Damien House - Principal AI Platform Architect & Fractional CTO - Capability Statement

Introduction

Enterprise AI agent platforms do not fail because organizations lack capable models. They fail because they lack the governed substrate those models need to operate safely: controlled context surfaces, enforced tool boundaries, tenant-isolated memory, auditable authorization paths, and human-in-the-loop checkpoints for privacy, security, compliance defensibility, and end-user safety. Without explicit context management and operating controls, AI accelerates data exposure, technical debt, and systemic risk.

That substrate is the problem Damien House solves.

As a principal AI platform architect, fractional CTO, and senior engineering advisor, Damien helps organizations move from fragile AI prototypes to production-ready systems that can be governed, secured, and scaled. He brings 20+ years of platform modernization, DevSecOps, Drupal architecture, and enterprise delivery together with active R&D in Agent Experience (AX), Model Context Protocol (MCP), context engineering, retrieval-backed systems, and agentic workflow governance.

His work centers on the operating problems that appear after the demo: multi-step workflows, tool routing, session state, failure recovery, retrieval quality, governance, observability, bounded autonomy, PII/PHI boundaries, and regulatory compliance. His R&D anchor, the **Context Control Center (CCC)**, treats the LLM context window as a governed enterprise resource with memory tiers, tenant boundaries, RBAC/ABAC policy enforcement, MCP governance, and audit trails.

The throughline is context control. Enterprise access rules, delivery governance, runbooks, and change controls must now apply one layer higher in the AI stack: managed context windows, explicit tool boundaries, namespace and tenant isolation, skill-level security review, PII/PHI-aware retrieval, and human-in-the-loop checkpoints.

For technical leaders, Damien brings staff-level judgment across multi-tenant services, MCP interoperability, harness engineering, skill governance, retrieval-backed systems, DevSecOps, and platform delivery. For product and business stakeholders, he creates a practical path from experimentation to production across FinTech, HealthTech, GovTech, Enterprise SaaS, and regulated sectors.

Services Overview

- **Fractional CTO & Technical Strategy:** Provide technical leadership for founders, product teams, and growth-stage organizations that need more than tactical development. Translate product ambition into architecture, sequencing, governance, and risk-reduction decisions without accumulating the governance debt that stalls AI initiatives in production.
- **AI Agent Platform Strategy & Technical Direction:** Define architecture for internal copilots, virtual assistants, multi-agent workflows, and AI-enabled tools. Identify platform primitives for safe execution: agent identity, tenant context, tool permissions, session state, evaluation paths, workflow checkpoints, and audit trails.
- **Prototype-to-Production Architecture:** Turn early concepts, AI-generated mockups, and fragile prototypes into versioned specifications, schema contracts, roadmap decisions, and production-ready designs using artifact-first execution and AI-augmented delivery while preserving security review, accessibility, and delivery discipline.
- **Context Control Center (CCC) Architecture:** Govern what knowledge, memory, tools, and system state each agent persona can access using permissioned retrieval, context compaction, memory hierarchies, review workflows, auditable context assembly, and policy-as-code enforcement.
- **MCP Tool Exposure & A2A Interoperability:** Define MCP-compatible tool surfaces, typed schema contracts, server registry patterns, auth boundaries, scoped credentials, A2A delegation models, and usage telemetry so agent capabilities compose without brittle point-to-point integrations.
- **Agent Skill Ecosystem & Marketplace Governance:** Architect reusable AI skill packages with ownership metadata, versioning, approvals, progressive disclosure loading, injection and side-effect vetting, usage analytics, and deprecation paths so teams can extend capabilities without prompt-library sprawl.
- **Harness Engineering, Evaluation & Compliance Defensibility:** Design harnesses for tool dispatch, session state, scratchpad isolation, retries, error recovery, workflow checkpoints, and escalation paths. Pair runtime controls with LLM-as-judge review, computational linters, schema validation, architecture constraints, and feedback sensors.
- **AI-Enabled Content Operations & Governance:** Apply AI to content estates where manual audits, accessibility remediation, metadata generation, migration planning, and editorial QA do not scale, preserving provenance and human review where quality, judgment, or compliance matter.
- **Engineering Leadership, DevEx & Platform Modernization:** Modernize infrastructure, standardize environments, improve CI/CD and documentation, strengthen security and code-quality workflows, and leave teams with safer releases and stronger foundations for agentic automation.

Methods and Tools

- **Design Science + Artifact-First Execution:** Clarify the operational problem, user consequence, and success condition before implementation. Each cycle produces working software plus durable context: schema definitions, policy-as-code constraints, interoperability protocols, architecture decisions, and operating assumptions.

