

Paul Warner III

Selected AI Systems Portfolio

Metro Atlanta, GA | pwarner333@gmail.com | <https://paulwarneriii.com>
<https://github.com/Pwarner96> | <https://www.linkedin.com/in/paul-warner-087198167>

Evidence-backed case studies for operational software, bounded automation, source-grounded decision support, and integration architecture. Status and limitations are explicit; employment, sole-authorship, customer, credential, and outcome claims are not included.



Scan or open <https://paulwarneriii.com>

System	Released status
KeyTurn Cleaning Company OS	Production Prelaunch - Controlled Launch
Government Contract Intelligence Platform	Functional Prototype / In Development - Not Deployed
Agency Operations OS	Functional Prototype / In Development - Local Only; Not Deployed
Tandem	Tested Local Product - MCP Server; Not Hosted
Project A.R.C.H.E.R.	Recovery Candidate - Empty Remote Repository

KeyTurn Cleaning Company OS

Status: Production Prelaunch - Controlled Launch

A multi-surface operating system connecting a public cleaning website, internal CRM, crew portal, Supabase data boundaries, and guarded automation controls.

Business problem

Move and rental-turn cleaning requires dependable handoffs across customer, operator, and crew surfaces without allowing automation to bypass human judgment or launch gates.

Role and evidence scope

Portfolio role scope: implementation architecture, workflow integration, safety-control framing, and evidence synthesis. Sole authorship, live operation, and business outcomes are not claimed.

Verified technologies

Next.js, React, TypeScript, Supabase, PostgreSQL, PWA

SANITIZED ARCHITECTURE / WORKFLOW

01 Role-specific surfaces: public website, internal CRM, and crew portal	02 Application boundary: typed workflows, routes, and server-side validation	03 Control boundary: authentication, authorization, approvals, and audit evidence	04 Data and adapters: Supabase/PostgreSQL plus guarded provider boundaries
---	---	--	---

Implemented workflow evidenced in source

1. Capture a scoped service request | 2. Review estimate and operating context | 3. Coordinate CRM and crew work | 4. Require approval before guarded actions

AI / automation boundary

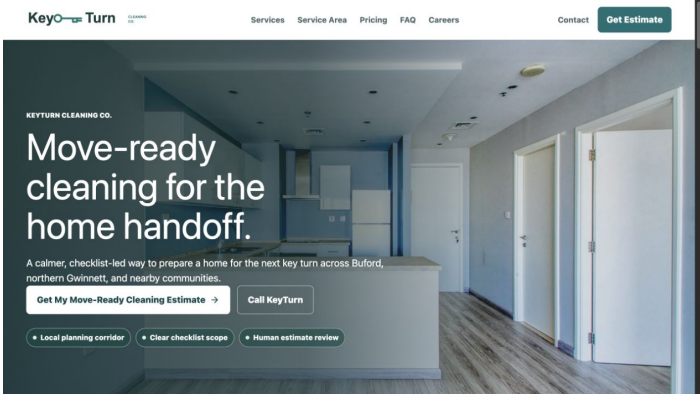
Typed tools, budgets, containment, audit evidence, approval gates, and deterministic fallback are represented in source. Live providers and autonomous external actions remain disabled.

Security and reliability controls

Surface-specific authentication and authorization | Row-level data boundaries and server-only secrets | Fail-closed staging and guarded write paths

Current limitations

Production prelaunch is not evidence of active business operations. | Messaging, payments, and autonomous external actions are not represented as active. Open the evidence-backed case study



Real public-site capture - production-prelaunch evidence only; operations and providers are not claimed as active.

Government Contract Intelligence Platform

Status: Functional Prototype / In Development - Not Deployed

A multi-tenant prototype that normalizes federal opportunities, applies explainable qualification rules, connects award context, and keeps pursuit decisions human-controlled.

Business problem

Small capture teams can lose pursuit time to notices that fail location, eligibility, timing, or scope checks only after detailed review.

Role and evidence scope

Portfolio role scope: source-grounded data architecture, explainable decision workflow, integration-boundary framing, and evidence synthesis. Sole authorship, deployment, and contract outcomes are not claimed.

