

Benjamin Stone

AI TRANSFORMATION & OPERATIONS LEADER | AGENTIC SYSTEMS | BUSINESS AUTOMATION ARCHITECT

Lehi, UT • benstone2038@gmail.com • 385-225-1199 • linkedin.com/in/benjamin-stone-lehi • benstone-ai.com • [View Verified Work →](#)

Finance and AI systems operator combining 10 years of Controller-level finance leadership with hands-on experience designing autonomous agents, governed workflows, integrations, verification systems, and production AI applications.

FINANCE -> OPERATIONS -> AI SYSTEMS -> AGENT ARCHITECTURE -> AUTOMATION -> VERIFICATION

● **FOCUS AREAS (AI TRANSFORMATION)**

Finance Systems, Controls & Audit-Ready Verification

SwarmSync.ai — Multi-Agent Marketplace

AI Platforms & Product Delivery — Cato + ConduitScore.com

● **SELECTED WORK & VERIFIABLE CLAIMS**

- Engineered InvoiceProof, AuditProof, VerifyAPI, and Conduit/Halo controls — a verification stack using hash-chained logs, cryptographic proofs, and approval gates to produce independently auditable action records for automated financial and agent-driven actions. [FOS-001]
- Architected SwarmSync.ai, an agent-to-agent marketplace enabling autonomous discovery, multi-agent negotiation, escrow settlement, and reputation scoring under production governance controls. [SS-001]
- Built Cato, a desktop AI operating system with multi-model orchestration and tool calling. [CS-001]
- Built and shipped ConduitScore.com, a live SaaS platform that diagnoses an organization's AI readiness and produces a scored remediation plan. [CS-002]

EXECUTIVE OPERATOR BRIEF • ARCHITECTURE & VERIFICATION

Turning Multi-Entity Finance Operations into an AI-Operated System

BUSINESS PROBLEM

- Manual month-end close and AP/AR/inventory/payroll processes consumed excess time and lacked standardized controls
- Automated financial and agent-driven actions lacked an independently auditable record
- No standardized way to prove an automated decision was authorized and unaltered after the fact

CONSTRAINTS

- Automated financial and agent-driven actions needed to be independently auditable without slowing the finance team down.
- Consequential actions (payments, GL postings) could not be allowed to execute without a human approval step — a wrong automated action in finance is expensive and hard to unwind.
- The system had to operate across already-fragmented business systems, accounting systems, and reporting tools rather than replacing them outright.

DECISION

Build a governed operating layer — not another point-automation tool — with an orchestration layer (Cato) that routes work to narrow specialist agents (Genesis), backed by a hash-chained, cryptographically verifiable audit trail (InvoiceProof/AuditProof/VerifyAPI/Conduit-Halo) and a mandatory human-approval gate before any consequential action executes.

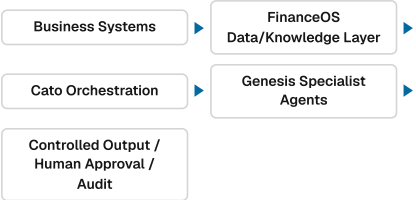
ALTERNATIVES REJECTED

One large general-purpose AI model handling the entire finance process end to end. — Harder to audit, harder to correct when it is wrong in one narrow area, and does not map to how finance teams already divide responsibility (AP, AR, close, treasury).

Traditional single-task automation (e.g. one RPA script per workflow). — Solves individual tasks but does not create a coordinated operating layer across the whole finance function, so the underlying fragmentation problem remains.

Fully autonomous execution with no human approval step. — Consequential financial actions carry too much downside risk to run unsupervised; the design keeps a human approval gate for anything beyond preparation and drafting.

SYSTEM



WHAT BEN PERSONALLY DID

Designed: The five-layer architecture: Business Systems → FinanceOS Data/Knowledge Layer → Cato Orchestration → Genesis Specialist Agents → Controlled Output/Human Approval/Audit.; The hash-chained, cryptographic audit-trail approach (InvoiceProof, AuditProof, VerifyAPI, Conduit-Halo) so any automated action stays independently checkable after the fact.