- **Agent Experience (AX):** Treat AI agents as operational participants needing explicit governance, not magical autonomy. Design action surfaces, permission models, governance boundaries, and context controls so agents assist without improvising against hidden state, sensitive data, production systems, or ambiguous business rules.
- **Context Control & Namespace Isolation:** Apply identity-first, namespace-isolated governance to the AI context layer. Context provisioning becomes controlled infrastructure with tenant-aware memory, source attribution, permissioned retrieval, review workflows, and auditable assembly.
- **MCP-Native Tool Governance:** Treat tool exposure as typed, auth-bounded, registry-governed infrastructure. Every tool needs a schema contract, authorization policy, usage telemetry path, review model, operational owner, and registry entry.
- **RBAC/ABAC Tool-Calling Governance:** Map enterprise identity, roles, attributes, content permissions, workflow state, and tool scopes into enforceable authorization rules that reduce prompt injection, privilege escalation, unauthorized data exposure, inappropriate autonomous action, and weak compliance defensibility.
- **Deterministic + Inferential Controls:** Combine schema validation, policy-as-code, architecture linters, test execution, static analysis, and release gates with LLM-as-judge evaluation, semantic similarity checks, answer-grounding review, and human approvals.
- **Retrieval, Context Compaction & Organizational Memory:** Design RAG pipelines, modality-aware chunking, source-grounded retrieval, provenance tracking, memory hierarchies, summarization, and state isolation that convert enterprise knowledge into reliable agent input.
- **Cloud-Native Delivery, DevEx & DevSecOps:** Improve delivery systems through standardized environments, CI/CD, automated testing, AI-assisted code review, dependency scanning, security checks, release documentation, and operational runbooks.
- **Human-in-the-Loop Governance:** Build escalation paths, approval gates, and accountability into workflows from the start, preserving human review where quality, privacy, compliance, judgment, or end-user safety matter.

Sample Project Types

- **Context Control Center (CCC):** Governed context-management architecture for tenant-aware memory hierarchies, context compaction, source attribution, permissioned retrieval, RBAC/ABAC policy enforcement, MCP tool governance, review workflows, and auditable context assembly across teams with different access policies.
- **Semantic Search / RAG Evaluation Framework:** Benchmark retrieval quality across heterogeneous enterprise knowledge by testing chunking strategies, embedding models, and vector, hybrid, or graph-augmented retrieval against domain-specific quality metrics.
- **Multi-Tenant Agent Platforms & Delivery Systems:** Backend services, workflow controls, deployment systems, and operating boundaries that separate agent identity, tenant context, memory, tool permissions, execution state, and escalation paths.
- **MCP-Ready Architecture & Skill Exposure:** Authenticated, schema-validated MCP-compatible interfaces for exposing structured content, reusable skills, and business actions through governed tool ecosystems, skill marketplaces, and A2A workflows.
- **Governed Agentic Workflow Prototypes:** Agentic patterns across content operations, engineering delivery, and workflow automation, including AI-driven content governance that expands audits from 50-page samples to 6,000+ item estates and reduces audit cost to roughly $400 in API compute while preserving human review and compliance traceability.
- **Enterprise Platform Modernization for AI Readiness:** Cloud migrations, containerized delivery platforms, CI/CD standardization, enterprise SSO, structured content models, governed access controls, reproducible deployments, and release practices that compress 3-to-6 month migrations into 3-to-4 week cycles.
- **Developer Tooling, Content Intelligence & Reusable Patterns:** AI-assisted code review, visual regression workflows, content audit automation, metadata generation, accessibility support, deployment scaffolds, release checklists, and documentation systems that reduce repetitive toil while keeping humans in control.
- **Cross-Disciplinary Product R&D:** Exploratory product and platform work that pairs design science, architecture strategy, and AI-assisted prototyping to test concepts quickly, gather evidence, and separate novelty from operational value.

About Damien House

Damien House is the Principal and Founder of 109.studio, based in Washington, DC. He has spent more than two decades delivering complex technical outcomes across startups, agencies, public-sector programs, and enterprise environments where scale, uptime, governance, privacy, and accessibility are non-negotiable. His work has modernized mission-critical public-sector platforms for the U.S. Department of Agriculture and the International Trade Administration, introduced early containerized delivery practices, and guided founders through architecture decisions that turn prototypes into governed production systems.

Damien combines executive-level technical judgment with direct implementation depth. He defines long-range architecture, challenges fuzzy requirements, tightens operating models with VP-level stakeholders, then works inside implementation: designing harness patterns, shaping RAG evaluation strategies, establishing platform boundaries, improving delivery pipelines, defining content architectures, and creating the artifacts that keep teams aligned.

Damien approaches problems as systems: architecture, operators, workflows, security models, data boundaries, consequences, and post-launch decisions. His view is practical and grounded: the hardest problems in enterprise AI are rarely model problems alone. They are platform, governance, and operating-model problems that require clear boundaries, reliable context, measurable quality, and accountable human oversight. In high-trust environments, Damien helps teams build AI systems that can be reviewed, explained, improved, and operated with confidence.