Nick Cerutti

nick.dicerutti@gmail.com | Buenos Aires, Argentina (Remote) | GitHub | LinkedIn | Portfolio

AI Infrastructure Architect / MLOps Engineer

Agentic AI Systems | Multi-Agent Orchestration | AI Agent Security & Governance

Summary

AI Infrastructure Architect / MLOps Engineer with 6+ years building production-grade **agentic AI systems**, multi-tenant agent platforms, and **AI agent security & governance** frameworks. Specialized in **multi-agent orchestration** (LangGraph, CrewAI, AutoGen) and Kubernetes-scale infrastructure on AWS Bedrock/AgentCore and Azure AKS — cutting LLM/cloud spend $200K+/yr and lifting SLO compliance 71%→96%. Creator of +5 open-source AI agent infrastructure projects spanning policy engines, belief-consistency runtimes, and self-optimizing multi-agent orchestration. All of them built in public on github.com/angelnicolasc

Education & Certifications

S21 University — Buenos Aires, Argentina
Bachelor in Computer Science, Minor in Business, GPA 3.92/4.0 — *Jan. 2018 – May 2021*

IAE Business School — Online
Master of Business Administration — *Feb. 2025 – Present*

Cloud Native Computing Foundation (CNCF) — Online
Certified Kubernetes Administrator (CKA) — *Licensed Aug. 2023*

DeepLearning.AI – Coursera — Online
MLOps Specialization (Andrew Ng) — *Mar. 2024*

Experience

Quantitative Systems Engineer — Jan. 2026 – July 2026
High-Frequency Proprietary Trading Firm (HFT) (Contractor) — *Remote*

- Designed and ran in production a real-time trading system for prediction markets — an event-driven pipeline (Python, Rust) pulling live data from centralized exchanges and on-chain sources, holding sub-10ms latency at peak load.
- Designed the risk-control layer for live trading: position limits, daily loss limits, and slippage checks — cutting execution slippage 14% with zero limit breaches while trading real capital.
- Built statistical models of market order flow to detect volatility clustering, and used them to size trades and set execution pace under fast-moving conditions.
- Built the position-sizing logic (Kelly-criterion-based) that scales bet size to model confidence, reducing tail losses in high-uncertainty periods without giving up long-run expected value.

AI Infrastructure & MLOps Engineer — April 2023 – Present
AI Infrastructure Advisory (Milestone-based Engagements) — *Remote*

- Brought in across 6+ fast-growing platforms (Series B to late-stage, 80–300 person engineering orgs) to architect and implement the core multi-tenant substrate for agent systems serving 250–450 concurrent instances with strict data-boundary enforcement across AWS (Bedrock, AgentCore) and Azure (AKS, Azure ML Studio, ACI) environments.
- Led incident response for a cost-explosion event in a production multi-agent system caused by uncontrolled tool-call loops ($40K+ projected daily spend); implemented circuit breakers, per-session spend caps, and retrospective governance controls within 72 hours with zero recurrence, while deploying offensive security red-teaming frameworks to pentest and simulate multi-agent adversarial exploits.
- Defined SLO frameworks for agentic workloads across 3 client platforms, introducing agent-native metrics (task completion rate, mean tool calls, hallucination escape rate) instrumented via App Insights, Kibana, and OpenTelemetry; drove SLO compliance from 71% to 96% over two quarters.
- Audited and built production retrieval and governance systems for AI agents supporting 1.8M+ tool invocations, implementing deterministic policy validation that achieved zero known bypasses across client deployments leveraging Pinecone and OpenSearch for enterprise-grade hybrid vector indexing.
- Designed and built Go/Python and TypeScript (Node.js) CLI middleware, Azure DevOps templates, and automation tooling that eliminated manual deployment steps for 4 distributed engineering teams, reducing release cycle overhead by ∼40%.
- Refactored distributed client codebases integrated with Anthropic SDK (Claude API), n8n, and Copilot Studio to enforce token optimization, caching strategies, and low-infra patterns, delivering an aggregate $200K annual reduction in cloud and LLM API spend.
- Standardized enterprise data lakes and model registries for 2 client platforms, migrating legacy data workflows into Databricks and Domino Data Lab while integrating traditional/classical supervised machine learning models (Scikit-learn, PyTorch, TensorFlow) alongside generative workloads.

AI Infrastructure & MLOps Engineer — June 2021 – March 2023
Iguazio AI (Contractor) — *Remote*

- Contributed to a MLOps pod, spearheading the enterprise deployment and evaluation infrastructure while ensuring pipeline reliability across containerized LLM and inference workloads in production.

- Deployed and operated a multi-GPU Triton Inference Server cluster with dynamic batching and model quantization for NLP and Computer Vision pipelines, defining inference SLOs and building Prometheus/Grafana alerting runbooks for OOM failures.

