AIR-GAPPED · ZERO-TRUST · HARDWARE-TETHERED

IRONGAP TECHNOLOGIES

www.iron-gap.com

Vault-OS
Technical Whitepaper

The architecture of a sovereign, on-premise enterprise AI knowledge vault — the encrypted enclave, hardware-tethered licensing, role-bound inference, multi-party collaboration, and the autonomous workflow & agent engine.

DOCUMENT	VERSION	SUPERSEDES	CLASSIFICATION
Vault-OS Platform	1.0.9.1	Phase 27.1.3	Confidential

CONTENTS

Table of Contents

Every entry below is a live link. In the PDF, click a row to jump to that section; each section header carries a marker back to this page.

What changed since the previous revision

The prior whitepaper documented the platform through phase 27.1.3, whose headline was the workflow engine. The table below is the delta, and each row names the section that covers it.

CHANGE	EFFECT	SECTION
Encrypted enclave	Customer-owned encrypted volume, built in the first-run wizard with their own master password	§04
Native database	PostgreSQL+pgvector bundled and run as a service — the last container dependency removed	§03
Group collaboration	Multi-party messaging with the assistant invited, voice/video notes, one-time media	§10
Speech-to-text	Local multilingual transcription, no remote model fetch	§09
Folder explorer	A department-scoped tree over the encrypted corpus	§14
Client-only access	Browsers are refused; the Vault-Ecosystem client is the sole entry point	§12
Seat & session enforcement	One active session per account; concurrency capped against the licence	§07
Letterless enclave	The volume never receives a drive letter at any point — not while being built, not while running	§04
Verified shutdown	Locking is checked and retried, not assumed; a still-readable enclave is reported rather than hidden	§04
Resumable first run	An interrupted ~22GB unpack continues instead of restarting, verified per file by size	§04
Verified pairing	The appliance publishes its certificate fingerprint on screen and over mDNS; the client asks you to compare before trusting	§12
Continuous startup	Setup, unlock and boot are one page over one socket, released only once the application is confirmed answering	§02
Group interaction model	Reactions, pins, edits, forwards, polls, per-member permissions, slow mode and server-enforced auto-delete	§10
~~Container execution node~~	Removed — running arbitrary containers contradicts the air-gapped design	§11
~~API / webhook gateway~~	Removed — an inbound integration surface contradicts the same	§11

01 Executive Summary

Vault-OS runs large language models, vector retrieval and multi-party collaboration entirely inside a customer-owned encrypted volume, on hardware the licence is cryptographically bound to, with no outbound network path required at any point.

Organisations handling classified, regulated or commercially sensitive material face a structural problem with commercial AI: the value of a model over your own corpus is highest exactly where sending that corpus to a third party is least acceptable. The usual mitigations — private endpoints, zero-retention promises, regional hosting — all still terminate in someone else's process on someone else's silicon. They are contractual controls, not physical ones.

Vault-OS takes the physical position instead. The models, the weights, the vector index, the database, the inference engines and the user interface all ship on the install media and execute on the customer's own machine. There is no telemetry endpoint, no licence phone-home, no model download at first run, and no degraded mode that reaches for the internet when something is missing. A deployment that never touches a network behaves identically to one that does.

0	9	19	33
OUTBOUND CALLS	CLEARANCE TIERS	WORKFLOW NODES	ASSISTANT TOOLS

What separates this from "an LLM running locally"

Everything around the model. An encrypted enclave created with the customer's own master password and mounted only while the product runs. A licence bound to the TPM, system UUID and MAC of one specific machine. A nine-tier role model that scopes what the assistant may read on a per-question basis, applied to the retrieved context rather than to the model's willingness to answer. A durable ingestion pipeline, an agentic workflow engine, and a destruction path that enumerates its targets from the live catalogue rather than a list in source that drifts.

PROPERTY	POSITION
Network dependency	None between cold boot and a served answer. No telemetry, no activation call-back.
Data at rest	Inside a BitLocker-encrypted volume unlocked by the customer's master password. Document content and group messages additionally AES-256-GCM encrypted at the application layer.
Licence binding	TPM + system UUID + MAC. RSA-OAEP + AES-256-GCM hybrid envelope, ECDSA signed by IronGap.
Inference	Bundled Ollama, with vLLM and TensorRT-LLM behind a common engine interface.
Access	Vault-Ecosystem client over mDNS-discovered TLS. Browser access refused by design.
Containers	None anywhere in the product.

02 Design Philosophy & Threat Posture

Three principles decide most arguments in this codebase: assume the network may not exist, make failures loud, and prefer containment to configuration.

▸ Assume the network is hostile, because it may be absent

Most secure-by-design software assumes a network that is present but untrusted. Vault-OS assumes one that may be absent entirely. That is a stronger constraint and it propagates everywhere: dependency resolution happens at build time, models are staged into the install media, the transcription model loads with remote fetching explicitly disabled, and the licence is a file the customer physically carries to the machine rather than a token an authorisation server issues.

▸ Failures must be loud

The failure mode this product treats as most dangerous is not a crash. It is a subsystem that reports success while doing nothing. Each of the following happened during development, and each is now defended by a check that asks the live system what is true rather than trusting the code path that just ran.

SILENT FAILURE	WHAT IT LOOKED LIKE	DEFENCE NOW
Model seeding	Logged "Default models seeded." immediately after skipping the copy. The vault booted with no inference at all.	The live store is inspected after seeding; the log reports what is actually present.
Burn protocol	Reported a destroyed vault while tables it had never been told about retained everything.	Targets enumerated from `pg_tables`, never a constant.
Packaged build	A 40-minute build succeeded around a binary seventeen hours stale.	The build refuses when any source file is newer than the binary.
Schema bootstrap	Migration files silently not found inside the compiled binary; the database came up with no schema.	Assets resolved through a single path helper that knows the packaged layout.

> **DESIGN RULE**
>
> Verification asks the resulting state, never the intent. "The copy function returned" is not "the models are there", and the difference has shipped.