Decided: That consequential actions require an explicit human approval gate rather than full autonomy.; That specialist agents (Genesis) should be scoped narrowly per finance function (AP, AR, close, treasury) instead of one general-purpose model.

Configured: Approval-gate thresholds and routing rules connecting Cato's orchestration layer to the underlying accounting structure.

Coded: The verification stack itself — InvoiceProof, AuditProof, VerifyAPI, and Conduit-Halo — including the hash-chaining and cryptographic proof logic.

Tested: Verified that tampering with a logged action after the fact is independently detectable via the hash chain.

Operated: Ran the underlying finance operation this architecture generalizes from directly, as Controller of a ~50-person, multi-million-dollar business for 10 years (Bella Lash Extensions, 11/2011-11/2021), including full-cycle accounting and a 3-day month-end close.

RESULT

- Gave automated finance and agent-driven actions an independently auditable, tamper-evident record instead of an unverifiable log line. [FOS-001]
- Consistently delivered financial statements, reconciliations, accruals, and supporting schedules within a 3-day close window for a ~50-person, multi-million-dollar organization. [FOS-002]

EVIDENCE

- [FOS-001-EV1] **DOCUMENT:** Internal repository/documentation for the InvoiceProof/AuditProof/VerifyAPI/Conduit-Halo verification stack (access restricted; architecture described in the operator brief) — no independent public artifact yet
- [FOS-002-EV1] **DOCUMENT:** Professional employment history — Controller, Bella Lash Extensions, Vineyard, UT (11/2011-11/2021), as stated on Benjamin Stone's resume; no independent third-party artifact available

WHAT FAILED OR REMAINS INCOMPLETE

- The InvoiceProof/AuditProof/VerifyAPI/Conduit-Halo verification stack has no independent, third-party-hosted evidence artifact yet — it is SELF-REPORTED with only an internal, access-restricted document as evidence, not a public URL or repo a reader can check directly. A public reference implementation or third-party audit is the remaining gap.

• BUSINESS PROBLEM

- No standard way for AI agents to autonomously discover each other, negotiate, and settle payment with trust and accountability
- Agent-driven marketplaces lacked the reputation and governance controls needed for production use

• CONSTRAINTS

- Autonomous agents needed a way to prove identity and authorization before a transaction, not just after the fact.
- The marketplace had to run under production governance controls, not exist only as a research prototype.
- Verification and audit records needed to be portable and checkable by parties outside SwarmSync itself, not just an internal log line.

• DECISION

Build SwarmSync as the trust and transaction layer for autonomous agents — a governed rail that checks identity/authorization before an action, generates evidence, performs verification, and produces a signed proof receipt for every transaction — rather than a plain marketplace with no verification layer.

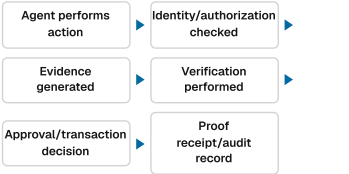
• ALTERNATIVES REJECTED

A plain agent marketplace with no built-in identity/authorization or verification layer. — Leaves businesses with no way to know whether an agent's output can be trusted or whether money/data should actually be released — the core problem the marketplace exists to solve.

Centralized manual review of every agent transaction. — Does not scale to an autonomous, agent-to-agent marketplace and reintroduces the human bottleneck the system is meant to remove.

Relying on proprietary, single-vendor protocols instead of open standards. — Would fragment the agent ecosystem instead of letting different vendors' agents interoperate — the reason this work also produced open IETF drafts and W3C AIVS participation.

• SYSTEM



• WHAT BEN PERSONALLY DID

Designed: The end-to-end trust/transaction flow: agent action → identity/authorization check → evidence generation → verification → approval/transaction decision -> proof receipt;; ProofRail as the governed transaction rail underneath the marketplace, recording what happened, why it was permitted, what provider responded, and what proof exists afterward.

Decided: To require identity/authorization checks and evidence generation before any transaction is allowed to settle.; To pursue open standards (IETF, W3C) alongside the product so the trust model is not locked to one vendor.

