ABU BAKAR WASE

Forward Deployed Engineer | AI Engineer | Senior Full Stack Engineer

Hamburg, Germany | +49 176 17 13 1474 | abubakar.wase2712@gmail.com | LinkedIn | GitHub

EU Work Authorized | 30-Day Notice | Open to Remote & EU Relocation

PROFESSIONAL SUMMARY

High-agency Senior Full Stack AI Engineer with 6+ years of owning problems end to end, from spotting the bottleneck to building, deploying and onboarding the people who use the fix. I helped Fashion Cloud move from startup to scale-up by rebuilding its data pipeline, media processing and content delivery, and by automating brand onboarding, customer support and internal workflows with deterministic AI agents, so it serves a much bigger customer base with less manual work and lower operating cost. I pair classic engineering (distributed systems, SLOs, cost-aware AWS) with agents where evals and guardrails decide what ships. Before that I spent two years embedded in client teams, from pre-seed MVPs to unicorns.

TECHNICAL SKILLS

Agentic AI: Claude Agent SDK, LangGraph, LangChain, OpenAI Agent SDK, CrewAI, AutoGen, Claude, MCP, Multi-Step Agent Workflows, Tool-Using Agents, Deterministic Playbooks & Rules-as-Code, Confidence Gates & Human-in-the-Loop, Agent Observability (Langfuse), Claude Code, Cursor, Agent Rules & Skills, AI Coding Agents

Architecture & Design Patterns: Agent Harness Design, Agent Evaluation Harnesses, Harness Engineering, Graph Engineering (LangGraph State Machines), Loop Engineering, Human-in-the-Loop Feedback Loops, System Design, Distributed Systems, Microservices, Event-Driven Architecture, Event Streaming & Replay, Domain-Driven Design, Orchestration & Choreography, Micro-Frontends, REST API Design, Reactive Programming

Generative AI / ML: RAG Pipelines (Hybrid BM25 + Vector Search, Reranking), Prompt Engineering & Versioning, Structured Outputs, Embeddings, Semantic Search, LLM Evaluation, Golden Sets & Rule-Based Validation (Golden Rules), LLM-as-Judge, Regression Gating in CI, Classifiers (Jev, LoRA Vision Classifier), Transfer Learning, LoRA / PEFT Fine-Tuning, Hugging Face, Vision Classification

Languages: TypeScript, JavaScript (ES6+), Python

Frontend: React, Next.js, Angular, Ionic, Redux Toolkit, RxJS, SCSS, Storybook

Backend & Databases: Node.js, NestJS, Express, FastAPI, GraphQL, Kafka, Auth0, PostgreSQL, MongoDB, Redis

Cloud & DevOps: AWS (Lambda, ECS, Step Functions, EventBridge, Kinesis Data Streams, KMS, S3, SQS, SNS, CloudFront, MediaConvert, Cognito, API Gateway, Bedrock), Docker, Kubernetes, Terraform, Spacelift, Infrastructure as Code

Observability: Datadog, CloudWatch, Langfuse, SLOs & Error Budgets

Testing: Jest, Mocha, React Testing Library, Jasmine, Karma, TDD

Tools: Git, CI/CD, GitHub Actions, CircleCI, SonarCloud, Jira, Linear, Scrum, Shape-Up

Customer-Facing Delivery: Customer Workshops, Requirements Discovery, Stakeholder Management, Rapid Prototyping, MVP to Phased Rollout, Customer Integrations (CRM, POS, sFTP), On-Call & Incident Response, Technical Documentation

