

USMAN ALI SIDDIQUI

Technical Lead — Agentic AI Engineering & Distributed Systems

Germany | German citizen | usmanalisiddiqui94@gmail.com | +49 176 4767 0947

maanizstudio.com | linkedin.com/in/usman-ali-siddiqui-744585132 | github.com/usman250994 | mani9418.medium.com

PROFESSIONAL SUMMARY

Technical lead and distributed-systems engineer with **10+ years** building cloud-native platforms across FinTech, HealthCare and startups, and an **MSc in Computer Science (AI)**. Senior engineer on a global microservices platform at WS Audiology, and concurrently technical lead at a startup where I own architecture, hiring and delivery end-to-end. Deep in Domain-Driven Design, Event-Driven Architecture and AWS, with Python as my applied-AI language (RAG, LangChain, LangGraph). Focused on **agentic AI engineering**: I designed a stateful, spec-driven multi-agent lifecycle in LangGraph in which agents exchange structured, contract-based results, shipped a production app with it, and rolled it out to a team, lifting throughput **~40%** with humans accountable for every merge.

CORE COMPETENCIES

- **Generative & Agentic AI**: Generative AI (Gen AI), LangGraph, LangChain, Multi-Agent Orchestration & Stateful Workflows, Human-in-the-Loop Design, Spec-Driven Development, Agent Tool & API Contracts (MCP, Structured Outputs & Errors), LLM Integration, RAG, Semantic Search & Embeddings, Vector Search (FAISS), Context Engineering, LLM Evaluation, LLM Cost Optimisation & Model Routing, Prompt Design, Deep Learning & Neural Network Fundamentals, GitHub Copilot, Claude, OpenAI
- **Architecture**: Microservices, Domain-Driven Design (DDD), Event-Driven Architecture (EDA), Saga Pattern, State Machines, Distributed Systems, REST API Design, System Design
- **Cloud & DevOps (AWS-first)**: AWS CDK, Lambda, API Gateway, DynamoDB, Cognito, Step Functions, EventBridge, SES, SNS, Secrets Manager, KMS, WAF, CloudWatch, EC2, S3; Docker, Infrastructure as Code, CI/CD (GitHub Actions), Fail-Fast Delivery, Multi-Environment Releases (Staging/Production); Azure
- **Observability & Quality**: OpenTelemetry, Dash0, Grafana, Alerting & Incident Response; Unit, Integration, E2E and Contract (Bounded-Context) Testing, Coverage & Code-Quality Gates
- **Languages & Frameworks**: TypeScript, JavaScript, Node.js, NestJS, React, React Native, Go, C# (.NET), Angular, Python (FastAPI, LangChain, LangGraph, Pydantic, SQLAlchemy, asyncio, pytest, Pandas/Scikit-learn)
- **Data & Messaging**: PostgreSQL, MongoDB, DynamoDB, SQL Server, Redis, Kafka, GraphQL
- **Leadership**: Technical Leadership, Requirements Engineering, Hiring (Screening, Shortlisting, Technical Interviewing), Mentoring, Stakeholder Management, Engineering Standards & Protocols, Agile/Scrum, SAFe

WORK EXPERIENCE

Arbeitsly — Technical Lead (Head of Engineering scope)

Jan 2026 – Current

Reverse-recruiting platform startup | Germany | Part-time (~15 hrs/week), concurrent with full-time role

- Own the full technical strategy as sole technical decision-maker — architecture, tooling, development protocols and feature prioritisation — leading a core engineering team of **4** within a **10-person** cross-functional group.
- Run requirements engineering with non-technical founders and operations: surface system gaps and bottlenecks, funnel them into scoped, sequenced features and implementable specs, and unblock delivery through pairing and review.
- Inherited a tangled, tightly coupled codebase; re-architected it by separating services along clear boundaries and reworking legacy features so new ones could integrate cleanly, then delivered features end-to-end on containerised services on AWS, used daily by **100+ active users** across Europe and the UK.
- Built and own a **fail-fast delivery pipeline** in GitHub Actions — unit and E2E tests on every PR, coverage and code-quality gates before merge and production promotion, isolated staging and production environments — cutting rollback/recovery time on bad deployments by **~90%**.
- Introduced OpenTelemetry-based observability across all environments with Dash0 (logs, errors, alerting) to catch failures before they escalate, cutting production downtime from roughly **twice a month to under once**.
- Defined the technical hiring bar and advised founders on hiring order; screened, shortlisted and interviewed candidates, then onboarded and mentored hires to independent delivery.
- Rolled out my multi-agent orchestrator (architect, developer and reviewer agents; human approval at spec and merge) as the team workflow, replacing ad hoc prompting: throughput up **~40%**, fewer unreviewed errors.

WS Audiology — Senior Software Engineer

Oct 2021 – Current

Erlangen, Germany | 12,500-person company; ~150-person SAFe programme within multi-region R&D

- Senior engineer on a team owning a core microservice of a global hearing-device fitting platform for hearing care professionals (**~20,000 fitting sessions/month**).
- Design, build and test microservices using DDD, EDA and state machines — covered by unit, integration, E2E and bounded-context contract tests and monitored in Grafana — deployed with Docker to Kubernetes, contributing to cloud-to-IoT device communication across **6 production releases**.
- Step in as technical team lead during absences, owning direction on complex features and unblocking stalled delivery; represent the team in cross-team architecture refinement, shaping feature design before implementation.
- Introduced AI-assisted, spec-driven development: prototyped custom GitHub Copilot skills in the product repository and drove adoption through knowledge-sharing sessions — now the team's default workflow, used by **20+ engineers**.
- Dependency updates across the team's microservices consumed engineering time every sprint; automated them in the CI pipeline, removing the manual workload.