Containment over configuration

Where a setting could allow the product to reach outside its boundary, the boundary is enforced structurally instead. The bundled inference engine listens on a private port (`41434` , deliberately not the well-known `11434`) so that "something is already listening, adopt it" can never silently attach to an unrelated engine the operator happens to have installed, with its own model store and lifecycle. The model store is install-relative rather than the shared user profile location, so two installations on one machine never observe each other.

Scope

Defended

- Data exfiltration over the network
- Theft of the physical disk
- A terminated employee retaining access
- An operator exceeding clearance via the assistant
- Licence duplication onto other hardware
- Retroactive tampering with the audit record

Not claimed

- An attacker with sustained kernel-level privilege on the host *while the enclave is mounted*. Once unlocked for legitimate use, the volume is readable by sufficiently privileged local code.

The window is reduced — the volume is **locked** on graceful shutdown, not merely detached — but it is not eliminated, and we do not claim otherwise.

03 System Architecture

One elevated process owns the entire runtime. No container runtime, no orchestrator, and no inter-service network hop on the critical path.

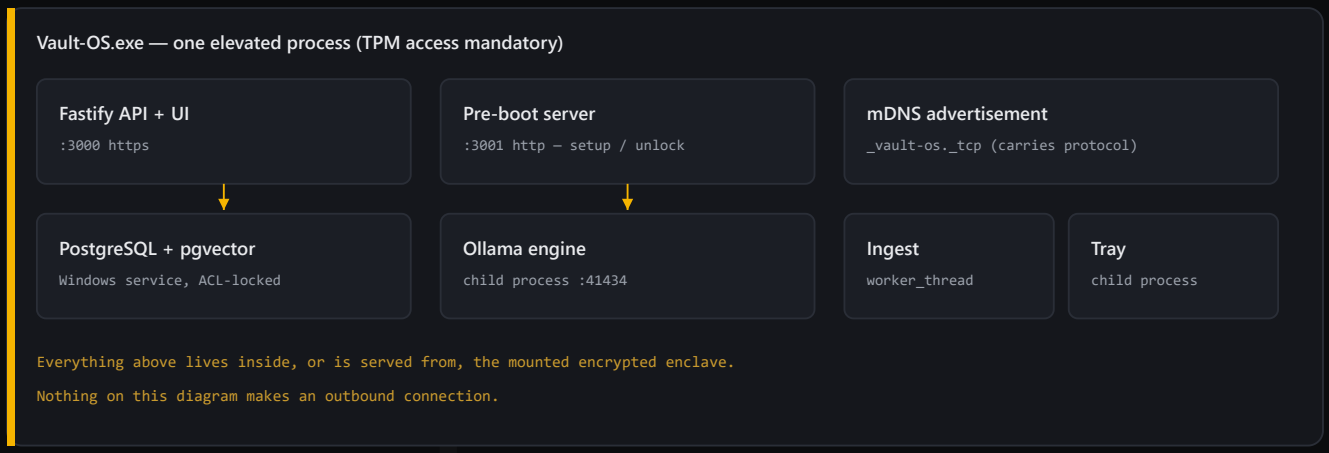


FIGURE 3.1 — PROCESS TOPOLOGY

Why PostgreSQL runs as a service, not a child process

`postgres.exe` refuses to start under an account carrying administrative privilege, and Vault-OS always runs elevated because TPM access requires it. The Service Control Manager is the mechanism that lets an elevated process start a service without that elevation propagating into the service's own security context. The database is registered under `NT AUTHORITY\NetworkService` with its data directory ACL-locked.

Two ports, never one

The pre-boot flow — first-run setup, and every subsequent master-password unlock — runs on its own port in plain HTTP, because at that point the TLS material lives inside a volume that is not yet mounted. The running application serves HTTPS on the real port.

> WHY THIS IS NOT ONE PORT
>
> They were once a single port handed off sequentially. That failed in production as an `EADDRINUSE` crash at the final unlock, when a customer retried quickly enough to catch the operating system not having released the just-closed socket.

> CLIENT IMPLICATION
>
> A client must read the advertised `protocol` from the mDNS record rather than assuming HTTPS, and must follow the navigation across the handoff. Certificate pinning keys off the *current* host:port, not whatever port the client first reached.

Live network re-binding

The listening interface is an administrator-configurable setting stored in the shared `system_config` table, defaulting to all interfaces — appropriate for the central-server-plus-LAN model the product is sold into. The TLS certificate covers the machine's addresses and is regenerated in place when a new network appears, without a restart, so a machine moving between networks does not serve a certificate that no longer matches the address clients reach it on.

04 The Enclave

Two different things carry the word "vault", and conflating them has cost more engineering time than any other single issue in this product. One is an *archive*. The other is a *volume*.

package.vault / system.vault — an ARCHIVE

An AES-256-GCM encrypted archive holding the entire product payload: database binaries, inference engine, language models, the speech-to-text model, the built interface, the tray application and the schema.

The installer copies it verbatim — a raw stream copy, no decryption — and extracts only `bin/`, because the executable must exist as a real file before anything can run.

enclave.vhdx — a VOLUME

A real BitLocker-encrypted virtual disk, created on the customer's machine and locked with their own master password. This is what makes the enclave theirs rather than ours.

Linux uses LUKS2 on a loop device; macOS an APFS sparse bundle. Both keep the `.vault` suffix.

WINDOWS CONSTRAINT

On Windows the volume file must end `.vhdx`. `diskpart` derives a virtual disk's format from the file extension and rejects anything that is not `.vhd` / `.vhdx`, so the product's own `.vault` suffix cannot be used there. Recognition accepts both, so an installation made before this change still mounts.

The archive is deliberately NOT deleted after extraction — it is the durable factory copy the enclave can be rebuilt from, and is refreshed from the live volume before every graceful shutdown.

FIGURE 4.1 — FROM INSTALL MEDIA TO A CUSTOMER-OWNED ENCRYPTED VOLUME

Why the volume is built in the wizard, not at first boot

Building it requires the master password, and the first-run wizard is the only place that password exists. Step one sets the master password; step two activates the licence; step three builds the volume. It is therefore created with the real password from the outset — there is no factory key, no temporary passphrase, and no rekey step that could be interrupted halfway leaving a volume in an indeterminate state.

