

PKA — INTERNAL SYSTEMS

Atlas

Internal Library & Team Chat Platform

Full Build Reference

Version 5 • May 6, 2026

Stack: PostgreSQL 18 • Next.js 15 • React 19

Host: Gondor • LAN: 192.168.0.75

CONFIDENTIAL — J2 BOOKKEEPING / PKA

Overview

Atlas is PKA's internal document library and team-chat hub — a Next.js web app that ingests, deduplicates, full-text-searches, and tags PKA's documents, deliverables, and scanned files, plus hosts persistent multi-thread chat with the PKA AI team agents. As of 2026-05-02 it is a **client/server application backed by Postgres on Gondor**, accessible from both Rohan and Gondor concurrently.

- **Source:** `D:\PKA\Atlas\` on Gondor — app code at `D:\PKA\Atlas\app\`
- **Run:** Auto-starts on Gondor login via `start-atlas.vbs` (port 80)
- **URL:** `http://localhost` on Gondor; `http://192.168.0.75` from Rohan/LAN
- **Database:** Postgres 18.3 on Gondor, `atlas` DB on `192.168.0.75:5432`

Migration History

Atlas was originally built as a local-first SQLite app. On **2026-05-02**, it was migrated end-to-end to Postgres 18.3 to enable safe concurrent multi-machine use. Full migration log:

Mem note `57373241-e6f2-4f4a-9702-e0331a6aa58a` .

Current Stats (as of 2026-05-06)

Metric	Value
Documents	5,810
Document locations	6,446
Document-client links	5,443 (52 clients)
Chat messages / threads	710 messages • 15 threads
Active chat agents	20 total • 14 active from chat-agents.ts
Triage queue	1,385 untagged/unassigned docs
Storage	~395 MB • 1,087 SHA-256 keyed files

Active Team Agents

ID	Name	Role	Color	Sort
larry	Larry	Orchestrator	#25A354	0
nolan	Nolan	HR Manager	#6366F1	1
pax	Pax	Senior Researcher	#8B5CF6	2
vox	Vox	Content Marketing	#8B5CF6	3
riv	Riv	Integrations	#14B8A6	4
ledger	Ledger	Accounting & Finance	#10B981	5
wren	Wren	WordPress Design	#3B82F6	7
pixel	Pixel	Canva Brand Design	#F97316	8
corvus	Corvus	Fantasy Lore	#78716C	9
atlas	Atlas	Full-Stack Developer	#0EA5E9	10
reid	Reid	Operations Strategist	#475569	12
ash	Ash	Diablo 4 Theorycrafter	#DC2626	14
cord	Cord	Data Connector & Spreadsheets	#06B6D4	15
rex	Rex	Tax Strategist	#D97706	16
slate	Slate	FinOptimal & Google Sheets	#64748B	17

Process rule: When a new team member is hired, add them to

`D:\PKA\Atlas\app\lib\chat-agents.ts` AGENTS array. `seedChatAgents()` upserts on next page load. Use the next sort number and a distinct hex color.

Stack

Layer	Choice
-------	--------

Database	PostgreSQL 18.3 (Windows service <code>postgresql-x64-18</code> , data dir <code>D:\PKA\postgres-data</code>)
Full-text search	Postgres <code>tsvector</code> + GIN index + sync trigger
Backend	Next.js 15 API routes (Node.js)
Frontend	Next.js 15 + React 19 + TypeScript + Tailwind
DB driver	<code>pg</code> 8.20 (async/await, connection pool)
Text extraction	pdf-parse, xlsx (SheetJS), mammoth (docx)
File storage	Local filesystem on Gondor (SHA-256 keyed paths)
AI integrations	<code>@anthropic-ai/sdk</code> 0.92 + Claude CLI subprocess
External APIs	QuickBooks Online OAuth (BvA + reporting)

Connection Details

Connection	Value
App DSN (Gondor)	<code>postgresql://atlas:needles6@localhost:5432/atlas</code>
App DSN (Rohan/LAN)	<code>postgresql://atlas:needles6@192.168.0.75:5432/atlas</code>
Superuser DSN	<code>postgresql://postgres:needles6@192.168.0.75:5432/postgres</code>
Postgres binaries	<code>C:\Program Files\PostgreSQL\18</code>
Postgres data dir	<code>D:\PKA\postgres-data</code>
SMB mount (Rohan)	<code>\\192.168.0.75\pka</code> as <code>z:\</code> (user: <code>pka-share / needles6</code>)

`.env.local`

```
D:\PKA\Atlas\app\.env.local -- DATABASE_URL,  
ANTHROPIC_API_KEY, MEM_API_KEY, QBO creds
```