- Led decomposition of monolithic FastAPI inference service into event-driven microservices (Docker, PostgreSQL, Kafka), slashing cold-start deployment from 18 min to under 4 min and enabling independent scaling of AI serving components.
- Owned schema and indexing strategy for behavioral analytics datasets consumed by 3 cross-functional product teams; restructured indexes and materialized views drove p99 query latency from +900ms to 120ms.
- Engineered GitOps-driven CI/CD platform (GitHub Actions, ArgoCD) with automated health-check probes and canary rollout policies, sustaining 99.9% uptime across production containerized environments.

Junior Infrastructure & Data Intern Jan. 2020 – May 2021
Algeiba *Buenos Aires, Argentina*

- Assisted the core infrastructure team in containerizing legacy monolithic applications using Docker, facilitating smoother transitions to development and staging environments.
- Wrote and maintained Python and Bash automation scripts to clean, parse, and ingest semi-structured log data into staging databases, accelerating team debugging pipelines.
- Configured Prometheus/Grafana monitoring for server health, memory allocation, and API response latencies, and optimized relational database schemas and SQL queries for internal analytics dashboards.

Selected Open-Source Projects (Proof of Work)

Drako | *Python, Policy-as-Code, Runtime Governance, AI Agent Security* 2025

- Built a comprehensive open-core AI agent security & governance platform for the full lifecycle: static scanning, policy-as-code, and deterministic runtime enforcement (no LLM in the decision loop).
- Designed and implemented a 97-rule deterministic policy engine across 16 categories (Security, Governance, Compliance/EU AI Act, Determinism, FinOps, Multi-Agent, etc.), achieving sub-2ms overhead with zero false positives from LLM evaluation.
- Developed static analysis capabilities including reachability analysis, Agent BOM generation, advisory mapping (OWASP LLM, MITRE ATLAS, real CVEs), baseline tracking, SARIF output, and CI/CD integration (GitHub Action + pre-commit hooks).
- Architected production runtime enforcement with 13-stage pipeline: ODD enforcement, DLP (Presidio), Human-in-the-Loop, circuit breakers, cryptographic audit trails, intent fingerprinting, magnitude/spend controls, and out-of-process proxy mode.
- Added specialized features: desktop MCP/agent scanning (Claude, Cursor, VS Code, etc.), autopilot config generation from scans, observability dashboard, FinOps tracking, and framework-specific rules (CrewAI, LangGraph, AutoGen, etc.).

Epica | *Rust, PyO3, Axum, Cryptography, Model Context Protocol (MCP), Python SDK* 2026

- Architected a modular open-source Rust workspace spanning 12 crates and 2 CLI tools to enforce mathematical consistency and state coherence over LLM agent cognitive steps.
- Implemented property-based testing (256+ cases) to validate strict AGM belief revision postulates, blocking semantic contradictions at runtime with sub-2ms verification overhead.
- Engineered a tamper-evident audit ledger using BLAKE3 Merkle trees and Ed25519 signatures, developing a standalone CLI (epica-verify) for offline, third-party cryptographic verification.
- Developed a highly optimized PyO3-powered Python SDK alongside an Axum-based MCP server implementing 16 native routes to expose secure, belief-driven tools to external host systems.
- Built a reproducible benchmark harness (epica-bench) evaluating synthetic multi-step agent trajectories against trajectory calibration (T-ECE) and Free Energy metrics.

Graymatter | *Go, Embedded Storage (bbolt), Vector Index, MCP Server, TUI (Bubble Tea)* 2026

- Designed a zero-dependency, persistent memory runtime for AI agents, achieving a 90–97% reduction in repeated token consumption via hybrid retrieval and atomic fact consolidation.
- Architected an embedded storage layer combining bbolt (ACID KV) with chromem-go (pure Go vector index), implementing a durable background reindexing queue to survive embedder failures.
- Built a hybrid recall pipeline integrating vector similarity, BM25 scoring, and exponential recency weighting via Reciprocal Rank Fusion (RRF) optimized for Top-K system prompt injection.
- Developed a full Model Context Protocol (MCP) server supporting stdio/HTTP transports for native integration with Claude Code, Cursor, and custom multi-modal agent environments.
- Engineered a real-time Bubble Tea TUI dashboard for local observability, featuring concurrent safety, fuzz testing, 73.5%+ coverage, and compilation into a single static ∼10MB binary (CGO=0).

Forge | *Python, Multi-Agent Orchestration, Self-Optimizing Runtime, Observability & FinOps* 2026

- Built an open-source universal harness for multi-agent systems that auto-wraps existing agent flows (without framework migration) to add production-grade observability, persistent memory, budget controls, and self-evolution capabilities.
- Designed a modular orchestration platform with auto-detected adapters for LangGraph, CrewAI, AutoGen, and generic async callables. Implemented a shared runtime, event bus, and protocol-based extension model for standardized execution across heterogeneous agent frameworks.
- Implemented comprehensive production observability and FinOps layer including OpenTelemetry tracing, per-agent and per-model cost tracking, real-time budget enforcement with hard/soft limits, and REST + SSE APIs for live operational monitoring.
- Developed a hybrid cross-run memory system combining vector retrieval, knowledge graph persistence, symbolic rules, provenance tracking, and conflict-aware versioning to enable intelligent context injection across agent executions.
- Architected a snapshot-safe self-evolution loop that automatically proposes and evaluates mutations in prompts, parameters, models, and topology, with built-in rollback, A/B testing, and circuit-breaker safeguards.