◆ RECOVERY

An interrupted enclave build used to be unrecoverable: the half-made volume remained and every retry met "vault already exists". A `.building` sidecar now distinguishes an interrupted volume from a finished one, and first-run setup is entitled to replace an unfinished one — a successful preparation ends the wizard permanently, so an unfinished volume can only ever be debris.

Where things live at runtime

```
IronGap/Vault-OS/                plain, unencrypted
  Vault-OS.exe   Vault-OS_tray.exe
  system.vault                   the encrypted factory archive
  enclave.vhdx                   the encrypted volume
  vault_dropzone/                outside on purpose — an inbox has to be
                                 reachable while the vault is locked
  logs/                          outside on purpose — the crashes that
                                 matter most happen before the volume mounts
  .env.vault   LICENSE.vault
  _uninstaller/

<mounted volume, e.g. Y:\>       everything below is encrypted
  db_data/  vault_models/  vault_archive/
  native_postgres/  ollama_engine/  vault-ui/  tray/  db/
  models_by_name/                whisper/ stays here; manifests/ and blobs/
                                 move into ollama_engine/models on first boot
```

INVARIANT

Shipped-asset directories have more than one owner. `models_by_name/` holds the inference engine's store *and* the transcription model. Seeding must take only the subdirectories it owns — handing over the parent relocates the transcription model into a store that will never read it, and speech-to-text then fails on every installation while reporting nothing but a single mild warning at boot.

Deliberately outside the enclave

PATH	WHY IT STAYS OUTSIDE
`vault_dropzone/`	An ingestion inbox that only exists while the vault is mounted is not an inbox. Contents are transient: files are ingested and the originals moved into the encrypted archive.
`logs/`	The crashes that matter most — wrong master password, BitLocker failure, missing volume — happen *before* the enclave mounts. A log inside it would be unreadable in exactly the cases someone needs to read it.
`backups/`	Already independently encrypted. A backup stored inside the volume it backs up does not survive that volume being lost, and is therefore not a backup.
Update staging	An incoming product update has to be stageable and applicable without a mounted enclave.

05 Boot Chain & Hardware Tethering

From a cold, powered-off machine to a serving vault, with the failure branch at every step being a loud halt rather than a degraded start.

```
Vault-OS.exe starts  (elevated, TPM mandatory)
  ├─ no .env?            → generate one; master password requested next
  ├─ no LICENSE.vault?   → first-run wizard on :3001
  │     1. master password        → seals the environment vault
  │     2. licence activation     → USB auto-detect or manual upload
  │     3. build the enclave      → create + BitLocker + unpack archive
  │     then restarts itself
  ├─ sealed?             → unlock server on :3001, master password requested
  ├─ mount enclave volumes
  ├─ ensureLiveVaultDirs()        db_data, vault_models, vault_archive
  ├─ nativeDb.ensureRunning()     initdb → createdb → init + migration
  ├─ self-healing migrations      the only path that touches live data
  ├─ spawn tray as a child process
  ├─ start the inference engine   :41434
  └─ serve on :3000 (https), advertise over mDNS
```

Hardware fingerprint

Identity is derived from the TPM endorsement material, the platform-native system UUID and the primary MAC address, composed and hashed. The export produced for licence issuance is a signed document the customer sends to IronGap; the licence returned is addressed to that fingerprint and will not yield a usable payload elsewhere.

> ANTI-CLONE GUARANTEE
>
> A missing or non-matching licence is **fatal** by default. A stolen drive or cloned image refuses to boot. Where a TPM is present but its key cannot be read, the vault drops to an explicit, logged software-lock fallback rather than silently weakening.

Activation without a network

Activation accepts a licence from removable media — watched for automatically — or from a manual upload in the wizard. A USB-sourced activation is followed by a guarded secure wipe of the source drive. A manual upload never is, because there is no way to know the upload came from disposable removable media rather than the customer's working disk.

Continuous enforcement

TERM	ENFORCED WHERE
Contract dates	Checked at boot and on privileged routes; clock integrity is validated against the monotonic progression recorded in the audit chain, so winding the clock back is detected.
Feature grants	Route-level gates. A revoked feature stops answering, it does not merely disappear from the interface.
Seat count	Checked against active accounts at the moment of hire.
Concurrency	An in-memory semaphore caps simultaneous AI operations — chat, workflow, ingestion — against the licensed ceiling.

06 Cryptographic Subsystem

Independent key domains protect the sealed environment, the enclave volume, data at rest, the audit ledger and the licence. Compromise of one does not cascade into the others.

ASSET	MECHANISM	PROTECTS
Master password	scrypt → AES-256-GCM	The sealed environment (`.env.vault`) and the enclave volume
Enclave volume	BitLocker / LUKS2 / APFS	Everything at rest: database, models, archived files
AES content key	SHA-256 → AES-256-GCM	Document chunks and group messages
Audit key	ECDSA secp256k1	Hash-chained ledger signatures
Recovery key	AES-256-GCM	Encrypted database backups
Package key	AES-256-GCM	The product archive; handed over by the installer under its own elevated token
Licence pair	RSA-OAEP + ECDSA	Licence confidentiality and authenticity
Session tokens	HMAC-SHA256	Authenticated sessions

▸ Environment sealing

```
# .env.vault — authenticated, salted, single-line payload
SALT : IV : AUTHTAG : CIPHERTEXT   (all hex)

scryptSync(masterPassword, salt, 32) → aes-256-gcm key
decipher.setAuthTag(authTag)  // GCM integrity — tamper = decrypt failure = halt
```

The plaintext `.env` is deleted the instant sealing succeeds. The recovery key and audit signing key are generated once at first seal and never regenerated on a re-seal.

Two licence keypairs, deliberately

ECDSA signing key

Vendor-only, forever. Only IronGap can *issue* a licence.

RSA decryption key

Ships inside every installation. Every instance can *read* a licence addressed to it.

