

# NAVEEN SINGH

Senior Angular Architect · Frontend Platform Lead · Generative AI Integration Specialist

Rishikesh, India | +91-8077486316 | [n\\_aveen@outlook.com](mailto:n_aveen@outlook.com) | Remote — Available Globally

## PROFESSIONAL SUMMARY

Senior Angular Architect with 8+ years owning frontend platform strategy for enterprise banking and insurance institutions. Expert in Angular internals (Ivy, zone-less architecture, @defer blocks, change detection lifecycle, Signals-first state), large-scale migrations (v2–v19), micro-frontend platform design, Web Vitals governance, and cross-squad engineering standards — with a proven record of 40% load-time improvements on \$50M+/month transaction platforms. Defines architectural direction and frontend operating standards for 12-engineer organizations across 3 product squads. Differentiator: production Generative AI integration experience building RAG platforms, LLM pipelines, and hallucination mitigation systems on Angular frontends in regulated financial environments.

## KEY PERFORMANCE HIGHLIGHTS

<b>40% Performance Gain — \$50M+/Month Platform</b>  Angular v9→v19 migration (4s → 2.4s) — zone-less + OnPush + Signals eliminated CD cycles; TBT reduced 380ms → 95ms on banking dashboard	<b>Web Vitals: LCP &lt;2.5s, CLS &lt;0.1, INP &lt;200ms</b>  CI-enforced Lighthouse performance budgets org-wide — PRs exceeding thresholds blocked at pipeline, eliminating performance regressions
<b>35% Release Dependency Reduction</b>  Module Federation MFE platform decoupled 3 squads across 12 engineers — independent zero-downtime deployments for all teams	<b>30% Bundle Size Reduction</b>  Source-map-explorer + build optimizer + @defer blocks + lazy chunking reduced initial JS payload, directly improving TTI and Core Web Vitals

## TECHNICAL SKILLS

<b>Angular Internals</b>	Ivy Renderer, Zone-less Architecture, @defer Blocks, Change Detection Lifecycle, Signals (fine-grained reactivity), Custom Schematics, Angular DevTools Profiling, DI System (hierarchical injectors, injection tokens), Hydration-aware SSR
<b>Angular Ecosystem</b>	Angular (v2–v19), TypeScript, RxJS (custom operators, higher-order observables), Standalone Components, Angular Material, Angular CDK, Analog.js, NgRx Signal Store
<b>Frontend Platform</b>	Micro-Frontend Architecture, Module Federation (Webpack/Vite), Nx Monorepo, Performance Budgets (CI-enforced), Web Vitals (LCP/CLS/INP), Bundle Analysis, Tree-shaking, Design Systems
<b>Generative AI &amp; LLM Integration</b>	RAG Architecture, LangChain, Prompt Engineering, Hallucination Mitigation, Token Budget Optimisation, LLM-as-Judge (GPT-4o), HITL Pipelines, Streaming LLM (SSE), Semantic Reranking, CoT Reasoning
<b>AI Tooling &amp; Scale</b>	FAISS (IVF-PQ), Google Gemini API, Claude API (Anthropic), LangSmith, Arize Phoenix, Drift Detection, Azure Document Intelligence, Pinecone
<b>Governance &amp; Ops</b>	ADR Formalization, Architecture Guild, SonarQube, Lighthouse CI, Docker, CI/CD Pipelines, Auth0, RBAC, Jasmine/Karma, Agile/Scrum

## ENTERPRISE ANGULAR PLATFORM STRATEGY

Platform ownership at Primus Software Corporation — the frontend operating standards, tooling, and governance that 12 engineers across 3 product squads build on:

- ▶ **PLATFORM BLUEPRINT** Authored the Angular Platform Blueprint adopted across all frontend applications — covering change detection strategy (OnPush → Signals), @defer-first lazy loading model, Signals-first state, performance budgets (LCP/CLS/INP/ bundle), version governance cadence, CI enforcement hooks, and custom schematic standards. Blueprint reduced onboarding time by 30%, eliminated architectural drift across 3 squads, and became the single source of truth for all frontend technical decisions.
- ▶ **ZONE-LESS ARCHITECTURE** Eliminated zone.js overhead via OnPush mandate + Signals-based fine-grained reactivity. Profiled via Angular DevTools: re-render count ~42 → ~11 per interaction cycle; session memory ~310 MB → ~195 MB; TBT ~380 ms → ~95 ms. Profiled baselines captured before and after migration — not estimated.
- ▶ **DEFER & LAZY LOADING** Implemented @defer blocks across all heavy component trees — deferring non-critical UI (data tables, charts, modals) until viewport intersection. Contributed directly to 30% bundle reduction and measurable INP improvements by reducing main thread blocking on initial parse.

