

WILLIAM L. HOOD

SENIOR ENTERPRISE AI ENGINEER | AGENTIC SYSTEMS | CLOUD-NATIVE ARCHITECTURE | AI GOVERNANCE

Taos, New Mexico | williamlhood@gmail.com | Remote within the United States

PROFESSIONAL SUMMARY

Senior enterprise AI engineer, software architect, consultant, educator, and technology entrepreneur with more than 40 years of experience spanning application development, databases, APIs, distributed infrastructure, cloud platforms, customer delivery, and artificial intelligence.

Designs and implements AI applications as complete operational systems—connecting models with enterprise data, APIs, retrieval, workflows, security boundaries, evaluation, human review, and evidence stakeholders can inspect.

Hands-on practitioner with OpenAI, Anthropic, Codex, and Claude across codebase discovery, architecture, implementation, integration, testing, troubleshooting, documentation, and Git-based review. Combines extensive Docker and Kubernetes design and implementation experience with AWS-native engineering, serverless systems, MCP integrations, relational and NoSQL databases, and practical AI governance.

Career-long systems perspective enables work across application, data, infrastructure, networking, security, cloud, and operational boundaries rather than treating AI as an isolated application layer.

CORE CAPABILITIES

Enterprise AI Engineering

- Enterprise AI application and platform architecture
- OpenAI and Anthropic models and APIs
- OpenAI Codex and Claude agentic development
- Agentic workflows and multi-specialist orchestration
- MCP servers, tools, and integrations
- RAG, vector knowledge, and governed retrieval

- Natural-language-to-SQL analysis
- Evidence-bounded reporting and source attribution

Cloud-Native Engineering

- AWS architecture and implementation
- Lambda, Boto3, EC2, S3, Glue, Athena, DynamoDB
- Serverless and event-driven systems
- Docker and Kubernetes architecture and implementation
- Linux-based systems
- Cloud resource discovery and configuration analysis
- Portable and scalable service architectures

Software, Data & Integration

- Python
- REST APIs
- SQL and relational database systems
- NoSQL architectures
- Managed relational services
- Data migration and modernization
- Transactions and embedded business rules
- Structured and unstructured enterprise data

AI Governance & Engineering Discipline

- Human-in-the-loop controls
- Guardrails and bounded tool access
- Evaluation and verification
- Evidence contracts
- Source attribution
- Approval and confirmation workflows
- CI/CD and Git/GitHub workflows
- Testing, observability, and operational evidence
- Repository analysis and bounded implementation planning

Consulting & Technical Enablement

- Enterprise discovery
- Solution architecture
- Customer prototyping
- Proofs of concept
- Technical demonstrations
- Stakeholder communication

- Technical curriculum development
- AI-assisted engineering enablement

PROFESSIONAL EXPERIENCE

ST21

Enterprise AI Architecture & Engineering 2025 & 2026

Designed and implemented enterprise AI capabilities connecting governed reasoning workflows with structured data, documents, cloud resources, APIs, and operational systems.

- Built ingestion and discovery workflows for text, PDF, CSV, and mixed enterprise data, including file inventory, schema discovery, column identification, table resolution, and source-bounded analysis.
- Implemented governed retrieval and natural-language-to-SQL analysis with saved result sets, evidence contracts, row limits, source attribution, and neutral reporting controls.
- Designed specialist AI workflows integrating cloud resources, service-management platforms, issue tracking, collaboration repositories, network-security systems, and secure web-access services.
- Established draft, review, confirmation, approval, and human-authorization boundaries for AI-assisted system actions.
- Designed and implemented MCP servers, APIs, and enterprise integrations exposing bounded tools, data, and operational capabilities to AI workflows.
- Applied OpenAI and Anthropic models, APIs, Codex, and Claude to architecture, implementation, evaluation, integration, and governed enterprise application development.
- Used DynamoDB extensively for AWS-native application state, metadata, workflow coordination, and enterprise AI data requirements.
- Integrated S3, Glue, Athena, Lambda, identity services, retrieval services, and infrastructure metadata into enterprise AI workflows.
- Implemented AWS resource discovery and configuration-drift analysis using baseline comparison, decision capture, human approval, and controlled remediation paths.
- Applied extensive Docker and Kubernetes architecture and implementation experience to portable, scalable application and specialist-service designs.
- Established repeatable engineering practices covering repository analysis, scoped planning, implementation, testing, deployment, verification, observability, and auditable evidence.

inPro.ai

AI Solutions Architect / Consultant | 2019–Present

Architect AI and automation solutions combining large language models, enterprise information, databases, APIs, workflow orchestration, and cloud services.

- Design conversational AI, RAG, agentic workflows, knowledge systems, and AI-enabled business processes.
- Apply OpenAI and Anthropic models and APIs to enterprise applications, integrations, evaluations, prototypes, and customer demonstrations.
- Translate business and technical requirements into working prototypes, proofs of concept, and production-oriented architectures.
- Design systems connecting AI with enterprise data, APIs, databases, cloud resources, and workflow platforms.
- Guide operational transformation by connecting business objectives with process improvement, architecture decisions, governance controls, and implementation priorities.
- Develop practical AI governance approaches preserving human authority, bounded use, evidence, accountability, risk controls, and appropriate review.
- Work directly with stakeholders to turn emerging AI capabilities into understandable and implementable solutions.

Sahara

AI Application Architect / Developer

Develop an AI-enabled software application using OpenAI Codex and Claude as primary engineering partners.

