Marko Milenković

Senior Backend Engineer & Tech Lead — Java / Spring Boot

📍 Vizantijski bulevar 8/7, Nis 18108, Serbia 📞 0648292074 | ✉️ mmarko88@gmail.com
🔗 LinkedIn | 🌐 Professional Developer 🎂 06-05-1988



PROFILE SUMMARY

Senior Backend Engineer & Tech Lead with 12+ years of professional experience in Java backend development. Worked with Java web services and relational databases on numerous fin-tech projects involving big-data processing, with deep focus on REST API design, platform migrations, CI/CD engineering, and application security. Focused on sharing knowledge and helping professional Java developers optimize their work and solve problems using best practices, new technologies, and methodologies — writing clean, secure, and performant code in an agile environment. Led teams of up to 8 engineers.

EMPLOYMENT HISTORY

Senior Backend Engineer & Tech Lead — Investment Cafe

Badin Soft (2018–2022) → Wiser Technologies (2022–Present), Niš

2018 — Present

Senior backend engineer and tech lead of one of the teams building the **Investment Cafe** platform — a multi-tenant document management and investor communication system serving BlackRock's alternative investments clients globally (up to ~100k end customers). Contributed to the migration from a legacy Flash application to a modern Spring Boot/React architecture under tight deadlines, delivering on schedule.

- **Production stability & reliability** — Kept the Investment Cafe platform stable under production load. Eliminated recurring SQL Server **deadlocks** (welcome-email, user-creation, permission-sync, login-logging) by root-causing the lock contention and fixing it with targeted index/primary-key redesign, moving hot triggers out of SQL into Java, and offloading low-priority work to async thread pools. Fixed a mail-server connection exhaustion with proper SMTP socket timeouts. Stabilized and sped up the document-visibility permission cache (**~4–7×** faster refresh on the dependent API). Root-caused and fixed a high volume of production defects across the platform.
- **Distribution-list performance** — Owned multi-year performance work on the high-volume distribution-list pipeline (lists of 80k–100k+ recipients): eliminated production timeouts and cut key operations by **7–32×** — recipient loading **~13 min → 6 s** (QA-verified ~24×), filter queries **~148 s → ~5 s**, and list save **~117 s → ~7 s** — via N+1 elimination, DTO projections, indexed views, temp-table joins, and bulk-insert batching.
- **Email & mail delivery** — Migrated send-mail into a dedicated contact service and owned the email/Email Archive pipeline: notifications, postal/hard-copy mailing, attachment handling, and template-variable substitution.
- **Excel import/export** — Led key parts of the platform's Excel import/export capability (contacts, investors, fund-investors, wire instructions, users). Led the performance rework of the import framework: replaced slow row-by-row saving with chunked, temp-table bulk inserts (a reusable bulk-persistence layer using SQL MERGE), making investor import **~40× faster** and eliminating gateway timeouts that previously failed large imports; restored 10k-row import capacity across all importers and root-caused the bulk-insert ID-generation bottleneck. Built the import validation layer (email format/length, case-insensitivity), templates, and DTO structure.
- **Antivirus (ZScaler)** — Designed and built the platform's antivirus scanning capability that screens every uploaded/emailed file for malware before it enters the system. Defined a pluggable antivirus-scan service interface and implemented the ZScaler integration end-to-end: an OkHttp-based client with tuned connection pools, a configurable block-on-suspicious policy, encrypted API credentials, and wiring into the document-upload and send-mail flows.
- **Audit-trail services** — Built the platform's audit-tracking capability: audit REST APIs (for organizations and resource uploads) backed by SQL Server triggers and audit tables that record who/what/when on every create/update/delete, wiring the acting user into trigger context via stored procedures. Later extended audit coverage (Fund Terms) and fixed a data-integrity bug by widening the audit value columns to a max-length type and redesigning the supporting index.
- **File-upload security hardening (PenTest/TVAS)** — Owned file-upload validation across every file-accepting endpoint (email, distribution mail, resource upload, all Excel imports). Built content-based MIME/extension validation on Apache Tika — refactored the entire detection module to match file content against filename and consolidated the allowed/forbidden-type rules — plus a config-driven extensible allow-list service. Hardened ZIP handling with per-entry validation, zip-in-zip rejection, and zip-bomb thresholds (uncompressed size / entry count / compression ratio), and switched validation from full byte-array loading to streaming to eliminate memory spikes on large uploads. Remediated PenTest/TVAS findings, including a High-severity path-traversal in the resource-upload endpoint (`../` /absolute/backslash filenames rejected).
- **Visibility & data-access performance** — Redesigned the platform's core fund–investor visibility view from a non-indexed 6-table join (recomputed on every request) into schema-bound **indexed/materialized views**, split by role with covering indexes, so entitlement checks became index seeks over precomputed rows (refresh path also **2.5–8× faster**); it underpins visibility across distribution-list filtering, investor/fund and resource-upload services, and shipped behind a config flag for safe rollout. Also drove related data-access wins: DTO-projected the heavy access-rights query to kill N+1 fan-out (**~9× local / ~3× QA**) and added columnstore indexing + query recompilation to a hot category-lookup query (**up to 10×**).

