AR. YASH ARORA

AI Systems Engineer · LLM Orchestration, Systems Integration & Audit

Najibabad, Uttar Pradesh, India | +91 99972 03210 | arch.yasha08@gmail.com | linkedin.com/in/ar-yash-arora | shaktiyukti.in

> **Open to:** Part-time & remote engagements, hourly rate · Available immediately · Flexible hours across time zones

PROFESSIONAL SUMMARY

AI systems engineer working at the **orchestration and integration layer**: building tools, connecting them into systems that close a full problem loop, then **auditing whether the connected parts actually reinforce each other**. Design **multi-agent architectures** and **multi-provider LLM routing** with automatic failover; enforce **data classification, guardrails and spend limits in code rather than policy**; make **build-vs-buy decisions from primary-source evidence** — including rejecting a category-leading vendor after testing disproved its claims. Cost engineering is a core discipline: **fail-closed budget controls cut a worst-case pipeline cost from ₹239 to ₹6.34**, and a routing audit caught an **11.5× model overspend**. Licensed architect by training — constraints first, then structure.

CORE COMPETENCIES

AI Systems & Orchestration: Agentic workflow architecture, multi-agent orchestration, AI agents, agent tool-calling, autonomous task execution, agent memory & persistent state, systems integration, pipeline orchestration, workflow automation, human-in-the-loop design, end-to-end system audit, capability-tier routing

LLM Engineering: LLM application development, multi-provider abstraction, automatic failover, model routing & selection, prompt engineering, system prompt design, context engineering, structured output & schema-constrained JSON, LLM evaluation, adversarial red-teaming, output guardrails, prompt-injection surface reduction, LLMOps, token cost optimization, inference latency profiling, content-addressed caching

AI Tools & Platforms: Anthropic Claude & Claude Code, OpenAI ChatGPT & Codex, Google Gemini & Google AI Studio, Groq, Ollama (cloud & local), Kimi, DeepSeek, Qwen, Mistral, GPT-OSS, Nemotron, Llama, ElevenLabs voice cloning, F5-TTS, Whisper / faster-whisper ASR, Google Video Intelligence, Remotion, Motion Canvas, FFmpeg, Apify

Engineering: Python, TypeScript, JavaScript (Node.js), Bash, REST API design & integration, OAuth 2.0, service-account JWT, webhooks, rate limiting, exponential backoff, idempotency, distributed locking, concurrency control, subprocess isolation, ETL & data pipelines, web scraping, schema validation, JSON / CSV / YAML

Web & Cloud: Next.js, React, Astro, Express, Tailwind CSS, GSAP, Framer Motion, responsive design, SEO & Open Graph, GitHub Actions, CI/CD, cron scheduling, Git, Vercel, Cloudflare Pages & DNS, Google Cloud Platform, serverless functions, secrets management, zero-downtime deployment

Reliability & Quality: Test-driven development, regression & integration testing, cache-replay testing, secret-hygiene testing, fault injection, graceful degradation, observability & run telemetry, structured logging, error taxonomy design, root-cause analysis, performance profiling, memory-pressure diagnosis, checksum-based change detection

Security & Governance: Data classification policy, PII & sensitive-data gating, credential-leakage prevention, least-privilege scopes, supply-chain hardening (SHA-pinned actions), secrets management, fail-closed budget enforcement, cost modelling & unit economics, vendor terms & compliance review (DPDP/GDPR-aware)

Leadership & Judgement: Technical decision ownership, build-vs-buy analysis, vendor evaluation, architecture review, scope control, risk assessment, red-team facilitation, technical writing & documentation, process design, stakeholder communication, mentoring through written method

Architecture (Licensed): AutoCAD, Revit, BIM, SketchUp, V-Ray, Enscape, ArcGIS, Photoshop, Illustrator, design documentation, AEC automation, computational & parametric design