Verified technologies

Next.js, React, TypeScript, Supabase, PostgreSQL, SAM.gov, USAspending

SANITIZED ARCHITECTURE / WORKFLOW

01 Public-source adapters: SAM.gov and USAspending records	02 Ingestion boundary: normalization, deduplication, and versioning	03 Decision services: deterministic qualification and bounded summaries	04 Review surface: tenant data boundaries and human GO / HOLD / NO-GO authority
---	--	--	--

Implemented workflow evidenced in source

1. Collect and preserve public source records | 2. Normalize, deduplicate, and version notices | 3. Apply explainable fit and risk checks | 4. Present evidence for a human pursuit decision

AI / automation boundary

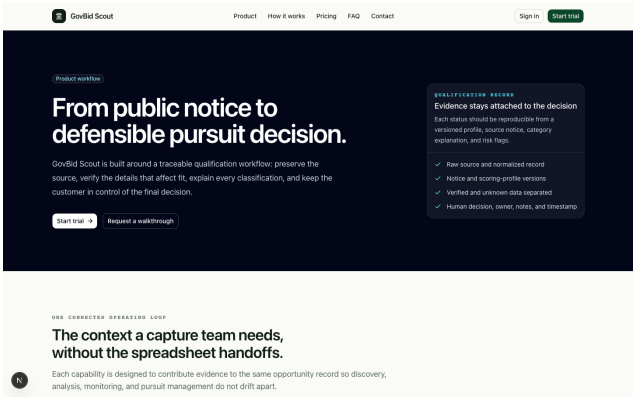
The core pursuit brief and qualification path are deterministic. Optional model-enhanced summaries are bounded and do not replace source evidence or the human pursuit decision.

Security and reliability controls

Tenant-scoped row-level security and server authorization | Raw-source preservation, audit trails, and retry-safe job boundaries | Controlled provider adapters with deterministic fallback

Current limitations

No production service, live billing, provider email, or production database is claimed. | Illustrative records are not represented as live opportunity coverage. Open the evidence-backed case study



Real local-prototype capture - source and human-decision controls; not a deployed service.

Agency Operations OS

Status: Functional Prototype / In Development - Local Only; Not Deployed

A local prototype for lead intake, owner qualification, client and project records, approvals, change requests, support, and organization-scoped portal access.

Business problem

Client-service work becomes hard to operate when sales, onboarding, delivery, approvals, and support lose a shared source of truth.

Role and evidence scope

Portfolio role scope: operations-workflow architecture, access-boundary framing, integration design, and evidence synthesis. Sole authorship, client work, and hosted production operation are not claimed.

Verified technologies

Next.js, React, TypeScript, Supabase, PostgreSQL, Playwright

SANITIZED ARCHITECTURE / WORKFLOW

01 Role-specific surfaces: lead intake, operator CRM, and scoped client portal	02 Application workflows: qualification, projects, approvals, changes, and support	03 Authorization boundary: auth claims, server role checks, and organization membership	04 Data boundary: Supabase/PostgreSQL row-level controls and validation
---	---	--	--

Implemented workflow evidenced in source

1. Capture and qualify a lead | 2. Convert approved work into client and project records | 3. Coordinate milestones, tasks, files, and decisions | 4. Expose only organization-scoped records in the client portal

AI / automation boundary

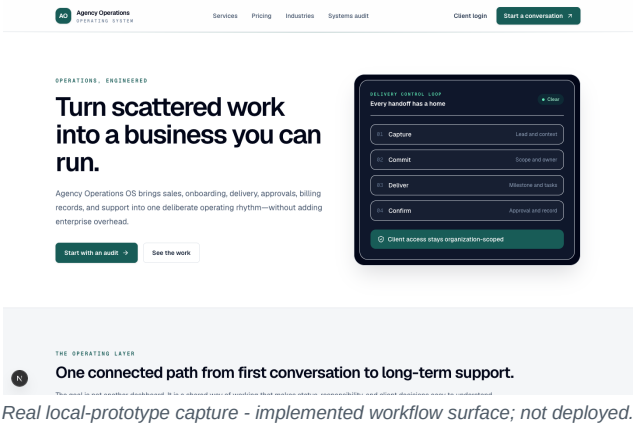
No AI runtime is verified. The system is represented as an operations and integration prototype, not as an AI product.

Security and reliability controls

Authentication claims and server-side role checks | Organization-membership boundaries | Database row-level controls as the final data boundary

Current limitations

The application is local-only and not deployed. | No active clients, public registration, or production providers are claimed. Open the evidence-backed case study



Real local-prototype capture - implemented workflow surface; not deployed.

Tandem

Status: Tested Local Product - MCP Server; Not Hosted

A tested local MCP server and collaboration hub connecting Claude Code, Codex, and a human through shared planning, cross-review, and delivery acceptance.

