key and see its usage.	HTML to PDF API
Signed callbacks	Webhooks carry an HMAC-SHA256 signature over the timestamp and body. necodoc logs each delivery attempt.	Webhooks
Data		
Residency	Records in your Postgres; PDFs in your S3-compatible bucket, Azure Blob container or disk, in the region you choose.	ADR 0023
Downloads	Files download through signed links that expire. The audit log records each link that is issued.	Generation log
Retention	An admin sets how long generated PDFs, and optionally their input data, are kept.	Generation log
Audit and logging		
Audit log	Every configuration and access change, sign-ins including failures, impersonation and API key use, with before and after diffs. A database trigger keeps it append-only. Exports to CSV.	Audit log
Generation log	Every PDF with its input, output, template version, caller and a timing breakdown; re-runnable against the exact version.	Generation log
Operational logs	JSON logs, Prometheus metrics and OpenTelemetry traces, with one request id end to end. Logs and traces never carry field values.	Observability
Change control and output		

CONTROL	HOW NECODOC DOES IT	PROOF
Versioning	Each template has one working draft. Published versions are numbered and can't change. Publishing needs its own permission.	Publishing
Review gate	When an admin requires it, only an approved, unchanged draft can be published.	Review
Regression tests	Named test cases with assertions, and a page-by-page visual and text diff against the published version.	Test cases
Output integrity	PDF/A-2b or 2u, PDF/UA-1, and PAdES B-B or B-T certification with your certificate. Each generation records the profiles it applied.	Archival output
Privacy and vendor risk		
Sensitive fields	Fields marked sensitive are masked in audit diffs and in other people's log entries.	Personal data
Erasure	An admin finds everything stored about a data subject and erases it. necodoc writes an erasure report.	Personal data
Exit	MIT source, standard Postgres, PDFs in your storage, templates as data. No vendor holds your data.	Licence

10

Audit and observability

Audit log. Every configuration and access change, sign-ins including failures, impersonation and API key use, with before and after diffs, the actor, IP address and request id. A database trigger keeps the rows append-only. The log exports to CSV.

Generation log. Every PDF with its source, caller, template version, input, output, output profile, worker, attempts and a timing breakdown (queue wait, Liquid, Chromium, bands, stamping, finishing). Any entry can be re-run against the exact version it used.

Operations. JSON logs on stdout with one request id from the API call through the queue to the worker. Prometheus metrics on a separate port: HTTP, render durations by phase, queue depth and age, worker slots. OpenTelemetry traces with the standard `OTEL_*` variables, off unless configured. Secrets are redacted from logs, and logs and traces never carry field values.

> LIMIT The audit log is tamper-resistant, not tamper-evident: there is no hash chain, and a database administrator can disable the trigger. Ship it to your SIEM if you need proof against your own administrators. Audit entries are kept forever. File downloads aren't audited. Only the creation of a signed link is. The metrics port has no authentication and must not be exposed.

SPECS Audit log · Generation log · Observability · ADR 0032 · ADR 0036

11

Data protection

- **Storage drivers.** Generated PDFs go to an S3-compatible bucket (AWS S3, MinIO, Wasabi, Cloudflare R2 and the like), an Azure Blob container, or a disk volume every process shares. You choose the region; encryption at rest is your storage service's.
- **Downloads** stream through the API, or through signed links that work without a session until they expire (up to 7 days). Minting a link is audited.
- **Sensitive fields.** Fields can be marked sensitive, and signatures and keys that look like a SIN, tax id, passport, birth date, bank account or card number are treated as sensitive automatically. Their values are masked in audit diffs and in other people's generation log entries.
- **Erasure.** Admins search everything stored about a data subject, by a subject key the caller sends or by any stored value, and erase it: inputs, PDFs and stored objects, packets, webhook delivery logs and share-link responses. The erasure report lists what was deleted; the audit entry records counts, never the search term.
- **Retention.** An admin setting deletes outputs, and optionally inputs, older than N days while keeping the log metadata.

> LIMIT Audit entries and logs you've shipped elsewhere can't be erased. Erasure matches exact values only, so callers should send a subject key. Signed links can't be revoked before they expire, except by deleting the entry.

SPECS Personal data masking and erasure · ADR 0023 · ADR 0039

12

Compliance: what helps and what stays yours

necodoc doesn't make a carrier compliant with anything. These are the features that help with each requirement, and what remains the carrier's work.