- **Document Search / Summary ("Document Hub")** — Led a multi-year performance program on the platform's most-used document summary/search endpoint. Eliminated gateway (504) timeouts and cut the main endpoint from **~29 s → ~3 s (~9×)** under load by rewriting the SQL — dropped an investor-freeze subquery in favor of the permission cache, removed redundant joins, made joins conditional on the active sort, added subquery paging + clustered indexes, and consolidated the duplicated document-summary stored procedures. Further optimized permission checks (**up to ~2.3×** in consolidate mode) and moved category/fund-spec filtering out of Java into the DB procedure to stop round-tripping report IDs.
- **Build & test performance (developer productivity)** — Champion of fast developer feedback loops — one thread of a sustained focus on performance across the platform (100+ performance/cache/index commits). Led a multi-year overhaul of the CI/build and test suite after PR build time had doubled (12 → 25 min): separated unit vs integration tests (class-naming convention `IntTest/ ResourceTest`) and split them into dedicated Maven modules, parallelized them with forked JVMs and DB isolation, and slashed per-test overhead — removed the slow/deprecated GreenMail server and eliminated blocking `Thread.sleep` /flaky tests, optimized BCrypt password hashing in tests by caching it where hashing strength wasn't important, cut Spring-context reloads (`@DataJpaTest` over `@SpringBootTest`, shared base classes), and batch-persisted fixtures — shaving ~10 min off the local build. Also added a Maven wrapper and migrated the PR pipeline from Windows to Linux — **build step 59% faster, checkout 89% faster** — plus Sonar/JaCoCo coverage and release-pipeline tuning, all directly shortening the feedback cycle for the whole team.
- **develop → develop-plus consolidation (Carlyle)** — Led a 10-month project to reconcile two long-diverged branches: `develop` (shared across all clients) and `develop-plus`, a fork carrying Carlyle-specific logic — notably a more complex permission-processing model. The shared branch had drifted 10–15 versions ahead; migrated all Carlyle-specific logic onto it and retired the separate branch, completing the consolidation successfully.
- **Spring Boot 3 & platform modernization** — Core engineer on the Spring Boot 3.5 / Jakarta EE migration of the standalone API: migrated the core REST module and enabled additional modules, reworked the Spring Security & servlet-filter configuration (redirection/authentication, Swagger/OpenAPI auth), fixed date/timezone defects across fund, document, and resource-upload endpoints, applied Hibernate query optimizations, and carried the Spring Boot major-version upgrade (2.7 → 3.5).
- **IC Core modernization** — Core engineer modernizing the legacy back-end platform: migrated it to **TomEE 10 + Spring Boot 3.5 / Jakarta EE**, structurally migrated remaining services and REST endpoints to Spring MVC (retiring legacy EJB), reconfigured SOAP (Apache CXF) and Artemis JMS, migrated AWS SDK v1 → v2, and modernized/pruned the dependency tree (Java 17, Jakarta cleanup), extracting shared tooling into a dedicated repo.
- **Security & TVAS hardening (enterprise)** — Major contributor to the enterprise TVAS/CWE remediation program: closed authentication bypasses in the JWT filter, a user-association check, and SSO validation (CWE-287/306) and required JWT auth on MFA endpoints; secured the JMS trust boundary with fail-closed authentication and HMAC-SHA256 message signing (CWE-862); added defense-in-depth SQL-statement and Logback-XML validation (CWE-94), ZIP disk-exhaustion protection (CWE-400), and path-traversal/privilege-escalation fixes; disabled public WSDL exposure; resolved 13 SonarQube BLOCKER/CRITICAL findings.
- **Mentoring & team practices** — Mentored and up-skilled the team, introducing and championing engineering practices through pair programming and rigorous code review (300+ PR review comments) — raising code quality and consistency across the codebase.

Tech: Java 8/17, Spring, Spring Boot 3, Hibernate 6, Jakarta EE, MSSQL, Liquibase, Apache POI, OpenAPI 3 (springdoc), Jersey/JAX-RS, Spring MVC, AWS S3/SDK v2, Azure Pipelines.

Senior Backend Engineer & Team/Tech Lead — Direktna Banka

Badin Soft, Niš

2017 (H2)

Team lead for the bank's core lending applications — **WWS** (loan granting), **Recall** (client debt management), and **PozajmiceTR** (account-level debt granting) — which my team fully owned. Maintained and modernized these systems and introduced an adapter integrating a new payment-card processor.