- Mentor engineers through code reviews, architecture discussions and pairing; contribute to hiring and technical interviewing.

HelloFresh — Full Stack Developer (part-time), Germany

May 2021 – Sep 2021

- Built a B2B application frontend in React and TypeScript with backend APIs in C# (.NET Core); delivered an internal ticketing desktop application using React and Electron.

University of Rostock — Research Contributor, EU Dementia Care Project

Oct 2019 – Sep 2021

Germany — part-time research role held concurrently with Master's studies

- Built a Python application for motion tracking and prediction of hand-movement patterns to improve automated patient-monitoring accuracy, applying computer vision and machine learning.

Expando — Lead Software Engineer

Jul 2018 – Jul 2020

Pakistan — concurrent with 10Pearls until Oct 2019, then part-time remote from Germany

- Led a team of **4 engineers** building a greenfield service marketplace platform (TypeScript, React, Node.js, MongoDB, AWS EC2/S3), owning full-stack architecture across backend, frontend and mobile.
- Designed a pub-sub architecture for real-time communication on a carpooling platform, and cut third-party API costs with Redis-based caching for the Google Distance Matrix API.

10Pearls — Full Stack Software Engineer

Nov 2017 – Oct 2019

Pakistan — left to relocate to Germany

- Delivered scalable B2B/B2C applications for international clients using TypeScript, React, Node.js, Angular, MongoDB and GraphQL, collaborating with cross-functional and offshore teams.

Resco Pvt. Ltd — Software Engineer, Pakistan

Mar 2016 – Oct 2017

- Developed and deployed hardware-integrated enterprise systems (biometric attendance, canteen billing, visitor management) for HBL Bank and Dolmen Group (C#, SQL Server, RFID, Angular); the software still serves **~10,000 users daily**.

AGENTIC AI & CLOUD PROJECTS

Agentic AI Framework — Multi-Agent Orchestrator (LangGraph)

2026

- Architected a stateful multi-agent development pipeline orchestrated in LangGraph with three specialised personas: **Defi** (architect — resolves ambiguity through dialogue, then breaks the feature into an approved spec and subtasks), **Mason** (developer — implements from curated context and opens the PR) and **Elina** (reviewer — validates against requirements and returns structured findings that Mason acts on).
- Designed agent-facing contracts: agents and tools exchange typed, JSON-structured results and errors, so each agent can self-correct and the next step acts without human re-interpretation. LangGraph state holds only what is needed to resume (current node, issue/PR refs, approval flags) with checkpointing; GitHub remains the durable source of truth.
- Built human-in-the-loop authority into the graph as explicit interrupts: no implementation starts without spec approval, and the merge decision always stays with a human. Adopted as the team development workflow at Arbeity.
- Benchmarked cost per feature: Anthropic Console (pay-as-you-go) and AWS Bedrock both ran **~€40** on a medium feature; moved to local orchestration that escalates only what needs human judgment, cutting cost per feature **~4x** at comparable quality and speed. Next: MCP server for tool access, evaluation on PR open, hosted FastAPI/Postgres worker.

Private repository | Write-up: [I Hired a Robot Dev Team \(AI in Plain English\)](#)

Tally — AI-Powered Personal Finance Navigator | AWS

2026

- Solo-designed, built and shipped to production in **~2 months** alongside a full-time role: **11 production features** delivered through the multi-agent orchestrator, each reviewed by the agent pipeline before human sign-off and merge.
- Chose a serverless AWS backend (Cognito, API Gateway, Lambda/Go, DynamoDB) over a Docker/Kubernetes footprint for near-zero idle cost and auto-scaling; EventBridge-scheduled Lambda jobs and SES emails; React Native frontend; OpenAI for natural-language financial insights and spending analysis.

Private repository | [maanizstudio.com](#)

smart-system — Grounded Document Q&A (RAG) | Python, FastAPI, LangChain, FAISS

2026

- Built a document Q&A service where every answer is grounded in retrieved content rather than model priors: semantic search over embedded chunks in FAISS, answers returned as structured JSON with confidence score and source page, and a NOT FOUND guardrail instead of guessing. FastAPI with Pydantic validation, a typed error hierarchy returning structured JSON errors, React/TypeScript frontend, and pytest coverage across retrieval and scoring.

Code: [github.com/usman250994/smart-system](#) | Write-up: [I Built a RAG-Powered Smart PDF Assistant \(AI in Plain English\)](#)

Open-Source Distributed Systems Work | Kafka, NestJS, Go, AWS

2025 – 2026

- Published reusable boilerplates for event-driven microservices (Kafka producer-consumer, TTL error handling) and Go serverless clean architecture, cutting setup time for greenfield projects; built EventBridge and Step Functions implementations of the same workflow to compare choreography against managed orchestration.

Write-ups: [Go serverless clean-architecture boilerplate](#) | [EventBridge or Step Functions? I built both](#)

EDUCATION & LANGUAGES

MSc Computer Science (Smart Computing / AI) — University of Rostock, Germany

Oct 2019 – Sep 2021

- Thesis: Neural Network Interpretability & Rule Extraction (XAI) — applied K-Means clustering and decision trees to extract human-interpretable rules from black-box models, using Python (Pandas, Scikit-learn, TensorFlow).

BSc Computer Science — Institute of Business Administration (IBA), Pakistan

Feb 2013 – Dec 2016

Languages: English (fluent, working language) | German (B1+)