- ▶ **DI ARCHITECTURE** Deep Angular DI system governance — standardized use of hierarchical injectors, injection tokens for configuration, and tree-shakeable providers across the platform. Eliminated service duplication across 3 squads and enforced lazy-loaded module isolation patterns via scoped injector hierarchies.
- ▶ **SCHEMATICS & GOVERNANCE** Built custom Angular schematics for scaffold generation — enforcing OnPush, Signals state, barrel exports, and @defer-first patterns at generation time, eliminating structural review round-trips. Defined and enforced version governance roadmap (v6→v19) with zero production regressions across all major transitions.
- ▶ **RXJS ARCHITECTURE** Designed advanced RxJS operator abstractions — custom pipe-able operators (retry-with-backoff, poll-until, debounced-save) and higher-order observable chains reducing state management bug surface area and providing type-safe async patterns org-wide.
- ▶ **SSR, HYDRATION & WEB VITALS** Implemented hydration-aware SSR for critical routes — faster LCP on initial load, managed hydration mismatches via server/client state reconciliation. CI-enforced Lighthouse performance budgets block any PR exceeding LCP <2.5s, CLS <0.1, INP <200ms thresholds.
- ▶ **DESIGN SYSTEM** Architected enterprise component library (Angular CDK primitives) adopted across all 3 squads — 25% duplication reduction, ~40 engineering hours saved per sprint, full accessibility compliance.

## PROFESSIONAL EXPERIENCE

**Associate Technical Lead (Senior Angular Architect & Frontend Platform Lead)** · **Primus Software Corporation** · Nov 2019 – Present · Rishikesh, India

**Stack:** Angular 6–19, TypeScript, RxJS, Signals, Standalone Components, @defer, Module Federation, Nx, SSR, REST APIs, Docker, CI/CD, SonarQube, Lighthouse CI

- ▶ **PLATFORM OWNERSHIP** Owned end-to-end Angular platform strategy for a \$50M+/month banking platform serving 1,000+ daily active users — defining version governance, performance standards, architectural conventions, and tooling across a 12-engineer, 3-squad organization.
- ▶ **ANGULAR MIGRATION** Executed Angular v9→v19 migration — zone-less change detection (OnPush + Signals), @defer blocks, Standalone Components, Ivy optimization's — delivering 40% load-time improvement (4s → 2.4s) and eliminating 4 years of technical debt.
- ▶ **MFE PLATFORM** Designed Module Federation micro-frontend platform — decoupling 3 squad release cycles, reducing inter-team dependencies by 35%, enabling independent zero-downtime deployments.
- ▶ **GOVERNANCE & EFFICIENCY** Defined engineering standards via Architecture Guild and ADRs; drove org-wide AI-augmented workflow adoption (Gemini/Claude code gen, test scaffolding) — ticket lifecycle down 15%, throughput up ~20%, validated via SonarQube.
- ▶ Grew 3 junior engineers to mid-level within 18 months; elevated test coverage 30% → 45%, reducing production incidents by 20%. Primary C-suite technical liaison at 95% sprint adherence.

**Software Engineer — Angular** · **Cartel First** · Jan 2017 – Oct 2019 · Gurugram, India

**Stack:** Angular 2–6, TypeScript, RxJS, REST APIs, Jasmine/Karma, Git, Agile/Scrum

- ▶ Engineered enterprise Angular applications for high-volume financial workflows serving 5,000+ concurrent users — full regulatory compliance via secure APIs and RBAC.
- ▶ Built 20+ component UI library saving ~40 engineering hours/sprint; optimized lazy loading cutting bundle size by 20% and improving TTL.

## KEY PROJECTS

**Intelligent Document Processing Platform (IDP)** — Generative AI Integration · Banking & Insurance · Production · 10k–50k Documents

**Stack:** Angular 18, TypeScript, RxJS, Signals, @defer, Azure Document Intelligence, RAG, FAISS (IVF-PQ), LangChain, LangSmith, Python, Auth0

- ▶ **ANGULAR FRONTEND** Built complete Angular 18 frontend — multi-panel document viewer with @defer-loaded heavy panels, real-time confidence score overlays, source-attribution UI mapping every AI claim to its exact source passage, HITL review queues, and LLM response streaming via SSE + Angular async pipe + RxJS Observable chains. Signals-based state management ensured zero unnecessary re-renders across complex panel interactions.
- ▶ **UPLOAD & INDEXING UX** Engineered reactive upload and indexing pipeline in Angular — drag-and-drop multi-file upload with chunked progress tracking, real-time indexing status updates via polling observables, and error recovery flows. Used RxJS switchMap/concatMap orchestration to manage concurrent document processing without race conditions.
- ▶ **GENAI INTEGRATION** Architected token-budget management system (sliding window chunking, per-document pre-counting, chunk-boundary detection) — ~30% reduction in Generative AI operational overhead across the full production document corpus.