Tech: Visual C#, MS SQL Server, ASMX & WCF web services, Windows services, Windows Task Scheduler.

Senior Backend Engineer & Team/Tech Lead — CREALOGIX E-Banking

Badin Soft, Niš

2017 (H1)

Maintained a 20+ year-old e-banking platform for Swiss banks (TKB, BKB, AKB): fixed client-reported issues on the v2.0 architecture while other teams built the new v3.0. Worked within the application's custom persistence framework and a pure-JS (functional-style) UI, with client-reported bug fixing as the team's primary focus.

Tech: Spring, Oracle DB, Git/Bitbucket, Jira.

Senior Software Developer / Tech Lead — Tranxactor New Zealand Limited

Badin Soft, Niš

2014 — 2017

Backend engineering and team leadership across the Tranxactor loyalty & transaction-processing platform, serving **1M+ members**.

- **Tranxactor HQ** — Three-layer application: an Oracle database layer, a Java API layer (Spring, Spring Boot, MyBatis, Jersey), and an Angular GUI. Focused on the database and API components — conceived, designed, and built secure, high-performance RESTful API services in collaboration with the team.
- **BAT (Batch Action Tracking)** — Designed and implemented from the ground up. Tracks database batch jobs — start, in-progress updates, and completion — with a subscription model that emails subscribed users about events during a job's execution. Includes the **RMU** reporting submodule, which emails scheduled reports from the database to specific users at a given time.

- **Transaction Engine** — Team lead on a module that processes a very high volume of transactions in real time (**~500–600k transactions/day**) — the core engineering challenge. Implemented numerous new features, fixed engine bugs, and supported other developers on their tickets. *(Java, PL/SQL, Tomcat servlets.)*
- **HDA (Help Desk Assistant)** — Backend team lead and architect for a white-labelled help-desk tool used by loyalty-programme support staff. It gives a single dashboard view of a member's activity, with drill-down into transaction history, reward earning/burning, eVoucher issuing/redemption, member communications (email, push, SMS) with live preview, memos and notes, and audit history — branded per client (logos and colour scheme). Designed the application architecture and REST API interfaces, working in agile sprints. *(Java, Spring, Oracle, Angular front-end.)*
- **Voucher Module** — Designed and developed, with a team, a microservice that issues and redeems vouchers in real time during transaction processing. *(Tomcat, JSP/servlets, Hibernate, Jenkins, SVN, Maven, PL/SQL.)*
- **Tiering ("Friends With Benefits")** — Designed and built from the ground up a fully configurable group-reward module: members grouped together (e.g. a group of 4 needing at least 3 to purchase on the same day) earn rewards for coordinated spending. When a group member made a purchase, all other members received real-time notifications; also designed and implemented the daily job that issued the earned rewards.

Tech: Java, Spring, Spring Boot, MyBatis, Jersey, Hibernate, Oracle / PL/SQL, Angular, Tomcat (JSP/servlets), Maven, Jenkins, SVN.

EDUCATION

- Bachelor studies, Faculty of Science and Mathematics, University of Niš, Department of Computer Science. *(2007–2011)*
- Master studies, Faculty of Science and Mathematics, University of Niš, Department of Computer Science. *(2011–2016)* — Master's thesis: *Prodavnica* (a store-management application applying Domain-Driven Design, CQRS, and the MVVM pattern).

SKILLS & TECHNOLOGIES

- **Programming Languages:** Java (Java EE, Swing), SQL, JavaScript, C/C++
- **Frameworks:** Spring (Core, MVC, Boot, Security, Batch, Cloud Netflix, Cloud Gateway, WebFlux), EJB, Jersey/JAX-RS, GWT
- **Web Services:** REST, SOAP, Microservices architecture, JMS, Apache Kafka, OAuth2
- **Databases:** Oracle, MS SQL Server, PostgreSQL, JDBC, Spring Data JPA, Hibernate, MyBatis, PL/SQL, Liquibase
- **Build & DevOps:** Maven, Gradle, Jenkins, Docker, Terraform, AWS EBS, Azure Pipelines, Artifactory
- **Tools & IDEs:** IntelliJ IDEA, Eclipse, NetBeans, Oracle SQL Developer, SQL Server Management Studio, Git, SVN, Jira
- **Servers:** Oracle WebLogic, Apache Tomcat, TomEE, Docker
- **Methodologies:** Agile, Scrum, Kanban
- **Languages:** Serbian (native), English (fluent)

BLOG & WEBSITE

- **professionaldev.dev** — technical articles on Java, best practices, performance, and knowledge sharing.