- Use Codex and Claude for codebase understanding, architecture discovery, implementation planning, coding, debugging, testing, and iterative refinement.
- Convert business ideas and user requirements into functional application capabilities through human-directed agentic engineering.
- Define implementation scope, constraints, and acceptance expectations before directing agent work.
- Evaluate proposed solutions, inspect changes, test behavior, and retain engineering accountability for accepted output.
- Use Git-based workflows and implementation evidence rather than accepting generated code without review.
- Developed a repeatable model for AI-assisted application engineering, customer prototyping, and technical enablement.

SELECTED ENGINEERING EXPERIENCE

Network Partners, Inc.

Founder / Principal | 14 Years

Founded and led a technology consulting company delivering software, networking, wireless, security, infrastructure, and cloud-transition solutions.

- Delivered thousands of technology projects spanning applications, servers, databases, networks, wireless systems, security, infrastructure, and cloud environments.
- Owned customer outcomes from discovery and requirements through architecture, implementation, troubleshooting, documentation, support, and long-term operation.
- Designed secure LAN, WAN, WLAN, VoIP, and highly available environments for commercial and federal customers.
- Developed broad systems-level troubleshooting experience across application, data, infrastructure, networking, and security boundaries.

SoftNAS

Co-Founder / Technical Lead

Co-founded and helped engineer a commercial Linux/ZFS virtual NAS platform for cloud environments.

- Designed a virtual storage architecture aggregating AWS EBS volumes into RAID-based storage systems.
- Worked across Linux, ZFS, storage, networking, databases, web applications, deployment, infrastructure, and customer support.
- Helped transform complex infrastructure capabilities into an installable and supportable software product.
- Participated in deployment through AWS Marketplace.

Manufacturing Systems Engineering

Designed a SQL-based manufacturing system supporting:

- Tool-and-die operations
- Production planning
- Inventory forecasting
- Operational visibility
- Relational business data
- Embedded operational rules

AWS Genomic Compute

Designed an AWS-based genomic-compute utility using:

- SNS
- Lambda
- Python

- Boto3
- Dynamically provisioned EC2 resources
- Long-running compute workloads

Designed the system to coordinate event-driven requests with dynamically created compute capacity for jobs inappropriate for short-duration serverless execution.

DATABASE & DATA ENGINEERING EXPERIENCE

Database experience spans systems predating widespread relational database adoption through modern relational, managed, cloud-native, and NoSQL architectures.

Experience includes:

- SQL
- Relational database design
- Transactions
- Data migration
- Embedded business rules
- Application/database integration
- Managed relational database services
- NoSQL architecture
- DynamoDB
- Data modernization
- Structured and semi-structured enterprise information
- Retrieval and analytical workflows

AGENTIC AI ENGINEERING

Hands-on experience using OpenAI Codex and Claude as active engineering agents rather than simple code-generation utilities.

Typical workflow:

Understand → Specify → Plan → Implement → Test → Review → Verify

Use AI agents to:

- Explore unfamiliar repositories
- Identify architecture and dependencies
- Investigate existing implementation patterns
- Decompose requirements into bounded work
- Produce implementation plans
- Modify coordinated sets of files

- Build integrations
- Diagnose errors
- Run tests
- Review implementation evidence
- Produce documentation

The engineer retains responsibility for requirements, architecture, constraints, security, testing, acceptance, and production outcomes.

AI GOVERNANCE

Design AI systems around explicit human authority and inspectable evidence.

Governance approaches include:

- Bounded tool access
- Role and permission controls
- Source attribution
- Evidence requirements
- Human review
- Confirmation before consequential actions
- Controlled remediation
- Neutral reporting
- Evaluation
- Auditability
- Reproducibility
- Decision capture
- Separation of analysis from action

The goal is not simply to make AI capable of taking action, but to make its actions explainable, reviewable, authorized, and operationally safe.

EDUCATION & TECHNICAL ENABLEMENT

University of New Mexico–Taos

Adjunct Instructor

Designed and taught courses including:

- Computer Science
- Networking
- Data and Databases
- Python Programming

- Cloud-related technologies

Created practical curricula emphasizing system-level understanding and real-world application.

New Apprenticeship

AWS / Python Instructor

Taught AWS architecture and Python to a professional apprenticeship cohort through hands-on engineering and practical application.

Developed technical instruction designed to make complex systems understandable and usable.

AI-Assisted Engineering Enablement

Develop repeatable methods for teaching professionals how to use Codex and other AI engineering agents while preserving:

- Human judgment
- Engineering accountability
- Technical ownership
- Verification
- System understanding
- Responsible use of automation

EDUCATION

Bachelor of Science, Mathematics

University of Texas at Arlington | 1985

Minor in Computer Science

TECHNICAL ENVIRONMENT

AI & Models: OpenAI, Anthropic, Codex, Claude, ChatGPT, Gemini, Amazon Bedrock

Agentic Systems: MCP, multi-agent workflows, specialist orchestration, tool integration, RAG, governed retrieval

Development: Python, JavaScript, PHP, HTML/CSS, SQL, REST APIs

AWS: EC2, Lambda, S3, SNS, Glue, Athena, DynamoDB, Boto3

Containers & Platforms: Docker, Kubernetes, Linux

Data: SQL, relational databases, managed relational services, DynamoDB, NoSQL, structured and unstructured data

Engineering: Git, GitHub, CI/CD, repository analysis, testing, troubleshooting, observability, documentation

Infrastructure: LAN, WAN, WLAN, VoIP, secure networking, high availability, cloud architecture

PROFESSIONAL THROUGH-LINE

Programming since 1980 and a technology professional since 1982.

Across more than four decades, the technologies have continually changed, but the engineering work has remained consistent:

Understand the system. Find the real problem. Design the path. Build it. Test it. Verify it. Explain it. Own the outcome.

Agentic AI is the latest—and potentially most powerful—extension of that engineering discipline.