Stanislav Alekminskii

Backend & Distributed Systems · Hands-on Technical Leadership

Istanbul, Türkiye · English C1 · stan@stasal.dev · stasal.dev · LinkedIn · GitHub

Backend engineer working in Java and Kotlin on event-driven financial systems: transaction processing, financial reporting, and trading. Takes technical ownership of core backend scope: system design, production incidents, and reusable integration infrastructure used by other teams. Took over from a departing lead four times, on joining a project or partway through it.

EXPERIENCE

EPAM — Senior Software Engineer

2018–present

Financial transaction processing — one of Europe's largest retailers

Mar 2025–present

Joined a distributed three-developer team to replace the departing project lead and took over technical leadership of the core backend and processing components of a financial posting system. It handles **~4M events/day from 7 sources** (POS, property/rent, and other financial transactions), with expansion to ~70 sources planned within six months.

- Identified a scaling limitation in the streaming aggregation model: related accounting lines could expand past the configured stream limit. Designed a replacement built on bounded chunk reads, stateful postings, and size- and time-based finalization, and made the case to the Product Owner for redesigning before growth made the limit critical. The implementation is nearing completion.
- Restructured the core aggregation processor, whose logic had been concentrated in a single service that was hard to maintain and extend: introduced abstraction layers, reworked parts of the algorithm, and moved suitable computation into MongoDB aggregation pipelines. One problematic message no longer breaks processing for all sources.
- Resolved a production incident within hours, on the live system: a collection without retention had grown to **~80M documents** and the database could no longer keep up with the daily transaction stream. Added indexes, removed obsolete data, and introduced a retention procedure for the collection; processing returned to normal, and later stabilization removed the need for daily close monitoring.
- Built from scratch a reusable adapter library for source-system integration: a documented multi-module Maven project with a Spring Boot starter, shared components, source-specific tests, and consumer-contract tests. Now used by three teams, with a fourth adopting it. Documentation is the first line of support, and adopting teams raise almost no questions.
- Keyed Kafka messages by `batchId`, falling back to `eventId`, so related accounting lines stay together while load spreads across partitions. Co-designed and implemented a separate MongoDB partition-key strategy and Kafka-to-Mongo partition mapping, so broker and database scaling stay decoupled.
- Influenced product scope as the Product Owner's regular, often daily, technical counterpart: in some cases showed that existing functionality already covered the need, and the Product Owner dropped the proposed feature before refinement.

Financial reporting and external-user access — same retail client

Jul 2023–Mar 2025

Kafka/MongoDB microservice system aggregating financial reports from multiple systems for franchise stores, published through a portal. Four developers, QA, and DevOps.

- Designed and developed services for the reporting system; soon after joining, became a permanent participant in the project lead's and architect's regular technical sync.
- Took over technical-lead responsibilities for the team when the lead moved to another project: design, development, candidate interviews, and code-review practices. Carried out the closing knowledge transfer to another team on site in Amsterdam.
- In parallel during 2024, took over an access-management implementation for external business-partner users and owned its entire backend: researched Microsoft Graph API, reworked the inherited implementation where needed, and built the controllers and services, with PostgreSQL for state. By year end the team had a working MVP integrated with the reporting portal.

Confidential pharmaceutical client

2022–2023

Java development automating parts of chemists' work in drug development. Most experienced engineer in a three-developer team that picked the project up after the previous team left.

Trading platform — one of Russia's largest investment banks 2018–2022

Trading system that had grown out of traders' Excel spreadsheets with VB macros. About five developers, eight analysts, one QA, and a technical lead; no dedicated architect role on the project. Kotlin in production.

- Joined to replace a departing lead/senior engineer. When the planned handover was cut to 2–3 days, learned the system quickly and took over development and code review.
- Co-designed and implemented a new flow publishing trader quotes directly to an exchange through QuickFIX/J. Diagnosing an exception in that work led to an upstream QuickFIX/J contribution.
- Toward the end of the project, acted as team lead in practice: prepared roadmaps for Product Owners, discussed timelines, and screened incoming analyst requests with the technical lead before they reached developers, protecting limited development capacity from uncontrolled scope growth.
- Introduced an explicit task flow (definition, refinement, estimation, development, review, deployment) in the core development stream; the team worked by it and refined it from 2019 to 2022. Reviewed code extensively and interviewed developer candidates.

Digital Mind Development — Software Engineer → Team Lead 2012–2018

Java EE/Oracle systems for a network of aluminium plants. Moved from .NET to Java on joining; took over the projects when the previous technical lead left, and became formal Team Lead.

- Designed and implemented a Hibernate-based reporting engine with runtime entity configuration through dynamically generated XML mappings. A defect found along the way became an upstream Hibernate ORM fix.
- As Team Lead: designed new components, defined and distributed development work, validated results, reviewed code, and worked directly with the customer and business analysts.
- Developed software standardizing planned and unplanned repair records for electrolyzer equipment across the plant network (team of about four developers and two QA), including work on site at plants.
- Designed the JSF frontend, including templating, for a foreman/technologist workplace application, took part in its development, and joined on-site deployment at the new Boguchany Aluminium Plant.
- Initiated TeamCity deployment pipelines, development/review workflow improvements, and task- and review-tracking integrations.

Earlier experience: Freelance, 2011–2012: Java audio-recording plugin for IBM Lotus. Siberia-Intellect, 2007–2008 and 2009–2011: practical training, ASP/.NET and C# development, IT support.

AI-ASSISTED ENGINEERING 2025–2026

Self-initiated experimental work alongside project delivery at EPAM.

- Built an automated review loop for agent-written code: a draft pull request gets a Copilot review plus several background Claude Code review passes, each with its own focus and a fixed JSON output. A consolidator rates the findings on one scale and returns a verdict to the implementing agent, which applies the fixes.
- Created an umbrella repository joining the project's components as Git submodules (chosen over subtree after evaluating both), with agent instructions and pre- and post-commit hooks. Moved project-specific engineering conventions from prose into checkable agent rules.

OPEN-SOURCE CONTRIBUTIONS

- **Hibernate ORM:** fixed invalid SQL generated when an HQL query combines a left join to an unrelated entity with an implicit join in the select clause (HHH-11337). Found on Oracle while building the reporting engine; applied upstream with a regression test. Pull request #1696
- **QuickFIX/J:** connection failures reported to session-state listeners now carry the affected socket address, so a failure can be traced to a host. Reviewed with the maintainers and merged for the 3.0.0 release. Pull request #451

TECHNICAL SKILLS

Backend: Java, Kotlin, Spring Boot, Spring Framework (Web MVC, Security, Data JPA/JDBC), Hibernate, Java EE (EJB, JSF, JPA)

Data & messaging: Kafka, MongoDB, PostgreSQL, Oracle, Avro, FIX (QuickFIX/J)

Cloud, delivery & observability: Azure, Kubernetes, Docker, Argo CD, GitHub Actions, TeamCity, Grafana

Build & development: Maven, Git

AI-assisted engineering: Claude Code, GitHub Copilot

EDUCATION & CERTIFICATION

Siberian Federal University
Master of Computer Science and Technology, specialization: Microprocessor Systems
Bachelor of Computer Science and Technology, diploma with honors

Claude Certified Architect — Foundations · Credly