Brian Sam-Bodden

Former Redis Principal Applied AI Engineer · AI Systems Builder & Researcher · Java Champion

Scottsdale, Arizona · bsbodden@integrallis.com · LinkedIn · GitHub · ORCID · Integrallis

PROFILE

I am a hands-on, polyglot engineer, architect, and researcher. I founded Integrallis, where I serve as Lead Engineer and researcher, previously served as Principal Applied AI Engineer at Redis, and am also an author, instructor, international speaker, and Java Champion. In 2026 I published three papers on coding-agent context, agent self-verification, and filter construction.
I currently build the infrastructure that supports AI applications: local and embedded inference runtimes, model packaging and distribution, vector indexing and retrieval, durable agent memory and planning, evaluation, and storage-engine internals. I aim to take on challenging, high-impact problems that require research, architecture, hands-on implementation, product judgment, and close customer collaboration—and deliver robust systems, libraries, and developer platforms.

EXPERIENCE

Integrallis Software — Founder & Chief Scientist

AI SYSTEMS RESEARCH, PRODUCT ENGINEERING, AND ENTERPRISE CONSULTING · 2002–PRESENT

- Architected Models, an in-process Java 25 inference stack for open-weight models with memory-mapped weights, portable and selectively native execution, tool calling, embeddings, streaming, and Spring AI/LangChain4j adapters.
- Built Vectors, an embedded JVM vector-search and persistence engine with HNSW, Vamana, IVF-Flat, IVF-PQ, eight quantizers, atomic mmap storage, hybrid retrieval, and framework integrations. Its vectorized distance calculations run **4.4–6.8× faster** than the non-vectorized implementation in reproducible benchmarks, speeding up the core operation behind similarity search.
- Created ModelJars and modeljars.org, a way to package, distribute, and run local AI models in Java applications through a curated catalog, JVM runtime, and native cross-platform CLI.
- Created MFCQI, an evidence-based software-quality index that combines complexity, maintainability, architecture, duplication, and security into a gaming-resistant score and CI quality gates. It ships for Java/Kotlin and Python with build plugins, reports, badges, and optional LLM recommendations.
- Built pleat, the Rust library that ships the technique from my ribbon-filter paper: homogeneous and RocksDB-shaped ribbon filters with sequential, pleated, and parallel builds that produce bit-identical filters, verified byte-for-byte against the reference C++ implementation.
- Built and led a long-running consulting practice for Fortune 500 organizations including Citadel, Walmart Labs, Visa, and JPMorgan Chase, working hands-on as architect, technical leader, advisor, and mentor.
- Built production decision systems well before GenAI: served as Decision Systems Architect at Citadel; led Visa's rules and Hadoop integration for about 150 million transactions per day; prototyped Walmart's grocery-checkout rules; and guided underwriting and quoting systems at FBL Financial and Nationwide.

Redis — Principal Applied AI Engineer

APPLIED AI ENGINEERING AND AI DEVELOPER ADVOCACY · 2021–2026

- Designed and shipped core infrastructure across LangGraph Redis, LangChain Redis, RedisVL, RedisVL for Java, Redis OpenAI Agents, and Redis Agent Memory—spanning durable agent state, sessions and memory, vector and hybrid retrieval, semantic caching and routing, streaming, and coordination across Python and the JVM.
- Worked directly with Fortune 10, Fortune 50, and Fortune 500 customers on RAG, multi-agent systems, agent memory, vector retrieval, and semantic caching, shaping architectures and moving AI initiatives from exploration to implementation.
- Created Redis OM Spring, bringing object mapping, repository queries, full-text search, aggregation, and Redis Stack document models to Spring developers. It became a widely adopted, actively maintained project with more than 650 GitHub stars.
- Authored more than 190 pull requests across the Redis and redis-developer GitHub organizations, shipping features, performance improvements, compatibility upgrades, security fixes, release automation, documentation, and extensive tests across Java and Python codebases.
- Served as an AI Developer Advocate, driving adoption of Redis as a vector database and AI platform through courses, workshops, reference applications, technical articles, public APIs, and customer and partner work.

Earlier Engineering Leadership

Founded Binnacle, a distributed messaging and browser-notification platform. Earlier architecture work included pharmaceutical anti-counterfeiting systems at Cardinal Health, scientific-content acquisition and workflow platforms at CAS, and asynchronous retail-credit origination at JPMorgan Chase.

RESEARCH

- "Pleated Ribbon Construction: Approximate Sorting Recovers the Locality of a Full Sort" — Zenodo, 2026 (artifacts). Makes space-efficient ribbon filters **2.05–2.24× faster to build** at 100–400 million keys by replacing a full sort with a single counting

pass, roughly halving their build-cost gap to Bloom filters. The result was replicated on x86 and ARM and transfers to RocksDB's production ribbon builder, where construction is 1.8× faster at 100 million keys per filter with unchanged false-positive rates.