Coded: The discovery, negotiation, escrow-settlement, and reputation-scoring logic underlying the live marketplace at swarmsync.ai.

Tested: Verified the marketplace's discovery, negotiation, escrow, and reputation-scoring flows in production at swarmsync.ai.

Operated: Runs swarmsync.ai as Founder & Principal Systems Architect since 12/2021.

• RESULT

- Delivered a live, publicly reachable multi-agent marketplace at swarmsync.ai covering discovery, negotiation, escrow, and reputation scoring.

[SS-001]

• EVIDENCE

- [SS-001-EV1]

PUBLIC_URL: https://swarmsync.ai (production deployment demonstrating the multi-agent marketplace: discovery, negotiation, escrow, and reputation scoring)

• WHAT FAILED OR REMAINS INCOMPLETE

- No independent, formal third-party security audit or penetration test of the marketplace has been published yet — production usage and the public deployment at swarmsync.ai are the evidence available today, not an external audit report.

Taking an AI Verification Product from Architecture to a Running Commercial System

BUSINESS PROBLEM

- No single desktop environment let a user orchestrate multiple AI models and tools together
- Organizations had no standardized way to assess their AI readiness before investing in automation

CONSTRAINTS

- Needed to prove instructions, meaning, and behavior survive across different models, agents, and transformations — not just work once on one model.
- Had to turn a technical verification concept into something a non-technical buyer could actually act on (a score and a plan), not just a research finding.
- Two different delivery shapes were needed at once: a desktop AI operating environment (Cato) and a public, self-serve SaaS diagnostic (ConduitScore.com).

DECISION

Ship two connected products: Cato, a desktop AI operating system with multi-model orchestration and tool calling, and ConduitScore.com, a public SaaS diagnostic that scans, analyzes, and scores an organization's AI readiness and turns the result into a signed/verifiable score plus a recommended fix — rather than keeping the verification logic as an internal-only tool.

ALTERNATIVES REJECTED

Keep the verification/scoring logic as an internal tool only. — Would prove the technical concept but not that it ships as a commercial product a customer can actually buy and use — the goal was to prove product delivery, not just internal automation.

Build Cato as a single-model tool instead of multi-model orchestration. — Users needed to coordinate multiple AI models and tools from one place instead of juggling a separate app per model, which a single-model tool cannot do.

A one-time manual AI-readiness assessment delivered as a consulting report. — Does not scale, is not repeatable, and cannot produce a consistent signed/verifiable score the way an automated scan-analyze-score pipeline can.

SYSTEM



WHAT BEN PERSONALLY DID

Designed: The Scan → Analyze → Score → Signed/Verifiable Result → Recommended Fix → Commercial Action loop that ConduitScore.com runs; Cato's multi-model orchestration and tool-calling architecture.

Decided: To ship ConduitScore.com as a public, self-serve SaaS product rather than a bespoke internal assessment.

Coded: Cato's orchestration and tool-calling engine and the ConduitScore.com diagnostic scoring pipeline.

Tested: Verified ConduitScore.com's scan/analyze/score pipeline against real organizational inputs before public launch.

Operated: Runs ConduitScore.com as a live, publicly reachable SaaS product today.

RESULT

- Delivered Cato as a working desktop AI operating system with multi-model orchestration and tool calling. [CS-001]
- Delivered a publicly accessible AI-readiness diagnostic and remediation SaaS product at conduitscore.com. [CS-002]

EVIDENCE

- [CS-001-EV1] **DOCUMENT:** Internal product documentation/repository for Cato (desktop AI OS) — access restricted; no independent public artifact yet
- [CS-002-EV1] **PUBLIC_URL:** https://conduitscore.com (live AI-readiness diagnostic and remediation SaaS platform)

WHAT FAILED OR REMAINS INCOMPLETE

- Cato remains SELF-REPORTED with only an internal, access-restricted document as evidence — no public artifact exists yet. ConduitScore.com is live and publicly reachable, but no usage or conversion metric for it has been added to career-proof.json yet, so no quantified commercial result can be shown here until that data exists and clears the safe_to_publish gate.