# Anatoly Rudakov

Senior Java/Kotlin Backend Engineer | Agentic Systems, MCP, RAG

Belgrade, Serbia • anatoly.ru@gmail.com • linkedin.com/in/anatoly-rudakov • github.com/tolboy
Telegram @kodobeast • WhatsApp kodobeast

## PROFILE

Backend engineer with 10+ years of commercial experience building server-side systems in Java and Kotlin — telecom billing, ECM, integration platforms and mobile back ends. Strongest in integrating heterogeneous systems (REST/OpenAPI, Kafka, SOAP), decomposing monoliths into services, and taking solutions all the way into operation.

Since April 2025 I have been building my own products. Berloga-AI is a local-first platform for governed AI workflows on Kotlin / Spring Boot 4, taken from prototype to an installable Windows product. AI Overseer is a governed Agentic SDLC pipeline in Java 21, where a deterministic controller drives AI workers from different vendors through machine gates and a human gate.

**My AI work is systems engineering around models**: RAG, MCP and tool calling, Agentic SDLC, policy-based model routing, guardrails and evaluation gates. I integrate and operate models in production code rather than train them.

## CORE SKILLS

**Languages:** Java (8–25), Kotlin, SQL

**Backend:** Spring Boot 3/4, WebFlux + coroutines, Spring Security, Hibernate/JPA, Apache Camel/CXF, REST/OpenAPI

**Integration & messaging:** Kafka, RabbitMQ, webhooks, MCP (Model Context Protocol)

**AI engineering:** Spring AI, Ollama, Claude/GPT/Grok APIs, MCP, RAG on pgvector, tool calling, Agentic SDLC, guardrails, evaluation gates

**Databases:** PostgreSQL (+ pgvector), Oracle, MySQL, Flyway, Liquibase

**Infrastructure:** Docker / Compose, Kubernetes, Gradle/Maven, GitHub Actions, Linux

**Testing & observability:** JUnit 5, Mockito, Testcontainers, OpenTelemetry, Prometheus/Grafana

## PROFESSIONAL EXPERIENCE

### Berloga-AI (Freelance)

**Senior Backend Engineer (Java/Kotlin/AI)** | April 2025 — Present

**AI Overseer — an open-source Agentic SDLC control plane (August 2026 — Present).** Coordinates AI workers through implementation, tests, independent review, visual acceptance and human decisions; merging remains a human action.

- **Writer provenance and budgets:** selected independent reviewers against the full writer history and enforced declared cost bounds across effective profiles and retry attempts.
- **Layered acceptance:** a visual-QA role found a P1 defect after green browser scenarios; AI Overseer routed the fix, re-ran the gates and reached the human gate.

*Java 21+, zero third-party Java libraries; offline javac build. Source: github.com/tolboy/ai-overseer*

**Technologies:** Java 21+ (--release 21, offline build), Chrome DevTools Protocol, git worktree, Codex / Grok / Claude CLIs.

**Berloga-AI — governed AI workflows, local-first (April 2025 — Present).**

A local-first governed-AI platform on Kotlin / Spring Boot 4: policy-based routing between local and frontier models, MCP connectors and policy-based approval for tool calls, with chat defaulting to ASK. Shipped as an installable Windows product.

- **Governed action dispatcher:** a compare-and-set claim and idempotency key separate approval from side effects; ambiguous outcomes require review without blind retries.
- **Hybrid model routing as policy, not heuristics:** a private catalog cannot be escalated to the cloud, while token/cost preflight prevents silent bill growth.
- **Cut per-turn latency from 9.6–85 s to 1.0–1.8 s.** OTel spans exposed tool-catalog synchronisation on the request path; moved it to a coalesced background refresh.
- **Quality gates in CI:** unit tests, Testcontainers + pgvector and evaluation corpora block security and routing regressions.

**The Telegram connector is published as a standalone open-source project:** github.com/tolboy/telegram-mcp-tdlib — see the Open source section.

*Scale (as of 08.2026): ~49,000 lines of Kotlin in the core, 550 automated tests, 41 Flyway migrations, 3 modules plus MCP connectors.*

**Technologies:** Kotlin 2.3, Java 25, Spring Boot 4, Spring AI, MCP, PostgreSQL + pgvector, Docker, Testcontainers, OpenTelemetry.

### RZD International (Remote)

**Senior Software Developer** | January 2024 — April 2025

Back end for a sports-events mobile application and an online-store integration service; microservice integration over REST API, Kafka and SOAP.

- Designed the integration with a labour-resource management system: 5 services on Apache CXF (SOAP, WS-Security) and Kafka, synchronising employee and org-structure data.
- Built end-to-end access revocation for dismissed employees (50–200 per daily run) across 5 services: HR system → Kafka → Keycloak roles removed and sessions terminated; retry topics with a DLT.

**Technologies:** Java 21, Spring Boot 3.2+, Spring Kafka, Apache CXF, Keycloak, Hibernate, PostgreSQL, Flyway, Docker, Grafana, REST/SOAP.

### Nexign, Saint Petersburg

**Senior Software Developer** | April 2018 — October 2022

Solutions for telecom billing systems; re-engineered monolithic applications into microservices.

- Migrated part of a monolithic customer-management system to a microservice architecture, so subsystems could be deployed and scaled independently.
- Implemented complex business scenarios with Apache Camel: orchestration of external-system calls with compensating routes on step failure (saga); new scenarios are added through configuration.

**Technologies:** Java, Kotlin, Groovy, Akka, Apache Camel, MyBatis, Oracle DB, PostgreSQL, Liquibase, ZooKeeper, Docker, Kubernetes, Helm, SonarQube, Gradle, Maven.

### Korus Consulting, Saint Petersburg

**Software Developer** | April 2010 — February 2015

Electronic document management system built on the open-source ECM Alfresco.

- Optimised Alfresco's built-in search engine (Lucene) and configured Russian morphological analysis — search time dropped by 40% and result accuracy improved.
- Set up a CI/CD pipeline (Bamboo + Mercurial) with automated unit tests and deployment to Glassfish/Tomcat, significantly reducing release time.

**Technologies:** Java, Alfresco ECM, Lucene, GWT, MySQL, PostgreSQL, LDAP/SSO, Maven, Bamboo, Mercurial.

## OPEN SOURCE

**telegram-mcp-tdlib —** github.com/tolboy/telegram-mcp-tdlib (Kotlin, Spring Boot 4, MCP SDK, TDLib, Apache 2.0)

A Kotlin MCP server for Telegram: multi-account isolation, read-only defaults, policy-controlled writes and security CI.

## EDUCATION AND CERTIFICATION

**Bonch-Bruevich Saint Petersburg State University of Telecommunications** — Specialist degree (5-year programme, equivalent to MSc), Multichannel Telecommunication Systems (1999–2004)

**Saint Petersburg Polytechnic University** — professional training programme "Application Software Developer (Java)", state-issued certificate (2009)

## ADDITIONAL

**Languages:** Russian — native; English — B1+ (fluent reading and writing of technical documentation and correspondence).

**Work format:** Remote (CET/CEST), full-time or contract.