Folder Layout

```

D:\PKA\Atlas\
├─ app/                               Next.js application (canonical)
│  └─ app/                             Pages & API routes
│     └─ api/                          bva, chat-agents, chat-send, chat-threads,
│        │                             chat-uploads, clients, documents, file,
│        │                             projects, qbo, tags, team, triage
│        └─ browse/                    Bulk tagging list/grid
│           └─ clients/                 Client list + per-client docs
│              └─ doc/[id]/            Document detail
│                 └─ reports/          BvA reporting
│                    └─ tags/          Tag management
│                       └─ team/       AI team chat
│                          └─ triage/  Untagged document queue
├─ components/chat/                   AgentSidebar, ChatArea, MessageBubble,
│                                     MessageInput, ThreadList
├─ lib/                               db.ts, chat-db.ts, chat-agents.ts,
│                                     search.ts, qbo.ts, bva.ts, claude.ts
├─ scripts/                           ingest.mjs, watch-ingest.mjs,
│                                     watch-ingest-gdrive.mjs, reextract.mjs
├─ bva/                               BvA pipeline engine (Python)
├─ storage/                           SHA-256 keyed file library (~395 MB)
├─ start-atlas.vbs                    Auto-start on Gondor login (port 80)
└─ add-hosts.ps1                      One-time atlas.local hosts entry

```

Database Schema

documents (core table)

```
CREATE TABLE documents (
  id          BIGINT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  hash       TEXT NOT NULL UNIQUE,
  filename   TEXT NOT NULL,
  extension  TEXT,
  mime_type  TEXT,
  size_bytes BIGINT NOT NULL,
  storage_path TEXT NOT NULL,
  original_path TEXT NOT NULL,
  extracted_text TEXT,
  notes      TEXT,
  imported_at TIMESTAMPTZ NOT NULL DEFAULT NOW(),
  updated_at TIMESTAMPTZ NOT NULL DEFAULT NOW(),
  storage_mode TEXT NOT NULL DEFAULT 'local', -- 'local' or 'reference'
  status     TEXT NOT NULL DEFAULT 'ok',
  search     TSVECTOR -- auto-maintained by trigger
);
```

Full-Text Search Trigger

```
CREATE OR REPLACE FUNCTION documents_search_update() RETURNS TRIGGER AS $$
BEGIN
  NEW.search := to_tsvector('english',
    COALESCE(NEW.filename, '') || ' ' || COALESCE(NEW.extracted_text, ''));
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER documents_search_trg
BEFORE INSERT OR UPDATE OF filename, extracted_text ON documents
FOR EACH ROW EXECUTE FUNCTION documents_search_update();
```

Supporting Tables

- **document_locations** — every path where each deduped file was seen
- **clients** — name, slug, notes, gdrive_folder
- **projects** — linked to clients
- **tags** — flat tag system

- **document_clients / document_projects / document_tags** — M2M junctions, ON DELETE CASCADE
- **bva_runs** — BvA report run history (client_slug, period, status, output_path, log)
- **qbo_tokens** — singleton OAuth token row (CHECK id = 1)

Chat Tables

- **chat_agents** — seeded from `chat-agents.ts` AGENTS array via `seedChatAgents()`
- **chat_threads** — per-agent threads; `cli_session_id` enables Claude CLI `--resume`
- **chat_messages** — role, content, model, token counts
- **chat_attachments** — file uploads; `message_id` set when user sends

Pages & API Routes

Route	Method(s)	Purpose
<code>/api/clients</code>	GET	List clients with doc counts
<code>/api/clients/[slug]/upload</code>	POST	Upload to client G: folder + DB
<code>/api/tags</code>	GET, POST	List + create tags
<code>/api/tags/[id]</code>	PATCH, DELETE, POST	Manage tags (rename, merge, delete)
<code>/api/documents/[id]</code>	GET, PATCH	Document detail + edit notes
<code>/api/documents/[id]/tags</code>	POST, DELETE	Link/unlink tag
<code>/api/documents/[id]/clients</code>	POST, DELETE	Link/unlink client
<code>/api/documents/bulk-tags</code>	POST, DELETE	Bulk tag/untag
<code>/api/file/[id]</code>	GET	Stream file content
<code>/api/triage/count</code>	GET	Untagged doc count
<code>/api/qbo/*</code>	GET, POST	QBO OAuth + query
<code>/api/bva/*</code>	GET, POST	BvA adapters, runs, downloads
<code>/api/chat-agents</code>	GET	List agents (seeds on first call)
<code>/api/chat-threads</code>	GET, POST	List + create threads
<code>/api/chat-threads/[id]</code>	GET, PATCH, DELETE	Thread detail, rename, tool grant, archive
<code>/api/chat-uploads</code>	POST	Upload chat attachments
<code>/api/chat-send</code>	POST	Send message, spawn Claude CLI, SSE stream

Team Chat

Backends

1. **/api/chat-send (canonical)** — Spawns local Claude CLI with `--resume` `<cli_session_id>` for continuity. Streams SSE. Stale-session auto-fallback.
2. **/api/team/chat (legacy)** — Direct `@anthropic-ai/sdk`. Kept for older `team_conversations` table.

UI Components

- **AgentSidebar** — color-coded avatars, pulsing green dot for unread (localStorage seen-state)
- **ThreadList** — per-agent threads with rename/archive
- **ChatArea** — markdown rendering (ReactMarkdown + remark-gfm), tool-status indicators, auto-scroll
- **MessageBubble** — inline images/PDFs, download links for other types
- **MessageInput** — file upload + drag-and-drop

Adding a new agent: Add entry to AGENTS in `chat-agents.ts`. Fields: id (lowercase), name, role, color (hex), sort (next int), promptPath (points to `.claude/commands/<name>.md`), `defaultAllowedTools`. `seedChatAgents()` upserts on next page load.

