Alexander Tikadze

Senior .NET Engineer · AI Solutions Architect · Team Lead

+995 551 230 811 | a.tikadze@gmail.com | t.me/ATikadze | github.com/atikadze

PROFESSIONAL SUMMARY

Senior .NET Engineer and Solutions Architect with 8+ years of experience delivering scalable backend systems across diverse industries and domains, now architecting AI-native engineering delivery: the shared agent tooling, local environments, CI quality gates, performance benchmarks and delivery metrics that let human and AI-agent contributors go from an approved specification to a reviewable pull request safely. Specializes in Microservices, Domain-Driven Design, and Clean Architecture, from greenfield system design through to production performance optimization. Proven track record producing full C4 architecture documentation (L1-L4) and owning complex distributed systems end-to-end. Leads cross-functional teams of up to 10 developers across the full SDLC. Hands-on with Claude Code agent tooling, MCP servers and GitHub Actions.

TECHNICAL SKILLS

Backend: C#, .NET 8, ASP.NET Core, .NET Framework, REST API, EF Core, LINQ, xUnit, Moq, Autofac, Swagger

Architecture: Microservices, DDD, Clean Architecture, Onion, Hexagonal, CQRS, Outbox Pattern, Saga Pattern, C4

Cloud & Messaging: Azure (Service Bus, App Configuration, APIs, Blob Storage, App Insights, CosmosDB), AWS, RabbitMQ, Dapr

Databases: MSSQL, PostgreSQL, MySQL, MariaDB, MongoDB, ElasticSearch, CosmosDB, ClickHouse, Redis

AI Engineering: Claude Code (plugins, agents, skills, hooks), MCP servers, spec-driven development, AI code review, OpenAI Codex CLI

CI/CD & Testing: GitHub Actions, Docker Compose, Testcontainers, Playwright, ArgoCD, Bash, Python

Frontend: JavaScript, TypeScript, React, HTML/CSS

Tools: GitHub/GitLab/BitBucket, Jira, Postman, Visual Studio

PROFESSIONAL EXPERIENCE

Digicode

AI Solutions Architect — *Jan 2026 - Present*

- Own the AI-native engineering delivery system for a multi-tenant .NET 8 / React SaaS platform: fast feedback loops, a spec-to-PR workflow enforced by GitHub rulesets and required checks, and automated platform stewardship driven by Datadog, Azure and CI evidence, so engineers and AI coding agents move from approved spec to reviewable pull request safely.
- Designed and built a versioned Claude Code plugin marketplace (3 plugins: agents, skills, hooks, MCP servers) replacing per-repository copies across 2 application repositories, with release tags, a version-bump CI check and a one-script install.
- Built a local Docker environment around the Azure Service Bus emulator, taking queue-driven features from shared-environment-only to fully testable on a laptop (4 min cold start, 42 s restart); traced a silent cross-tenant conversion loss to local instances consuming shared queues and added a startup guard that blocks it.
- Introduced the platform's first automated performance benchmark: a month of generated traffic (200,000 views, 100,000 clicks, 10,000 conversions) seeded in one set-based SQL statement 20x faster than row-by-row, with a CI-enforced p95 limit of 4 s against a measured 1.9 s.
- Built a fixtures seeder that creates four realistic tenants in one command through the product's own services; 72 of 72 end-to-end flows pass on a freshly seeded stack.
- Made local and CI validation equivalent: moved backend CI from Windows to Linux on the developers' own test command across a 3,163-test suite, cutting the build step from 320 s to 205 s, extended local integration coverage from 7 to 14 suites and added per-PR test-failure summaries.
- Eliminated a cryptography test that failed about 1 in 256 CI runs, proven with a 5,000-round probe, and fixed a second test that passed for the wrong reason.
- Rewrote the AI agents' safety guard hooks around a shell-command parser that understands quoting, operators, heredocs and subshells, so guards judge what actually runs; accuracy rose from 60 to 84 of 84 test cases.

- Defined validation patterns for API, admin, MCP-tool and queue changes that tell an agent which checks each kind of change needs, and extended the spec format so performance requirements become measurable acceptance criteria checked by a benchmark.
- Established feedback-loop targets and a 30-day CI baseline (PR gate median 12.8 min, p95 14.7 min), automated as a weekly GitHub Actions report; delivered 22 tickets and 39 pull requests across 4 repositories.

Senior .NET Engineer | Solutions Architect *Sep 2024 - Dec 2025*

- Collaborated with system analysts to produce end-to-end system design documentation for cross-functional development teams, covering functional requirements, C4 diagrams (L1-L4), data schemas, state machine transitions, outbox event models, and acceptance criteria.
- Authored full architecture for an Automatic Refund Processing system integrating Google Play, App Store, and Microsoft Store webhooks, including ghost refund detection, retry logic, and escalation pipelines, eliminating manual processing for the majority of refund cases.
- Designed a modular Event Contest System (Announcement → Registration → Leaderboard → Results) with complex scoring multipliers, entry purchase flows, and snapshot-based client delivery via ExtendedGameData using RabbitMQ fanout exchanges.
- Architected event-driven distributed flows using RabbitMQ and the Outbox pattern, ensuring reliable, idempotent, retry-safe processing across multi-instance microservices.
- Led architectural restructuring of a large-scale AI platform built on microservices with Azure Service Bus and Dapr, removing redundant services and eliminating Saga pattern overhead to improve maintainability and system reliability.
- Drove significant backend performance improvements in production: higher throughput, reduced latency, and lower cloud resource consumption.
- Aligned backend architecture with the AI/ML team's inference pipelines and data processing workflows.

Cornerstone Technologies International

Dec 2020 - Sep 2024

Senior .NET Engineer | Team Lead | Solutions Architect

- Led multiple simultaneous projects as Architect, Team Lead, and Developer spanning diverse domains including Web3, e-commerce, delivery, payments, HR, and AI.
- Architected greenfield systems using Clean Architecture, DDD, and Microservices; managed teams of up to 10 developers including code reviews, mentorship, sprint planning, and coding standards enforcement.
- Owned the full Azure environment across multiple projects: databases, APIs, Blob Storage, monitoring via Application Insights, and proactive cost management.
- Designed and built Fraction Motors from scratch using DDD and blockchain technologies (Solidity smart contracts, ERC-20/1155, Solana, Ethereum, Ledger Crypto Wallet, IPFS) for decentralized fractional ownership of luxury assets.
- Refactored the Tulogy platform to Clean Architecture, introducing middleware-based request/response logging and improving system observability and long-term maintainability.
- Worked across six database engines (PostgreSQL, MySQL, MSSQL, MongoDB, Elasticsearch, CosmosDB) and delivered multiple third-party API integrations.
- Served as the primary client-facing technical contact: managed project scoping, sprint planning, status communication, and delivery milestones directly with stakeholders.

Wandio

Oct 2019 - Dec 2020

.NET Developer

- Led development on balance.ge, a high-traffic financial platform, owning end-to-end feature delivery and team coordination.
- Contributed to a full redesign of zoommer.ge, a large-scale e-commerce platform, integrating multiple payment providers and improving user experience across the platform.
- Built independent plugins and modules for nopCommerce, extending platform functionality for multiple e-commerce clients.

LANGUAGES

English (fluent) · Russian (fluent) · Georgian (native)