A single keypair for both purposes would mean the key that decrypts is also the key that forges — and it ships to every customer.

OPERATIONAL NOTE

Three artefacts must agree or licences fail with the characteristic symptom "decrypts but signature invalid": the vendor's private signing key, the public key bundled into the product, and the key held in the issuing service's environment. That symptom almost always means the last of the three has drifted.

The tamper-evident ledger

Every privileged action is written through a single logging path that builds a hash chain: each entry hashes the previous hash together with the actor, action and detail, so a retroactive edit breaks every downstream link. Each hash is additionally signed and mirrored to an append-only file. A transaction-level advisory lock serialises writers so the chain cannot fork under concurrency.

DELIBERATE TRADE-OFF

Vector **embeddings remain plaintext** — the index must perform cosine mathematics directly on the float arrays. Only human-readable chunk content is encrypted. Extracted entity values are likewise plaintext so the knowledge graph supports fast search. Both are recorded as explicit acknowledgements, not oversights.

07 Identity, Authentication & RBAC

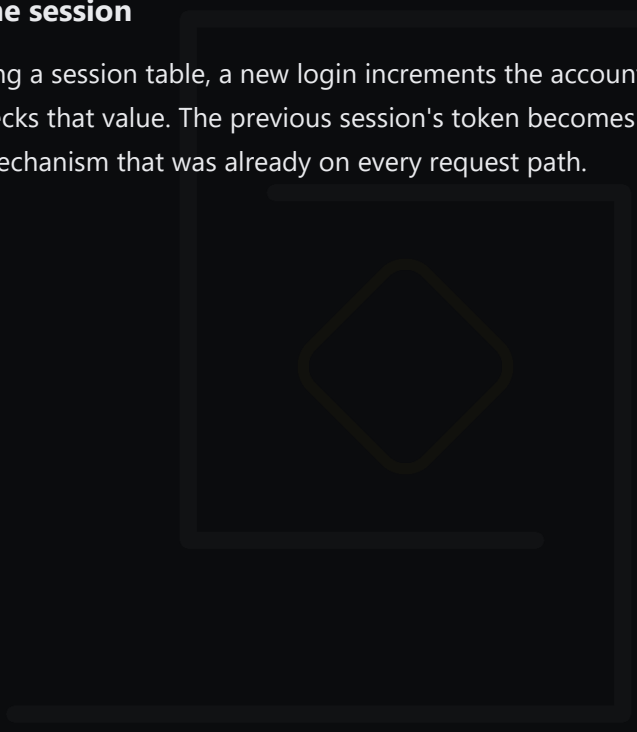
Authentication is stateless in transport but stateful in trust: a signed token carries the claim, and every protected request re-confirms the holder still exists, is not terminated, and presents the current token version.

▸ Two factors, and the second cannot stand alone

Password verification returns **not a session** but a single-purpose, short-lived pre-auth ticket bound to the account and its token version. Only that ticket plus a correct one-time code yields a real session, and every authorisation decorator rejects pre-auth-scoped tokens — so a ticket can never be replayed against a protected route. First login forces a password change before anything else is reachable.

▸ One account, one session

Rather than maintaining a session table, a new login increments the account's token version — and every request already re-checks that value. The previous session's token becomes invalid at the moment the new one is issued, through a mechanism that was already on every request path.



The hierarchy

TIER	ROLE	VRAM CEILING	NOTES
8	**Chairman**	Unlimited	Backup restore, signed updates
7	**admin**	Unlimited	System administration
6	**Board**	Unlimited	Burn authorisation
5	**VIP**	12 GB	Burn authorisation
4	**Senior_Manager**	4 GB	HR surface
3	**Analyst**	4 GB	
2	**Department_Head**	4 GB	HR surface, department-scoped
1	**Employee**	2 GB	
0	**user**	2 GB	

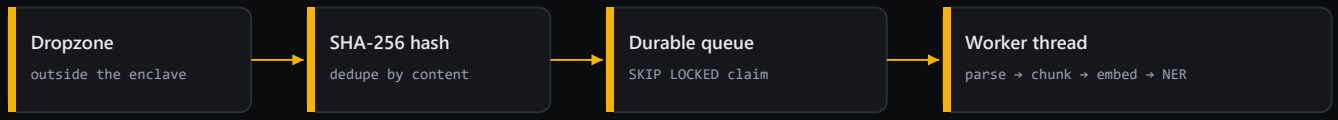
Clearance is both a permission boundary and a resource one. The VRAM ceiling governs which models a role may occupy the accelerator with; over-budget requests degrade gracefully and raise an audit event rather than failing opaquely. The eligible device list is resolved at runtime from the machine's actual detected hardware — an earlier version pinned specific GPU model names, which meant every request on any appliance with different silicon matched nothing at all.

◆ APPLIED TO THE ASSISTANT, NOT JUST THE INTERFACE

Retrieval filters by the asking user's department and clearance *before the model sees anything*, and the assistant's tools are filtered per user by the same rules. A user cannot obtain through conversation what they could not obtain through the interface, because the restriction is applied to the retrieved context rather than to the model's willingness to answer.

08 Knowledge Ingestion Pipeline

Two paths lead into the vault — an administrator uploading directly, and a watched folder for bulk intake. Both converge on the same worker and the same durable queue.



Idempotent: a hash already completed is discarded. Zero-trust: the source file is removed from the dropzone once ingested or recognised as a duplicate.

FIGURE 8.1 — INGESTION PATH

Processing

- Parsing runs in dedicated worker threads, never the main event loop.
- Formats: PDF, DOCX, CSV, TXT, Markdown, plus images.
- Images are downsampled before being handed to the vision model.
- Chunking is boundary-aware — 1000 characters with 200 of overlap, split on sentence endings rather than mid-word — yielding to the event loop periodically so garbage collection can run during a large ingestion.
- Documents past a length threshold receive an executive summary and named-entity extraction, populating the knowledge graph.