WORK EXPERIENCE

Full Stack Engineer | Fashion Cloud

Jan 2023 – Present | Hamburg, Germany | Full-time

- **AI support agent.** The support queue got 20+ repetitive tickets a week from enterprise brands. Each one needed a manual fix, so one on-call engineer spent the rotation on them while customers waited. Pitched and shipped a LangGraph + Claude agent on a custom harness to production in under 2 weeks. The LLM only classifies tickets and pulls out the details; tested playbooks behind confidence gates take the actions, and golden-set evals with Langfuse traces check every change. **Repetitive tickets cut from 20+ to under 5 a week**, freeing the on-call engineer.
- **Brand data pipeline.** Onboarding a brand took about 7 days of manual work, because each brand sent conflicting data in many formats. Delivered a new pipeline in under 8 months: Dagster orchestration with synchronous workflows on Lambda, Step Functions and ECS, covering sFTP ingestion, parsing, normalisation and a multi-strategy merge engine, with enrichment running in parallel once products publish. Change detection re-enriches only what changed. **Runs a 30M+ product catalog**, stress-tested on the largest brand and migrated with zero downtime.
- **Product data streaming (PDS, PCS).** At 500k product updates a day, EventBridge copied every event into each consumer's SQS queue; publishing hit rate limits and consumers fell back to 7h full syncs. Co-built the Product Data Service and Product Content Service on Kinesis: one publish to style, colour and size streams with 2-day replay and KMS encryption, relayed to filtered queues for internal and client use. **Publish throttling gone, legacy queue traffic down 84%**, 40M+ records a month for under $300.
- **Brand onboarding agent.** Even with the new pipeline, every brand still needed hand-written parse and mapping files. Built an autonomous agent on ONA, a third-party cloud agent platform, that writes these files and processes the brand. If data parity is below 85%, it fixes the mappings and tries again, then opens a PR once it passes. **350 brands onboarded, onboarding time down from 7 days to 1.**
- **In-house media processing.** Image variants ran through imgix, a costly third-party service. Replaced it with an in-house Sharp library on ECS, **cutting image costs from about $2.7k to $0.3-0.5k a month**. Also built a MediaConvert video library on EventBridge that **three teams** use with no extra wiring.
- **Data enrichment and classifiers.** Brands send incomplete product data, but the 2026 OKR was 100% data quality: no missing or invalid attribute in the database or in the data retailers pull from us. Built the Data Enrichment Service. It classifies garment images by type, colour and other attributes, and an LLM-as-judge classifier checks the results (first a LoRA-tuned Hugging Face vision model, now Jev, a probabilistic classifier from Typesafe AI). **100% data quality by Q2 2026, two quarters early.**
- **Content distribution (CDN).** Files were served through uncached S3 signed URLs, so every request went through the whole system and each byte was paid for twice (NAT and load-balancer egress). Rebuilt serving alone as ingestion, processing and distribution layers with TTL-cached CloudFront URLs. **Egress down from 73 TB to 3 TB, saving about $2-3k a month**, while serving about 50 TB and 500M+ requests a month.

- **Content API.** The highest-revenue API got slower as the company scaled, and it had no service-level target. Moved reads to a sharded read database kept in sync through events. Got management to adopt an SLO (p99.9 under 500 ms, 0.1% error budget) with Datadog burn alerts. **Revenue share grew from 23% to 36%** since taking it over.
- **Marketing Material.** Always-on workers made it expensive, async hops were hard to trace, and customers could not send data from their own systems. Rebuilt it as its own ECS service with Lambdas and orchestration, added external CRM and POS APIs. **Running cost down 60-70%**, and **4 of the biggest customers** send millions of products through the APIs.
- **Architecture committee.** The content team's code lived in two other teams' services, so every change waited for their review. AI-assisted coding made that queue even longer. Helped drive a company-wide, domain-driven split of services by team ownership (2024 OKR, delivered) and moved that code into one content service. **The content team now ships without cross-team review.**

Stack: *NestJS, Angular, RxJS, TypeScript, Python, MongoDB, PostgreSQL, AWS (Lambda, Step Functions, ECS, EventBridge, Kinesis, CloudFront, MediaConvert), Terraform, Docker, Datadog, Dagster, LangGraph, Claude, Langfuse*

Full Stack Engineer | VentureDive

Nov 2020 – Dec 2022 | Lahore, Pakistan | Full-time | Consultancy embedding engineers in US, UK and MENA client teams

Motive, San Francisco logistics unicorn, 500k+ fleet operators

- **Real-time fleet view.** FleetView's old dashboard polled for updates, so fleet owners saw delayed truck data. Rebuilt it from scratch as a real-time map view in Angular 10, with RxJS reactive state, WebSocket loading, live dashcam feeds and 85% unit-test coverage. This contributed to a **40% increase in customer satisfaction**.
- **Hours of Service compliance.** Under FMCSA rules, US fleets must prove that drivers take required breaks and log their hours correctly. Built compliance checks that combine ELD and truck ECU signals to auto-flag missed breaks and wrong timesheets.