REQUIREMENT	HOW NECODOC HELPS	STILL YOURS
OSFI B-10, third-party risk	Self-hosted MIT software: audit and access rights are yours, and the exit plan is your own Postgres and storage	Your third-party assessment and exit testing
OSFI B-13, technology and cyber risk	Runs inside your controls: your IdP, network, keys, logging and patch cycle	Operating and patching it
Bill 96, French first in Québec	English and fr-CA documents from one template, with the same fields and layout; translations are reviewed with each version	Translation quality and legal review; an untranslated string prints in the source language
Law 25 and PIPEDA, privacy	You choose the region; sensitive-field masking; erasure per data subject with a report	Privacy assessments, retention policy
AODA, accessible formats	Tagged PDF/UA-1 output from designed and HTML documents	Accessible content and alt text in templates

REQUIREMENT	HOW NECODOC HELPS	STILL YOURS
Records retention	PDF/A-2b or 2u output; retention settings for outputs and inputs	Your retention schedule

What we don't claim

- No SOC 2, ISO 27001 or "certified secure": we describe controls, not attestations.
- Not a "tamper-proof" audit log: it's tamper-resistant, as stated above.
- Not an e-signature service, and no ESIGN, eIDAS or UECA claims. Signing packages hand signers, routing order and box positions to your provider, which runs the signing.
- Not "data never leaves" without the AI caveat in Data flows.
- No packaged core-system connectors, prebuilt bureau form libraries, Word output or multi-step approval chains.

SPECS Archival, accessible and certified output · Document localization · Signing packages

---

13

Performance

Measured by `pnpm bench` through the public API, from the run finished 2026-10-12 at commit `b359f1296d`. Latency is the client's round trip, one request at a time. Rerun it on your own hardware with the same command.

15 ms
p50, one-page HTML to PDF

28 ms
p50, overlay on an existing PDF form

5,950
docs/min, 1 worker, batch

8,786
docs/min, 4 workers on one machine

Latency

SCENARIO	PAGES	P50	P95	WORKER RUN P50
HTML to PDF, 1 page	1	15 ms	18 ms	9 ms
HTML to PDF, invoice with header/footer	3	30 ms	35 ms	23 ms
Designed template, surety bond	2	21 ms	25 ms	14 ms
Overlay, AcroForm bond (Phoenix AZ)	1	28 ms	32 ms	21 ms
Overlay, flat bond (Rockville MD)	3	28 ms	30 ms	21 ms

PDF/A-2b with PDF/UA on the invoice adds 77 ms at p50 (30 ms to 107 ms).

Batch throughput

A JSON batch of 5,000 designed surety bonds, timed from the batch's creation to its last row, so it includes the queue. All workers share one machine.

WORKERS	SLOTS EACH	ROWS	FAILED	WALL TIME	DOCS/MIN
1	2	5,000	0	50.4 s	5,950
2	2	5,000	0	36.9 s	8,125
4	2	5,000	0	34.1 s	8,786

Cold start

MEASURE	N	P50	P95
Worker process start to ready	5	1.33 s	1.63 s
First render on a new worker	5	30 ms	35 ms
Warm render	25	16 ms	17 ms

Environment

ITEM	DETAIL
Machine	Apple M5, 10 cores, 32 GB, macOS 27.0
Software	Node v24.21.0, Chromium 151.0.7922.34, PostgreSQL 18.6 (Homebrew)
Storage	disk (local temp directory)
Settings	100 samples after 10 warm-up requests; RENDER_CONCURRENCY=2; latency on 1 worker
Commit	b359f1296dd1ac60eb6272ea55dd519680d1d92b

CAVEATS Everything ran on one machine: Postgres with default settings, the disk storage driver, calls over loopback, so there's no network, TLS, proxy or object storage latency. The documents are representative insurance documents, not a worst case. Workers sharing one machine's CPU say little about scaling across nodes: past one worker this laptop's Chromium capacity is the ceiling, not the queue.

SPECS Performance benchmark

14

Known limits

This table collects the known limits from the specifications.

AREA	LIMIT
Integration	No packaged core-system connectors; integration is over the REST API. SDKs aren't on package registries yet.
Deployment	No published image registry; the Helm chart hasn't run on a live cluster; OpenShift isn't tested; Helm brings no Postgres.
Regions	One Postgres per install, so one region per install. Run one install per region if residency requires it.
Throughput	Postgres carries the queue as well as the data and is the ceiling. Very high volumes will need a dedicated queue.
Recovery	A killed worker's job is retried after a missed heartbeat, 10 to 20 seconds later. A render timeout is a template error and isn't retried.
Batches	Up to 10,000 rows and 20 MB of input per batch.
Identity	No SAML single logout, IdP-initiated sign-in, signed AuthnRequests or encrypted assertions.
Audit	Tamper-resistant, not tamper-evident; no retention; downloads aren't audited.
Privacy	Erasure matches exact values; audit entries and shipped logs can't be erased.
Output	PDF/A for overlays depends on the uploaded form embedding its fonts. PDF/UA is only as good as the template's content. One certificate per server; no HSM or cloud KMS yet; signatures are PAdES B-B or B-T, not B-LT.
Change control	One review step; no multi-step approval chains.
Observability	The metrics port is unauthenticated; no spans for database queries or outbound calls.

15

Support and licence

Support: coming soon.

Licence: MIT. You can read, fork and modify the source. There is no licence fee per worker, document or seat; the cost is your infrastructure.

Exit: standard Postgres, PDFs in your storage, templates stored as data, and the source you already have. No vendor holds your data.

Source: github.com/mikevalstar/necodoc