> A BUG WORTH RECORDING
>
> The worker once asked for one model *by name*. On an installation running any other model, summarisation and entity extraction both failed silently and the entity map stayed empty forever, with nothing anywhere saying why. The configured model is now passed through.

Operational visibility

The pipeline exposes its own state: the configured watch directory, what is currently sitting in the dropzone awaiting pickup, and the live job queue with status filters and per-job error detail on failures. The queue had existed for many phases with nothing exposing it, which meant a failed ingestion was invisible unless someone noticed a document missing.

09 Retrieval & Inference Fabric

Vector retrieval in PostgreSQL, three interchangeable inference backends behind one interface, and local speech-to-text — none of which touch a network.

▸ Retrieval

768-dimension embeddings, HNSW indexed for approximate cosine similarity. Connections carry a statement timeout: a single pathological query holding a connection indefinitely will, in sufficient number, exhaust the pool and hang every unrelated feature in a way indistinguishable from the application itself being wedged.

Engines

ENGINE	LIFECYCLE	NOTES
Ollama	Boot-time daemon	Bundled and default. Model-less daemon that models load into, so it gets a boot lifecycle rather than a per-model spawn.
vLLM	Per-model spawn	Starts with a specific model to serve, spawn-and-verify-readiness.
TensorRT-LLM	Per-model spawn	Distributed execution across multiple accelerators.

All three stream token by token. Chat, vision and embedding roles are independently reassignable at runtime without re-ingesting anything already processed.

> RESIDENT-MODEL MANAGEMENT
>
> Every question runs an embedding call for retrieval immediately before the chat call. Instructing the engine never to *idle*-evict a model does nothing to stop it evicting one to make room for a *different* model — so on constrained hardware that pair displaced each other on every single message, and the chat model reloaded before every reply. The resident ceiling now covers embedding, chat/vision and the ingestion worker's model concurrently. It is a soft cap: real VRAM accounting still refuses a model that does not fit.

Speech-to-text

Voice messages are transcribed locally by a multilingual model shipped in the archive.

- **Remote model fetching is disabled unconditionally.** The library's default would have the first voice message on a customer machine reach out to a public model host — which an air-gapped appliance cannot do and must never attempt.
- **No language is pinned.** A hardcoded English setting forces English interpretation even with a multilingual model loaded; omitting it lets the model detect the spoken language across roughly ninety-nine of them.
- **Dependencies load lazily.** This module is reached by every route, so a broken transcription dependency once took the whole server down at import time rather than merely disabling voice.
- **Unavailability throws rather than returning empty text.** Empty text is also the correct result for a recording with no speech in it, and a caller must be able to tell those apart.

10 Collaboration & Group Chat

Vault-OS is not only a person talking to a model. It is several people talking to each other, with the model present as an invited participant that can see exactly as much as they collectively may.

Group messaging is modelled separately from AI chat sessions because their requirements are *opposed*: an incognito AI session must be able to vanish completely, and a group message must not. Conflating them would force one of those guarantees to give way.

Group roles are group-local

◆ DELIBERATE SEPARATION

A group's member role is independent of the organisation-wide clearance tier. Being a Department Head does not make you an administrator of a group you were invited into, and being an Employee does not stop you owning one you created. Organisational seniority and group ownership are genuinely different things, and collapsing them produces the wrong answer in both directions.

Attachments have a policy, not just a location

POLICY	BEHAVIOUR
`department`	Filed into the group's folder inside its department's tree.
`ingest`	Filed *and* put through the full ingestion pipeline, so the assistant can retrieve it later.
`ignore`	Kept with the message only. Never filed, never indexed.

A group's department is resolved from its members' own departments at creation and is a genuine access-control input rather than a label — it determines which department's folder tree the group's material lands in, and therefore who can later reach it.

Ephemeral and limited media

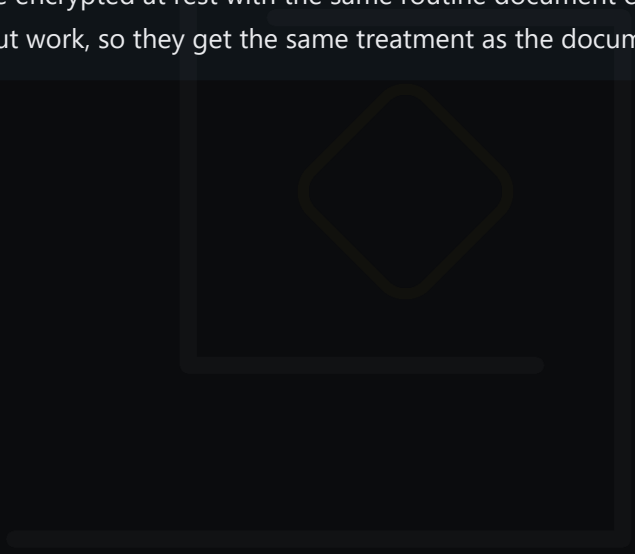
Attachments support one-time viewing and download budgets, both enforced server-side against a per-user view ledger rather than by client-side politeness. A consumed one-time item is refused to every subsequent viewer; an exhausted download budget refuses to serve at all, and the remaining count is returned as a response header so the interface can show it honestly rather than estimating.

Voice notes, video notes, images and video are all captured in the client and stored inside the encrypted volume — which means the existing backup and destruction paths already cover them without a new mechanism.

Summoning the assistant

Mentioning the assistant brings it into the conversation under an explicit context scope: the message alone, the group's history, or the wider vault subject to the asking user's own clearance. The scope is chosen deliberately rather than inferred, so pulling vault-wide context into a group conversation is always a decision someone made.

> ENCRYPTION
>
> Message bodies are encrypted at rest with the same routine document content uses. Group messages are people talking about work, so they get the same treatment as the documents they discuss.



11 Autonomous Workflow & Agent Engine

A directed-acyclic-graph runtime lets cleared operators compose local agents, vector search, database actions and sandboxed code into scheduled or event-driven automations that never leave the air gap.

Execution is a topological sort with retry and conditional routing. Nineteen node types are implemented, spanning agent loops, retrieval, parameterised queries, enrichment, routing, data reshaping, sandboxed execution, timing and human approval.

