

Andy Widjaja

Chicago, IL | andywidjaja@gmail.com | andywidjaja.com | linkedin.com/in/andy-widjaja-aws

Summary

Forward deployed tech lead who owns end-to-end AI deployments, from first prototype to stable production. Most recently set the technical strategy for a Fortune 50 health insurer's AI SDLC transformation (8,000–9,000 builders), owning the architecture and the quality bar for work delivered with a global SI partner and the customer's engineering teams. At AWS, built and shipped Delivery Agent (5,000+ global active users), led its 3.0 redesign and rewrite (3–7x faster), and published an open-source MCP server ([170,000+ total downloads](#)). Builds full-stack systems in Python, TypeScript, and Java, and measures agent quality with evals (ATLAS, accepted at AMLC 2026). Has presented to AWS VPs and to customer CIOs and VPs. Background spans sub-millisecond trading gateways to an 800M-document migration from on-prem to cloud. Builds daily with AI coding agents (Codex, Claude Code), and created Patina, an open-source cognitive agent framework.

Experience

AWS · AI Architect, Applied AI (Forward Deployed Engineering) | 2022–Present

Agent Infrastructure (2025–2026):

- Built and shipped Delivery Agent, a multi-step agent orchestration system for software delivery workflows (Python, TypeScript, MCP protocol, LLM reasoning). Scaled from 0 to 5,000+ global active users (90%+ org adoption) across 6 regions within 8 months and led LATAM and APJC technical roadshows (10 countries, 900+ engineers) to drive adoption from builders
- Led the redesign and rewrite of Delivery Agent 3.0's design engine and mobilized the team behind the 3.0 relaunch: modular architecture across five agent harnesses (Kiro IDE, Kiro CLI, Kiro Crew, Claude Code, Codex), cutting design runs from ~90 min. to 12–21 min. (3–7x faster). Built a dependency-graph scheduler that starts each parallel sub-agent stage as soon as its inputs are ready (nearly 2x faster on Kiro Crew), scoped agent tool permissions to trusted commands, integrated and reviewed five teammates' features into one engine, and closed the release security review's required findings
- Tech lead for a Fortune 50 health insurer's AI SDLC transformation (multi-million-dollar engagement) for 8,000–9,000 builders globally: led discovery with customer engineering leadership, set the technical strategy and a platform-agnostic agent orchestration architecture (18 ADRs) across thousands of repositories, set the quality bar for work delivered with a global SI partner, and escalated a scope and role misalignment across the customer, the SI partner, and AWS to leadership; the engagement was extended through Q1 2027
- Co-authored ATLAS, an automated evaluation framework measuring agent output quality via precision/recall/F1. Deployed as CI/CD quality gate across 2,500+ concurrent users. Peer-reviewed paper accepted to the AMLC 2026 Gen-AI Autonomous Agents Workshop
- Published [document-loader MCP server](#) (awsllabs, open source): structured content extraction from PDF/DOCX/XLSX/PPTX/images. [170,000+ total downloads](#). Shipped a path-containment security fix hardening file access against traversal
- Designed agent migration architecture for a global financial data provider: LangGraph agents and MCP tools to AgentCore Runtime with state migration strategy
- Co-authored [AWS Machine Learning Blog post](#) on event-driven ambient agents with human approval (AgentCore)

Enterprise AI Systems (2022–2025):

- Shipped AI-powered document intelligence platform for a global insurer: 800M documents migrated on ECS Fargate with per-document SHA-256 validation and self-healing retries, 165K daily ingestion, multi-region (NAMER/EMEA/APAC). OpenSearch + Textract pipeline with sensitive payment detection ([published on the AWS blog](#))
- Developed production RAG system for a financial market infrastructure provider (public-facing): LLM on Amazon Bedrock, cross-region inference, 60+ concurrent users, financial regulatory compliance
- Built authorization system for a global fintech using Amazon Verified Permissions: 30x cost reduction, multi-tenant RBAC/ABAC

Legal & General Investment Management America · Full Stack Developer | 2018–2022

- Built serverless event-driven portfolio performance calculation platform for quantitative trading team (Lambda, ECS, Kafka)

Performance Trust Capital Partners · Lead Developer | 2013–2018

- Built proprietary bond analytics engines (CMOs, MBS, munis) and ML models for pricing/asset classification. Decomposed legacy monolith into 20+ microservices (Redis, RabbitMQ, Docker)

R.J. O'Brien & Associates · Trading Systems Developer | 2010–2013

- Built CME Globex trading gateway: 25,000 orders/day, 150,000 contracts, sub-second execution (FIX/FAST protocol). Custom routing gateway for Morgan Stanley

Earlier Career: Newedge/Société Générale, Dell, SEI

Open Source

Patina: Cognitive Agent Framework (2026) | github.com/Sanctum-Origo-Systems/patina

Cognitive agent framework implementing belief graphs, graduated autonomy, and persistent memory via MCP with three-layer architecture: messaging gateway, Claude Agent SDK runtime, and cognitive MCP server. 38,000+ lines Python, 1,500+ tests.

- Created belief extraction pipeline: LLM-based entity/relationship extraction from unstructured messages into a queryable graph
- Implemented and validated graduated autonomy: agent decision authority expands or shrinks per-domain from measured confidence, exercised against 40+ real decisions
- Implemented MCP server exposing 30+ tools for memory, prioritization, profiling, and introspection to maintain portability across different agentic clients

Autoloop: Governed AI Build Pipeline (2026) | github.com/Sanctum-Origo-Systems/autoloop

Config-driven pipeline that triages, decomposes, and implements GitHub issues autonomously under human oversight. One-command setup, language-agnostic, running in production.

- Built an observer that catches features passing CI but not working in production, auditing the builder's output against actual runtime behavior
- Built governed execution model: protected paths prevent the pipeline from modifying its own infrastructure or safety boundaries
- Processes issues one at a time in dependency order to avoid merge conflicts when implementing multiple issues autonomously
- `autoloop init` sets it up on any repo with existing CI, with no framework to adopt
- Measures its own output: 176 merged PRs from 213 autonomous implementation runs, 74% first-attempt success at \$1.88 per run, and 16% later edited by a human, tracked in a [published eval report](#) with metric definitions

Education, Publications & Speaking

BS Computer Science, University of Wisconsin-Madison

Publications: "[High-throughput enterprise-scale document migration on AWS](#)" (AWS Migration & Modernization Blog, 2026); "[Building ambient agents with Amazon Bedrock AgentCore](#)" (AWS Machine Learning Blog, 2026)

Speaking: re:Invent 2023–2025, LATAM Agentic AI Tour 2026 (4 countries), APJC Agentic AI Tour 2026 (6 countries)