Meridian | *Rust, Python, CUDA, vLLM, Inference Scheduling* 2026

- Architected a phase-aware scheduling layer for vLLM that separates reasoning-model inference into think-decode and output-decode workloads, each governed by independent SLOs, KV-eviction policies, and entropy-driven budget control, without forking or modifying the underlying vLLM scheduler.
- Implemented dual-queue dispatch so output-decode requests drain the GPU batch ahead of think-decode traffic, running a TPOT-relaxed think scheduler at a 2.5× batch budget alongside a TTOT-strict, stream-priority output scheduler to shield user-visible latency from reasoning-phase batch pressure.
- Built a three-tier phase-aware KV block manager that evicts completed think-tokens first and shields output-critical blocks from eviction, paired with EAT/RPDI entropy-convergence signals that trigger early think-termination instead of relying on a static token budget.
- Developed a reversible, drop-in vLLM plugin that attaches via attribute delegation with zero source modification, plus disaggregated KV-transfer hooks (offload/ingest) compatible with NIXL and Mooncake-style prefill-decode fabrics.
- Engineered a Rust-core, CUDA-kernel, and PyO3-bound Python workspace with CI/GPU-runner test coverage across the scheduler, block manager, and entropy probe, SLSA Level 2 provenance attestation on every release, and a synthetic-plus-real-vLLM A/B benchmark harness for TTOT-focused evaluation.

Technical Skills

Languages: Go, Python, TypeScript, Rust, SQL, C++, CUDA (proficient)

Cloud, MLOps & Orchestration Platforms: AWS (Bedrock, AgentCore, CloudWatch), Azure (Azure ML Studio, AKS, ACI, App Insights, Azure DevOps), Databricks, Domino Data Lab, n8n, Copilot Studio, Bot Maker

ML/AI Frameworks & Data Systems: PyTorch, TensorFlow, Scikit-learn, MLflow, Pinecone, OpenSearch, ClickHouse, PostgreSQL, bbolt, chromem-go, Data Lakes & Data Warehouses

Agentic AI & Multi-Agent Systems: Multi-Agent Orchestration (LangGraph, CrewAI, AutoGen), Agent-to-Agent Protocol (A2A), MCP Server & Client Development, Agentic Loop Design, Computer-Use Agents, Context Engineering, Persistent & Episodic Memory Systems, Hybrid RAG (Vector + Keyword + Recency + Graph), Tool-Use & Function-Calling Pipelines, Anthropic SDK (Claude API)

AI Agent Security & Governance: Policy-as-Code, Offensive Security (Agent Red Teaming & Pentesting), Runtime Tool Call Validation, Deterministic Evaluation Loops, Reachability Analysis, Agent BOM, Desktop Agent Scanning, EU AI Act & NIST AI RMF Compliance, Prompt Injection Mitigation

LLM Infrastructure & Inference: vLLM (tensor parallelism, multi-GPU serving), Triton Inference Server (dynamic batching, model ensembles), Model Routing & Tiering, AWQ/GPTQ Quantization, KV-Cache Optimization, Inference Fleet Observability, Local Runtimes (Ollama, llama.cpp), Reasoning Model Integration (o-series, DeepSeek-R1), Prompt Caching & Token Budget Control, Streaming & Structured Output Contracts, LoRA / QLoRA Fine-tuning Pipelines

Scale, Reliability & Distributed Systems: Multi-Tenant Agent Platforms, Kubernetes at Scale (HPA, VPA, node autoscaling), SLO/SLA Definition & Error Budget Management, Incident Response for Agentic Systems (cost explosions, drift, security breaches), Chaos Engineering, Capacity Planning, Event-Driven Architectures, Distributed Systems Design, REST/HTTP + stdio + SSE Transports, Schema & Index Engineering, Async Worker Pools, Zero-Dependency & Single-Binary MCP Servers, Offline-First Agents, WASM-Compatible Tooling

Observability, DevOps & Developer Tooling: OpenTelemetry, Prometheus/Grafana, Kibana, TUI Observability (Bubble Tea), Agent Trace & Span Instrumentation, Real-time Memory Analytics, Token Spend Tracking, Knowledge Graph Extraction, Audit Trails, Evaluation Loops, GitHub Actions, ArgoCD, GitOps Pipelines, MLflow Monitoring, Claude Code / Cursor / VS Code MCP Integration, CLI Tooling, Pre-commit Hooks, SARIF, CI/CD for Agentic Workflows