Per-thread tool grants: `PATCH /api/chat-threads/[id] { grantTool: "Edit" }` adds tools on the fly for that thread.

Ingestion Pipeline

```
npm run ingest "D:\PKA\Team Inbox"           # local mode (copies files)
node scripts/ingest.mjs "G:\..." --reference-only # reference mode
npm run watch                                # watch Inbox dirs (2s debounce)
npm run watch:gdrive                          # poll G: drive every 15 min
npm run reextract                             # retry NULL extracted_text
```

Pipeline Steps

1. SHA-256 hash the file. If found, add document_location row only (dedup).
2. Copy to `storage/<prefix>/<hash><ext>` (local) or leave in place (reference).
3. Extract text: pdf-parse, SheetJS, mammoth, or raw read. Strip NUL bytes. Truncate at 1 MB.
4. Insert into documents (RETURNING id). FTS trigger builds search column.
5. Insert document_locations row.

Skipped: files >100 MB; .exe/.dll/.iso/.zip/.rar/media; Google Drive shortcut files.

BvA Reporting Engine

Python engine at `D:\PKA\Atlas\bva\`. Generates Budget vs Actual workbooks for J2 clients.

- **Adapters:** loop_tv (Eagle II LLC / Loop TV), fuzebox (FuzeBox AI LLC)
- **Run history:** stored in bva_runs with stdout/stderr log and output file path
- **Web UI:** /reports — upload wrangler files, start run, download xlsx

QuickBooks Online Integration

`lib/qbo.ts` — OAuth + API against QBO sandbox. Singleton token in qbo_tokens. Auto-refreshes within 5 minutes of expiry.

Portability (Fresh Machine Setup)

1. Install Postgres 18+ from postgresql.org (Windows installer).
2. Create role + DB: `CREATE ROLE atlas WITH LOGIN PASSWORD '...'; CREATE DATABASE atlas OWNER atlas; GRANT ALL ON SCHEMA public TO atlas;`
3. Apply schema: `node apply-schema.mjs`
4. Migrate data (if porting): `node migrate-data.mjs <path-to-atlas.db>`
5. For LAN: `listen_addresses = '*'` in postgresql.conf; add CIDR to pg_hba.conf; open firewall TCP 5432.
6. Copy `D:\PKA\Atlas\` (exclude `node_modules`, `.next`, `data/atlas.db*`, old backups/).
7. `cd app && npm install`
8. Set `DATABASE_URL` in `app/.env.local`.
9. `npm run dev` or configure `start-atlas.vbs` for auto-start on login.

Known Follow-Ups (open as of 2026-05-06)

1. **Stale chat_agents rows** — lane, maya, lux (old seed); finn, flux, sage (orphan placeholders). Cosmetic. Clean by reconciling chat-agents.ts with CLAUDE.md roster and archiving orphan threads.
2. **Gondor jimmi account has no password** — security exposure, unrelated to Atlas.
3. **Password reuse** — needles6 shared across pka-share, postgres superuser, atlas role. Acceptable for now; diversify when more users come online.
4. **xlsx replacement** — SheetJS removed from npm; switch to exceljs when scoped as a project.
5. **2026-05-16 cleanup** — delete SQLite atlas.db + WAL/SHM + Gondor-local backup. Rohan backup preserved as long-term archive.
6. **Sidebar polish** — typing indicators, agent presence, keyboard shortcuts.

Accepted Vulnerabilities (Deferred)

- **postcss XSS via </style> stringify** (moderate) — bundled in Next.js 15; no safe auto-fix available. Atlas doesn't process untrusted CSS. Resolves when Next.js bumps bundled postcss.
- **xlsx prototype pollution + ReDoS** (high) — SheetJS npm unavailable; no fix. Atlas only parses trusted internal files. Future fix: switch to exceljs.

Cross-References

- Migration log: Mem `57373241-e6f2-4f4a-9702-e0331a6aa58a`
- Phase 1 chat origin: Mem `69b2b71a-ffa4-44c4-90b9-46aed6515121`
- Phase 2 chat (uploads): Mem `543f6fce-87d7-448e-9038-5f2ae3996089`
- Cross-machine parity: Mem `a60a4f39-bce0-4e64-b9e4-3e7532d7da5a`
- AI Assistant in a Box: Mem `0eca7b79-4d94-4ce0-8d7d-b46c3f612b94`
- CLAUDE.md / Rohan setup: Mem `d4c8e08a-b5a2-4bed-ba93-08a5f830534a`
- Whitepaper Maintenance Rule: Mem `8413f291-7f19-4f8a-b490-abd6066f3ff0`
- PKA Team Whitepaper Rule: Mem `4398dbe4-b13d-4f41-8bcc-984e2362e61e`

Atlas — PKA Internal Library • Version 5 • May 6, 2026 • Confidential — J2 Bookkeeping / PKA