Business problem

Two coding agents need a common process for independent proposals, task ownership, review responsibilities, and the human's next decision.

Role and evidence scope

Portfolio role scope: MCP integration, workflow architecture, controls, and verification. Evidence includes the Python workflow core, agent adapters, human CLI, and protocol tests.

Verified technologies

Python, SQLite, MCP, Claude Code Channels, Codex app-server, WebSockets

SANITIZED ARCHITECTURE / WORKFLOW

01 Participants: human CLI, Claude MCP adapter, and Codex driver	02 Local hub: authenticated loopback HTTP, agendas, and long-poll events	03 Workflow core: role checks, blind proposals, cross-review, and human gates	04 Durable state: SQLite records and event log; persisted Codex dispatch state
---	---	--	---

Implemented collaboration workflow

1. Collect independent proposals and merge the plan | 2. Request partner review and human plan approval | 3. Execute dependency-ready tasks with cross-review | 4. Verify delivery and require human acceptance

AI / automation boundary

MCP tools and the Codex driver coordinate existing coding agents. The MCP handshake, 19 tools, channel notifications, protocol 1.1 conformance, and 80 unit tests passed. Both lead selections reached delivered with two scripted participants and scripted human decisions, without real model calls.

Security and reliability controls

Per-participant tokens and hashed token lookup | Agenda-only event wakeups; FYI notes do not trigger turns | Codex denies approval/input requests; human questions use the hub | Defaults: 20 dispatches per hour, 900-second turns, and 5-300-second failure backoff

Current limitations

Local-only MCP product; operated through tools and the CLI, with no hosted service or graphical UI required. | A complete product run with both real agents through human acceptance is not verified. | Retry needs an actionable wake or startup reminder; app-server or transport failure requires restart. | Claude push relies on the development-channels flag; participant identity is not tenant isolation.
Open the evidence-backed case study

MCP product interface

Agents interact through MCP tools; the human uses CLI commands for status, agenda, plan approval, decisions, and delivery acceptance. The architecture above represents the implemented local service, not a simulated application screen.

Verification scope

Fresh MCP handshake, 19 tools, channel push, conformance, 80 unit tests, and both scripted lead-mode CLI workflows passed. Historical real-Codex plus simulated-Claude checks ended at awaiting acceptance; those real-agent checks were not rerun for this release. No productivity or business outcome metric is claimed.

Project A.R.C.H.E.R.

Status: Recovery Candidate - Empty Remote Repository

An authenticated empty remote with a matching local recovery candidate that has not passed restoration, provenance, secret, scope, or publication review.

Business problem

Potential source must be recovered without treating an empty remote or an unvalidated local candidate as evidence of a working application.

Role and evidence scope

Portfolio role scope: recovery-governance framing and approval-gate documentation only. No implementation role, application capability, ownership, or deployment claim is made.

Verified technologies

Withheld - the recovery candidate is not a validated implementation.

EVIDENCE / RECOVERY WORKFLOW - NOT APPLICATION UI

01 Evidence record: confirm the authenticated empty remote	02 Candidate review: reconcile local evidence without treating it as product proof	03 Approval gate: complete secret, provenance, and intended-scope review	04 Decision: restore only verified source or document non-restoration honestly
--	--	--	--

Recovery review workflow - not application functionality

1. Confirm the remote condition with authenticated evidence | 2. Reconcile candidate history and working-state evidence | 3. Review secrets, provenance, intended scope, and bootstrap safety | 4. Restore only approved evidence through review, or document why restoration is withheld

AI / automation boundary

No AI function, implementation, architecture, or deployed behavior is claimed for the recovery candidate.

Security and reliability controls

Owner approval before any restoration | Secret and provenance review before staging source | No screenshot or application-interface claim

Current limitations

The remote repository is empty. | Feature, technology, architecture, screenshot, ownership, and deployment claims remain withheld.
Open the evidence-backed case study

EVIDENCE FOUND	Authenticated empty remote plus a matching local recovery candidate.
NOT VALIDATED	Implementation, technology, architecture, screenshot, ownership, and deployment.
APPROVAL GATE	Secret, provenance, intended-scope, and restoration review before any source action.
SAFE RESULT	Restore only verified evidence through review, or document why restoration remains withheld.

Evidence diagram only - not a product screenshot, design mockup, deployed interface, or implementation proof.