- ▶ **COMPLIANCE** Designed 5-layer hallucination mitigation pipeline (source-constrained retrieval → confidence scoring → semantic reranking → attribution UI → HITL escalation) — unverified LLM outputs reduced to near-zero; every financial AI output audit-ready and legally defensible in regulated environments.
- ▶ **OBSERVABILITY** Deployed LangSmith observability layer tracking token usage, retrieval latency, and response quality in production — drift detection drove a prompt tuning roadmap achieving 15% query success improvement over 3 months, directly shaping two product development cycles.
- ▶ **PERFORMANCE** Resolved FAISS retrieval latency across the full document embedding set via IVF-PQ indexing with query-time caching — bringing p95 retrieval from 3–4s to sub-500ms with under 2% recall degradation. Engineered stateful multi-turn context management with compressed memory and turn-based eviction across 20+ conversational turns.

**Project Mycroft** — Full-Stack RAG Platform · Independent Build · Private & Proprietary · Prototype Stage

**Stack:** Angular 19, TypeScript, RxJS, Signals, @defer, LangChain, Google Gemini API, FAISS (IVF-PQ), Python (FastAPI), SSE, Auth0, SCSS

- ▶ **FULL-STACK ARCHITECTURE** Independently designed and built a full-stack RAG platform from zero — Angular 19 frontend, Python/FastAPI backend, LangChain orchestration layer, and FAISS vector store — supporting PDF, DOCX, CSV, and web URL ingestion. Zero-to-working-prototype in 72 hours; full demo-ready build completed in 6 weeks with no prior Generative AI engineering background.
- ▶ **ANGULAR + GENAI UX** Angular 19 frontend built with Signals-first state, @defer-loaded panels for heavy components, real-time token-streaming chat (LangChain + SSE + async pipe) achieving sub-200ms TTFT, multi-format upload pipeline with live progress tracking, web URL crawling, query history with session persistence, confidence score indicators, and source passage highlighting per answer.
- ▶ **AI QUALITY** Designed intent-aware prompt routing with query classification (factual, comparative, summarization, multi-hop), few-shot injection per intent class, and Chain-of-Thought (CoT) reasoning — verified via GPT-4o LLM-as-judge scoring loop achieving 98%+ alignment with a 500+ human expert benchmark dataset without manual review.
- ▶ **DATA PIPELINE** Built heterogeneous document ingestion pipeline — PDF text extraction, DOCX structure parsing, CSV tabular flattening, and web page crawling with content normalization. Pre-embedding cleaning pipeline reduced false-positive retrieval rate by 40% across all input types.
- ▶ **RETRIEVAL ENGINEERING** Implemented adaptive retrieval engine with dynamic K selection (3–12 based on query complexity scoring), semantic threshold filtering, and MMR diversity — improving multi-hop query answer completeness by ~35% versus fixed-K baseline. Platform is private and proprietary — available for demonstration on request.

THOUGHT LEADERSHIP & CONTINUOUS LEARNING

◆ Knowledge Sharing	Delivered internal Angular platform talks at Primus — Signals migration playbook, custom schematics strategy, and @defer-first loading model adopted org-wide across 3 squads and 12 engineers
◆ Open Source	Shares Angular architecture patterns, Signals migration guides, @defer best practices, and micro-frontend playbooks.
◆ Mentorship	Structured 18-month engineering growth programme — grew 3 junior engineers to mid-level through architectural pairing, code standards enforcement, and design review cycles
◆ Currently Advancing	NgRx Signal Store (deep adoption), Nx Monorepo at scale, Analog.js (Angular meta-framework), Angular CDK advanced patterns, Advanced RAG & Generative AI pipelines, Fine-tuning LLMs

- ▶ **DEMONSTRATED VELOCITY** Ramped from zero Generative AI engineering knowledge to a full demo-ready RAG platform in 6 weeks — using the same systematic approach applied to every Angular major version: read the specification, trace the internals, build a spike, then scale to production quality. This methodology has been applied across Angular v6 through v19, the full Generative AI engineering stack, and React during a critical cross-team release.
- ▶ **FRAMEWORK BREADTH** Contributed production-ready code to a React-based Micro-Frontend within 48 hours of first access during a critical release — Angular architectural instincts (SOLID principles, component contracts, Observer/Facade patterns, hierarchical DI) transferred directly. Deep platform expertise is framework-agnostic by design.