PROFESSIONAL EXPERIENCE

AI Systems Engineer — Orchestration, Integration & Audit

Independent · 2025 – Present · Remote

- **Own end-to-end technical direction** of a multi-workspace AI automation estate — architecture, vendor selection, spend policy and release standards — as sole decision-maker for what ships and what is refused.
- Architected the **Workflows–Agents–Tools (WAT) framework** separating LLM reasoning from deterministic tested execution, after diagnosing why chained AI degrades: five steps at 90% per-step accuracy succeed only 59% of the time.
- **Led the progression from isolated tools to integrated systems to system audit** — built the orchestration layer where agents hand off and share state, then instituted **end-to-end audits catching integration failures that every component-level test passed**.
- Built a **multi-provider LLM abstraction** with a Provider interface and **FallbackChain**: errors are classified retryable vs terminal and failover occurs only when safe, with per-attempt telemetry — eliminating load-bearing single API keys.
- Designed a **capability ladder** matching task to cheapest sufficient tier (deterministic Python → small LLM → frontier model → local-only for confidential data); a routing audit found **11.5× overspend** from a premium model doing a mid-tier job.
- Engineered **fail-closed cost controls** — pre-flight duration ceilings, resolution floors, hard caps that refuse rather than warn, and a **content-addressed cache making repeat analyses free** — reducing worst-case pipeline cost **₹239 → ₹6.34**.
- Moved **~90% of routine inference off premium models** onto cost-appropriate tiers via an 18-model routing catalogue, reserving frontier models for judgement.
- Enforced **governance as code**: an automated classifier blocks sensitive payloads pre-flight; spend gates demand written justification. Built after observing that **a documented rule nothing enforces is a preference** — the policy had been restated five times and still applied inconsistently.
- Reduced **prompt-injection attack surface** by architecturally separating data ingestion from tool-calling authority.
- Ran an **instrumented benchmark that falsified my own documented capacity assumption**, proving latency tracks output volume not input size — and revised standing guidance rather than defending the original claim.
- Instituted **evidence discipline** as standard: every claim carries a measurement, source and falsifier; figures labelled measured vs estimated; predictions recorded before testing.

Founder & Technical Lead — Shakti Yukti (Architecture × AI Studio)

Shakti Yukti · 2026 – Present · shaktiyukti.in

- Founded and run an **Architecture × AI studio**; set positioning, built internal tooling, shipped the **live production website** — a practice designed to operate beyond its headcount on its own AI back-office.
- **Led build-vs-buy evaluation of the AI presentation-tool market and recommended against the category leader**: primary-source testing showed lossy export and brand templates that did not survive, corroborated by a third-party tool built specifically to work around the defect. Chose a **zero-cost internal pipeline** over per-seat licensing.
- **Rejected a technically superior animation framework** (Motion Canvas, MIT-licensed) after quantifying the migration at **1,632 lines** — the marginal runtime gain did not justify the rewrite. Declined paid cloud render services on the same cost-to-benefit basis.
- Engineered a **chunked video render pipeline** with process-lifetime isolation after root-causing stalls to **memory pressure, not scene complexity** — splitting 800-frame jobs into 200-frame subprocesses and muxing audio once, cutting render to **8m47s** and avoiding a hardware upgrade.
- Established a **concurrency policy from measurement**: profiling revealed a cliff where 4 parallel workers hung on long jobs, so parallelism is capped at 2 above a measured frame threshold.
- Authored the studio's **tool-building methodology** — staged delivery, pre-registered predictions, an independent **red team briefed to refute rather than improve**, and seam gates verifying the joins between parts. Applied across **12 documented red-team cycles**.
- Introduced **MD5 freeze anchors** on locked artifacts so unintended changes are detectable by checksum rather than review.
- Ran an **AI-assisted front-end workflow on Google AI Studio** with formal reconciliation: AI Studio owns the front-end, hand-built backends deploy independently, and each output is diffed against the previous to catch silent drift before integration.
- Set a **free-first cost ladder** requiring written justification before spend — **delivering a six-part production tool at zero external cost**. Success metric: each tool must consume less of my time than the last.

Full-Stack Engineer — Production Web & Automation Systems

Independent · 2025 – 2026

- Built a **hybrid local/cloud publishing system**: the same TypeScript codebase runs as a local library and as a **GitHub Actions cloud worker firing on a 30-minute cron**, so scheduled work executes even with the authoring machine offline — **chosen over an always-on server to eliminate idle compute cost**.
- Designed a **single-source-of-truth JSON queue synchronised through Git**, bridging offline authoring with guaranteed online execution, with status committed back after each run.
- Implemented **platform adapters behind one typed publish contract** — LinkedIn, Instagram, Threads and X in the scheduler, plus **Facebook Reels and YouTube publishing paths** — with **pre-dispatch locks guaranteeing idempotency** so concurrent runs cannot double-publish.
- **Hardened the CI supply chain**: third-party actions pinned to exact commit SHAs, workflow permissions scoped to least privilege, all secrets held in Actions Secrets, and a **$0 spending cap that pauses the workflow rather than overrunning**.
- Shipped a **live production Next.js/React application** on Vercel with a custom Cloudflare domain; solved link permanence structurally with a **redirect indirection layer** so printed QR assets never need reprinting.
- Executed a **zero-downtime DNS migration inside a shared production zone**, scoped to one record under an explicit blast-radius constraint.
- Built a **React + Express content engine** with a research pipeline, topic selection and draft-review UI, and resolved CI cancellation collisions via **namespaced concurrency keys**.