NODE	PURPOSE	CONTAINMENT
`llm_agent`	ReAct-style agent loop over the local model	Bounded iterations; RBAC-filtered tools
`pgvector_search`	Semantic retrieval	Scoped to the executing identity
`postgres_query`	Parameterised query	Never string-interpolated
`js_sandbox`	Inline transformation logic	Isolated context, no filesystem
`run_isolated_task`	Bounded sub-task execution	argv mapping, no shell
`file_system_write`	Write an artefact out	Path-traversal jailed to the dropzone
`switch_router / conditional_router`	Multi-way and binary branching	Pure over prior node output
`data_transform`	In-graph reshaping library	Pure functions only
`approval_gate`	Human sign-off	Blocks the branch until cleared
`time_window_gate / delay`	Time-of-day bounds; deferral	—
`for_each`	Iteration over a collection	Bounded fan-out
`vram_purge / notify`	Accelerator reclaim; operator notification	—

REMOVED DELIBERATELY

Earlier revisions described a *container execution node* and an *API/webhook gateway module*. Both have been removed. A node that runs arbitrary containers and a gateway that invites inbound integration each contradict the air-gapped design they were supposed to sit inside. Neither will be reintroduced.

Triggers

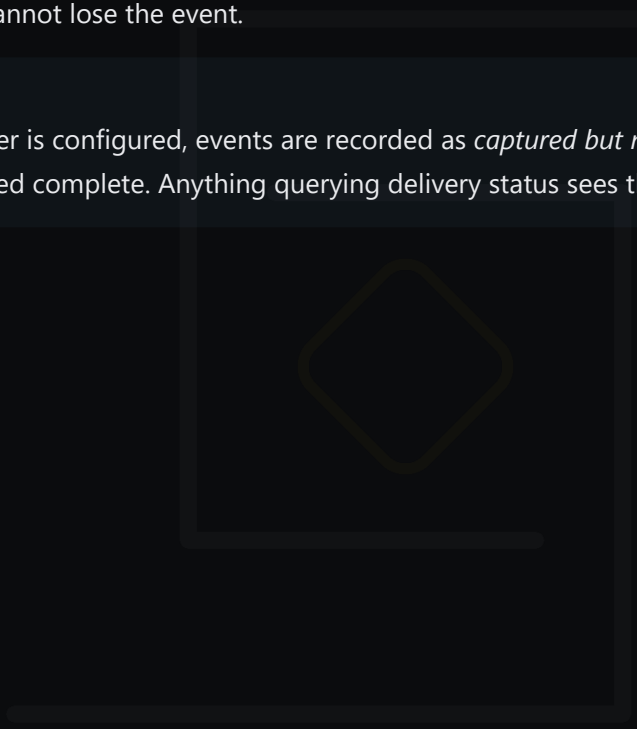
Workflows fire on a cron schedule, at a one-shot date and time, or from database events delivered over a dedicated notification channel held by a permanently checked-out connection.

Reliable eventing

Events use a transactional outbox: the event row is written in the same transaction as the state change that caused it, then dispatched asynchronously with retry and backoff. A crash between "state changed" and "event delivered" therefore cannot lose the event.

HONEST BOOKKEEPING

Where no dispatcher is configured, events are recorded as *captured but not delivered* rather than optimistically marked complete. Anything querying delivery status sees the truth.



12 Access & the Vault-Ecosystem Client

Vault-OS has no public web address, and it is not reachable from a browser — including a browser running on the appliance itself. Access is through the Vault-Ecosystem client application.

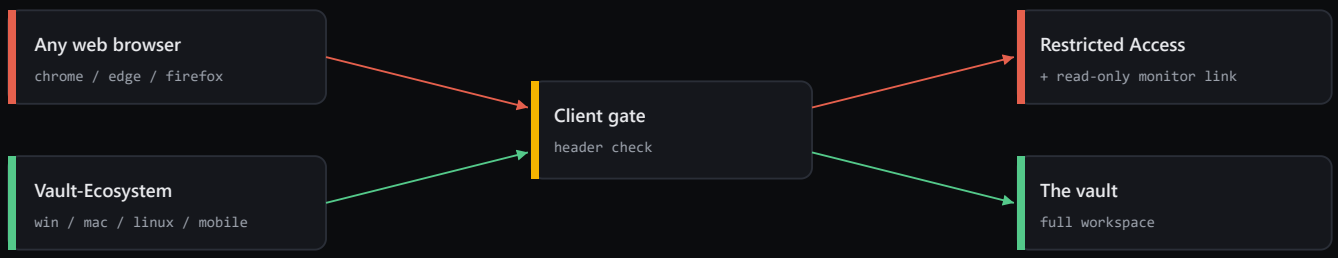


FIGURE 12.1 — EVERY REQUEST PASSES THE SAME GATE

Why a dedicated client

- **Extensions can read the page.** A browser is a large, extensible application; any installed extension with page access can read what the vault just decrypted. That surface is outside the appliance's control and outside the customer's.
- **Pinning only works if it cannot be waved away.** Vault-OS presents a self-signed certificate. In a browser that is a warning users learn to click through; in a dedicated client it is a pin, trusted on first use, with unexpected changes surfaced rather than shrugged off.
- **Habit is the realistic threat.** The common failure is not an attacker but someone typing the appliance's address into a browser because that is what they always do.

> STATED PRECISELY
>
> The gate identifies the client by a header, which a determined person with developer tools could replicate. It is **not** presented as a cryptographic control, and it is not what stands between an attacker and the data — the enclave, the licence binding and the role model are. What it reliably removes is the casual path, which is the one that actually gets used. There is deliberately **no loopback exemption**: a same-machine exemption sounds narrow but would readmit precisely the access path the gate exists to close.

Discovery and platforms

Nodes advertise themselves on the local network; the client browses for them and lists what it finds. There is no address to type. The advertisement carries the protocol, so the client correctly follows the appliance from its plain-HTTP pre-boot phase to its HTTPS running phase.