Stack: *Angular 10, RxJS, TypeScript, WebSockets, Google Maps API, IndexedDB, Jasmine, Karma*

Daraz, Alibaba subsidiary, B2C e-commerce, 2M+ active users

- **Auth service split.** Sign-in lived in the monolith and was slow, held up by backend auth and synchronous API calls. Moved sign-in (frontend and backend) into a standalone Auth0 service that runs as a widget in the React micro-frontend platform. Shipped it as an MVP first, then in phases. **Sign-in time cut from about 2.3 s to 0.5 s.**

Stack: *React, Redux Toolkit, Micro-Frontends, Auth0, Node.js, JWT, React Testing Library*

ApexChat, CRM and virtual assistant platform, 1k+ clients, 1M+ messages/day

- **React portal and design system.** The management UI was tied to a legacy .NET monolith that served both frontend and backend from one server. Rebuilt it as a React SPA portal (data tables, forms, dashboards) with Redux Toolkit and Formik/Yup. Created the design system with the client team and documented it in Storybook, so new pages reuse ready-made components.

Stack: *React, Redux Toolkit, Formik, Yup, SCSS, Storybook, React Testing Library, SonarCloud*

Zabardast, pre-seed startup of a Careem co-founder

- **Idea to pre-seed MVP.** The founder needed a working product for delivery riders before the pre-seed round. As the only engineer, turned the idea into requirements and milestones, then built the mobile-first Ionic app, the NestJS backend and the web app with AWS Cognito. **MVP ready in 3 months and used in the raise.**

Stack: *Ionic, Angular, NestJS, TypeScript, AWS Cognito*

Nugttah, point-based loyalty platform, 100k+ users, 200+ brands

- **Marketing automation.** The marketing team could not schedule or automate campaigns on their own. Redesigned the SMS and push module on AWS SNS/SQS with CRON jobs, CRM integration and a campaign dashboard. **Active users rose 23%.**
- **POS integration.** A POS integration hardcoded for one client was one of the company's biggest bottlenecks. Turned it into a reusable module for the Foodics and MARN POS providers, **enabling enterprise clients including Dunkin' Donuts and Starbucks**. Also planned the move to microservices, starting with auth.

Stack: *Angular, RxJS, TypeScript, Node.js, Express, MongoDB, AWS (S3, SNS, SQS, Cognito)*

Frontend Engineer | ADDO AI

Mar 2020 – Oct 2020 | Lahore, Pakistan | Full-time

- **Investor dashboard.** Investors comparing US states had to pull data from several open-data APIs. Built the interactive D3.js, Chart.js and ECharts views for Mapology, a private dashboard of county-level data (PM2.5, population, traffic, industrial activity, disasters). Investors can zoom from one county out to national averages.

Stack: *D3.js, Chart.js, ECharts, JavaScript, REST APIs*

Full Stack Engineer | DevFactori

Mar 2019 – Mar 2020 | Lahore, Pakistan | Full-time

- **Attendance module.** Built the full attendance module for New York Sports Science Lab (NBA-affiliated), with sprint demos.
- **Bidding engine.** On a PCB bidding platform, bids placed at the same moment could collide. Helped design the bidding engine with locks and timers, so bids at the same moment resolve safely and auctions close on time.

Stack: *Node.js, Express, MongoDB, Mongoose, Angular, Redis, AWS S3, CircleCI*

AWARDS

Hackathon Winner | nemox, Munich | Nov 2026 | 6 hours, team of two

- **Feedback Autopilot.** Customer feedback at nemox waited for a developer to triage and fix it by hand, and new developers were slow to learn what customers meant. Given the problem on the day, built with a teammate an agent that does a developer's first hour: it reads the feedback and session data, reproduces the bug, writes the fix and opens a PR, merging only fix types proven safe and routing the rest to a developer. Added an onboarding map of each screen's purpose, real usage, fixes and current state. Python, LangGraph, FastMCP, PostHog, Claude Code. **Won with a live demo, from customer complaint to merged fix.**

EDUCATION & LANGUAGES

BSc Computer Science | COMSATS University Islamabad, Lahore, Pakistan | 2015 – 2019 | English (C2), German (A2)