Co-Founder — Shakti Kreeda

Shakti Kreeda · 2025 – Feb 2026

- Co-founded a design and AI-systems venture; built an automated multi-platform content pipeline integrating **Meta Graph API**, YouTube and Telegram with speech-to-text and automated video clipping. **Concluded the venture on evidence** in Feb 2026.

Junior Architect

MAC · 2023 · 6 months · Bijnor, Uttar Pradesh

- Delivered design development, construction documentation and client presentations on live projects using AutoCAD, Revit and SketchUp.

SELECTED WORK

Shakti Yukti Studio Platform · shaktiyukti.in

Live · Astro islands · Tailwind · GSAP · Cloudflare Pages

Production studio site shipping minimal JavaScript; all visual composites produced offline at zero asset cost. Front-end iterated in **Google AI Studio** with a diff-based reconciliation step before integration.

LLM Routing & Cost Governance Policy

18-model catalogue · 4-class data policy · enforced in code

Routing law deciding which model may process which data class and which model should do which task. **Providers disqualified on evidence** — one free tier banned because its terms permit human review of API inputs, another restricted for an unobtainable data-processing agreement. Automated classifier blocks non-public payloads pre-flight, including an **aggregate-sensitivity rule** for public data that composes into PII.

The Researcher — Evidence & Credibility Instrument

Multi-stage vision + LLM pipeline · accepted into production use

Maps who asserts a claim, who opposes it, and each source's independence — built to **assemble evidence and deliberately issue no verdict**. **Scope cut after a live pilot exposed a real defect**: the tool treated an accused party's denial as credible evidence, so verdict logic was removed rather than patched. Chains profile retrieval, deterministic sampling, OCR, speech transcription and LLM narrative, with keyless offline test paths and per-unit checkpointing.

Explanation-Video Production System

Chunked render · 8m47s · quality-gated · ₹0 external spend

Programmatic video pipeline with subprocess isolation, orphan-process guards, FFmpeg assembly and **AAC priming-padding correction**. Every beat passes an **8/8 quality gate**, including audio-sync verified to a **0.0005-second delta** and frame-level checksum comparison proving unrelated changes left the render bit-identical.

Hybrid Local/Cloud Publishing Scheduler

TypeScript · GitHub Actions cron · 6 platforms · $0 cap

One codebase running locally and as a cloud worker, sharing a Git-synchronised JSON queue. Publishes to **LinkedIn, Instagram, Threads, X, Facebook and YouTube** through typed adapters, with idempotent dispatch locks, SHA-pinned actions, least-privilege permissions and a hard $0 spend cap that pauses rather than overruns.

Identity Platform & Permanent-Link Architecture · about.aryasharora.vip

Live · Next.js · React · Vercel · Cloudflare DNS

Identity application built on a **permanence contract**: printed QR codes encode only the owned domain root, every destination resolving through an indirection layer. Deployed into a shared DNS zone under strict blast-radius constraints.

WAT Framework — Agentic Automation Architecture

Multi-workspace · agent memory · self-improvement loop

Three-layer architecture: documented workflows define intent, an agent orchestrates and recovers, deterministic tested code executes. **Persistent agent memory** resumes work across sessions without re-deriving context; every failure is diagnosed, fixed, verified and written back into the method so the defect cannot recur.

EDUCATION & CREDENTIALS

Bachelor of Architecture (B.Arch.)
AIT-SAP, Greater Noida, Uttar Pradesh, India · 2020 – 2025

Higher Secondary Certificate (Class XII)
Dhruva Public School, Delhi, India · 2018 – 2020

Licensed Architect — India
Registered architectural practitioner

Continuous Self-Directed Engineering Practice
Software, AI and systems engineering · self-taught and shipped to production · 2025 – present