

Akshay Palimkar

Forward-Deployed / Applied AI Engineer — Enterprise AI Solutions

Ships production agentic · MCP · RAG systems · 15 years deploying enterprise software into demanding, regulated reality

palimkar.akshay@gmail.com · +1-647-561-1202 · linkedin.com/in/akshaypalimkar · github.com/palimkarakshay · npmjs.com/~akshaypalimkar · Ontario, Canada · open to US / UK / EU relocation with sponsorship (TN-eligible)

SUMMARY

A two-halves engineer the AI-deployment world is short on: I **build and ship production AI systems** — an open-source MCP server on npm, a multi-provider agent platform with an LLM-as-judge eval gate, RAG pipelines, self-healing production CI — **and I have 15 years deploying enterprise software into hard, high-stakes environments** (Toyota, MAN Energy Solutions, Capgemini), translating frustrated-stakeholder requirements and messy data into delivered, supported systems. That intersection — *applied-AI engineer + enterprise-deployment veteran in the SAP/Clean-Core domain the AI industry is racing to connect to* — is the Forward-Deployed / Solutions profile. Every claim is publicly inspectable at github.com/palimkarakshay.

WHAT I BRING TO A FORWARD-DEPLOYED / SOLUTIONS / APPLIED-AI ROLE

- **Deploy AI into real enterprises and earn trust** — 15 yrs embedded with business stakeholders (daily production support, satisfaction-scored delivery, Kaizen savings). This is the customer-discovery, messy-data, and compliance reality these roles actually screen for.
- **Production agentic / MCP / RAG engineering, shipped & published** — not slideware. abap-mcp is live on npm; RAP Dojo is in production.
- **Reliability & eval discipline** — LLM-as-judge eval gates, a 16-code failure-mode taxonomy, token-budget circuit breakers, prompt-injection hardening, self-healing error → triage → auto-PR pipelines.
- **Full-stack, in their stack** — TypeScript / Next.js / React / Vercel (AI SDK + AI Gateway) + Python + ABAP. Solo-operate ~50 repos and 13 live apps on cost-guardrailed free-tier infra.

SELECTED AI ENGINEERING (OPEN SOURCE UNLESS NOTED; LINKS LIVE)

abap-mcp — MCP server for SAP ABAP · github.com/palimkarakshay/abap-mcp · npm [abap-mcp](https://www.npmjs.com/package/abap-mcp)

Gives AI coding agents offline ABAP analysis, Clean-Core readiness checks and RAP scaffolding — no SAP system needed. Designed a **dual-parse Cloud-readiness diff** that separates real migration blockers from pre-existing defects; built a **self-validating RAP scaffolder** (generator + linter share one parser, every output round-tripped through abaplint at Cloud level). 7 tools, 80 tests, weekly auto-upgrade CI.

Multi-Vendor AI Agent Platform — "issue → shipped PR"

Autonomous **triage** → **plan** → **execute** → **maker-checker review** → **auto-merge** loop with fallback ladders across **5 LLM providers**, gated by a **prompt-regression eval suite**: deterministic offline stub, **LLM-as-judge** scoring, and a 16-code failure-mode taxonomy.

Property Due-Diligence App — live in production

One address → a web-grounded, cited, weighted-risk report across 36 dimensions (Astro SSR on Vercel, Gemini + Search grounding, Zod-validated structured output). Engineered a **self-healing pipeline**: runtime errors → Postgres (deduped) → daily AI-triage cron → gated, source-only auto-fix PRs behind a kill-switch.

Production MCP toolkit & tool-lint · github.com/palimkarakshay/mcp-kit

Hardened MCP starter (TypeScript + Python/FastMCP) with dual transports, bearer auth, retryable-error taxonomy, and a novel **tool-description lint** that treats model-facing docs as lintable code (hard-fails CI on credentials-in-inputs).

RAP Dojo — Clean Core Developer Academy · live: rapdojo.lumivara.tech

17-module interactive ABAP → RAP academy with a live in-browser linting loop. Engineered a **provider-agnostic AI router on the Vercel AI Gateway** (multi-model, rate-limited, token-budget circuit breaker) grounded by a self-hosted SAP-docs MCP.

Also: a native Android AI-coding terminal (Kotlin; vendored Termux engine + cross-compiled SSH; runs the whole fleet from a phone); a content-agnostic LMS engine with crash-safe Vercel Workflow content drafting; an AI-usage observability dashboard; dual-provider encrypted offsite backup/DR.

ENTERPRISE DELIVERY EXPERIENCE (CUSTOMER-FACING, AT SCALE)

SAP Technical Lead / ABAP Consultant — Toyota Motor Manufacturing Canada · 2020–Present

Architected A2A interfacing on SAP PI/PO 7.5 (EDI/IDocs) for scheduling, orders, shipping, invoices. Ran daily production support alongside the business; delivered a sales-process **Kaizen saving ~C\$100k/year** and automated the Annual Pricing Review workflow.

ABAP Lead / Workflow Developer — Capgemini · 2018–2019

Built Fiori approval apps (PO / RFQ / PR) with custom multi-step workflows; lifted the annual **OTACE client-satisfaction score from 3.7 → 4.1 / 5**.

ABAP Developer — ACG Worldwide · 2016–2018

Migrated and optimized **2,500+ custom developments** during an EHP8 / Suite-on-HANA upgrade; rollouts across Croatia, Brazil, Thailand.

SAP ABAP & Workflow Consultant — MAN Energy Solutions · 2013–2016

Implemented master-data governance workflows in a multi-client landscape (ALE/IDoc/EDI); harmonized and de-duplicated master data across **75,000+ customers and 225,000+ suppliers**.

ABAP Developer — Wind World India · 2011–2013

Delivered an SAP R/3 implementation + post-go-live support; BAPI goods-movement interfaces, LSMW/BDC conversions.

CERTIFICATIONS & EDUCATION

- **SAP Certified Development Specialist** — ABAP for SAP HANA 2.0 · **SAP Certified Development Associate** — ABAP w/ NetWeaver 7.50
- **B.Eng, Industrial Electronics** — Dr. B.A.M. University, 2009 (*supports TN / O-1 work-authorization paths*)

TECHNICAL SKILLS

- **AI / LLM** — MCP server & client engineering · multi-model orchestration & fallback (Claude, Gemini, OpenAI, Mistral, OpenRouter) · Vercel AI SDK & AI Gateway · RAG (embeddings + BM25 + RRF, Qdrant) · agentic loops · **LLM-as-judge & eval harnesses** · structured output (Zod / JSON-Schema) · prompt-injection hardening · self-healing CI · token/cost controls · local LLMs (Ollama)
- **Languages** — TypeScript · Python · ABAP · JavaScript · SQL
- **Frameworks & Infra** — Next.js 16 · React 19 · Astro · Tailwind · FastAPI · Vercel · Cloudflare (Workers AI, Vectorize, R2) · Neon/Postgres · Docker · GitHub Actions
- **SAP / ABAP** — RAP · CDS · Fiori · Clean Core · ATC migration · ABAP Cloud / S/4HANA · PI/PO (EDI/IDoc) · BTP · abapGit · abaplint

Note: I am an AI-augmented engineer — I orchestrate agent fleets to ship at high leverage, and I can read, extend, and defend every line they produce. Happy to demonstrate live.