CLIENT	TECHNOLOGY	STATUS
Windows desktop	Electron	SHIPPING Bundled inside the Vault-OS installer
Android	Flutter	SHIPPING
Windows	Flutter	BUILT Electron remains the recommended desktop client
macOS · Linux · iOS	Flutter	IN DEVELOPMENT Requires the respective platform to build and sign

The mobile and Flutter clients are a secure shell around the appliance's own interface rather than a separate reimplementation, so every capability in this document is present the moment the session loads.

13 Data Lifecycle, Burn & Update

Backup

Full-system encrypted backups cover the database, logs and ingested files. Each backup is protected by its own key, never reused across backups: a manually created backup's key is shown once at creation and never stored, while a scheduled backup's key is wrapped and revealed exactly once. Backups live *outside* the enclave, because a backup kept inside the volume it backs up does not survive that volume being lost. Restore is gated at the highest role tier.

The burn protocol

An authenticated, re-authenticated, licence-gated destruction path for circumstances where the appliance must not survive: files are shredded to NIST SP 800-88, identifying columns are overwritten, and every table is truncated.

> WHY THE TARGET LIST IS ENUMERATED, NOT WRITTEN DOWN
>
> The burn once truncated a fixed list of eighteen tables named in source. It had never been told about the group chat tables, the notification inbox, assistant memory, backup metadata, or five others — all of which survived a "completed" burn. Worse, a single stale name would raise an error and roll the whole transaction back, destroying *nothing at all* after the operator had been told the vault was gone. The list is now enumerated from the live catalogue, so it cannot drift behind the schema.

Shutdown

Graceful shutdown retires the tray, refreshes the factory archive from the live volume, and **locks** the encrypted volume rather than merely detaching it. Detaching alone leaves the volume unlocked in the operating system's view — the encryption is present but the key is still resident, which is not the state a powered-down appliance should be in.

Updates

Updates arrive as signed patches applied deliberately by the highest role tier, staged outside the enclave so an update can be applied without the volume mounted. There is no automatic update channel, because an automatic update channel is both a network dependency and an inbound code path into an air-gapped machine.

Retention

Configurable retention policy is administered through the same key/value configuration service the network binding and watch-directory settings use, rather than each setting acquiring its own bespoke table.

14 Operator Console

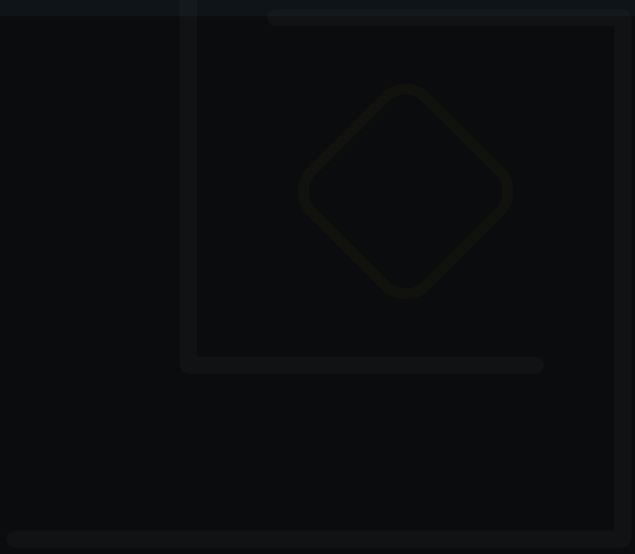
The interface is served by the appliance itself, built at package time and extracted into the encrypted volume during first-run setup.

SURFACE	PURPOSE
Sessions	The main assistant workspace: streaming responses, tool invocation, per-role scoping, forced tool selection by mention.
Groups	Multi-party messaging with the assistant as a participant; voice notes, video notes and one-time media.
Archive · Explorer	Ingested documents, and the folder tree organising them by department and clearance. One tree rooted at a single company node whose direct children *are* departments.
Ingest · Pipeline	Manual intake, and the live view of the durable ingestion queue with per-job failure detail.
Map	The extracted entity graph across the corpus, navigable to the files each entity came from and back.
Loops · Telemetry	The workflow canvas, execution history, and the cross-workflow approval queue.
Hardware · Console	Live accelerator and system telemetry over server-sent events; the read-only system log.
Overseer	Accounts and roles, the audit ledger, system configuration, backups, network sharing, retention, storage — and the burn control, isolated at the foot of the panel away from routine settings.

Details that matter more than they look

- Round video messages draw their own controls — a progress ring traced around the circle's own edge — because a native control bar renders rectangular regardless of the element's shape and clips against a circular frame.
- The right-click message menu is in-app rather than the browser's, and is position-clamped so it cannot open off-screen near a viewport edge.
- Dropdown options are given explicit, *opaque* colours: option elements are drawn by the operating system, so a translucent dark background composites against the platform's white popup and lands back where it started — unreadable.
- The interface and the assistant's replies follow the language the user writes in, with an explicit per-account preference available.

> CAPACITY IS VISIBLE, NOT IMPLICIT
>
> Seat and concurrency pressure against the licence is surfaced in the interface through a shared component, so an administrator sees the ceiling approaching rather than discovering it when someone is refused a session.



15 API Surface Reference

Every route is authenticated unless explicitly noted, rate-limited at the edge, and validated against a schema before reaching a handler.