- "Who Grades the Grader? The Verification Gap in Self-Testing Code Agents" — Zenodo, 2026, with Jyoti Vasudev, Josh Meyers, and Eric Kasper (artifacts). Measures how often agent-reported successes fail withheld tests: 18.5% on HumanEval, 28.9% on LiveCodeBench, and 41.5% on RGRBench, with failures persisting under stronger test authors and reviewers.
- "What Context Does a Coding Agent Actually Need to Act?" — arXiv, 2026. Finds that compressed context matched whole-file issue resolution at about one-third the tokens.

SELECTED ENGINEERING

Models + ModelJars

Java-native execution and distribution for open-weight and Apple Foundation Models, with qualified backends, content-addressed downloads, native CLIs, and Spring AI and LangChain4j integrations.

Vectors

Embedded JVM vector search with durable local storage, metadata filtering, quantization, hybrid retrieval, semantic routing and caching, and reproducible performance and recall tests.

LangGOAP

Goal-Oriented Action Planning for LangGraph and Deep Agents: deterministic planning, constraint optimization, automatic replanning, checkpointing, and human approval without an LLM in the execution loop.

MFCQI — Software Quality Index

Evidence-based software-quality scoring across complexity, maintainability, architecture, duplication, and security, with Java/Kotlin and Python CI gates, build integrations, and reports.

pleat

Rust ribbon filters with pleated construction: tunable false-positive rate, arbitrary hashable keys, batch queries, validated serialization, and differential tests against the reference implementation. About 2.3× faster to build than arrival order at 10 million keys in the crate's own benchmarks.

BarúDB

A from-scratch Rust LSM-tree research system covering concurrent memtables, compaction, Bloom filters, cache-aware indexing, compression, recovery, and a concurrent client/server protocol.

Redis OM Spring

Created a Spring-native data layer for Redis JSON, search, aggregations, vector similarity, and hybrid search through familiar repositories, annotations, and fluent APIs.

Redis LangGraph Checkpointers & Stores

Built Redis checkpoint savers and stores for durable LangGraph state and memory, with sync/async APIs, vector-searchable storage, caching middleware, and high-availability support.

LangChain Redis Integration

Built and maintained Redis's LangChain integration for vector retrieval, semantic caching, persistent chat history, async execution, and Sentinel deployments.

Spring AI Contributions

Developed Redis semantic-caching and chat-memory contributions for Spring AI, including similarity-based response reuse, expiration, conversation persistence, and integration tests.

EDUCATION & RECOGNITION

- **Harvard University** — Master of Liberal Arts (ALM) in Data Science and Artificial Intelligence, 2026. Focus: generative AI, reinforcement learning, graph computing, and graph neural networks.
- **Ohio Wesleyan University** — Bachelor of Arts in Computer Science; Bachelor of Arts in Physics.
- **Java Champion** — recognized by the global Java Champions program for technical and community leadership.

DEVELOPER ADVOCACY, WRITING & SPEAKING

- Delivered hundreds of conference talks and hundreds of multi-day technical courses internationally, including an 11-year run on the No Fluff Just Stuff conference tour. Paired production libraries with live coding, documentation, and direct feedback from developers, partners, and enterprise customers.
- Selected GenAI talks: Design Patterns for Multi-Agent Systems — DevNexus 2026, Secrets of Agentic Memory Management — Jfokus, and What Is RAG? Retrieval-Augmented Generation Explained Simply; complete video catalog. The agentic-memory talk was also presented at Microsoft JDConf 2026; other sessions cover agent memory, RAG, vector databases, Redis, Spring, Java, Ruby, and Rust.
- Author, *Beginning POJOs: Lightweight Java Web Development Using Plain Old Java Objects*; co-author, *Enterprise Java Development on a Budget*; contributor, O'Reilly's *97 Things Every Project Manager Should Know*.

SIGNATURE CAPABILITIES

- **API and library design:** coherent developer-facing abstractions, stable extension points, composable domain models, framework integrations, compatibility strategy, and release discipline across open-source ecosystems.
- **AI and data-systems architecture:** local model execution and distribution, vector indexing and persistence, durable agent state and memory, retrieval, semantic caching and routing, evaluation, and model qualification.

- **Performance and correctness engineering:** storage and indexing internals, memory layout, quantization, reproducible performance and recall benchmarks, reference-equivalence tests, pre-registered experiments, claims ledgers, and published null results.
- **Technical and ecosystem leverage:** product and open-source roadmaps, reference architectures, documentation, courses, talks, and user feedback loops that help teams adopt difficult technology safely.