GROUP	REPRESENTATIVE ROUTES	GATE
Auth	/auth/verify · /auth/totp-verify · /auth/reset-initial	Rate-limited; pre-auth ticket required for the second factor
Inference	/ask · /ask/stream · /search · /tools	Authenticated + licence feature
Files	/files · /files/:id/download · /files/:id/burn	Role-scoped; burn requires Board or VIP
Groups	/groups · /groups/:id/messages · /groups/:id/media	Membership checked server-side per call
Ingestion	/ingest · /upload · /ingestion/jobs	Administrator
Folders	/folders/root · /folders/:id/children	Authenticated; writes need an exact department match
Workflows	/workflows · /workflows/:id/execute · approvals	Elevated clearance
HR	/hr/employees · /hr/capacity-status	Manager or above
System	/system/backups · /system/update · /admin/burn	Administrator; restore and update require Chairman
Settings	/settings/network-config · retention · storage	Administrator

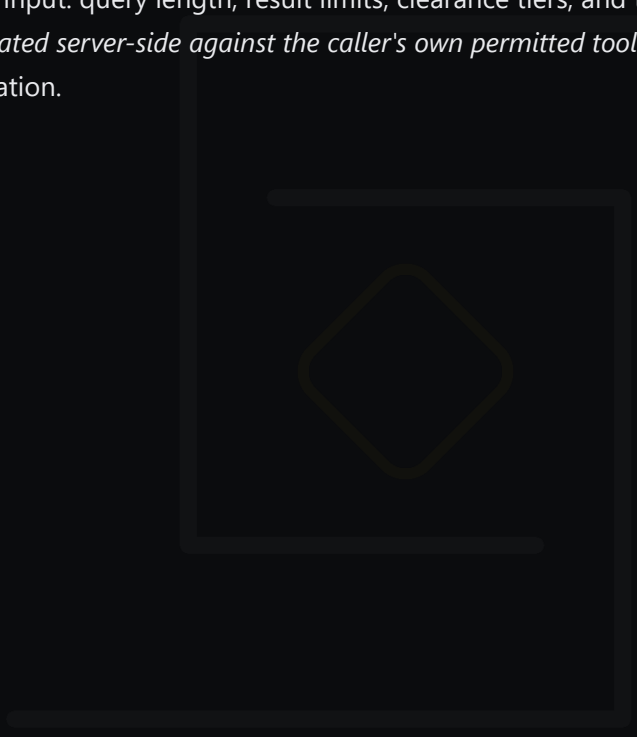
Two contracts, not one

Distinct from the HTTP surface above, the assistant's **internal tool contract** is what the model itself calls during a turn. Each of the thirty-three tools is independently filtered per user and logged, regardless of whether the underlying HTTP route would otherwise be reachable by that role.

> RENDERING ALLOW-LIST
>
> Only an explicit allow-list of tool results receives a rich preview card in the interface. Anything shaped like personal data, security records or vault content is deliberately excluded from a rendering path never designed to sanitise arbitrary tool output.

Validation

Schemas bound every input: query length, result limits, clearance tiers, and the mention-forced tool list — which is capped and *re-validated server-side against the caller's own permitted tools* before it is trusted, exactly like every other tool invocation.



16 Verification Status

This section states what has been verified and what has not, at the same level of detail. A roadmap listing only strengths would misrepresent the product to exactly the customers least able to absorb a surprise.

AREA	STATUS	BASIS
Windows install → licence → enclave → boot	VERIFIED	Exercised end to end on real hardware with a real TPM
Logic, command construction, invariants	VERIFIED	Over one thousand automated tests
Inference, retrieval, ingestion	VERIFIED	Running on installed builds
Automated end-to-end install harness	ABSENT	Requires real TPM hardware and administrative rights. Every enclave defect to date was found by a person installing the product.
Linux enclave backend	NEVER EXECUTED	Written against documented tool behaviour. Treat as a draft.
macOS enclave backend	NEVER EXECUTED	As above; requires Apple hardware.
Speech-to-text on an installed build	NOT YET	Both its original check and its regression test run from source.
Recently added interface surfaces	PARTIAL	Compiles and is unit-tested; most has not been seen running against a live backend.

◆ WHY THIS IS IN THE DOCUMENT

On this product the most expensive defects all *looked* fine: a stale binary that packaged successfully, models that logged "seeded" after being skipped, a destruction routine that reported destroying data it never touched. "Builds cleanly" is not "works", and the distinction is maintained deliberately throughout.

Direction

- An unattended virtual-machine harness that runs the installer and asserts the system reaches a serving state — this would have caught every enclave defect found so far.
- Linux and macOS enclave backends executed and hardened on real hardware.
- File provenance through the workflow engine, so a workflow's outputs link back to the documents that produced them.
- Ingestion progress surfaced in the explorer, not only the pipeline view.

A Schema, Keys & Glossary

Principal tables

DOMAIN	TABLES
Identity	users · active_logins · user_usage_stats · ephemeral_tickets
Knowledge	files · documents · entities · ingestion_jobs · vault_folders · generated_artifacts
Interaction	sessions · session_files · chat_messages · user_memory
Collaboration	chat_groups · chat_group_members · chat_group_messages · chat_group_invites · chat_group_media · chat_group_media_views · user_inbox
Automation	workflows · workflow_nodes · workflow_edges · agent_executions · webhook_outbox
Governance	security_logs · hr_actions · tool_invocations · backups · system_config · installed_packages

Two-tier migration discipline

migration.sql	A destructive fresh-install script. Opens by dropping tables. **Never safe against live data.**
self-healing block	Idempotent creates and column additions, run on every boot. The only path that touches an existing installation.
the rule	Every schema change belongs in both. In only the first, it works on a new install and does nothing on every existing one. In only the second, a fresh install and an upgraded one end up with different schemas.

Glossary

TERM	MEANING
Archive	The AES-256-GCM encrypted product payload shipped on the install media and retained as the durable factory copy.
Volume / enclave	The encrypted virtual disk created on the customer's machine with their own master password.
Master password	Set in step one of first-run setup. Seals the environment secrets and locks the enclave. Not recoverable without it or the one-time recovery key.
Dropzone	The plain folder outside the enclave into which documents are dropped for automatic ingestion.
Burn	The authenticated, licence-gated destruction of vault contents to NIST SP 800-88.
Client gate	The check that refuses browsers and admits only the Vault-Ecosystem client.
Vault-Ecosystem	The client application through which Vault-OS is reached. Discovers appliances on the local network; required for access.

Document control

Product version	1.0.9.1
Development phase	29.8.9
Supersedes	Vault-OS Technical Whitepaper, Phase 27.1.3
Classification	Confidential & Proprietary
Distribution	Named recipients under NDA